

universidade de aveiro



**Universidade de Aveiro**

Departamento de Electrónica e Telecomunicações

# **Redes Neurais Aplicadas à Segmentação e Classificação de Leucócitos em Imagens**

**Pedro João Soares Rodrigues**

Dissertação de Mestrado apresentada à Universidade de Aveiro  
como requisito parcial para a obtenção do grau académico de  
Mestre em Engenharia Electrónica e Telecomunicações

Aveiro  
Janeiro de 2000

# Redes Neurais Aplicadas à Segmentação e Classificação de Leucócitos em Imagens

Tese realizada sob a supervisão do  
**Prof. Doutor Armando José Formoso de Pinho**  
Professor Auxiliar do  
Dep. de Engenharia Electrónica e Telecomunicações  
Universidade de Aveiro

## Agradecimentos

Ao Prof. Doutor Armando Pinho pela sua permanente disponibilidade para acompanhar a feitura desta dissertação, através de esclarecimentos, sugestões e correcções que se revelaram preciosos. Desejo manifestar-lhe ainda o meu apreço pela tolerância humanamente demonstrada.

Ao Prof. Doutor Campos Neves por me ter despertado o interesse por esta área científica e pela amizade que sempre me devotou.

Ao laboratório de análises clínicas de Maria Montanha e Laura Soares que facultou, desinteressadamente, todos os elementos necessários à obtenção das imagens.

A todos os meus colegas e amigos que, de alguma forma, contribuíram para a realização deste trabalho.

Aos meus pais  
À memória da minha avó

## Resumo

A identificação automática de leucócitos representa um factor de combatividade e de qualidade para os modernos laboratórios de análises clínicas. No entanto, as classes de leucócitos que os actuais equipamentos conseguem identificar ficam aquém do necessário. Com esta tese pretendeu-se desenvolver métodos baseados em redes neuronais que pudessem aprender a segmentar e a classificar automaticamente qualquer das classes de imagens de leucócitos.

Começou por fazer-se um estudo sobre as características das imagens de sangue humano adquiridas pelo microscópio óptico. A natureza dessas imagens cria muitos problemas a um sistema de visão artificial. As abordagens por nós propostas para a segmentação e classificação dos leucócitos foram no sentido de ultrapassar esses problemas. Ao basearem-se em redes neuronais, permitem a aprendizagem destas tarefas através de exemplos, o que constitui uma vantagem muito grande relativamente a outros métodos. No entanto, o ponto frágil das redes neuronais reside na eficácia do treino a que elas são sujeitas e que por vezes é difícil de conseguir. Os estudos e implementações que foram efectuadas com as redes neuronais permitiram minimizar alguns desses problemas. Como principal conclusão podemos dizer que seguindo as abordagens apresentadas, a utilização de redes neuronais para resolver os problemas da segmentação e classificação de leucócitos é perfeitamente possível, desde que os padrões de treino sejam bem representativos do universo das imagens de leucócitos e a convergência da rede neuronal seja conseguida.

**Palavras Chave:** redes neuronais; reconhecimento de padrões; análise de imagens, segmentação; classificadores; leucócitos.

## Abstract

Automatic identification of leucocytes represents a combativeness and quality factor for clinical laboratories. However, the leucocytes classes that modern equipment can identify are faraway from real necessities. With this thesis we intended to develop neural networks based methods which could learn how to segment and classify automatically every leucocytes classes in images.

We started with a study of the characteristics of the human blood images which were acquired by the optic microscopy. The nature of those images creates many problems to an artificial vision system. The approaches that we propose for the leucocytes segmentation and classification intend to surpass those problems. As those approaches are based on neural networks, they permit learning these tasks using examples, which is an advantage comparing to other methods. However, a weak point of neural networks is due to the training effectiveness that they have to pass through and that sometimes is difficult to achieve. Studies and implementations that were effectuated with neural networks permit to minimise some of those problems. The main conclusion that we draw from this work is that, according to the presented approaches, using neural networks to solve leucocytes segmentation and classification is perfectly possible if the training patterns are sufficiently representatives of the leucocytes images universe and if the neural network convergence is achieved.

**Key words:** neural networks; pattern recognition; image analysis; segmentation; classifiers; leucocytes.

## ÍNDICE

|  |    |
|--|----|
| <b>Introdução</b> .....  | 1  |
| 1.1 Técnicas actualmente existentes para resolver este problema, suas vantagens e desvantagens ..... | 3  |
| 1.2 Objectivos do trabalho .....   | 6  |
| 1.3 Aspecto geométrico das células .....   | 8  |
| 1.4 Características do ambiente de desenvolvimento .....   | 9  |
| 1.4.1 Hardware .....   | 9  |
| 1.4.2 Software .....   | 11 |
| 1.5 Organização da tese .....  | 11 |
| <b>Redes neuronais aplicadas à análise de imagem</b> .....   | 13 |
| 2.1 Introdução .....   | 13 |
| 2.2 Redes Neuronais Biológicas (constituintes) .....   | 15 |
| 2.3 Redes Neuronais Artificiais .....  | 18 |
| 2.3.1 Funções de activação .....   | 19 |
| 2.3.2 Arquitecturas de redes neuronais .....   | 24 |
| 2.3.3 Aprendizagem neuronal .....  | 26 |
| 2.3.3.1 Aprendizagem supervisionada .....  | 27 |
| 2.3.3.2 Aprendizagem não supervisionada .....  | 28 |
| 2.3.3.3 Aprendizagem reforçada .....   | 29 |
| 2.3.4 Backpropagation .....  | 30 |
| 2.3.4.1 O Algoritmo Backpropagation .....  | 31 |
| 2.3.4.2 Regra delta generalizada .....   | 32 |
| 2.3.4.3 Passos da Backpropagation .....  | 34 |

|                                    |  |           |
|------------------------------------|--|-----------|
| 2.3.4.4                            | Actualização dos pesos por lotes/tempo real (batch/on-line)  | 34        |
| 2.3.4.5                            | A inicialização dos pesos                                    | 35        |
| 2.3.4.5.1                          | Inicialização aleatória                                      | 35        |
| 2.3.4.5.2                          | Inicialização Wessels Barnard                                | 36        |
| 2.3.4.5.3                          | Inicialização Nguyen-Widrow                                  | 36        |
| 2.3.4.6                            | Outros métodos de actualização dos pesos                     | 37        |
| 2.3.5                              | Técnicas e constituintes neuronais mais adequados            | 40        |
| <b>O problema da segmentação</b>   |  | <b>43</b> |
| 3.1                                | Descrição do problema da segmentação                         | 43        |
| 3.2                                | Proposta para a resolução do problema da segmentação         | 46        |
| 3.2.1                              | Cor e Pré-processamento                                      | 48        |
| 3.2.2                              | Janela de varrimento   | 52        |
| 3.2.3                              | Extracção de cada leucócito na imagem                        | 55        |
| 3.3                                | Gerador de padrões de treino                                 | 58        |
| 3.4                                | Estrutura da rede neuronal backpropagation usada             | 61        |
| <b>O problema da classificação</b> |  | <b>63</b> |
| 4.1                                | Descrição do problema  | 63        |
| 4.2                                | Descritores  | 65        |
| 4.2.1                              | Descritores de regiões                                       | 66        |
| 4.2.1.1                            | Momentos   | 67        |
| 4.2.1.2                            | Momentos invariantes à escala                                | 69        |
| 4.2.1.3                            | Momentos invariantes à rotação (I)                           | 70        |
| 4.2.1.4                            | Escalonamento dos momentos para as entradas da rede neuronal | 72        |
| 4.2.1.5                            | Momentos invariantes à rotação (II)                          | 73        |
| 4.2.2                              | Descritores de textura                                       | 76        |
| 4.2.3                              | Descritores primários  | 78        |
| 4.3                                | Descritores entregues à rede                                 | 79        |
| 4.4                                | Segmentação do Núcleo/Citoplasma                             | 80        |
| 4.5                                | Descrição do classificador neuronal                          | 81        |
| 4.5.1                              | Treino neuronal  | 83        |

|  |     |
|--|-----|
| 4.5.1.1 Validação Cruzada.....                                   | 83  |
| 4.5.2 Classificação .....  | 84  |
| <b>Resultados experimentais</b> .....                            | 87  |
| 5.2 Resultados da segmentação.....                               | 88  |
| 5.3 Resultados da classificação.....                             | 98  |
| <b>Conclusões e trabalho futuro</b> .....                        | 101 |
| 6.1 Principais conclusões sobre o processo de segmentação .....  | 102 |
| 6.2 Principais conclusões sobre o processo de classificação..... | 103 |
| 6.3 Trabalho futuro .....  | 104 |
| <b>Apêndice A</b> .....  | 107 |
| Implementação da Backpropagation em C++.....                     | 107 |
| <b>Apêndice B</b> .....  | 111 |
| Derivação da regra delta generalizada .....                      | 111 |
| <b>Bibliografia</b> .....  | 117 |



## Capítulo 1

### Introdução

A visão artificial é uma subclasse da inteligência artificial. Embora a definição de inteligência seja algo difusa e sempre discutível, podemos constatar que numerosos autores a definem como a capacidade que um sistema tem de interagir com o meio que o circunscribe. Assim, esta definição pela sua falta de clareza permite considerar que certas espécies de organismos vivos primitivos possam ser tomados como sistemas inteligentes ou que uma simples calculadora possa ser da mesma forma apelidada de inteligente, e de facto estas últimas afirmações poderiam ser validadas ou não, dependendo do contexto de inteligência em que elas fossem proferidas. Outros há que preferem denominar de sistemas inteligentes só aqueles que imitam acções humanas mais complexas, como a síntese de voz ou o reconhecimento de cenas visuais. Nesta perspectiva podemos então dizer que esta tese se encontra imersa nos domínios da inteligência artificial.

O sangue humano é uma substância fluída constituída por células e por plasma<sup>1</sup>. A população celular existente no sangue divide-se em três classes: leucócitos<sup>2</sup>, hemácias<sup>3</sup> e plaquetas<sup>4</sup>. A cardinalidade relativa e absoluta de cada um destes grupos está directamente associada com a existência ou não de anomalias num indivíduo. A classe dos leucócitos, no caso de um indivíduo são, está ainda dividida em cinco subclasses: basófilo, eosinófilo,

---

<sup>1</sup> A quase totalidade do plasma sanguíneo é constituído por água, na qual se encontram dissolvidos proteínas, sais, substâncias nutritivas e produtos de degradação.

<sup>2</sup> Também denominados por glóbulos brancos. Estas células desempenham funções imunitárias importantes. Ao longo desta tese permutaremos entre as duas designações (glóbulos brancos / leucócitos).

<sup>3</sup> Também designadas por glóbulos vermelhos ou eritrócitos, as hemácias têm como tarefa principal a distribuição de oxigénio e substâncias nutritivas pelo organismo, além da remoção de produtos de degradação.

neutrófilo, linfócito e monócito. Contudo, existem situações anómalas nas quais podem ser visualizados no sangue de um paciente outros tipos de leucócitos. Os leucócitos são produzidos na sua maior parte pela medula óssea e aí permanecem durante um período de amadurecimento até serem difundidos para a circulação sanguínea. Porém, em determinadas afecções estes podem, ainda imaturos, ser lançados na circulação. Alguns destes jovens leucócitos têm o nome de mieloblastos.

Os tipos de análises ao sangue executados nos actuais laboratórios de análises clínicas são inúmeros, encontrando-se quase todos mais ou menos automatizados. Uma das análises realizadas nestes laboratórios consiste na diferenciação percentual dos diversos tipos de leucócitos existentes numa amostra de sangue. Na Tabela 1.1 podemos constatar os valores percentuais típicos de uma análise deste teor a uma pessoa sã. Porém, até há bem pouco tempo, o equipamento existente no mercado dedicado a esta área da hematologia apenas estava capacitado para fazer a diferenciação de três tipos de leucócitos. Recentemente surgiram no mercado máquinas capazes de diferenciar as cinco classes normais de leucócitos. No entanto, as restantes, como é o caso dos mieloblastos, continuam a ter que ser discriminadas por acções visuais, executadas por pessoal especializado. Estas acções consistem na contagem de cem leucócitos através de um microscópio óptico e, à medida que se vai procedendo à identificação, vai-se registando a quantidade de leucócitos de um e de outro tipo, obtendo-se assim a relação percentual entre os glóbulos brancos encontrados. Todavia, este trabalho é muito moroso e cansativo. Assim, pretendemos aduzir nesta tese soluções que permitem a elaboração de um sistema que seja capaz de diferenciar da forma mais autónoma possível, a partir de imagens sanguíneas adquiridas por um microscópio, as diversas classes de leucócitos, inclusive os mieloblastos.

---

<sup>4</sup> Também conhecidas por trombócitos, participam na coagulação sanguínea.

**Tabela 1.1** Percentagens de leucócitos existentes no sangue de uma pessoa sã.

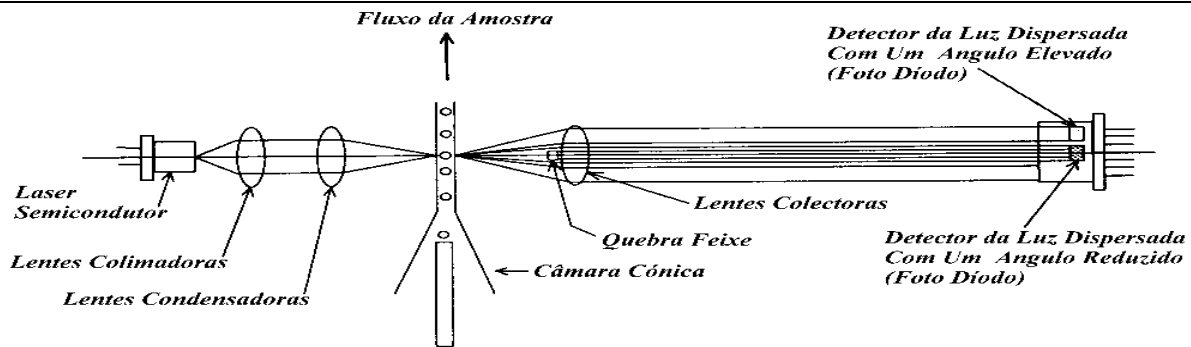
| Leucócitos |             |           |             |            |           |              |
|------------|-------------|-----------|-------------|------------|-----------|--------------|
|            | Eosinófilos | Basófilos | Neutrófilos | Linfócitos | Monócitos | Mieloblastos |
| %          | 2.5         | 0.5       | 52.0        | 40.0       | 5.0       | —            |

### **1.1 Técnicas actualmente existentes para resolver este problema, suas vantagens e desvantagens**

Como já referimos, os automatismos actuais não vão além da diferenciação de cinco classes de leucócitos: basófilos, eosinófilos, neutrófilos, linfócitos e monócitos. Esta limitação fica a dever-se ao princípio usado na sua diferenciação. Esse princípio recorre ao uso de reagentes químicos para inibir as células que não interessam distinguir e a um mecanismo de detecção baseado na interacção da luz com as células.

No equipamento que analisámos, o SF-3000 da Sysmex, este mecanismo dava pelo nome de Citometria de Fluxo [Bessman, 1988]. O funcionamento geral é o seguinte: depois de uma amostra de sangue ser aspirada e diluída numa porção de reagente, é injectada numa câmara cónica especial que tem como função manter uniformemente constante a densidade de células por unidade de volume, de maneira a aumentar a precisão na contagem diferencial e o seu grau de repetibilidade. Em seguida, as células sanguíneas são atingidas por um feixe de luz *laser* originando uma dispersão luminosa cuja intensidade é proporcional ao tamanho dos leucócitos. Um pequeno ângulo de dispersão luminosa, entre 1-6°, fornece informação sobre o tamanho das células; uma dispersão luminosa com um ângulo entre 8-20° proporciona informação acerca da densidade intracelular e do tamanho do núcleo. Estes feixes de luz resultantes da dispersão são projectados num dispositivo semiconductor que os converterá em sinais eléctricos a serem posteriormente tratados e sintetizados. Este mecanismo encontra-se esquematizado na figura. 1.1.

**Figura 1.1** Sistema de Citometria de Fluxo para diferenciação de leucócitos usado no Sysmex SF-3000.

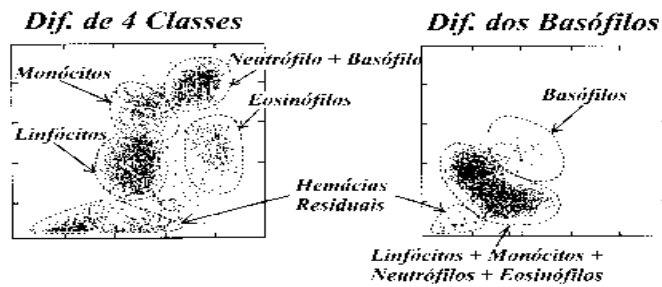


A informação adquirida é então mapeada num sistema de duas coordenadas em que um dos eixos representa o tamanho das células e o outro a sua densidade (Fig. 1.3). Isto permite a criação de grupos representativos das classes de leucócitos ou conjunto de classes (Fig. 1.2). No equipamento (SF-3000-Sysmex) que nós estudámos, a diferenciação das cinco classes de leucócitos é feita pelo conjunto de duas acções. Na primeira acção é adicionado um reagente específico que vai permitir a diferenciação dos monócitos, linfócitos, eosinófilos e de um grupo heterogéneo que engloba os neutrófilos e os basófilos. Daí a necessidade da segunda acção, pois esta irá permitir a diferenciação dos basófilos e por consequência a dos neutrófilos. Estas duas acções diferem apenas no tipo de reagente misturado na amostra de sangue. Na figura 1.2 podemos observar os diferentes grupos originados pelas duas acções no plano tamanho/densidade. Podemos verificar aqui a importância dos reagentes que vêm cobrir a lacuna motivada pela pequena dimensão do vector características de tamanho e densidade. A este facto está associada a limitação de glóbulos brancos possíveis de classificar, pois o investimento no estudo e fabricação de outros reagentes capazes de possibilitar a diferenciação de outras classes de leucócitos pode não ser justificado devido à escassez de casos com classes anormais de glóbulos brancos.

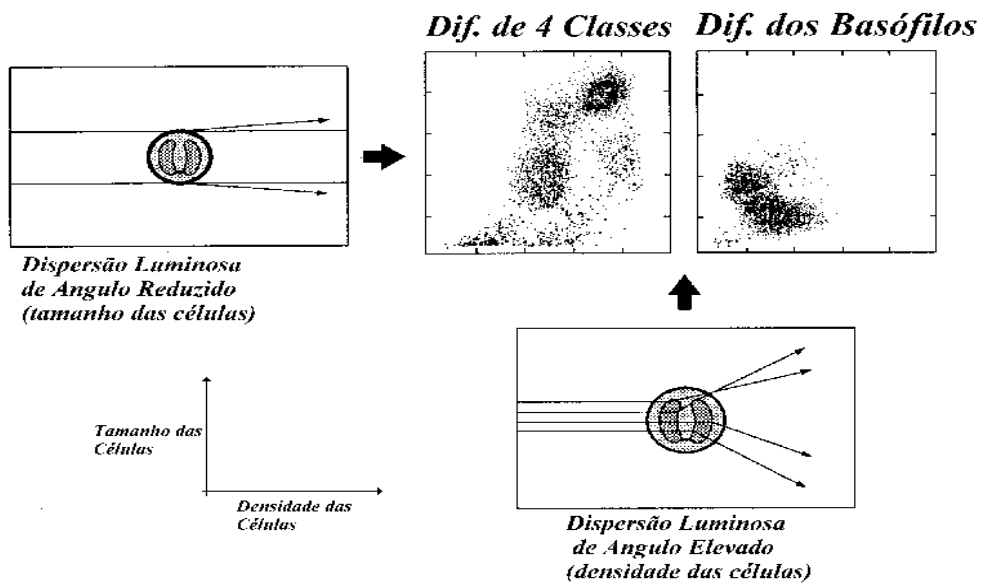
O tempo de execução da análise está relacionado com o período de reacção dos reagentes químicos usados, que é da ordem dos 8.5-segundos para a primeira acção e 31.5-segundos para a segunda. Trata-se assim de um sistema eficiente em termos de precisão e rapidez, mas, como já referimos, limitado no número de leucócitos que pode distinguir.

A recorrência ao uso de reagentes encarece significativamente a manutenção do sistema, sendo este também um senão a acrescentar ao referido atrás.

**Figura 1.2** Grupos representativos das classes de leucócitos sintetizados a partir da informação da dispersão luminosa. Uma intensidade mais escura representa uma quantidade de leucócitos superior no respectivo grupo.



**Figura 1.3** Construção de mapas bidimensionais que permitem classificar os leucócitos.



## **1.2 Objectivos do trabalho**

Como vimos na parte introdutória deste capítulo, a diferenciação automática de leucócitos é o cerne deste trabalho. De uma forma geral, os diversos passos do processo consistem no seguinte: depois de ser feita a colheita de sangue a um indivíduo, esta é centrifugada<sup>5</sup> de modo a separar o volume plasmático do volume celular. Este, por sua vez, é adicionado a um composto químico a que se dá o nome de *corante*. Esta aglutinação será espalhada numa lâmina de vidro, dando origem a um esfregaço seco que irá permitir que as células sejam visualizadas num microscópio óptico. Este primeiro procedimento é externo ao nosso sistema, de maneira que, ainda que possível de realizar em trabalho futuro, não constitui nesta tese objecto de automação. No passo seguinte a lâmina é colocada num microscópio óptico, as imagens são então obtidas por uma câmara de vídeo CCD<sup>6</sup> a cores e transferidas para um PC onde uma carta digitalizadora as converte e as entrega ao nosso software. As imagens adquiridas apresentam informação visual dos constituintes a nível celular. Glóbulos brancos, glóbulos vermelhos e plaquetas sanguíneas são algumas das células possíveis de encontrar numa imagem deste tipo. Uma vez que as células que nos interessam classificar são os glóbulos brancos, teremos de dispor de um processo que os detecte e isole na imagem, por forma a que possam ser posteriormente classificados. Este processo toma o nome de segmentação<sup>7</sup>. A natureza visual existente neste género de imagens pode ser observada na figura 1.4. Repare-se que, tanto numa como noutra imagem, a célula central representa um eosinófilo circundado por glóbulos vermelhos. O primeiro problema significativo com que nos deparamos encontra-se na grande variabilidade em termos de cor que este tipo de imagens apresenta; isto acarreta dificuldades no processo de segmentação dos glóbulos brancos. A questão põe-se da seguinte forma: como determinar quais os pixels que constituem o leucócito e quais os que não o constituem? Esta decisão, pelo motivo que já referimos atrás, não pode ser tomada unicamente através da luminância e crominância do próprio pixel, mas terá também de ser tomada com base na distribuição espacial dos pixels vizinhos conexos e não

---

<sup>5</sup> Depois de inserir o sangue num tubo de ensaio, este é colocado numa centrifugadora a alta rotação, para que, por acção da força centrífuga daí resultante, o plasma seja separado das componentes celulares, pois as densidades destes dois líquidos são distintas.

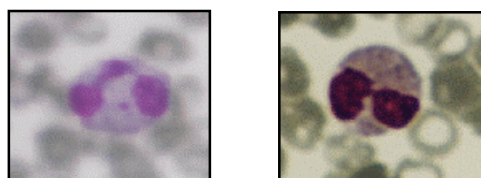
<sup>6</sup> Terminologia vulgarmente usada para designar câmaras de vídeo que utilizam como órgão detector de imagem um CCD Charged-Coupled Device.

conexos. De facto, encontrar um método que reconheça a pertença ou não de um pixel a um leucócito é uma tarefa que, se não impossível, é, no mínimo, custosa. Por estes motivos o nosso trabalho recai sobre o uso de redes neuronais. Estas apresentam a vantagem de poderem aprender a partir de exemplos. Temos assim como primeiro objectivo o desenvolvimento e implementação computacional de uma técnica, que, servindo-se de redes neuronais, consiga, de uma forma autónoma, segmentar as múltiplas classes de leucócitos existentes nas imagens.

---

**Figura 1.4** Imagens de leucócitos obtidas num microscópio óptico com uma ampliação de 40×10.

---



---

Depois de os glóbulos brancos existentes numa imagem terem sido detectados pelo processo de segmentação, necessitamos de os classificar e contar. Este trabalho é executado pelo processo de classificação, que terá de identificar correctamente as diversas classes de glóbulos brancos. Se observarmos os dois eosinófilos nas imagens da figura 1.4, concluimos que a mutabilidade geométrica entre as duas células é elevada, tornando evidente a subtileza das características visuais que as definem. Também aqui voltamos a ter dificuldade em descrever e transferir para um algoritmo o conhecimento presente no nosso cérebro, que nos permite diferenciar as diversas classes de leucócitos. Novamente recorreremos ao emprego de redes neuronais, de forma a que estas possam aprender as ténues características geométricas que cada classe de leucócitos possui e consequentemente classificar correctamente os glóbulos brancos. Sem dúvida que, seja qual for o processo usado, o sucesso da classificação

---

<sup>7</sup> Designação geralmente dada às técnicas de extracção de regiões de interesse em imagens.

depende em grande medida da qualidade da segmentação, estando condenada à partida uma tentativa de classificação de uma região que tenha sido parcial ou equivocadamente segmentada.

Assim, o nosso segundo objectivo é o estudo e a elaboração de um mecanismo de classificação de leucócitos, baseado em redes neuronais. Temos então como objectivo global o desenvolvimento e implementação, recorrendo a suportes computacionais, de um sistema que a partir de imagens sanguíneas execute autonomamente a diferenciação percentual dos leucócitos presentes nessas imagens. Devemos referir que todo o desenvolvimento que fizemos sobre os dois processos, segmentação e classificação, foi sempre no sentido de ambos poderem aprender a segmentar e classificar respectivamente qualquer das classes de leucócitos existentes no sangue. No entanto, por serem relativamente escassos os casos clínicos que implicam a existência de leucócitos anormais no sangue, não nos foi possível treinar o nosso sistema com essas classes de células. Daí que tenhamos restringido o treino do nosso sistema às cinco classes de leucócitos normais. De qualquer forma, e mediante os resultados que obtivemos com as cinco classes, somos levados a pensar que com mais ou menos dificuldades o sistema poderia aprender a segmentar e classificar outras classes.

### **1.3 Aspecto geométrico das células**

Vamos agora olhar para as principais características geométricas que mais sobressaem quando observamos as várias classes de leucócitos. Na figura 1.5 temos expostas as cinco classes de leucócitos normais. Dentro destes cinco tipos de leucócitos podemos agrupar como granulócitos os basófilos (a), os eosinófilos (b) e os neutrófilos (c), uma vez que apresentam na sua superfície uma textura granular; enquanto os linfócitos (d) e os monócitos (e), pelo motivo inverso, são agrupados como agranulócitos. Esta característica importante na diferenciação dos leucócitos é frequentemente perdida durante a aquisição das imagens, o que faz com que se tenha de dar mais ênfase a outros atributos. Por exemplo, o linfócito (d) é facilmente distinguível dos outros pelo seu tamanho e pela tonalidade uniforme e escura do seu núcleo, não exibindo citoplasma. O monócito (e) é caracterizado por um núcleo algo regular. O neutrófilo (c) exhibe um núcleo particionado irregular e uma área de citoplasma considerável. O núcleo do eosinófilo também é particionado mas menos irregular que o

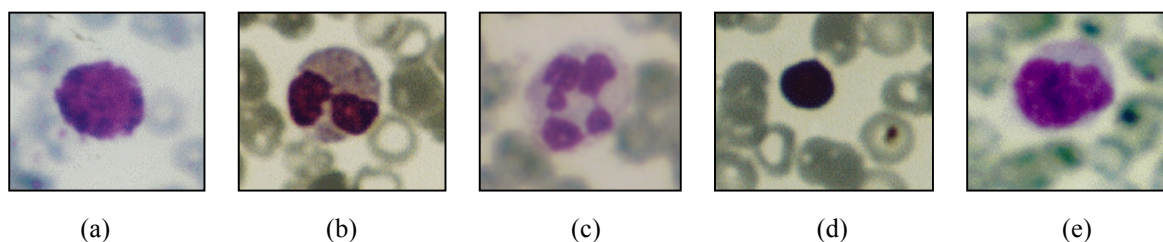
núcleo do neutrófilo e a granulação do seu citoplasma é mais evidente que no neutrófilo. O basófilo apresenta um núcleo denso, não tem citoplasma e, ao contrário do linfócito, expõe uma granulação intensa em toda a sua superfície.

Como podemos constatar, estas caracterizações acabam por ser algo vagas. De facto, por vezes, a nós próprios ou mesmo a especialistas com anos de experiência surgem dificuldades enormes em classificar correctamente algumas células que aparecem nas imagens. Nesta situação é mais importante não classificar do que classificar erradamente, uma vez que se está a executar uma contagem relativa e não uma contagem absoluta do número de leucócitos.

---

**Figura 1.5** Imagens de Leucócitos. (a) Basófilo; (b) Eosinófilo; (c) Neutrófilo; (d) Linfócito; (e) Monócito.

---



## **1.4 Características do ambiente de desenvolvimento**

### **1.4.1 Hardware**

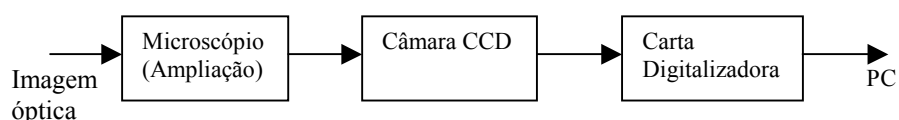
O suporte físico utilizado no desenvolvimento deste sistema divide-se em dois grupos. O primeiro agrupa os dispositivos necessários à aquisição das imagens (Fig. 1.6); o segundo alberga os algoritmos neuronais e de processamento de imagem; trata-se de um vulgar PC que serve perfeitamente para demonstrar o funcionamento do protótipo. No primeiro conjunto podemos encontrar como dispositivo de partida o microscópio óptico, órgão fundamental na qualidade das imagens recolhidas. Acoplado ao microscópio temos ainda a câmara de vídeo a

cores CCD, responsável por converter as imagens ampliadas pelo microscópio em sinais eléctricos do tipo Y/C<sup>8</sup>. Este sinal é injectado numa carta de digitalização de imagens que se encontra inserida num slot PCI do PC.

---

**Figura 1.6** Diagrama de blocos da aquisição de imagens neste protótipo.

---



#### **Hardware utilizado:**

##### **Microscópio Óptico**

- Nikon Labophot;
- Triocular;
- 5 Objectivas;
- Ampliação Max. - 100×10.

##### **Câmara CCD**

- Sony - SSC-DC50P;
- Resolução Horiz. (PAL) - 470 Linhas;
- Pixels (CCD) - 752(H)×582(V).

##### **Cartas de Aquisição**

- Matrox - Meteor;
- MRT Video Port.

---

<sup>8</sup> Y/C Luminância e Crominância separadas. Técnica usada na codificação de imagens de vídeo a cores, que tem como objectivo melhorar a qualidade das imagens em termos de crominância relativamente à vulgar codificação

## PC

- Intel - PII-300Mhz;
- 128Mbytes - RAM.

### 1.4.2 Software

O software usado no desenvolvimento das rotinas integradas no protótipo foi escrito na linguagem de programação orientada ao objecto C++. O compilador utilizado foi o Visual C++ 6.0 da Microsoft. A interface com o utilizador é do tipo gráfica por janelas.

A escolha desta linguagem para a implementação do sistema deveu-se essencialmente à grande estruturabilidade e reutilidade desta, permitindo assim facilmente a portabilidade para outros sistemas operativos. Trata-se de uma linguagem poderosa que permite decompor facilmente um problema em subgrupos de partes relacionadas a que chamamos objectos. Esta característica torna os programas mais claros e evolutivos, facilitando o desenvolvimento do software. O sistema é constituído por dois módulos: um módulo tem como função o treino da rede neuronal *backpropagation*; o outro módulo constitui o sistema de segmentação e classificação de leucócitos.

## 1.5 Organização da tese

Esta tese encontra-se dividida em 6 capítulos e dois apêndices.

No segundo capítulo é feita uma apresentação sobre a rede neuronal que irá ser usada nos mecanismos de segmentação e classificação dos leucócitos. Tendo-nos particularizado numa rede neuronal sem realimentação (*feedforward*), são referidos, explicados e discutidos os seus constituintes e os mecanismos de aprendizagem neuronal que lhe foram associados.

Em paralelo são também emitidas algumas ideias sobre a adequação desses mecanismos neuronais ao nosso problema.

O terceiro capítulo começa por mostrar em pormenor as dificuldades inerentes à acção de segmentação, para de seguida serem descritos os maquinismos por nós desenvolvidos para a realização dessa acção. A integração da rede neuronal no processo, uma discussão sobre o sistema de cor a utilizar e a forma de gerar os padrões de treino para a aprendizagem neuronal são alguns dos pontos focados neste capítulo.

O quarto capítulo é dedicado ao funcionamento do mecanismo de classificação. Também este capítulo é iniciado com a exposição detalhada dos abrolhos deste processo. Segue-se uma descrição e justificação dos constituintes usados na elaboração deste maquinismo. São também explicados e discutidos os descritores geométricos integrados neste processo. É explanado o mecanismo de separação citoplasma-núcleo. Por fim, é feita a apresentação de uma série de pontos relacionados com o classificador neuronal, como o modo de treino, decisão final de classificação, etc.

Com o quinto capítulo são mostrados os resultados obtidos com os métodos implementados (segmentação e classificação). Os resultados da segmentação são apresentados e discutidos com base nas imagens sanguíneas processadas pelo nosso protótipo e nas quais se podem observar as capacidades deste mecanismo a detectar as regiões leucócito. Os resultados conseguidos com o processo de classificação serão neste capítulo expostos recorrendo a uma matriz de confusão.

No sexto e último capítulo são apresentadas as conclusões finais para o processo de segmentação e para o processo de classificação, bem como os principais pontos de execução futura para a continuidade do trabalho.

No apêndice A é mostrada e explicada a estrutura de classes usada para a implementação da rede neuronal em C++, enquanto no apêndice B é explicada a dedução matemática da *regra delta generalizada*.

## Capítulo 2

# Redes neuronais aplicadas à análise de imagem

### 2.1 Introdução

Neste capítulo faremos um estudo e justificação dos diversos mecanismos constituintes de uma rede neuronal com uma topologia *feedforward*, já que este tipo de rede foi o escolhido durante a fase de investigação e procura de técnicas a adoptar para a concretização do sistema de diferenciação de leucócitos. Devemos referir ainda que as técnicas neuronais que vamos descrever serão apenas aquelas que foram ensaiadas por nós e que mostraram possível aptidão na resolução do problema que é tema desta tese.

De uma forma geral, a procura de técnicas que permitissem a segmentação e a classificação eficientes dos glóbulos brancos passou pelo estudo de matérias que vão desde a lógica *fuzzy* [Heske & Heske, 1996], *neuro-fuzzy* [Nauck & Klawonn & Kruse, 1997], redes neuronais *pulsed-coupled* [Lindblad & Kinser, 1998] a ferramentas clássicas de processamento de imagem. Decidimos então adaptar e integrar as redes neuronais no nosso sistema, fundamentalmente devido à sua capacidade de aprendizagem, generalização e relativa facilidade de implementação. Esta escolha foi feita de uma forma mais ou menos empírica, uma vez que seria temporalmente impossível implementar e testar todas as técnicas por nós estudadas.

Como já referimos, as redes neuronais irão ser usadas tanto no problema da segmentação como no da classificação dos leucócitos. Nos capítulos 3 e 4 descreveremos o modo de integração das redes neuronais nestes processos. Neste capítulo trataremos da rede neuronal *backpropagation*.

A partir do momento em que nascemos, inicia-se uma incessante interacção entre nós e o meio que nos rodeia, resultando numa constante aprendizagem que abrange desde as situações mais triviais às mais complexas. Nos primeiros anos de vida aprendemos de uma forma inconsciente a classificar correctamente objectos; por exemplo, no caso de uma mesa, sabemos obviamente que existem mesas de variadíssimos formatos. No entanto, conseguimos perfeitamente substantivá-las como tal, mesmo que se trate de uma que nunca antes tenhamos observado. Isto significa que o nosso cérebro conseguiu aprender os atributos gerais que caracterizam uma mesa e que recusou os atributos específicos que estão presentes em cada tipo de mesa. Para nós, executar este tipo de tarefa é fácil, mas, e para uma máquina?... Como é que poderíamos ensiná-la a aprender e posteriormente a tomar decisões com base nessa aprendizagem? É com esta antiga e humana ideia de fazer máquinas *pensantes* que surgem as redes neuronais.

De facto, não se trata de uma pretensão meramente fútil. Máquinas com capacidades cerebrais humanas encontram aplicabilidade em múltiplos sectores: no reconhecimento de voz, síntese de voz, reconhecimento de padrões, no controlo dinâmico de sistemas, na medicina, na previsão de séries financeiras, etc..

Como dissemos, as redes neuronais artificiais tiveram a sua origem na tentativa de o homem perceber e simular o funcionamento do cérebro humano. Essa cruzada iniciou-se por volta dos anos 40 e foi fortemente acelerada pelo desenvolvimento de suportes físicos tais como computadores, VLSI<sup>9</sup>, ASICS<sup>10</sup>, que permitem a emulação eficaz de tais redes, e consequentemente pelas descobertas de novos e mais eficientes modelos de aprendizagem neuronal. Estes modelos não são mais do que representações matemáticas com características adaptativas que permitem o processamento da informação. De notar ainda que muitos destes modelos neuronais tiveram inspiração directa no que se conhece das estruturas e funcionamento das redes neuronais biológicas. Desta forma, o sucesso e a eficiência, a todos os níveis, de algumas redes artificiais encontram-se também intimamente ligados (condicionados) às descobertas científicas no campo da neurobiologia. De qualquer forma, há que ter em conta que os circuitos neuronais que hoje conseguimos compreender e implementar estão ainda muito distantes das complexas estruturas cerebrais, que formadas por

---

<sup>9</sup> Very Large Scale Integration.

<sup>10</sup> Application Specific Integrated Circuit.

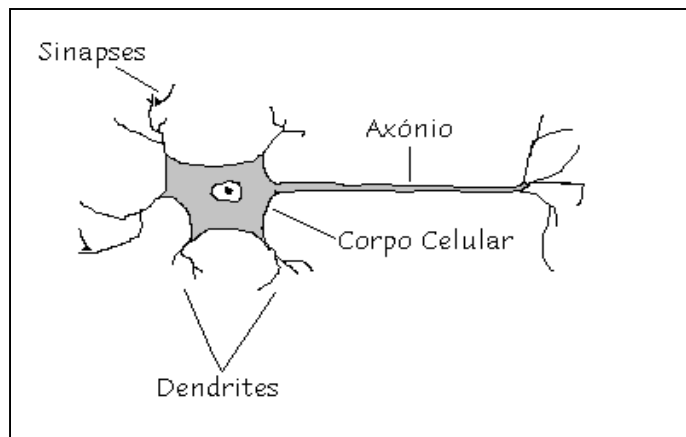
cerca de 10 biliões de neurónios e 10 triliões [Damásio, 1995] de sinapses (conexões) levam alguns investigadores a duvidar se algum dia conseguiremos interpretar e compreender na totalidade o cérebro humano. Existem, no entanto, outros que se mostram mais optimistas, ostentando que, apesar da aparente complexidade dos circuitos neuronais, um dia seremos capazes de perceber o seu funcionamento; e justificam-no da seguinte forma: apesar de estarmos a falar de uma super rede de 10 biliões de neurónios e de cada neurónio ter em média 1000 sinapses, cada neurónio está interligado com um pequeno grupo de outros neurónios e não com os restantes grupos. Sendo assim, a actividade de um neurónio está condicionada ao conjunto de neurónios vizinhos. Por sua vez, estes grupos interagem com outros grupos cercanos formando sistemas. A localização destes grupos no sistema determina a sua participação na função do sistema. Assim, e em última instância, o cérebro é um aglomerado destes sistemas. Desta forma hierárquica o funcionamento do cérebro poderá ser um dia compreensível.

## **2.2 Redes Neuronais Biológicas (constituintes)**

As redes neuronais que formam o cérebro são constituídas por neurónios. Estas células são a base de toda e qualquer actividade cerebral. O seu tipo de organização e funcionamento nas diferentes regiões do cérebro origina actividades tão vulgares e complexas como as motoras ou as da visão. As componentes principais de um neurónio são o corpo celular, as dendrites e o axónio (Fig. 2.1).

**Figura 2.1** Desenho genérico de um neurónio humano com os seus constituintes principais: Axónio, corpo celular e dendrites. Pode ainda observar-se as sinapses, que proporcionam as conexões entre o axónio de um neurónio e as dendrites de outros neurónios.

---



Quando um neurónio se torna activo (dispara), é propagada uma corrente eléctrica a partir do seu corpo celular e ao longo do axónio. Quando esta corrente atinge as sinapses que proporcionam a junção entre o axónio de um neurónio e as dendrites de outros, desencadeia-se a libertação de substâncias químicas conhecidas por neurotransmissores. Estes, por sua vez e dependendo do tipo de sinapse (estimuladora ou inibidora), vão determinar o grau de transmissão de um impulso através da respectiva dendrite para o neurónio seguinte. O conjunto de impulsos que chegam ao corpo celular do neurónio seguinte faz com que mecanismos internos do corpo celular decidam com base nestes impulsos se irá, ou não, ser disparado um novo impulso ao longo do axónio deste neurónio. É este o mecanismo base do funcionamento neuronal. Muitas das características funcionais proporcionadas pelos neurónios cerebrais foram transpostas para os neurónios artificiais, tornando-se estas fundamentais no seu desempenho. As mais comuns são<sup>11</sup>:

- A unidade de processamento pode receber muitos sinais de outros neurónios, em analogia com o corpo celular dos neurónios cerebrais.

- A intensidade com que os sinais chegam aos neurónios é determinada por parâmetros numéricos a que chamamos pesos. Esta função, no caso dos neurónios biológicos, é desempenhada pelas sinapses.
- A unidade de processamento opera numericamente os sinais que recebe, sendo a soma a operação mais comum.
- Em determinadas circunstâncias, dependendo do resultado obtido na computação dos sinais de entrada, a saída do neurónio sofre uma alteração do seu valor; paralelamente o neurónio cerebral faz propagar um sinal eléctrico através do seu axónio.
- A saída de um neurónio pode estar conectada a muitos outros neurónios.
- A representação do conhecimento reside nos pesos ou nas sinapses no caso dos biológicos.
- O valor dos pesos pode ser alterado durante a fase de aprendizagem, de forma a adquirir um determinado conhecimento.
- Tolerância a ruídos. As redes biológicas são capazes de processar informação correctamente, mesmo que esta tenha sofrido alguma alteração relativamente à informação apresentada na fase de aprendizagem. Esta vantagem também é verificável nas redes artificiais.
- Versatilidade na possibilidade de aprender. Tal aspecto pode ser observado em situações de disfunções em certas regiões cerebrais que passam a ser executadas por outras regiões que substituem funcionalmente as infuncionais. Também as redes artificiais apresentam simetrialidade [Bishop, 1995] na representação do conhecimento, ou seja podem representar um mesmo conhecimento em variadíssimos arranjos em termos de pesos (sinapses).

---

<sup>11</sup> As características aqui apresentadas baseiam-se em [Fausett, 1994].

### 2.3 Redes Neuronais Artificiais

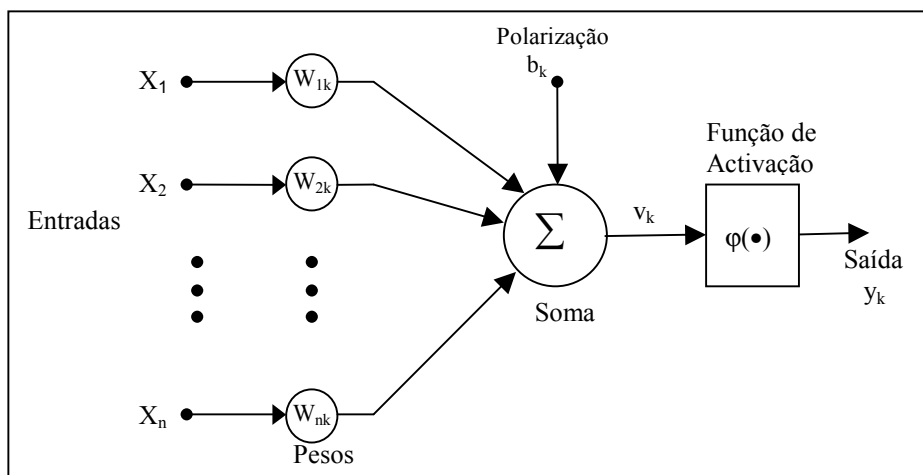
Já vimos que as redes neuronais são formadas por neurónios. Vamos agora ver como é que podemos imitar artificialmente o funcionamento de um neurónio biológico.

Na figura 2.2 podemos ver as três partes que constituem um modelo de um neurónio artificial típico. São elas: o conjunto de pesos, análogo às sinapses biológicas, a unidade de soma e a unidade de activação. Estas duas últimas podem ser integradas numa só unidade de processamento.

Cada sinal injectado numa determinada entrada  $x_j$  de um neurónio  $k$  é multiplicado por um peso  $w_{jk}$ . O resultado deste produto é somado pela unidade de soma aos  $n$  restantes produtos, da seguinte forma:

$$u_k = \sum_{j=1}^n W_{jk} x_j \quad (2.1)$$

**Figura 2.2** Modelo de um neurónio, constituído por um conjunto de pesos, uma unidade de soma e uma função de activação.



O valor obtido pela unidade de soma é sujeito a uma função geralmente não linear, chamada função de activação, que limita superior e inferiormente o valor obtido na unidade soma. Os valores proporcionados pela função de activação são vulgarmente limitados a intervalos de  $[0, 1]$  ou  $[-1, 1]$ . Note-se que o modelo da figura 2.2 possui também um elemento de polarização (*bias*) que permite adequar o valor gerado pelos sinais de entrada ao valor de transição da função activação. Repare-se na equação 2.2 que traduz a soma de  $u_k$  com o valor de polarização  $b_k$ . O valor resultante  $v_k$  é o argumento da função activação  $\varphi(\cdot)$ .

$$v_k = u_k + b_k \quad (2.2)$$

O valor de saída,  $y_k$ , é então dado pela equação :

$$y_k = \varphi(v_k) \quad (2.3)$$

Fazendo as necessárias substituições podemos apresentar a equação geral que traduz a computação de um neurónio deste tipo:

$$y_k = \varphi\left(\sum_{j=1}^n W_{jk} x_j + b_k\right) \quad (2.4)$$

### 2.3.1 Funções de activação

Vamos agora ver os vários tipos de funções de activação mais usados em redes neuronais. Algumas delas foram implementadas e usadas no nosso protótipo com vista a conseguir discriminar a mais eficaz no âmbito da nossa aplicação.

Consideremos uma função  $\varphi(v)$ , definida da seguinte forma:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (2.5)$$

Este tipo de função de activação adquire o nome de função binária (*threshold*). No entanto, esta função não pode ser usada com redes *backpropagation*, já que este método de treino exige que a função de activação seja diferenciável em todo o seu domínio, o que não é o caso desta.

Existe um tipo de função normalmente usado em problemas ditos de regressão. Trata-se da função identidade (equação 2.6). Os problemas de regressão podem beneficiar, pelo menos na fase de treino, do uso deste tipo de função, pois os valores pretendidos na saída da rede neuronal podem variar continuamente em vez de tomarem somente dois valores, como é o caso das saídas bipolares ou binárias. Porém, normalmente, só os neurónios de saída são constituídos com esta função, já que uma rede neuronal que contenha apenas neurónios com funções de activação identidade não consegue aprender a resolver certos problemas. Por exemplo, não pode categorizar classes que se encontrem separadas por mais do que um hiperplano<sup>12</sup> no espaço de classificação, conforme provado por [Minsky & Papert, 1969], provando-se também que uma rede neuronal composta por funções deste tipo e com várias camadas pode ser sempre estruturada numa rede de uma só camada.

A função identidade é a seguinte:

$$\varphi(v) = v. \quad (2.6)$$

Um tipo de função de activação muito usado é a sigmóide assimétrica descrita pela equação 2.7., que apresenta um contradomínio no intervalo de [0, 1]. O parâmetro  $\sigma$ , existente no denominador da função, permite ajustar a *rapidez* com que a função transita do limite inferior para o limite superior.

---

<sup>12</sup> Generalizando, designaremos, ao longo desta tese, por hiperplano as fronteiras limitadoras das classes no espaço de classificação, independentemente da sua dimensão.

$$\varphi(v) = \frac{1}{1 + \exp(-\sigma v)} \quad (2.7)$$

Alguns algoritmos de treino neuronal, como o *backpropagation*, exigem que a função de activação seja diferenciável e monótona. A função sigmóide contempla estas características e, além disso, a sua derivada pode ser apresentada na seguinte forma, o que a torna menos pesada em termos computacionais, já que reutiliza o valor calculado na função  $\varphi$  no ponto  $v$ :

$$\varphi'(v) = \sigma \varphi(v) [1 - \varphi(v)] \quad (2.8)$$

As funções de activação podem ser facilmente escalonadas em termos do seu domínio de saída, de forma a adequarem-se a uma determinada aplicação. Por exemplo, a sigmóide bipolar apresenta um contradomínio no intervalo de  $[-1, 1]$ . A equação seguinte representa-a:

$$\varphi(v) = \frac{2}{1 + \exp(-\sigma v)} - 1 = \frac{1 - \exp(-\sigma v)}{1 + \exp(-\sigma v)} \quad (2.9)$$

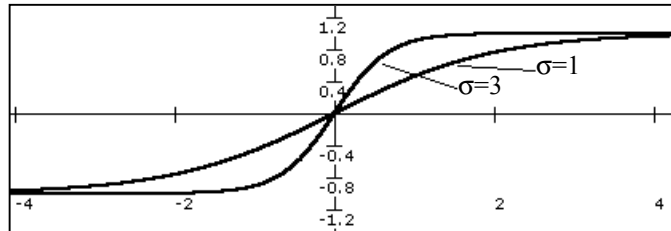
sendo a sua derivada:

$$\varphi'(v) = \frac{\sigma}{2} [1 + \varphi(v)] [1 - \varphi(v)] \quad (2.10)$$

Na figura 2.3 podemos comparar os gráficos de duas sigmóides bipolares com distintos valores para o parâmetro *sigma* ( $\sigma$ ).

**Figura 2.3** Representação gráfica de duas funções sigmóide bipolar, uma com *sigma* igual a 1 e a outra com *sigma* igual a 3.

---



Uma outra função semelhante à sigmóide bipolar é a tangente hiperbólica. Esta função é exactamente igual à sigmóide com *sigma* a valer dois:

$$\varphi(v) = \frac{\exp(v) - \exp(-v)}{\exp(v) + \exp(-v)} = \frac{1 - \exp(-2v)}{1 + \exp(-2v)} \quad (2.11)$$

A derivada da tangente hiperbólica é :

$$\varphi'(v) = [1 + \varphi(v)] [1 - \varphi(v)] \quad (2.12)$$

Embora estas duas últimas funções não lineares sejam as mais usadas nas redes *feedforward*, que utilizam o algoritmo de treino *backpropagation*, as outras funções podem também ser usadas dependendo do seu desempenho em cada tipo de problema. Porém, de acordo com a experiência que obtivemos com a nossa aplicação, os resultados não variaram muito com o tipo de função de activação, tendo a tangente hiperbólica e a sigmóide bipolar registado de uma forma pouco sensível os melhores desempenhos.

Uma outra função usada como activação é a arcotangente. Esta função atinge mais lentamente os seus valores assintóticos do que a tangente-hiperbólica ou do que a sigmóide bipolar.

A função arcotangente é representada da seguinte forma:

$$\varphi(v) = \frac{2}{\pi} \arctan(v) \quad (2.13)$$

A derivada da função arcotangente é:

$$\varphi'(v) = \frac{2}{\pi} \frac{1}{1+v^2} \quad (2.14)$$

Dissemos atrás que determinados problemas apelidados de regressivos podem ser favorecidos pelo uso de funções lineares na última camada [Bishop, 1995]. Porém, o uso de funções, que não tendem assintoticamente para um valor constante, mas que apresentam um certo grau de não linearidade, pode também ajudar na resolução de problemas não-regressivos [Fausett, 1994]. É o caso da seguinte função apelidada de não-saturação e definida assim:

$$\varphi(v) = \begin{cases} \log(1+v) & \text{se } v \geq 0 \\ -\log(1-v) & \text{se } v < 0 \end{cases} \quad (2.15)$$

e tendo como sua derivada:

$$\varphi'(v) = \begin{cases} \frac{1}{1+v} & \text{se } v \geq 0 \\ \frac{1}{1-v} & \text{se } v < 0 \end{cases} \quad (2.16)$$

### 2.3.2 Arquitecturas de redes neuronais

Uma rede neuronal<sup>13</sup> é tipicamente formada por neurónios conectados entre si através de pesos. Os pesos representam o nível de inibição ou não dos sinais que são propagados de uns neurónios para os outros. É durante a fase de treino que estes parâmetros são alterados por forma a obter o desempenho correcto da rede; ou, dito de outra forma, é na combinação destes valores que irá ficar representado um determinado conhecimento aprendido pela rede durante o treino.

As estruturas organizacionais dos elementos de uma rede tipificam a sua arquitectura. Os neurónios são unidades que processam sinais que lhes chegam por meio das ligações e que são provenientes de padrões de entrada ou das saídas de outros neurónios. É vulgar agrupar os neurónios de uma rede neuronal em camadas, podendo resultar em uma das duas tipologias seguintes: em redes do tipo multi-camada ou em redes de camada única. As redes de camada única são constituídas pela camada de neurónios de entrada e pela camada de neurónios de saída. No entanto, a camada de entrada não é normalmente contabilizada, pois não executa qualquer tipo de computação, apenas existindo como órgão receptor dos sinais de entrada. Assim, as redes multi-camada possuem pelo menos uma camada que se situa entre a camada de neurónios de entrada e a camada de neurónios de saída. Estas considerações aplicam-se normalmente às redes *feedforward*, embora possam ser adaptadas a outras redes.

Uma outra característica que também pode definir a arquitectura de uma rede neuronal é a direcção com que os sinais são transmitidos de um neurónio para outro ou para ele mesmo. Por exemplo, no caso das redes *feedforward* a direcção dos sinais é única e sempre da camada de entrada para a de saída. Outras há, como as BAM<sup>14</sup> [Kosko, 1988], nas quais os sinais se propagam nos dois sentidos, tanto da camada de saída para a entrada como da camada de entrada para a saída; ou como as redes competitivas em que todos os neurónios podem estar ligados a todos, inclusive podem possuir *loops* de pesos para eles mesmos. Geralmente, uma rede do tipo *feedforward* apresenta cada neurónio de uma determinada camada ligado através de pesos a cada um dos outros da camada anterior, não existindo ligações entre neurónios de uma mesma camada.

---

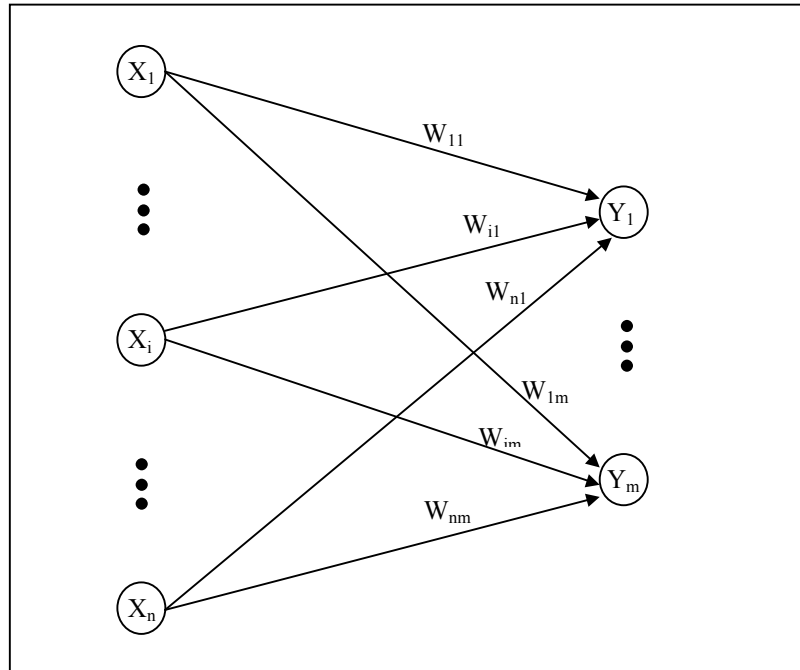
<sup>13</sup> A partir deste ponto da tese, designaremos simplesmente por redes neuronais as redes neuronais artificiais.

<sup>14</sup> BAM (Bidirectional Associative Memory).

---

**Figura 2.4** Rede neuronal do tipo *feedforward* de uma só camada

---



Na figura 2.4 pode observar-se uma rede *feedforward* de camada única. Neste exemplo, a camada de saída é formada pelo primeiro neurónio de saída  $Y_1$  até ao  $m$ -ésimo neurónio. A unidade de saída  $Y_1$  recebe sinais desde os neurónios de entrada  $X_1$ ,  $X_i$  até ao  $n$ -ésimo, o mesmo acontecendo com todos os neurónios de saída até ao  $m$ -ésimo. O peso  $W_{i1}$  proporciona ao neurónio  $Y_1$  a recepção do sinal enviado pela unidade de entrada  $X_i$

Cada unidade de uma rede *feedforward* de uma só camada pode apenas dividir o espaço  $n$ -dimensional de entrada através de um hiperplano [Minsky & Papert, 1969], limitando assim os tipos de problemas capaz de resolver.

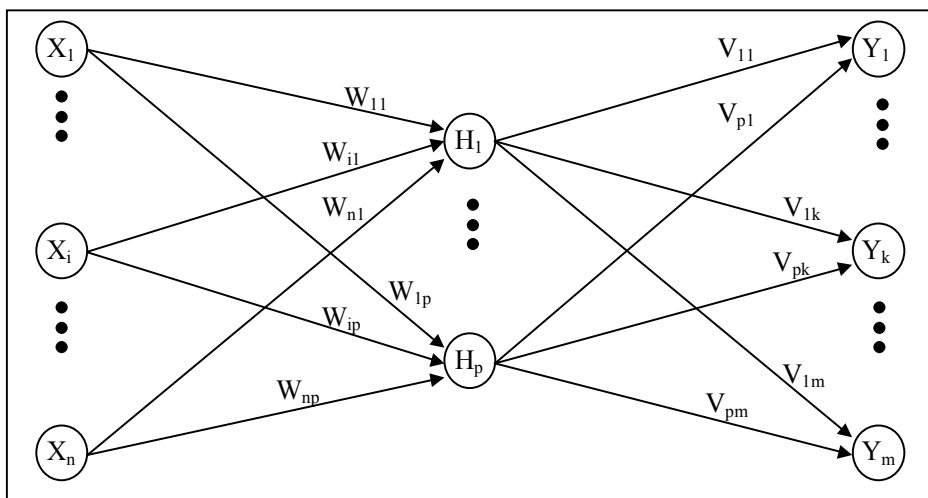
Esta limitação conhecida também por problema da não separabilidade linear é solucionado por redes multi-camadas. A rede *feedforward* da figura 2.5 é uma rede com arquitectura multi-camada. Trata-se efectivamente de uma rede com duas camadas já que

possui uma camada de neurónios intermédia colocada entre a camada de entrada e a de saída. Esta camada é vulgarmente designada por camada escondida.

---

**Figura 2.5** Rede neuronal do tipo *feedforward* com as camadas de entrada e de saída intercaladas por uma camada escondida (neurónios  $H$ ).

---



No nosso protótipo a arquitectura de redes usada foi uma multi-camada *feedforward*, já que a complexidade intrínseca às classes de entrada fazia prever logo à partida a necessidade de particionar o espaço de classificação em mais do que um hiperplano.

### 2.3.3 Aprendizagem neuronal

Este ponto também é chave na caracterização de uma rede neuronal. Os métodos ou algoritmos usados para ajustar os pesos de uma estrutura neuronal de forma a atingir a aprendizagem pretendida encontram-se distribuídos por três classes: métodos supervisionados, métodos não supervisionados e métodos de aprendizagem reforçada. Vamos

então ver em que consiste cada um. Há pelo menos dois pontos comuns aos três métodos. Primeiro ponto: todos funcionam com base em exemplos dados pelos padrões de treino que são colocados na entrada da rede neuronal. Segundo ponto: nem sempre, no final de um treino, a rede fica capaz de resolver os problemas correctamente. Nesta situação a aprendizagem não foi bem conseguida; há então que reiniciar o processo, tentar perceber as dificuldades e corrigir as falhas.

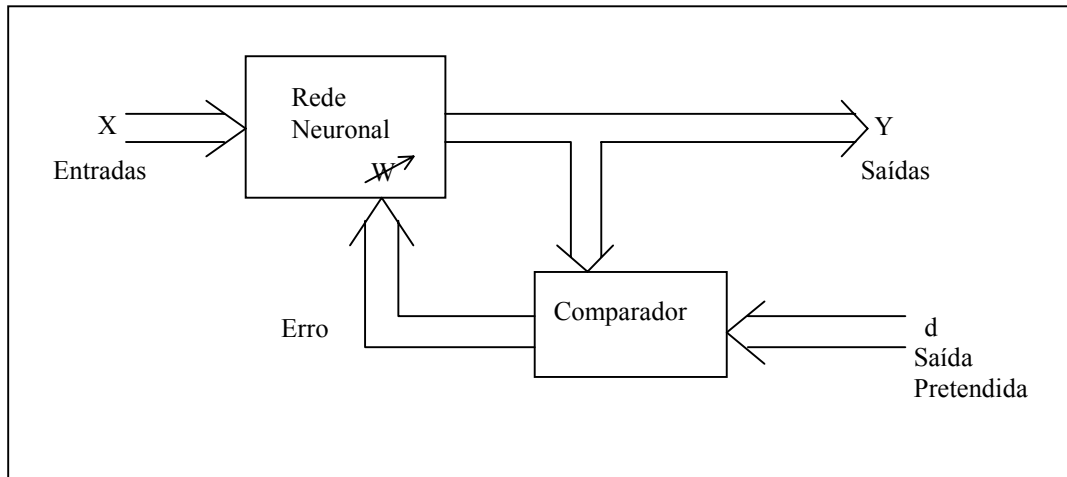
### 2.3.3.1 Aprendizagem supervisionada

Neste tipo de aprendizagem cada padrão de treino colocado na entrada da rede está sempre associado ao seu par correspondente na saída da rede. Desta forma, durante o treino a rede tem sempre disponível para cada padrão de treino de entrada um padrão de treino de saída que representa a resposta pretendida para aquele vector de treino de entrada e que a rede deverá aprender. Designemos por  $\mathbf{x}^{(k)}$  o  $k$ -ésimo vector de treino de entrada e por  $\mathbf{d}^{(k)}$  o correspondente vector de treino desejado na saída. Então a sequência de  $n$  pares de vectores de treino apresentados à rede durante a aprendizagem será  $(\mathbf{x}^{(1)}, \mathbf{d}^{(1)})$ ,  $(\mathbf{x}^{(2)}, \mathbf{d}^{(2)})$ , ...,  $(\mathbf{x}^{(k)}, \mathbf{d}^{(k)})$ , ...,  $(\mathbf{x}^{(n)}, \mathbf{d}^{(n)})$ .

Durante o processo de treino os pesos  $W$  são ajustados com base no erro que existe entre o vector  $\mathbf{y}^{(k)}$  obtido na saída e o vector desejado  $\mathbf{d}^{(k)}$ , para um vector de entrada  $\mathbf{x}^{(k)}$ , de forma a aproximar o valor obtido do valor desejado. Encontra-se na figura 2.6 um modelo representativo deste tipo de aprendizagem.

**Figura 2.6** Modelo representativo da aprendizagem supervisionada.

---



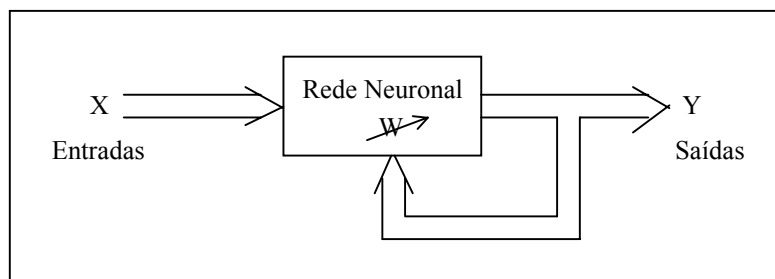
### 2.3.3.2 Aprendizagem não supervisionada

Neste tipo de aprendizagem não existe qualquer tipo de informação do erro entre a saída obtida e a saída pretendida, uma vez que não é conhecida a saída desejada. Durante a aprendizagem decorre um processo chamado “auto-organizativo” que tenta descobrir similaridades e dissimilaridades nos padrões de treino, de maneira a agrupá-los em classes de características mais ou menos comuns. Um exemplo de uma rede deste tipo são os mapas de Kohonen [Kohonen, 1989].

Na figura 2.7 encontra-se esquematizada a aprendizagem não supervisionada.

**Figura 2.7** Modelo representativo da aprendizagem não supervisionada.

---

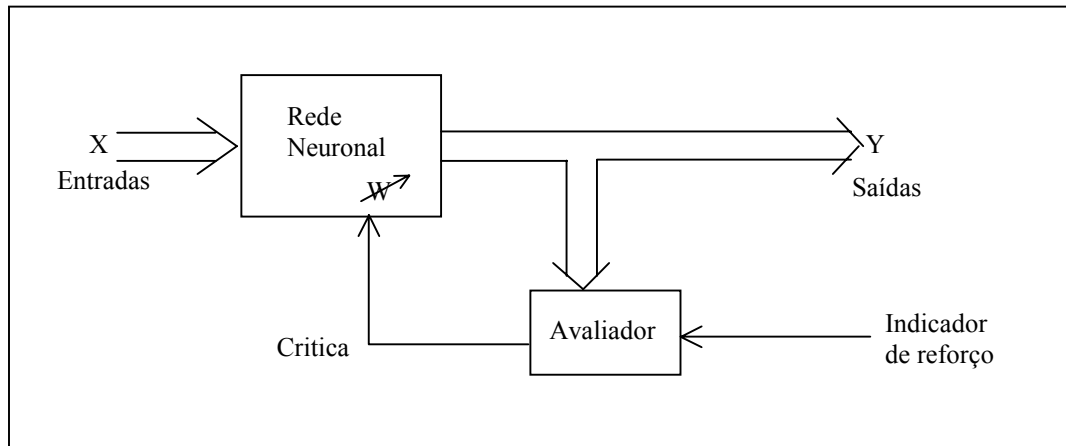


### 2.3.3.3 Aprendizagem reforçada

Este método de aprendizagem [Barto et alii, 1983] encontra alguma similaridade com a aprendizagem supervisionada, já que a rede recebe um sinal de *feedback* acerca da validade do padrão de saída gerado, tal como pode observar-se na figura 2.8. No entanto, este sinal não tem uma relação tão directa com o padrão de treino pretendido, como acontece na aprendizagem supervisionada. Este tipo de aprendizagem é usado em situações em que se torna difícil conhecer com exactidão os padrões de saída pretendidos. Por exemplo, se quisermos construir um controlador que permita manter o equilíbrio de uma bicicleta, torna-se difícil e pouco prático obter os padrões de saída pretendidos para corrigir num qualquer momento um desequilíbrio. Daí que seja mais fácil dizer à rede se uma saída gerada produz uma boa compensação ou não. A este tipo de *feedback* damos o nome de crítico, pois ele resulta não da diferença entre o padrão gerado e o pretendido, mas da validade do padrão gerado para resolver um determinado problema. O ajuste dos pesos, neste caso, consiste em reforçar o valor de cada peso, conforme este contribua, de acordo com o sinal crítico, para a resolução do problema.

**Figura 2.8** Modelo representativo da aprendizagem reforçada.

---



#### 2.3.4 Backpropagation

No processo de detecção das células na imagem necessitamos de tomar a decisão, pixel a pixel, se estes pertencem a um glóbulo branco ou se pertencem a qualquer outra região da imagem. Esta tarefa será assegurada por um classificador neuronal que durante a fase de treino aprende a reconhecer os padrões formados pelo pixel a avaliar e pelos seus pixels vizinhos que caracterizam a pertença ou não desse pixel ao leucócito. Assim, durante a fase de aprendizagem, percorremos uma imagem de treino e simultaneamente fornecemos à saída da rede neuronal a respectiva imagem alvo, que assegura uma correcta correspondência entre os padrões de entrada e de saída.

Durante o processo de classificação dos leucócitos são extraídas características das regiões que foram assinaladas durante a execução do mecanismo de segmentação. Essas características são traduzidas numericamente por descritores de regiões, descritores de textura e por outros descritores geométricos. Os valores obtidos com estes descritores são fornecidos

à entrada de uma rede neuronal, de forma a que esta possa proceder, depois da fase de aprendizagem, à correcta classificação do respectivo leucócito. A aprendizagem é então conseguida através da apresentação à rede neuronal de um conjunto de pares de vectores características/leucócito.

O género de situação com que nos deparamos quando olhamos para o problema da segmentação e classificação das células sanguíneas sugere-nos estarmos perante uma forma de aprendizagem neuronal supervisionada, já que, durante o treino, para cada padrão de entrada temos um conhecido e exacto padrão de saída. Esta pareceu-nos ser a forma mais apropriada de treinar a rede. Isto porque existe uma facilidade relativa de encontrarmos as correspondências entre os padrões de entrada e os de saída. Por exemplo, no caso da segmentação é relativamente fácil recorrendo à ajuda de um especialista em hematologia criarmos uma imagem alvo, que não é mais do que uma imagem igual à de entrada mas com as regiões que constituem os leucócitos pintadas com um tom singular. Ou ainda, continuando a recorrer a um especialista, classificar as células existentes na imagem de treino e codificar essa informação em forma de vector a ser entregue à saída da rede de classificação.

Dito isto, já sabemos que usamos na nossa implementação uma rede neuronal com uma estrutura que permite o treino supervisionado. Restava agora escolher o algoritmo que permite a uma rede com esta topologia ajustar os seus pesos. Esta opção caiu no algoritmo *backpropagation* não só pela sua tradicionalidade, mas essencialmente pela sua facilidade de implementação, versatilidade, modelização e relativa eficiência computacional.

#### 2.3.4.1 O Algoritmo Backpropagation

Sabemos que o interesse de redes neuronais de uma única camada é relativamente pequeno, já que apenas podem ser empregues em problemas que não exijam mais do que um hiperplano a separar o espaço de classificação. Sendo assim, houve necessidade de encontrar técnicas que permitissem a aprendizagem a redes compostas por mais do que uma camada. Foi então que vários investigadores entre os quais [le Cun, 1985], [Parker, 1985], [Rumelhart

& Hinton & Williams, 1986] apresentaram um processo que permite ajustar os pesos de redes *feedforward* com mais do que uma camada. Este processo sofreu vários aperfeiçoamentos e mutações ao longo dos anos seguintes por vários investigadores, com vista a reduzir o tempo necessário de treino e aumentar a capacidade de aprender (encontrar mínimos globais e generalizar). Este método conhecido por *backpropagation* ou por *regra delta generalizada* baseia-se na técnica de descida segundo a direcção do gradiente que tenta minimizar um determinado erro fornecido por uma função que traduz a diferença entre os valores de saída originados pela rede e os valores pretendidos. O termo *backpropagation* significa a propagação do erro no sentido da saída da rede para a sua entrada.

O treino de uma rede usando o algoritmo *backpropagation* pode dividir-se em três etapas seguintes. A primeira, conhecida por *feedforward*, consiste na computação das saídas da rede para um vector de treino de entrada. A seguinte etapa compara o vector de saída resultante da computação anterior com o vector pretendido e retro-propaga o erro associado para as camadas anteriores; isto é, no sentido das unidades de entrada. Na última etapa os pesos são ajustados de acordo com o erro propagado na etapa anterior. Esta sequência é então repetida durante vários ciclos de conjuntos de padrões de entrada e saída até que o erro global entre os padrões de saída e os pretendidos permita que os resultados desejados sejam minimamente conseguidos.

#### 2.3.4.2 Regra delta generalizada

Consideremos a seguinte rede *feedforward* multi-camada  $U$  constituída por  $n$  camadas, sendo  $U_n$  o subconjunto que agrupa as unidades da última camada e  $U_1$  o subconjunto que agrupa as unidades de entrada e  $U_i$  a  $i$ -ésima camada:

$$U = U_1 \cup \dots \cup U_i \cup \dots \cup U_n \quad (2.19)$$

Se de uma rede multi-camada se tratar,  $n$  terá que ser igual ou maior que 3. Supondo ainda que cada camada pode possuir o seu tipo de função de activação  $\varphi_i(\cdot)$  e tendo em conta a equação (2.4), então a saída  $Y$  de um neurónio  $k$  da camada  $i$ , para  $i$  maior ou igual a 2 é dada por:

$$Y_{(i,k)} = \varphi_i(v_{(i,k)}) = \varphi_i\left(\sum_{j=1}^{m_{i-1}} W_{(i,k,j)} Y_{(i-1,j)} + b_{(i,k)}\right) \quad (2.20)$$

Uma vez que o processo de *feedforward* tenha sido completado em consonância com a equação (2.20), os pesos poderão ser ajustados de acordo com a equação seguinte:

$$\Delta W_{(i,k,j)} = \alpha \delta_{(i,k)} Y_{(i-1,j)} \quad (2.21)$$

e no caso das *polarizações*:

$$\Delta W_{(i,k,0)} = \alpha \delta_{(i,k)} \quad (2.22)$$

onde  $\alpha$  é um escalar com valores compreendidos entre 0 e 1 e conhecido por coeficiente de aprendizagem. Este reflecte a magnitude com que o peso é actualizado.  $\delta$  é calculado segundo (2.23), na qual  $T$  representa o valor pretendido na unidade de saída  $k$ .

$$\delta_{(i,k)} = \begin{cases} \varphi'_i(v_{(i,k)}) (T_k - Y_{(n,k)}) & \text{se } i = n \\ \varphi'_i(v_{(i,k)}) \sum_{j=1}^{m_{i+1}} \delta_{(i+1,j)} W_{(i+1,k,j)} & \text{se } 2 \leq i < n \end{cases} \quad (2.23)$$

### 2.3.4.3 Passos da Backpropagation

1. Inicializar os pesos.
2. Executar ciclicamente desde o passo 3 ao 7 até se verificar uma condição de paragem, por exemplo ter-se atingido um erro mínimo ou um número de iterações máximo.
3. Aplicar na entrada e saída da rede um par de padrões de treino, podendo este ser seleccionado do conjunto de padrões disponível através de um método sequencial ou de um método estatístico.
4. Usando a equação (2.20), obter todos os valores de saída da rede, para o padrão de entrada apresentado no passo 3.
5. Usar a definição (2.23) para calcular e retro-propagar a todas as camadas da rede (com excepção da primeira, obviamente) o erro entre o vector de saída obtido em 4 e o pretendido para o vector de entrada apresentado em 3.
6. Com base nas equações (2.21) e (2.22), calcular os termos de correcção para os respectivos pesos.
7. Com os termos obtidos em 6 actualizar os pesos, por exemplo, com a seguinte equação, na qual  $p$  representa a iteração actual:

$$W_{(i,k,j)}^p = W_{(i,k,j)}^{(p-1)} + \Delta W_{(i,k,j)} \quad (2.24)$$

### 2.3.4.4 Actualização dos pesos por lotes/tempo real (batch/on-line)

Os passos e as equações apresentadas nos dois pontos anteriores permitem deduzir que as actualizações dos pesos são executadas padrão após padrão. Este processo é conhecido por actualização *on-line* (ou tempo real). No entanto, a formulação inicial do algoritmo da *backpropagation* indica que a actualização deve ser feita época após época, sendo o valor efectivo de actualização igual à média dos valores parciais de actualização obtidos para cada um dos padrões que constituem uma época de treino. Este procedimento, como de resto constatamos, pode de facto aumentar as possibilidades de sucesso da aprendizagem

relativamente à primeira opção, principalmente se a distribuição dos padrões de treino durante a aprendizagem não for a mais adequada. Porém, com este método o processo de aprendizagem torna-se mais lento à medida que o conjunto de padrões de treino aumenta, uma vez que é necessário percorrer todo o conjunto de padrões de treino para que os pesos sejam actualizados.

A apresentação aleatória dos padrões de treino tende a minimizar os problemas de convergência do método *on-line*, podendo-se assim tirar partido da maior velocidade que este método oferece.

#### 2.3.4.5 A inicialização dos pesos

A inicialização dos pesos é o primeiro passo a executar quando se pretende treinar uma rede com este tipo de algoritmo. Este passo, de facto, detém alguma importância, pois poderá determinar a capacidade de se conseguir atingir em termos de erro um mínimo global ou mínimo local, ou pelo menos influenciará a rapidez com que um destes mínimos será atingido.

##### 2.3.4.5.1 Inicialização aleatória.

Os valores iniciais dos pesos não devem ser elevados nem devem ser muito próximos de zero, pois tanto uma situação como a outra produzem lentidão e dificuldade na aprendizagem. Isto acontece porque os valores de actualização dos pesos dependem da derivada da função de activação do neurónio da camada posterior e do valor da função de activação do neurónio da camada anterior. Assim, é boa medida atribuir aos pesos, aleatoriamente, valores compreendidos entre  $-0.5$  e  $0.5$ .

#### 2.3.4.5.2 Inicialização Wessels Barnard<sup>15</sup>.

Uma forma de tornar mais eficaz a inicialização aleatória de pesos consiste em ajustar a gama de valores a gerar, para cada conjunto de pesos associado a cada neurónio de uma determinada camada.

O intervalo de valores a gerar aleatoriamente deverá ser:  $\left[ \frac{-3}{\sqrt{Z_{(i,k)}}}, \frac{3}{\sqrt{Z_{(i,k)}}} \right]$  onde

$Z_{(i,k)}$  representa o número de unidades da camada anterior ( $i-1$ ) que estão conectadas ao neurónio  $k$  da camada  $i$ .

#### 2.3.4.5.3 Inicialização Nguyen-Widrow

Uma outra técnica de inicialização aleatória de pesos, conhecida por inicialização Nguyen-Widrow [Nguyen & Widrow, 1990] e desenvolvida para redes *feedforward* com uma camada escondida, é a seguir descrita. Os pesos da camada de saída podem ser inicializados também como no primeiro método com valores entre  $-0.5$  e  $0.5$ . No entanto, os pesos da camada escondida sofrem um escalonamento de acordo com a equação (2.25), onde  $m$  representa o número de unidades de entrada e  $q$  o número de unidades da camada escondida. Este método pode em algumas aplicações trazer uma melhoria significativa da velocidade do treino.

$$W_{(k,j)}^t = \frac{0.7 \sqrt[3]{mq} W_{(k,j)}^{(t-1)}}{\|W_k^{(t-1)}\|} \quad (2.25)$$

---

<sup>15</sup> [Wessels & Barnard,1992]

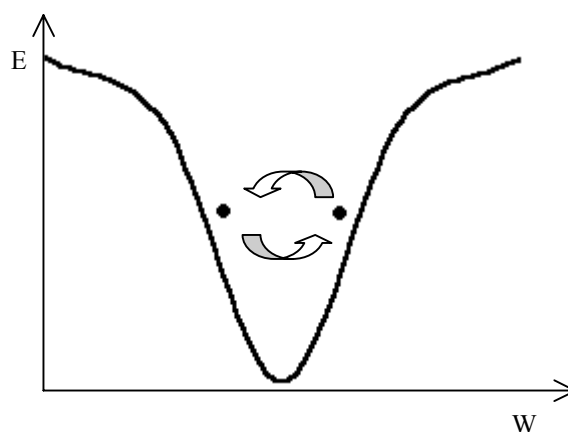
#### 2.3.4.6 Outros métodos de actualização dos pesos.

Existem múltiplas técnicas de actualização dos pesos que diferem mais ou menos da técnica descrita pela equação (2.21). No entanto, todas têm os mesmos propósitos: o de diminuir o tempo de aprendizagem e o de minimizar o erro. A equação (2.21) deduzida da regra delta generalizada pode apresentar alguns problemas, nomeadamente lentidão na convergência quando o valor de  $\alpha$  é demasiado pequeno e impossibilidade de convergir para um mínimo devido às oscilações causadas por um valor de  $\alpha$  elevado (Fig.2.9).

---

**Figura 2.9** Se o factor de aprendizagem for demasiado elevado podem ocorrer oscilações que impossibilitam a backpropagation atingir o mínimo erro.

---

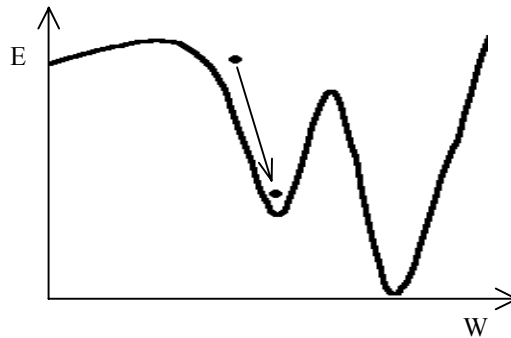


---

Um método muito usado que permite a  $\alpha$  ter valores elevados sem provocar oscilações na convergência é conhecido por método do *momento* [Rumelhart & McClelland, 1986b]. A ideia consiste em adicionar ao factor de correcção (2.21) um novo termo que não é mais do que o factor de correcção dos pesos calculado na iteração de treino anterior e que vai funcionar como uma componente de inércia que reduz as oscilações na convergência, já que tende a obrigar a que a direcção de descida do gradiente se mantenha a mesma da iteração (padrão) anterior. Isto também pode reduzir a probabilidade de a convergência atingir um mínimo local em vez de um mínimo global (figura 2.10) [Fausset, 1994].

**Figura 2.10** Ficar retido num mínimo local não conseguindo convergir para um mínimo global é um dos riscos da backpropagation.

---



Por outro lado, esta dependência do factor de correcção anterior pode fazer com que o erro em determinados padrões de treino, em vez de diminuir, aumente devido à direcção que lhe é imposta pelo termo do momento [Jacobs, 1988]. O grau com que esta componente é adicionada à componente de correcção é determinado por  $\mu$ , como pode verificar-se na equação (2.26). Os valores que vulgarmente são atribuídos a  $\mu$  encontram-se no intervalo entre 0 e 1.

$$\Delta W_{(i,k,j)}^p = \alpha \delta_{(i,k)} Y_{(i-1,j)} + \mu \Delta W_{(i,k,j)}^{(p-1)} \quad (2.26)$$

Uma outra técnica de actualização dos pesos que pode oferecer bons resultados é conhecida por *delta-bar-delta* [Jacobs, 1988], que permite que cada peso tenha o seu próprio factor de aprendizagem  $\alpha$ . Permite também que este factor possa sofrer alterações de iteração para iteração durante a fase de treino. As heurísticas para estas alterações são as seguintes:

- Se a derivada parcial do erro em ordem a um determinado peso mantém o sinal durante algumas iterações, então devemos aumentar o coeficiente de aprendizagem desse mesmo peso.

- Se a derivada parcial do erro em ordem a um determinado peso sofre variações no seu sinal relativamente às últimas iterações, então devemos diminuir o coeficiente de aprendizagem.

Assim, neste método, a actualização dos pesos na iteração  $p$  é executada da seguinte forma:

$$W_{(i,k,j)}^p = W_{(i,k,j)}^{(p-1)} + \alpha_{(i,k,j)} \delta_{(i,k)} Y_{(i-1,j)} \quad (2.27)$$

sendo  $\alpha_{(i,k,j)}$  modificado em cada iteração segundo a definição (2.30).

Consideremos então  $\Delta_{(i,k,j)}$  definido na equação (2.28) como o contributo dado, por um determinado peso, ao erro entre o padrão de saída e o pretendido. Ou dito de outra forma, a derivada parcial do erro em ordem a um determinado peso.

$$\Delta_{(i,k,j)} = \frac{\partial E}{\partial W_{(i,k,j)}} = -\delta_{(i,k)} Y_{(i-1,j)} \quad (2.28)$$

A seguir, na equação (2.29), associamos o valor da derivada actual com o valor das derivadas passadas, permitindo  $\beta$  ajustar a comparticipação de cada uma, valendo entre 0 e 1.

$$\bar{\Delta}_{(i,k,j)}^p = (1 - \beta) \Delta_{(i,k,j)}^p + \beta \bar{\Delta}_{(i,k,j)}^{(p-1)} \quad (2.29)$$

Servindo-nos das equações 2.28 e 2.29 podemos agora definir, parametricamente e com base nas heurísticas referidas atrás, o sistema que gere os coeficientes de aprendizagem de cada peso.

$$\alpha_{(i,k,j)}^{(p+1)} = \begin{cases} \alpha_{(i,k,j)}^p + \kappa & \text{se } \bar{\Delta}_{(i,k,j)}^{(p-1)} \Delta_{(i,k,j)}^p > 0 \\ (1 - \gamma) \alpha_{(i,k,j)}^p & \text{se } \bar{\Delta}_{(i,k,j)}^{(p-1)} \Delta_{(i,k,j)}^p < 0 \\ \alpha_{(i,k,j)}^p & \text{se } \bar{\Delta}_{(i,k,j)}^{(p-1)} \Delta_{(i,k,j)}^p = 0 \end{cases} \quad (2.30)$$

De acordo com a definição 2.30,  $\alpha$  irá sofrer um incremento igual a  $\kappa$  se o sinal da derivada parcial actual for o mesmo que o das derivadas anteriores. E irá sofrer um fraccionamento proporcional a  $\gamma$  se o sinal da derivada actual for contrário ao das derivadas passadas. Este método permite em muitos problemas diminuir o tempo de aprendizagem. Porém, às vezes os coeficientes de aprendizagem atingem rapidamente valores elevados, o que produz divergência na aprendizagem. Nesta situação devem repor-se os parâmetros da rede obtidos na iteração anterior e forçar a diminuição dos coeficientes de aprendizagem.

### 2.3.5 Técnicas e constituintes neuronais mais adequados

As técnicas e constituintes neuronais para redes *feedforward* aqui descritos foram quase na totalidade experimentados e integrados no nosso simulador. A sua associação aos processos de segmentação e classificação permitiu-nos seleccionar os mais adequados no âmbito dos dois métodos, embora alguns deles mostrassem desempenhos similares como foi o caso das funções de activação.

As várias funções de activação experimentadas não proporcionaram diferenças significativas no desempenho do treino neuronal. Acabámos por escolher a tradicional sigmóide bipolar tanto para a segmentação como para a classificação já que esta função é computacionalmente menos pesada.

O método de actualização de pesos *batch*, apesar de mais lento relativamente ao método *on-line*, mostrou-se mais consistente no que se refere à obtenção dos melhores resultados. Foi na aprendizagem para a segmentação que mais se manifestou essa

consistência, já que nesse caso o conjunto de padrões de treino é tão vasto que nem mesmo uma seleção aleatória dos padrões evita que a rede tenda, por vezes, a especializar-se em alguns subconjuntos desses padrões.

Os modos de inicialização dos pesos que escalonam estes em função do número de neurónios, como o *Nguyen-Widrow*, mostraram-se adequados para a obtenção dos resultados da segmentação e da classificação. No entanto, mesmo assim, é necessário fazer concordar os limites das gamas de valores com a dimensão da rede neuronal. Foi notória, no nosso trabalho, a necessidade de diminuição dos intervalos de valores à medida que a dimensão da rede neuronal crescia.

A técnica de aceleração da aprendizagem neuronal *delta-bar-delta* foi sempre a que permitiu tempos de treino mais reduzidos e, em simultâneo, bons resultados. De qualquer forma, as repetidas tentativas de treino que permitem beneficiar das diferentes inicializações da rede e ajustar os parâmetros relacionados, neste caso, com o *delta-bar-delta* são ponto chave para encontrar um conjunto de pesos aptos para a resolução dos problemas da segmentação e da classificação.



## Capítulo 3

### O problema da segmentação

#### **3.1 Descrição do problema da segmentação**

Como dissemos no capítulo de introdução, a primeira acção significativa que teremos de executar nas imagens que são captadas pela câmara CCD que se encontra acoplada ao microscópio óptico será a de detectar e assinalar nessas imagens as regiões que constituem os glóbulos brancos. De facto, esta acção torna-se essencial para o sucesso do processo posterior, o de classificação dos leucócitos, pois caso a detecção das regiões leucócito seja deficiente o mecanismo de classificação fica prematuramente condenado ao fracasso.

O problema é o seguinte: como determinar se um dado pixel pertence ou não a uma região de um leucócito? De facto, devido à natureza destas imagens, só é possível tomarmos essa decisão sem equívocos, se considerarmos também como adjuvante a essa decisão a região de pixels circundantes ao pixel considerado. Problemas existem onde a avaliação da pertença de um pixel a uma determinada região se consegue fazer com base apenas na luminância e crominância do pixel a avaliar (ou mesmo com base só numa delas). Porém, se olharmos para as imagens das células sanguíneas, verificamos facilmente que esta técnica não pode ser usada, uma vez que podemos encontrar pixels pertencentes a um leucócito com luminâncias e crominâncias idênticas a pixels que pertencem a outra qualquer parte da imagem que não um leucócito. Podem ainda acontecer outras situações ambíguas, pois nem todos os leucócitos apresentam coerência, em termos de cor e luminância. Mas os problemas não terminam aqui: a variabilidade com que as lâminas que suportam as amostras sanguíneas são

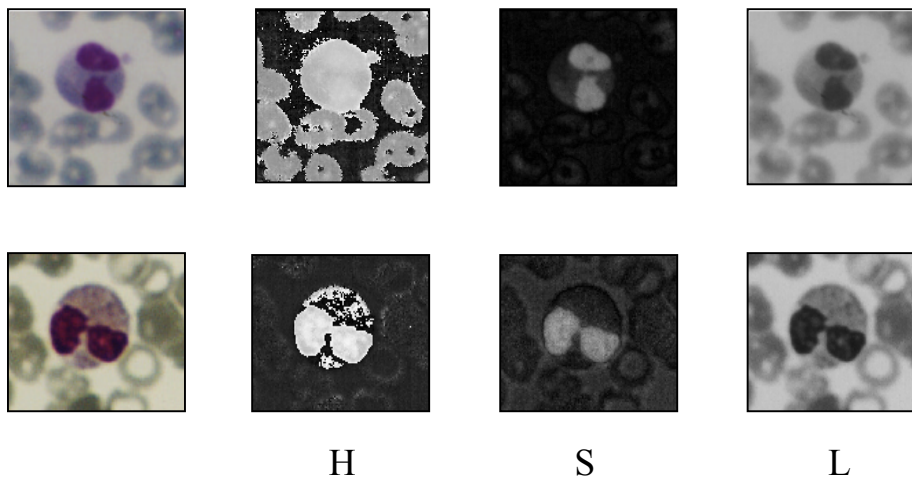
coradas provocam uma variabilidade na tonalidade das imagens que se constata dentro da mesma lâmina. Pretende-se que esta variância não afecte tanto o processo de segmentação bem como o de classificação.

Para se tornarem ainda mais evidentes as diferenças que podem ocorrer de lâmina para lâmina, observemos a figura 3.1, na qual temos na primeira coluna de imagens dois eosinófilos rodeados de glóbulos vermelhos. Cada eosinófilo encontrava-se em lâminas diferentes. Cada uma destas imagens foi decomposta em três outras que representam, da esquerda para a direita, o seguinte: a componente cor (Hue), a componente intensidade de cor (Saturation) e a intensidade de luz (Lightness). Note-se a disparidade bem evidente entre os planos de cor e os planos de saturação de cor.

---

**Figura 3.1** As imagens da primeira coluna correspondem a dois leucócitos da mesma classe, mas em lâminas distintas. As restantes três colunas representam a decomposição das imagens referidas, nos planos H, S, L. A comparação entre as duas linhas de imagens mostra bem as diferenças que podem surgir de uma amostra para outra.

---



Por conseguinte, fomos obrigados a arranjar soluções mais complexas que permitam decidir correctamente se um pixel pertence ou não a um leucócito, seja este de que tipo for. Torna-se óbvio que quando observamos uma célula deste tipo olhamos de uma vez só para o aglomerado de pixels que a constituem e é esse conjunto de pixels de que nós conseguimos ter percepção simultânea que nos permite delimitar correctamente a região que forma a célula. Na verdade, esta é uma tarefa vulgar e simples como outras que executamos no dia-a-dia, o que faz com que não nos apercebamos da complexidade que isto pode representar em termos do mecanismo funcional que está por detrás destas simples acções.

Fica assim claro que a decisão de pertença de um pixel a uma região do leucócito tem que ser tomada com base no maior número de pixels cercanos possível, embora dentro de um limite razoável. Esse limite é principalmente imposto pelas máquinas onde emulamos o protótipo. É que à medida que aumentamos o número de pixels para a tomada de decisão, a necessidade de memória e velocidade de processamento cresce exponencialmente. Uma outra dificuldade prende-se com o aumento da dimensionalidade do vector a analisar, o que faz com que uma aprendizagem neuronal eficaz se torne mais difícil de conseguir.

Mas aqui surgem outros problemas. Como traduzir para uma máquina se um pixel pertence ou não a um leucócito, com base na distribuição dos pixels vizinhos? Como considerar todos os arranjos de pixels possíveis?

Métodos estatísticos [Ripley & Hjort, 1995] ou *Fuzzy* [Heske & Heske, 1996] são técnicas possíveis de se usar. No entanto, exigem a interpretação e descodificação prévia da informação relacionada com o problema, o que por si só significa dificuldade em consegui-lo, constituindo uma fonte possível de erros e de faltas que podem comprometer o desempenho final do processo.

É na resolução deste tipo de dificuldades que as redes neuronais sobressaem. A sua capacidade de aprender através de exemplos atenua os problemas inerentes aos métodos anteriormente citados. É de facto esta capacidade que permite às redes aperceberem-se de características subtis e menos subtis que fundamentam a pertença de um determinado pixel à região leucócito.

### **3.2 Proposta para a resolução do problema da segmentação.**

Depois de pesquisarmos, estudarmos e analisarmos técnicas de segmentação, arquitectámos um mecanismo de segmentação que se apoia numa rede neuronal.

A ideia é deslocar, pixel a pixel, uma janela por toda a imagem a segmentar e ir avaliando se o pixel que se encontra no centro da janela está situado na região leucócito ou não. O conjunto de pixels circunscritos pela janela de varrimento irão constituir na fase de aprendizagem os padrões de treino de entrada.

Para sermos mais esclarecedores vamos, através das figuras 3.2 e 3.3, demonstrar esta ideia. A saída da rede neuronal será formada por uma só unidade que deverá tomar o valor de 0.8 ou -0.8, consoante o pixel central da sub-região a analisar pertença ou não à região leucócito. Na figura 3.2 observamos uma imagem de células a ser varrida linha a linha de cima para baixo, por uma pequena janela a que chamaremos janela de varrimento, que se lhe sobrepõe pixel após pixel. Esta janela limita e define para cada pixel uma sub-região na imagem a segmentar. Essa sub-região constitui o padrão de pixels que será entregue à rede neuronal. O objectivo é criar uma nova imagem (binária) assinalando os pixels que formam a região célula (Fig 3.4).

Repare-se no exemplo da figura 3.2; a janela pequena foi centrada sobre um pixel que claramente não pertence ao leucócito. Aqui a rede respondeu correctamente com o valor de -0.8. Já no exemplo da figura 3.3 encontramos a outra situação. O pixel que se encontra no centro da janela de varrimento pertence à região leucócito. Neste caso, a saída da rede deverá ser 0.8 assinalando correctamente a pertença do pixel central à região leucócito. Pixel após pixel vai sendo formada uma nova imagem, na qual vai ficando registada a decisão da rede neuronal, da forma que vamos descrever a seguir.

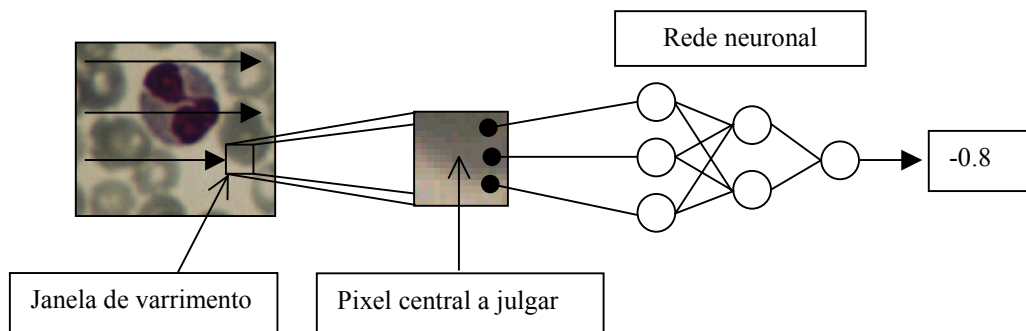
Consideremos a imagem a segmentar  $P$  e a imagem resultado  $I$  (figura 3.4); suponhamos ainda que o processo de segmentação está a decorrer e que a janela de varrimento se encontra centrada no pixel  $(i, j)$  da imagem  $P$ . Então a rede neuronal irá calcular um valor de saída que será função do conjunto de pixels abrangidos pela janela de varrimento. Se o valor de saída da rede se aproximar de 0.8, então o pixel  $(i, j)$  da imagem  $I$  ficará com uma tonalidade clara, indicando que esse pixel na imagem  $P$  pertence a um leucócito. Se o valor apresentado pela rede neuronal for mais próximo de -0.8, então o pixel  $(i, j)$  da imagem

resultado  $I$  mostrará uma tonalidade escura, indicando que esse pixel na imagem  $P$  não pertence a um leucócito.

---

**Figura 3.2** O percurso da janela de varrimento sobre a imagem permite à rede neuronal decidir sobre a pertença ou não dos pixels à região leucócito. Nesta situação a janela de varrimento encontra-se centrada sobre um pixel que não pertence ao leucócito. A rede neuronal processou acertadamente a sua saída para o valor de  $-0.8$ .

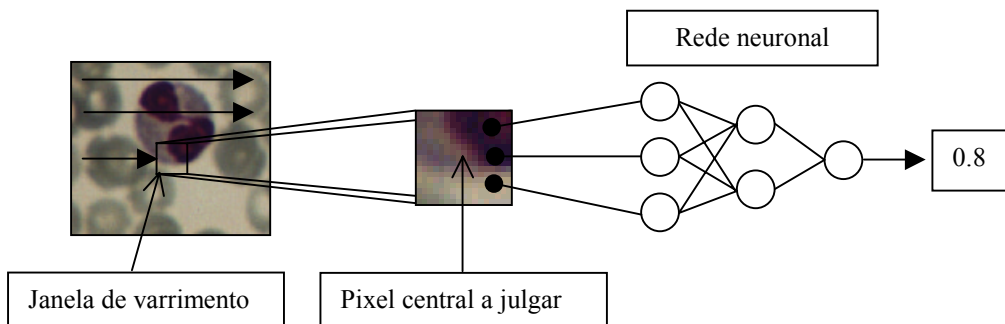
---



---

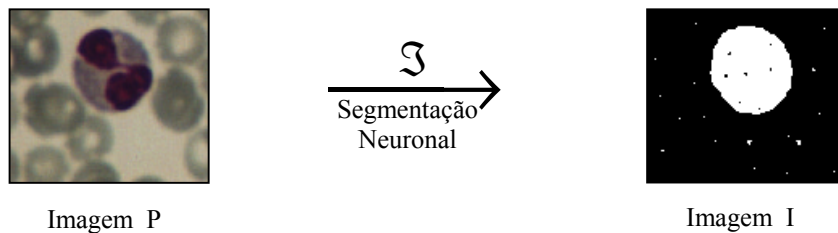
**Figura 3.3** O percurso da janela de varrimento sobre a imagem permite à rede neuronal decidir sobre a pertença ou não dos pixels à região leucócito. Nesta situação a janela de varrimento encontra-se centrada sobre um pixel que pertence ao leucócito. A rede neuronal processou acertadamente a sua saída para o valor de  $0.8$ .

---



**Figura 3.4** Depois de percorrer a imagem a segmentar  $P$ , pixel a pixel, é formada uma nova imagem  $I$ . Essa imagem assinala a branco os pixels que formam as regiões leucócito. Repare-se, no entanto, que alguns pixels se encontram erradamente assinalados tanto na região célula como na restante região.

---



### 3.2.1 Cor e Pré-processamento

Entendemos como fase de pré-processamento as acções que são executadas antes de entregar os vectores provenientes das imagens das células sanguíneas à rede neuronal.

Como vimos, as imagens com que trabalhamos são imagens a cores, uma vez que a cor facilita a caracterização das regiões que constituem os leucócitos. Esta justificação torna-se mais válida em situações em que temos glóbulos vermelhos fisicamente colados a glóbulos brancos. Se as imagens fossem em níveis de cinzento teríamos mais dificuldade em conseguir fazer com que a rede neuronal distinguisse sobre a pertença do pixel ao leucócito ou ao glóbulo vermelho.

O uso da cor também se revela importante no processo posterior ao da segmentação. No entanto, temos consciência de que o uso da cor pode trazer certas dificuldades que não apareceriam se as imagens fossem em níveis de cinzento. Nomeadamente, os problemas relacionados com a sensibilidade às variabilidades na cor de lâmina para lâmina e dentro da própria lâmina. Tentou contornar-se o problema da variabilidade da cor, fornecendo à rede um conjunto de padrões de treino o mais diversificado possível, com padrões extraídos de imagens com as mais variadíssimas tonalidades.

O problema da dimensionalidade do padrão está relacionado com o crescimento do espaço de classificação. É que, em geral, um aumento linear da dimensão do padrão de entrada implica um crescimento exponencial do espaço de classificação [Bishop, 1995]. Isto quer dizer que para termos uma rede com uma boa capacidade de generalização teremos de lhe fornecer durante o treino um suficiente número de padrões distintos que permitam à rede “traçar” correctamente por todo o espaço de classificação as combinações de hiperplanos delimitadoras das classes.

Ora, a exigência desse número de padrões tende a aumentar de acordo com o crescimento do espaço de classificação, o que obriga a possuir um conjunto de padrões de treino não redundantes que cresce exponencialmente com a dimensão do padrão. No entanto, dependendo do tipo de problema, a redução do tamanho do padrão pode trazer ainda mais dificuldades para a obtenção de bons resultados, já que informação discriminatória importante pode estar a ser omitida pela redução da dimensionalidade, o que obriga a uma busca de uma solução de compromisso.

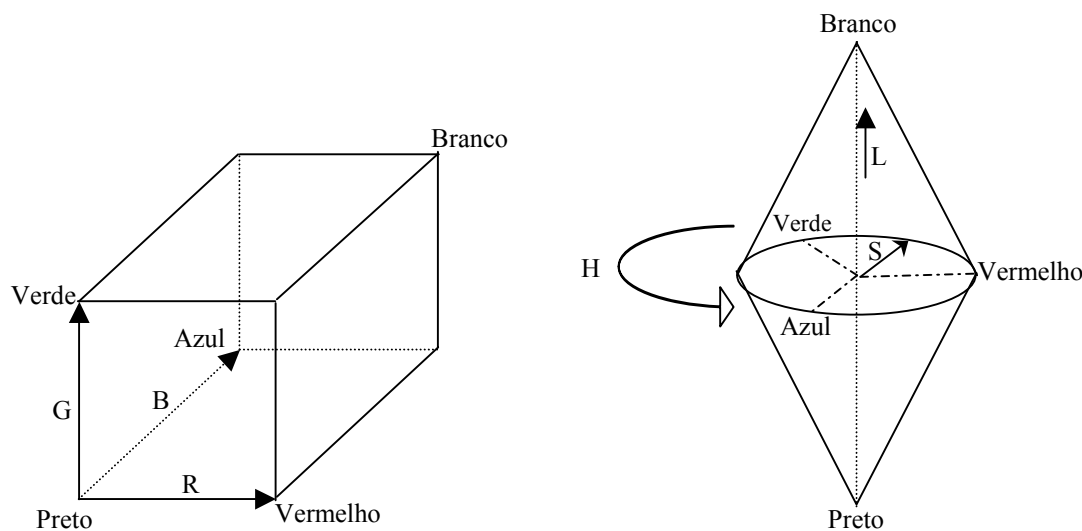
No nosso protótipo, as imagens são recebidas tanto através de ficheiros como através da carta digitalizadora, num formato RGB de 24 bits. Trata-se pois de imagens que são compostas por três planos de cor, vermelho, verde e azul. Cada pixel neste formato de imagem encontra-se codificado em 8 bits de intensidade vermelha, 8 bits de intensidade verde e 8 bits de intensidade azul. Este tipo de codificação pode implicar mais dificuldades para a rede aprender, já que uma pequena variação na tonalidade das imagens pode significar uma grande variação no mapeamento neste espaço de cores. Um sistema de cores que pode atenuar um pouco este efeito é o HSL (figura 3.5), já que este espaço de cores não é uma transformação linear do espaço RGB, mas sim uma transformação não linear [Gonzalez & Woods, 1992]. Em vez de existir uma componente verde, azul e vermelho, existe uma componente **H**(hue) que representa o espectro de cores do vermelho ao violeta, outra a intensidade de cor, **S**(saturation) e outra o brilho, **L**(Lightness) (ver figura 3.6). Assim, variações de brilho nas imagens traduzir-se-iam por variações significativas apenas na componente **L**, enquanto nas imagens RGB uma variação do brilho provoca geralmente uma variação significativa das três componentes. O mesmo poderíamos dizer sobre as outras duas componentes. Experimentalmente verificou-se que, de facto, no caso de imagens no formato

RGB a rede segmentava erradamente; quando treinávamos a rede com imagens no formato HSL o resultado da segmentação era melhorado.

---

**Figura 3.5** O espaço de cores HSL, apresenta-se como dois cones invertidos. Enquanto que o espaço de cores RGB apresenta-se como um cubo.

---



Para aumentar a eficácia da rede neuronal e para simplificar o treino, tentámos normalizar cada um dos planos de cor (RGB). O método usado consiste em determinar para cada plano (RGB) a média dos valores médios das imagens de treino. Às imagens a analisar é então calculada a diferença entre o seu valor médio e a média das imagens de treino. Essa diferença é posteriormente subtraída a todos os pixels da imagem a analisar, garantindo-se assim que o valor médio dessas imagens é para cada plano o mesmo valor apresentado pela média das imagens de treino. Este simples método melhorou bastante o resultado da segmentação em imagens que apresentavam tonalidades diferentes das imagens de treino.

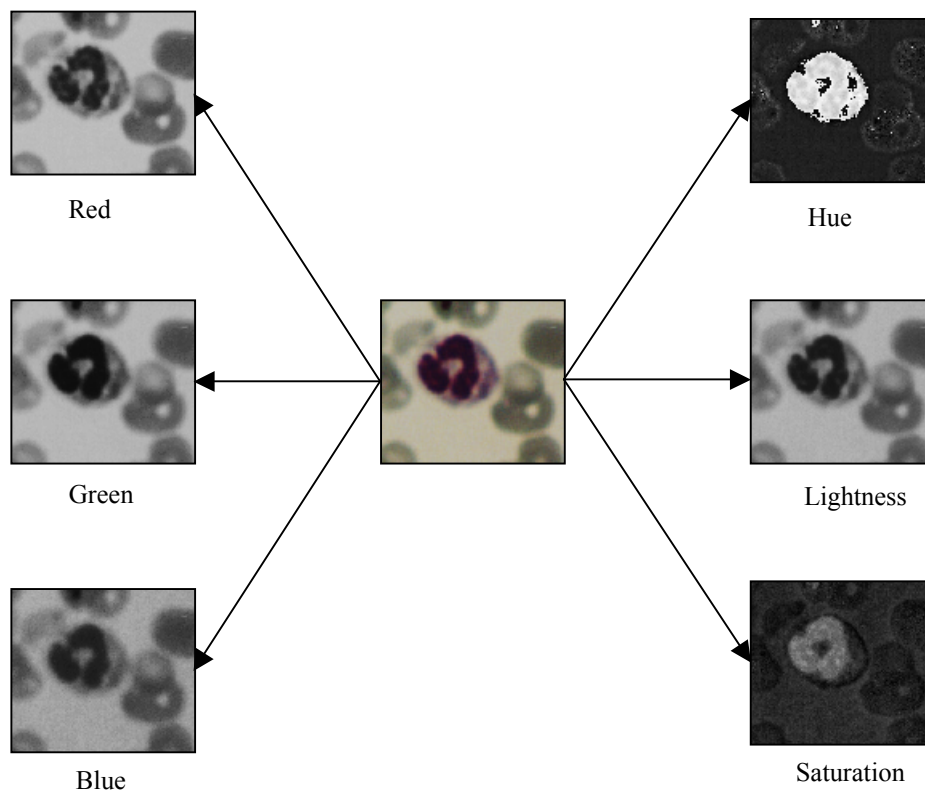
Mas claro que tendo este processo como base apenas a média das imagens, por vezes os resultados não eram os esperados.

De referir ainda que, para assegurar uma adaptação apropriada dos valores dos pixels à entrada da rede neuronal, é calculado um escalonamento, de forma a que a gama de valores típica, para cada um dos pixels, se situe no intervalo  $[-1, 1]$ .

---

**Figura 3.6** Uma imagem a cores como a imagem central pode ser representada por variadíssimos espaços de cor. As três imagens da coluna da esquerda representam as componentes RGB da imagem central. As três imagens da coluna da direita representam as componentes HLS da imagem central.

---



### 3.2.2 Janela de varrimento

Como vimos atrás, as imagens são percorridas pixel a pixel por uma janela que delimita uma sub-região centrada num pixel, o qual pretendemos classificar. Na verdade e de acordo com o desdobramento da imagem em três planos de cor, não existe uma só janela mas sim três, uma para cada plano. Uma estratégia adoptada por nós com vista a diminuir a dimensionalidade do padrão de entrada, minimizando os efeitos já referidos, foi a de considerar uma janela com um determinado tamanho que abrangesse uma sub-região considerável da imagem, mas na qual nem todos os pixels por ela abrangidos seriam entregues à rede neuronal.

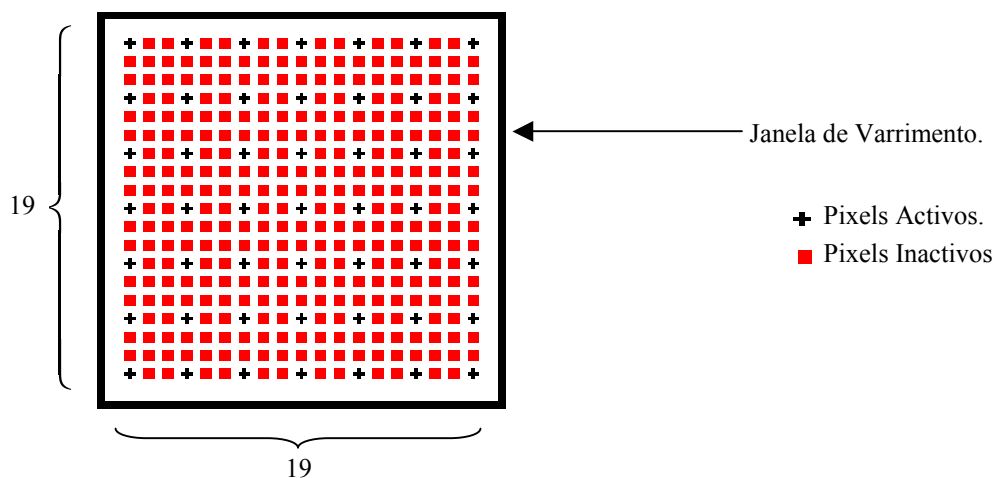
Esta abordagem implica alguma perda de informação, mas ela tem como suporte o facto de que, para a determinação da pertença de um pixel à região leucócito, informação muito detalhada de como evolui a região circundante não é determinante.

De facto, se não se adoptasse alguma medida de redução do tamanho dos padrões de treino, uma quantidade de memória da ordem dos Gbytes seria facilmente requerida pelos ficheiros que contêm os padrões de treino. Isto motivaria um constante acesso ao disco rígido da máquina que suporta o nosso protótipo, pelo menos durante a fase de aprendizagem, tornando o treino de uma lentidão incomportável. Um dos arranjos utilizados foi o da figura 3.7.

---

**Figura 3.7** Janela de varrimento com 19x19 pixels.

---



Como pode ver-se, a janela tem um tamanho de 19x19 pixels, mas todos os pixels entregues à rede (activos) encontram-se intercalados por dois pixels que não são injectados na rede (inactivos). Isto permite uma redução no número de entradas da rede de acordo com o seguinte: consideremos  $j$  como sendo o número de pixels laterais de uma janela quadrada e  $l$  como sendo o número de pixels inactivos que intercalam dois pixels activos consecutivos. Então, o número de pixels activos será dado por:

$$n\_Pixels\_Activos = \left( \text{Int}\left(\frac{j}{l+1}\right) + 1 \right)^2 \quad (3.1)$$

O número de entradas que a rede deverá ter será dado por:

$$n\_Entradas\_rede = n\_Pixels\_Activos * 3 \quad (3.2)$$

De acordo com estas equações, a janela da figura 3.7 terá 49 pixels activos. E o número de unidades de entrada que a rede neuronal terá que possuir será 147. Desta forma, consegue-se reduzir a dimensão do padrão de entrada, mantendo em simultâneo alguma informação sobre o aspecto das regiões próximas e menos próximas que circundam o pixel central (pixel a julgar). Se esta técnica não fosse utilizada, todos os pixels da janela da figura 3.7 seriam entregues à rede neuronal, fazendo com que o número de entradas da rede aumentasse para 1083 ( $19*19*3$ ) unidades, o que sem dúvida representa um acréscimo considerável na dimensão do padrão de entrada e dos problemas que daí advêm.

A escolha do tamanho da janela, além de estar relacionada com os problemas computacionais e com o número de padrões de treino necessários, está também relacionada com a abrangência de regiões mais distantes do pixel central que podem constituir informação importante para a segmentação. A procura de um tamanho de janela que proporcione uma segmentação correcta, mantendo as performances do sistema minimamente satisfatórias,

levou à experimentação de janelas com diferentes tamanhos e com diferentes números de pixels inactivos intercalados com pixels activos.

Se a janela for muito pequena, a rede não terá informação suficiente para julgar o pixel central, pois irá basear-se apenas no grau de tonalidades dos poucos pixels existentes. No entanto, se o tamanho da janela for considerável, a rede neuronal pode aperceber-se não só de tonalidades próximas do pixel central, mas também do aspecto geométrico que resulta da distribuição espacial de tonalidades mais distantes, resultando numa segmentação eficiente, mais imune às variações de tonalidades das imagens.

Por outro lado, como já dissemos, uma janela maior é limitada por questões de desempenho e hardware. Como é evidente, os tamanhos das janelas de varrimento são sempre relativos às escalas das imagens onde elas se encontram sobrepostas. Se as imagens das células estiverem numa escala pequena, então também não necessitamos de janelas muito grandes para conseguir abranger sub-regiões que facilitam a classificação. Se as escalas forem grandes, as janelas devem também aumentar para se conseguir abranger sub-regiões que com o seu aspecto ajudam a rede neuronal na classificação.

Podemos perguntar então por que é que não usamos imagens das células com escalas mais pequenas. A resposta é que há um limite: imagens das células numa escala demasiado pequena implica perdas e deformações significativas de características que podem ser fundamentais para uma segmentação e principalmente para a posterior classificação. Diminuir a escala das imagens implica diminuir a máxima frequência espacial relativa, o que provoca perdas de detalhes geométricos, que no processo de classificação dos leucócitos são determinantes.

Uma heurística que nos pareceu razoável para que a aprendizagem da rede se baseasse no aspecto geométrico da região que circunda o pixel a julgar seria considerar o tamanho da janela de varrimento igual ao diâmetro máximo que uma região leucócito possa ter, para uma determinada escala de imagens. Só que mais uma vez fomos limitados pelo hardware, não permitindo que esta heurística fosse cumprida à risca. Ficámos assim pela utilização de janelas que em vez de cobrirem o maior leucócito por completo, apenas cobrem aproximadamente  $\frac{1}{2}$  deste. Isto corresponde ao uso de janelas 25x25 para imagens que são obtidas num microscópio com uma ampliação de 10x40. O número de pixels inactivos entre

dois activos consecutivos foi obtido com base nos pontos já focados, principalmente o computacional e o eficaz, e situou-se no valor dois.

Apesar de todas estas reduções dos dados, tentando-se sempre suprimir os redundantes e manter os essenciais, a segmentação neuronal mostrou-se eficiente e só não foi possível obter melhores resultados, nomeadamente nos problemas de variabilidades de tonalidades, devido às limitações do nosso hardware que impossibilita o uso de estruturas neuronais de maior dimensão e a utilização de conjuntos de padrões de treino mais vastos, uma vez que se tratava de um vulgar PC.

Muitas outras abordagens no que respeita à redução da dimensionalidade poderiam ter sido usadas. Por exemplo, em vez de fornecermos à rede os valores de cada pixel activo, poderíamos fazer a média dos valores dos pixels conexos aos pixels a que chamamos activos e fornecer à rede esse valor. Ou então usar processos de extracção de características, como descritores de regiões ou de textura ou transformar a imagem para o domínio das frequências e eliminar as frequências mais elevadas. No entanto, não nos parece que estas possíveis abordagens apresentassem resultados muito diferentes daqueles que foram obtidos da forma já descrita, a qual é computacionalmente mais simples.

### 3.2.3 Extracção de cada leucócito na imagem.

As imagens que desejamos segmentar podem conter uma qualquer quantidade de leucócitos. Quer isto dizer que teremos de ter um mecanismo de extracção que nos permita retirar das imagens originais cada uma das regiões assinaladas pela segmentação neuronal. Este procedimento encontra justificação na necessidade de classificar cada um dos leucócitos encontrados durante a segmentação. Os pixels de cada uma das regiões glóbulo branco serão analisados pelo processo de classificação de forma a determinar que classe de leucócito define essa região. É por este motivo que a diferenciação dos leucócitos está dependente directamente da exactidão da segmentação.

Como já sabemos, o resultado da segmentação neuronal é registado numa nova imagem. Os pixels que a rede julga pertencer às regiões leucócito são assinalados a branco, os

restantes a negro. Nesta imagem binária, existem vários aglomerados de pixels. Alguns destes aglomerados são pequenas regiões que foram erradamente segmentadas pela rede neuronal; outros são de facto regiões que assinalam os leucócitos. As regiões demasiado pequenas, por exemplo formadas por um, dez, quarenta pixels, não podem representar leucócitos, pelo que são imediatamente eliminadas. Esta eliminação constitui, na verdade, um primeiro processo de classificação das regiões segmentadas, no qual só passam à fase seguinte as regiões que tiverem uma área mínima suficiente para poderem ser consideradas como possíveis leucócitos.

Consideremos então a imagem de células da figura 3.8, com a sua respectiva imagem resultado da segmentação neuronal. Repare-se que existem nesta imagem quatro leucócitos que foram minimamente bem segmentados pela rede neuronal. O processo de extracção de cada uma das regiões (leucócitos) funciona com base numa sequência de passos que descrevemos a seguir:

1. Percorremos a imagem resultado linha a linha de cima para baixo e pixel a pixel da esquerda para a direita.
2. Se durante o varrimento anterior for encontrado algum pixel a branco, esse pixel pertencerá forçosamente ao contorno de uma região segmentada. Então interrompemos o varrimento do passo anterior e, com base num algoritmo tipo “chain-code” [Jähne, 1997], assinalamos todos os pixels pertencentes ao contorno dessa região. Este contorno pode então ser transferido para a imagem original, permitindo a inspecção visual da qualidade da segmentação.
3. Devido aos erros da segmentação, existem pixels internos ao contorno encontrado no passo 2 que não foram assinalados. Como as regiões leucócitos são sempre regiões completamente preenchidas, esses pixels devem pertencer à região e por isso devem ser também extraídos para ajudarem no processo de classificação de leucócitos. Utilizando um processo de preenchimento (fill), todos os pixels internos ao contorno detectado no passo 2 são assinalados e, se o seu número permitir considerá-los como constituintes de uma região leucócito, serão entregues ao processo de classificação de leucócitos, para

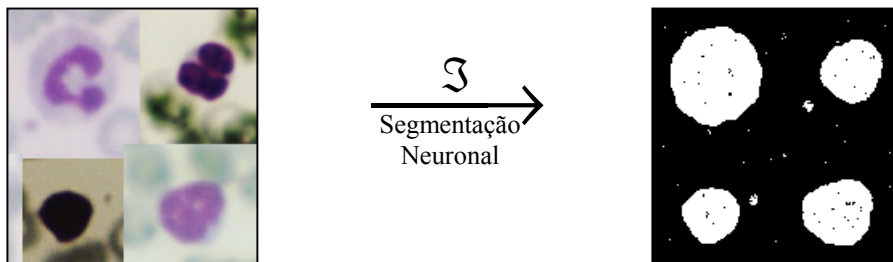
identificação destes. Esta entrega é feita facilmente através do mapeamento directo dos pixels assinalados na imagem resultado para a imagem original.

4. Para que a procura de outros leucócitos possa continuar, a região que se acabou de extrair deverá ser apagada da imagem resultado, evitando-se assim repetir a extracção de uma região já retirada anteriormente.
5. Voltar ao passo 1 até que a imagem seja toda percorrida.

---

**Figura 3.8** Resultado ainda em bruto da segmentação de uma imagem com quatro leucócitos.

---



### **3.3 Gerador de padrões de treino**

Vejamos agora como podemos obter os padrões de treino a serem entregues à rede para proporcionar a aprendizagem. Como já sabemos, o tipo de rede que vamos usar é uma *feedforward* treinada com um algoritmo *backpropagation*. Vimos no capítulo anterior que este tipo de rede se enquadra no tipo de aprendizagem supervisionada. Isto significa que por cada iteração de treino que executamos teremos que apresentar à rede um par de padrões de treino. Par este que é constituído pelo padrão de entrada e pelo padrão de saída. O padrão de entrada é aplicado às unidades da primeira camada da rede neuronal. O padrão de saída, também conhecido por padrão alvo, representa a resposta que se pretende que a rede neuronal aprenda para aquele padrão de entrada. Este padrão é aplicado ao comparador de erro, que no nosso protótipo adquiriu a forma de uma função<sup>16</sup>, a soma dos quadrados das diferenças.

$$E = \frac{1}{2} \sum_{k=1}^{m_n} (T_k - Y_{(n,k)})^2 \quad (3.3)$$

Este comparador devolve à rede neuronal informação sobre a distância entre o vector de saída pretendido e o obtido, para o actual padrão de entrada, permitindo ao processo de aprendizagem ajustar os pesos da rede neuronal de forma a minimizar esta distância.

Pelos motivos já mencionados, a diversidade de padrões de treino deverá ser a maior possível. Este requisito exige algum saber na geração dos padrões de treino. Primeiro: as limitações de hardware, temporal e espacial, impõem um número máximo de padrões. Segundo: os padrões escolhidos deverão representar da forma mais completa possível e sem redundâncias os arranjos que permitem a correcta classificação do pixel central.

O nosso gerador de padrões de treino serve-se de imagens de células (imagens de treino) para extrair os padrões de entrada e de imagens que nós designamos por resultado, para extrair os padrões de saída. Estas imagens resultado não são nem poderiam ser ainda as

---

<sup>16</sup> Embora esta função de custo se mostre suficientemente funcional para a maior parte dos problemas, outras funções que possam traduzir de uma forma mais apropriada (de acordo com a aplicação) o erro entre os padrões pretendidos e os obtidos podem ser empregadas.

imagens obtidas pelo processo de segmentação, mas sim imagens em que nós assinalámos (pintámos) manualmente a localização dos leucócitos.

A obtenção dos padrões de entrada resulta do varrimento de uma janela idêntica à já referida, ao longo de uma imagem de treino. Imagem esta que é por nós organizada e que contém diversos tipos de leucócitos, obtidos em distintas lâminas com diferentes colorações. De cada vez que a janela de varrimento é posicionada centralmente num pixel da imagem de treino, são transferidos para um ficheiro os valores dos pixels abrangidos pela janela. No entanto, para reduzir o tamanho do ficheiro e minimizar a possibilidade de redundância de padrões, podemos no nosso protótipo extrair da imagem de treino padrões não de todos os pixels mas apenas de alguns. A ideia é idêntica à utilizada na janela de varrimento e consiste em considerar na imagem de treino pixels intercalados por um  $n$  número de pixels que não serão considerados.

Analisemos o problema do tamanho dos ficheiros que guardam os padrões de treino. Consideremos uma imagem de treino com 68 leucócitos e um tamanho de 1270\*580 pixels. Se cada pixel originar um padrão de treino, teremos 736600 padrões. Se a janela de varrimento tiver 25x25 pixels com dois pixels inactivos intercalados com os activos, então a cada um dos 736600 pixels da imagem corresponderá um padrão com 243 Bytes (equação 3.2). Multiplicando o número de Bytes exigido pela janela de varrimento pelo número de pixels da imagem de treino, obtemos o tamanho do ficheiro que contém os padrões de treino e neste caso será de 178993800 Bytes (174798 KBytes)! Por exemplo, no nosso caso, a máquina que tínhamos disponível tinha apenas 128Mbytes de RAM. Isto faz com que o acesso ao disco seja permanente; resultado: um treino neuronal que demora meses!...

Se apenas gerarmos padrões pixel sim pixel não, linha sim linha não, o número de padrões é reduzido para 184150 e o ficheiro passa a ter um tamanho de apenas 44748450 Bytes (43699 KBytes), tornando pelo menos no nosso caso o treino já possível temporalmente. Certo é que todas estas tentativas de redução de informação constituem um acréscimo em termos de risco de falha na aprendizagem. Mas parece-nos que o facto de não considerarmos todos os pixels da imagem de treino não aumenta significativamente esse risco, pois pixels vizinhos acabam por proporcionar padrões de treino muito idênticos que em grande parte acabam mesmo por se tornar redundantes.

Mas para termos hipóteses de reduzirmos ainda mais o tempo de aprendizagem ou em alternativa aumentarmos outros factores como o tamanho da janela de varrimento beneficiando das vantagens que daí advêm, pensámos e criámos ainda outro método, voltando a diminuir o número de padrões sem prejudicar a aprendizagem neuronal ou até mesmo melhorando-a. O método foi o seguinte: em vez de gerar padrões por todas as regiões da imagem de treino, assinalamos manualmente, numa outra imagem (imagem piloto), as regiões sobre as quais os padrões de treino devem ser gerados. Desta maneira evitamos estar a gerar padrões repetitivos sobre regiões similares. Esta situação é mais evidente nas partes das imagens que constituem os glóbulos vermelhos e o fundo. Além disso, a ocupação de área destas regiões nestas imagens é superior à área ocupada pelas regiões leucócitos. Esta técnica da imagem piloto permite uma maior equalização entre o número de padrões gerados sobre estes dois tipos de regiões, aumentando a probabilidade de sucesso da aprendizagem neuronal. Esta selecção empírica das regiões significantes para a aprendizagem produz uma redução no número de padrões de treino para aproximadamente metade.

Estes métodos, que se baseiam na eliminação de informação redundante, quando associados, representam uma redução do número de padrões de pelo menos oito vezes sem prejudicar a aprendizagem neuronal e permitem conciliar as performances na aprendizagem com os limites computacionais.

Os padrões de treino de saída são obtidos a partir de uma imagem binária que nós manualmente construímos. A construção desta imagem binária resume-se à substituição de pixels numa réplica da imagem de treino de entrada. Nesta imagem, com a ajuda de um programa de manipulação gráfica, pintamos a branco os pixels pertencentes às regiões leucócitos. Quando geramos o ficheiro de padrões de saída, os pixels pintados a branco irão gerar padrões com o valor de 0.8; os restantes irão ser guardados com o valor de -0.8.

A correspondência entre os padrões de entrada e os de saída está perfeitamente sincronizada. Ou seja, se gerarmos um padrão de entrada pela sobreposição da janela de varrimento sobre a imagem treino de entrada  $I$  no pixel  $(k,l)$ , então a resposta a este padrão será encontrada na imagem treino de saída  $O$  no pixel  $(k,l)$ .

A apresentação iterativa sequencial ou aleatória à rede neuronal do conjunto de pares de padrões de treino, obtidos da forma descrita anteriormente, possibilita a aprendizagem para a segmentação neuronal. A apresentação dos pares de padrões de uma forma sequencial à rede

neuronal pode trazer dificuldades na aprendizagem, principalmente no método de actualização de pesos do tipo *on-line*, no qual os pesos são actualizados padrão a padrão. Isto porque a geração dos padrões faz-se de uma forma sequencial sobre as imagens de treino de entrada e de saída, provocando o aparecimento de seqüências mais ou menos longas de padrões pertencentes a uma mesma classe das duas possíveis (classe leucócito/ classe não-leucócito), podendo prejudicar a convergência da rede neuronal. Para se reduzir este efeito, o nosso algoritmo de treino foi equipado com um seleccionador aleatório dos pares de padrões de treino.

### **3.4 Estrutura da rede neuronal *backpropagation* usada**

Embora as redes neuronais *feedforward* com uma só camada escondida possam aprender a representar qualquer função [Haykin, 1999], na prática o uso de redes neuronais com mais do que uma camada escondida pode trazer melhorias na performance com que se atinge a convergência [Fausett, 1994]. Só que também aqui as limitações acrescidas pelo suporte físico acabam por tornar mais significativo o prejuízo que o benefício. Eliminámos assim essa hipótese.

Ainda que tivéssemos tentado usar, com o intuito de diminuir a estrutura neuronal e assim facilitar a sua convergência, estruturas não completas, atribuindo a três sub-redes os três planos H, L, e S respectivamente, ficou patente após experimentação que o desempenho neste caso era sacrificado. Optou-se então por uma estrutura de conectividade total entre camadas sucessivas de unidades.

Durante as sucessivas vezes que treinámos a rede neuronal, além de ajustarmos os vários parâmetros com ela relacionados, tentámos também modificar o número de neurónios da camada escondida. Com o intuito de a rede poder aprender a segmentar com base nas distribuições mais particulares de pixels contidos na janela de varrimento, foi aumentado o número de neurónios da camada escondida. Este procedimento, apesar de trazer ligeiras melhorias ao resultado final, acabou por ser abandonado, devido às elevadas exigências

computacionais. Essas exigências acabavam por reflectir-se tanto na fase de treino como na fase posterior de segmentação.

Não esqueçamos também que associados ao aumento do número de neurónios temos os problemas da dimensionalidade, que obrigam a treinar a rede com um maior número de padrões de treino. A nossa dificuldade na obtenção de um número de padrões de treino que acompanhasse o crescimento da rede também não contribuiu para a melhoria significativa dos resultados ao aumentar o número de neurónios da rede.

Assim, experimentalmente chegou-se a uma solução de compromisso ao adoptar apenas dois neurónios para a camada escondida. Este número revela que a segmentação é essencialmente feita com base no número de pixels que apresentam uma determinada cor e luminância, e não tanto na distribuição desses pixels na janela de varrimento. Esta desvantagem pode ser mais sentida em imagens que possuem tonalidades muito distintas das imagens de treino. Apesar de tudo, os resultados obtidos com este número de neurónios são satisfatórios.

## Capítulo 4

### O problema da classificação

#### 4.1 Descrição do problema

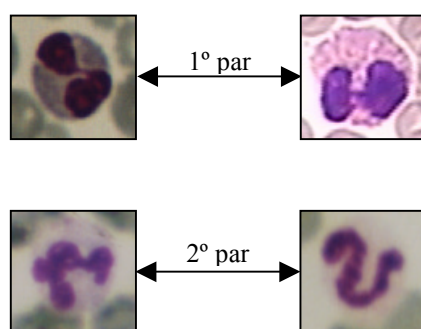
Quando um especialista em sangue explora opticamente uma lâmina à procura de leucócitos, executa uma tarefa comparável à da segmentação, já que realiza uma detecção visual das regiões que formam os leucócitos. Imediatamente a seguir à detecção do primeiro leucócito, o especialista observa-o mais pormenorizadamente e com base no seu aspecto geométrico determina a que classe de leucócitos pertence.

A dificuldade que esta tarefa apresenta, mesmo para um especialista, prende-se essencialmente com o facto de as características geométricas definidoras do tipo de leucócito serem pouco claras e mesmo ambíguas, como já havíamos referido no primeiro capítulo. Por exemplo, repare-se nos dois pares de células da figura 4.1. Qual deles agrupa células de

---

**Figura 4.1** Dois pares de leucócitos. No primeiro par temos dois eosinófilos e no segundo par temos dois neutrófilos.

---



classes distintas?

Na verdade e curiosamente, nenhum dos dois pares de imagens agrupa leucócitos de classes diferentes. No primeiro par podemos observar dois eosinófilos e no segundo par dois neutrófilos. Tornam-se assim evidentes as dificuldades na classificação dos leucócitos.

Se analisarmos a acção de um especialista, verificamos o poder generalista que o nosso cérebro aplica para conseguir uma correcta classificação. A independência à rotação, a independência à variação de tonalidades e até mesmo a independência à escala são apenas alguns dos pontos inerentes a essa generalização. Esta capacidade pode ainda ser observada doutra forma: por vezes, certas características aparecem esvanecidas. Por exemplo, nos dois eosinófilos apresentados no par 1 da figura 4.1, a granulação típica apresentada pelo citoplasma é notória na célula da direita. Esta característica leva de imediato um especialista a classificá-la como um eosinófilo. Essa granulação já não é de forma alguma evidente no eosinófilo da esquerda, obrigando a uma observação mais cuidada do aspecto do núcleo. Este é realmente difícil de descrever e por vezes passível de ser confundido com o núcleo de alguns neutrófilos.

Mais uma vez a dificuldade algorítmica, e até mesmo linguística, em descrever estas subtis características faz-nos recorrer às redes neuronais supervisionadas.

No capítulo anterior descrevemos a técnica que usámos para a detecção dos leucócitos nas imagens. Cada um dos aglomerados de pixels assinalados por esta são pressupostamente regiões leucócitos. Temos então que determinar com um mínimo de erro a qual classe de leucócitos é que pertence cada uma das regiões segmentadas. Assim, e depois de termos classificado 100 dos leucócitos presentes na lâmina que contém a amostra, podemos apresentar os resultados da contagem diferencial entre as diversas classes de leucócitos aí presentes.

Uma dificuldade que se nos colocou quando tomámos a decisão de usar, também na classificação, redes neuronais foi a forma de introduzir na rede a região leucócito. É que enquanto no problema da segmentação o conjunto de pixels a classificar tinha sempre uma cardinalidade constante, pois a janela de varrimento mantinha o seu tamanho ao longo do seu percurso na imagem, implicando também uma rede neuronal com um número de unidades de entrada constante, no processo de classificação o número de pixels que constituem a região leucócito varia significativamente, não permitindo uma transposição directa para as entradas

da rede neuronal. Embora esta dificuldade pudesse ser transposta, por exemplo, considerando uma área quadrada cujo tamanho garantisse a abrangência total da maior região leucócito possível de aparecer numa imagem. Então cada pixel dessa região quadrada corresponderia a uma unidade de entrada da rede neuronal, faríamos alinhar o centro de massa da região leucócito com o centro da área quadrada e entregar-se-ia o conteúdo desta área receptiva à rede neuronal. No entanto, esta abordagem parece-nos algo restritiva, pois estaríamos limitados a um tamanho máximo da região leucócito, não permitindo, numa perspectiva futura, a aprendizagem de quaisquer outras classes de leucócitos. E seria ainda computacionalmente ineficiente, devido ao tamanho excedente da área receptiva. Para já não falarmos das inúmeras dificuldades que teríamos a treinar a rede neuronal para obter as já referidas capacidades generalistas.

Decidimos, por isso, entregar à rede neuronal não a região leucócito directamente pixel a pixel, mas o resultado numérico fornecido por descritores. Assim, o número de entradas da rede neuronal será igual ao número de descritores, eliminando-se o problema acima referido e beneficiando-se ainda, desta forma, de uma redução na dimensão da rede neuronal [Bishop, 1995].

Estes descritores traduzem numericamente variadíssimos aspectos de uma textura ou de uma região. Tentou então seleccionar-se empiricamente os mais adequados para a resolução do nosso problema, já que uma boa escolha pode beneficiar a capacidade de generalização que a rede neuronal apresentará depois de treinada [Bishop, 1995].

## 4.2 Descritores

Os descritores que usámos para codificar numericamente os atributos geométricos das células dividem-se em três grupos: descritores de regiões, descritores de textura e descritores primários. Com os descritores de regiões pretende-se traduzir para a rede neuronal aspectos geométricos da região citoplasma (pelo menos nos leucócitos que o possuem) e aspectos geométricos da região núcleo. Os descritores de textura são aqui usados com o propósito

principal de codificar numericamente a intensidade de granulação que algumas classes de leucócitos apresentam, tanto ao nível do citoplasma como do núcleo.

Os descritores primários são constituídos por parâmetros como o perímetro do citoplasma, o perímetro do núcleo, rácios entre estes, etc.. Estes descritores permitem também uma caracterização numérica de importantes atributos geométricos das células, contribuindo para a facilidade da aprendizagem e generalização neuronal.

Houve ainda o cuidado de utilizar normalizações aplicadas a alguns descritores de regiões, com o intuito de conseguir a invariância à translação, à rotação e à escala [Masters, 1994]. Ao ter-se adoptado os descritores de regiões com mecanismos que os tornam invariantes à escala, a gama de valores resultantes destes é assim mais pequena, estável e coerente com o aspecto geométrico das células. Apesar disto, optou-se por integrar também no conjunto de variáveis a entregar à rede neuronal uma que representa a área das células (descritor primário), pois o tamanho relativo entre as várias classes de leucócitos constitui um atributo de diferenciação que em algumas situações de incerteza pode ser determinante para a correcta classificação. Isto pode ser feito, já que a escala das imagens fornecidas pelo microscópio se mantém sempre constante.

#### 4.2.1 Descritores de regiões

Existem vários modos de descrever numericamente figuras bidimensionais. Um método possível recorre à codificação do contorno fechado que limita a figura, como é o caso dos descritores de Fourier [Pratt, 1991]. Com estes métodos consegue-se codificar os detalhes das formas, razoavelmente bem; contudo são muito sensíveis ao ruído [Masters, 1994]. Ora, o processo de segmentação introduz em alguns casos deformações que se denotam principalmente ao nível do contorno da região. Por isso, decidimos empregar um outro método, que para descrever uma figura se baseia em toda a sua superfície e não apenas no seu contorno, tornando-se assim mais imune ao ruído. Este método pode ser implementado através dos momentos. Os momentos de pequena ordem são, relativamente aos de ordens elevadas, pouco sensíveis a detalhes geométricos e mais fáceis de calcular. A sua transdução

geométrica resume-se a um valor, facilitando a sua transposição para a entrada da rede neuronal e é possível através de normalizações torná-los invariantes à translação, rotação e escala. São assim adequados para a resolução do nosso problema.

Uma outra forma de descrever regiões é através de rácios entre os descritores primários da região. A equação seguinte serve-se do perímetro e da área da região para produzir uma descrição numérica desta.

$$C = \frac{p^2}{A} \quad (4.1)$$

Este descritor, chamado circularidade [Jähne, 1997], é também independente à rotação e escala, como é evidente, já que resulta da razão entre o quadrado do perímetro e a área da região a descrever.

#### 4.2.1.1 Momentos

A definição mais simples de um momento que usa informação bidimensional discreta é a seguinte:

$$M(p, q) = \sum_x \sum_y x^p y^q f(x, y) \quad (4.2)$$

na qual **p** e **q** representam a ordem do momento segundo a direcção dos **xx** e segundo a direcção dos **yy** respectivamente e  $x$  e  $y$  as coordenadas de um pixel da imagem  $f(x,y)$ . Esta imagem  $f(x,y)$  tanto pode estar representada de uma forma binária como em níveis de cinzento. Se a imagem considerada for binária, os momentos dependerão apenas das formas contidas na imagem; se for em níveis de cinzento, os momentos não dependerão apenas das

figuras contidas na imagem mas também da distribuição das amplitudes dos pixels ao longo dessas figuras.

Qual a opção a seguir para obter o melhor desempenho na classificação das células? À primeira vista, a utilização de imagens do núcleo e citoplasma em níveis de cinzento para o cálculo dos momentos parece ser a mais adequada, devido ao maior grau de informação que estas possuem. No entanto, como já sabemos, os pixels que constituem estas regiões estão sujeitos a variações e ruído, o que provoca instabilidade nos valores dos momentos. Por outro lado, o uso de imagens binárias do núcleo e do citoplasma reduz estes problemas, mas ignora a informação que provinha da distribuição dos pixels ao longo destas regiões. Pensamos que a melhor solução, e já que o problema da dimensionalidade da rede neuronal não se coloca com tanta importância como no caso da rede de segmentação, será a de calcular os momentos para as duas abordagens e deixar que a rede neuronal decida com base nestes dois momentos.

Outra tentativa possível para atenuar os problemas causados nos momentos pelo ruído das imagens em níveis de cinzento é a de calcular os centróides e fazer a normalização à escala com base nas imagens binárias (Equação 4.8) e só depois, usando estas normalizações, calcular os momentos das imagens em níveis de cinzento [Masters, 1994]. Isto faz com que os valores dos momentos se tornem menos sensíveis e em simultâneo dependam da distribuição dos pixels na figura.

Os valores fornecidos pela equação (4.2) são relativos à origem do referencial onde a figura se encontra, ou seja, são dependentes do local onde a figura se situa na imagem. Para se eliminar esta dependência, os momentos são calculados relativamente às coordenadas do centróide da figura incluída na imagem  $f(x,y)$ . As coordenadas do centróide são obtidas através das seguintes equações:

$$\mu_x = \frac{M(1, 0)}{M(0, 0)} \quad (4.3)$$

$$\mu_y = \frac{M(0, 1)}{M(0, 0)} \quad (4.4)$$

Os momentos referidos são obtidos pela próxima equação e dá-se-lhes o nome de momentos centrados.

$$U(p, q) = \sum_x \sum_y (x - \mu_x)^p (y - \mu_y)^q f(x, y) \quad (4.5)$$

#### 4.2.1.2 Momentos invariantes à escala

Para que os momentos não variem com o tamanho da figura, a equação (4.5) é normalizada de acordo com a equação seguinte [Masters, 1994]. Esta normalização é válida para imagens binárias e  $M(0, 0)$  representa a área total da figura contida na imagem.

$$V(p, q) = \frac{U(p, q)}{\sqrt{M(0, 0)}^{(p+q+2)}} \quad (4.6)$$

Se utilizarmos imagens em níveis de cinzento, teremos que ter em conta que a figura apresenta uma profundidade codificada em níveis de cinzento, como se de um volume se tratasse. Isto implica que  $M(0,0)$  deixa de representar uma área e passa a representar um volume, que é o somatório de todos os valores dos pixels que formam a figura. Neste caso, a equação usada para a normalização à escala é a seguinte [Masters, 1994]:

$$V(p, q) = \frac{U(p, q)}{\sqrt[3]{M(0, 0)}^{(p+q+3)}} \quad (4.7)$$

Como já foi referido atrás, os momentos de uma figura em níveis de cinzento podem ser calculados recorrendo aos centróides obtidos com os momentos da imagem binária dessa

figura. Sendo assim, a normalização à escala também terá que levar em conta este ponto, resultando numa outra equação [Masters, 1994] de normalização em que  $M(0,0)$  representa novamente a área total da figura:

$$V(p, q) = \frac{U(p, q)}{\sqrt{M(0, 0)^{(p+q+3)}}} \quad (4.8)$$

#### 4.2.1.3 Momentos invariantes à rotação (I)

A importância de se possuírem descritores invariantes à rotação é óbvia, pois qualquer tipo de leucócito presente numa imagem terá que ser bem classificado independentemente da sua orientação geométrica. Estudámos dois métodos que permitem tornar os momentos invariantes à rotação; porém, ambos podem apresentar alguns problemas, pois, de facto, este género de normalização é o mais difícil de obter.

Um método baseia-se na correcção da orientação actual da figura [Masters, 1994]. Os momentos são normalizados tendo em conta o ângulo que o eixo de maior inércia da figura faz com o eixo horizontal. É como se rodássemos a figura de maneira a que o seu eixo de maior inércia coincidisse com o eixo horizontal. Desta forma, os momentos tornam-se independentes da orientação inicial da figura. Os problemas inerentes a este processo aparecem principalmente quando em certas figuras o eixo de maior inércia é idêntico ao de menor inércia. Se isto acontecer, é muito provável que a identificação do maior eixo possa falhar, por exemplo devido a ruído, provocando a incorrecta normalização dos momentos.

O ângulo que o eixo de maior inércia faz com a horizontal pode ser obtido a partir dos valores próprios ( $\lambda_1$  e  $\lambda_2$ ) da matriz seguinte [Masters, 1994] cujos elementos são momentos de segunda ordem:

$$V = \begin{bmatrix} V(2, 0) & V(1, 1) \\ V(1, 1) & V(0, 2) \end{bmatrix} \quad (4.9)$$

Se considerarmos  $a=V(2,0)$ ,  $b=V(1,1)$  e  $c=V(0,2)$ , então a equação (4.13) permite-nos calcular o referido ângulo [Masters, 1994]:

$$r = \sqrt{(a - c)^2 + 4b^2} \quad (4.10)$$

$$\lambda_1 = \frac{1}{2}(a + c + r) \quad (4.11)$$

$$\lambda_2 = \frac{1}{2}(a + c - r) \quad (4.12)$$

$$\theta = \tan^{-1}\left(\frac{\text{MAX}[\lambda_1, \lambda_2] - c}{b}\right) \quad (4.13)$$

Temos, no entanto, que ter em atenção que existe ainda uma ambiguidade a ser resolvida. O ângulo obtido indica, de facto, o quanto é que temos que rodar a figura para atingir a horizontal; mas, se a esse ângulo adicionarmos  $180^\circ$ , a figura continua a manter-se na horizontal. Se a figura não for simétrica, os momentos irão depender desta adição angular. Para resolver este problema, tenta procurar-se o lado da figura com o aspecto mais *cheio*, rodando-se de seguida a figura  $180^\circ$  ou não de forma a que esse lado se situe sempre para a esquerda do referencial vertical. O lado mais *cheio* pode ser encontrado através do cálculo do momento de terceira ordem  $M(3,0)$ . Se este momento for negativo, significa que teremos de adicionar ao ângulo calculado em (4.13)  $180^\circ$ , pois significará que o lado mais *cheio* se encontrará à direita do referencial vertical.

Com o ângulo obtido através da equação (4.13), os momentos são transformados segundo a equação apresentada a seguir [Masters, 1994]:

$$M_{\theta}(p, q) = \sum_{r=0}^p \sum_{s=0}^q (-1)^{q-s} \binom{p}{r} \binom{q}{s} \bullet (\cos \theta)^{p-r+s} (\sin \theta)^{q-s+r} M(p+q-r-s, r+s) \quad (4.14)$$

#### 4.2.1.4 Escalonamento dos momentos para as entradas da rede neuronal

Para se obter um melhor desempenho da rede neuronal, é importante que a média dos valores de cada variável de entrada seja 0 e ainda que a gama de valores destas variáveis se situe dentro de um intervalo aceitável de forma a serem comensuráveis com a gama de valores de entrada da rede neuronal. A forma de se conseguir um escalonamento admissível pode passar por uma análise estatística, por vezes até empírica, dos valores obtidos para o conjunto de padrões de treino, estando então estes sujeitos a algum tipo de transformação, não-linear ou linear ou ambas. Uma transformação possível, e que usámos com os momentos, passa por sujeitar os valores a uma compressão logarítmica, ou, no caso de momentos que podem atingir valores negativos, a uma raiz de índice ímpar.

No nosso caso, depois de analisarmos o universo de valores inerentes ao conjunto de padrões de treino, chegámos à equação (4.15), que nos permite, para os momentos  $M(2,0)$  e  $M(0,2)$ , ter uma gama de valores que está em sua grande parte no intervalo  $[-1, 1]$ , estando assim de acordo com a gama de valores dos neurónios usados.

$$M = \log\left(\frac{m+0.015}{0.087}\right) \quad (4.15)$$

#### 4.2.1.5 Momentos invariantes à rotação (II).

Existe uma outra abordagem para resolver o problema da invariância à rotação. Nesta abordagem, a figura não tem que sofrer qualquer alteração à sua orientação, eliminando-se, por isso, os problemas já referidos e que daí advêm. Trata-se de um conjunto de variáveis [Masters, 1994] cujos valores não dependem da orientação que a figura detém. Porém, esta abordagem também apresenta alguns problemas, verificando-se experimentalmente alguma incoerência dos valores destas variáveis para diferentes orientações de uma mesma figura. No entanto, a disparidade de valores obtidos pelo método anterior, quando o eixo de maior inércia da figura não é evidente, faz com que este último método seja preferível, pelo menos nessas situações. De qualquer forma, devido à dificuldade em decidirmos por um dos dois sem primeiro efectuarmos experimentações, a nossa opção foi calcular os momentos usando ambos, deixando, assim, a rede neuronal aproveitar o melhor de cada um.

A primeira variável (4.16) pretende dar informação sobre o aspecto (excentricidade) da figura a partir da relação entre o eixo de maior inércia e o eixo de menor inércia. Repare-se que  $\lambda_1$  e  $\lambda_2$  são resultado das equações (4.11) e (4.12). As constantes da equação foram ajustadas de forma a garantir o correcto escalonamento de valores para a rede neuronal [Masters, 1994].

$$ASPECT = 2 \log \left( \frac{\frac{\lambda_1}{\lambda_2} - 0.99999}{1.2} \right) \quad (4.16)$$

Outra variável (4.17) independente à rotação baseia-se na soma dos dois eixos de inércia [Masters, 1994].

$$SPREAD = 2 \log \left( \frac{\lambda_1 + \lambda_2 - 0.158}{0.092} \right) \quad (4.17)$$

Um conjunto de variáveis independentes à rotação foram apresentadas por [Hu, 1962]. Estas variáveis utilizam momentos até à terceira ordem que foram normalizados à escala e translação.

$$h1 = V(2, 0) + V(0, 2) \quad (4.18)$$

$$h2 = [V(2, 0) - V(0, 2)]^2 + 4[V(1, 1)]^2 \quad (4.19)$$

$$h3 = [V(3, 0) - 3V(1, 2)]^2 + [V(0, 3) - 3V(2, 1)]^2 \quad (4.20)$$

$$h4 = [V(3, 0) + V(1, 2)]^2 + [V(0, 3) - V(2, 1)]^2 \quad (4.21)$$

$$\begin{aligned} h5 = & [V(3, 0) - 3V(1, 2)] [V(3, 0) + V(1, 2)] \\ & \{ [V(3, 0) + V(1, 2)]^2 - 3[V(0, 3) + V(2, 1)]^2 \} \\ & + [3V(2, 1) - V(0, 3)] [V(0, 3) + V(2, 1)] \\ & \{ 3[V(3, 0) + V(1, 2)]^2 - [V(0, 3) + V(2, 1)]^2 \} \end{aligned} \quad (4.22)$$

$$\begin{aligned} h6 = & [V(2, 0) - V(0, 2)] \{ [V(3, 0) + V(1, 2)]^2 - [V(0, 3) + V(2, 1)]^2 \} \\ & + 4V(1, 1) [V(3, 0) + V(1, 2)] [V(0, 3) + V(2, 1)] \end{aligned} \quad (4.23)$$

As equações seguintes permitem escalonar os seis descritores anteriores, de forma a adequá-los às entradas das redes neuronais [Masters, 1994].

$$nh1 = 10 \log(10h1 - 1.2) \quad (4.24)$$

$$nh2 = 3 \log\left(\frac{h2 + 0.0001}{0.0101}\right) \quad (4.25)$$

$$nh3 = 8 \log\left(\frac{h3 + 0.0005}{0.0025}\right) \quad (4.26)$$

$$nh4 = 2 \log\left(\frac{h4 + 0.0000001}{0.0000101}\right) \quad (4.27)$$

$$nh5 = 1000 \sqrt[3]{h5} \quad (4.28)$$

$$nh6 = 200 \sqrt[3]{h6} \quad (4.29)$$

#### 4.2.2 Descritores de textura

De entre os diversos processos existentes para descrever texturas, optámos pelos que recorrem a modelos estatísticos, pois pareceu-nos serem estes os mais adequados mediante a falta de regularidade de primitivas de textura que as nossas células apresentam.

Usámos, assim, um conjunto de variáveis [Masters, 1994] muito simples que permitem codificar numericamente alguns aspectos das texturas. Os valores fornecidos por estas variáveis provêm da região núcleo, da região citoplasma ou de ambas em simultâneo.

As variáveis que usamos foram as seguintes:

- **Desvio médio**

$$Desv\_Med = \frac{1}{n} \sum_{i=1}^n |reg[i] - media_{reg}| \quad (4.30)$$

Em que  $reg$  representa o conjunto de  $n$  pixels da região a considerar e  $media_{reg}$  a média das intensidades dos pixels dessa região.

- **Desvio padrão**

$$Desv\_Pad = \sqrt{\frac{\sum_{i=1}^n (reg[i] - media_{reg})^2}{n}} \quad (4.31)$$

- **Assimetria (Skewness)**

$$Skewness = \frac{\frac{1}{Desv\_Pad^3} \sum_{i=1}^n (reg[i] - media_{reg})^3}{n} \quad (4.32)$$

- **Curtose (Kurtosis)**

$$Kurtosis = \frac{1}{Desv\_Pad^4} \frac{\sum_{i=1}^n (reg[i] - media_{reg})^4}{n} - 3 \quad (4.33)$$

- **First neighbour contrast**

$$FNC = \frac{\sum_{i=1}^{n-1} |reg[i+1] - reg[i]|}{n-1} \quad (4.34)$$

Enquanto os anteriores descritores desprezam qualquer informação concernente às posições ocupadas pelos pixels, este último fornece o valor médio do módulo das diferenças entre pixels adjacentes, podendo esta diferença ser calculada tanto na direcção vertical como na horizontal.

- **Second neighbour contrast**

$$SNC = \frac{\sum_{i=1}^{n-2} |reg[i+2] - reg[i]|}{n-2} \quad (4.35)$$

Este descritor é idêntico ao anterior. No entanto, as diferenças são calculadas não entre pixels adjacentes mas com um pixel de intervalo.

Também estes descritores foram escalonados com base nos valores obtidos para o conjunto de padrões de treino.

### 4.2.3 Descritores primários

Chamamos descritores primários a parâmetros geométricos elementares que podem ajudar na classificação dos leucócitos.

Os descritores primários por nós usados são aplicados às regiões núcleo e citoplasma separadas por um processo de segmentação descrito no ponto 4.4. Esses descritores primários são seis:

- Intensidade ( $I_{Nu}$ ) média do núcleo
- Intensidade ( $I_{Ci}$ ) média do citoplasma
- Área ( $A_{Nu}$ ) do núcleo
- Área ( $A_{Ci}$ ) do citoplasma
- Perímetro ( $P_{Nu}$ ) do núcleo
- Perímetro ( $P_{Ci}$ ) do citoplasma.

Foram ainda usados rácios entre os descritores primários do núcleo (Nu) e os do citoplasma (Ci).

$$R_I = \frac{I_{Nu}}{I_{Ci}} \quad (4.36)$$

$$R_A = \frac{A_{Ci}}{A_{Nu}} \quad (4.37)$$

$$R_P = \frac{P_{Nu}}{P_{Ci}} \quad (4.38)$$

### 4.3 Descritores entregues à rede

A seguinte tabela mostra a que regiões das células foram associados os diversos descritores vistos anteriormente.

Tabela 4.1

|             | Eq.    | Núcleo<br>(Imagem binária) | Citoplasma<br>(Imagem binária) | Núcleo<br>(Luminância) | Citoplasma.<br>(Luminância) | Célula<br>(Luminância) |
|-------------|--------|----------------------------|--------------------------------|------------------------|-----------------------------|------------------------|
| $M_0(2,0)$  | (4.14) | ×                          | ×                              |                        |                             | $\times^{17}$          |
| $M_0(0,2)$  | (4.14) | ×                          | ×                              |                        |                             | $\times^{17}$          |
| Aspect      | (4.16) | ×                          | ×                              |                        |                             | $\times^{17}$          |
| Spread      | (4.17) | ×                          | ×                              |                        |                             | $\times^{17}$          |
| Nh1         | (4.18) | ×                          | ×                              |                        |                             |                        |
| Nh2         | (4.19) | ×                          | ×                              |                        |                             |                        |
| Nh3         | (4.20) | ×                          | ×                              |                        |                             |                        |
| Nh4         | (4.21) | ×                          | ×                              |                        |                             |                        |
| Nh5         | (4.22) | ×                          | ×                              |                        |                             |                        |
| Nh6         | (4.23) | ×                          | ×                              |                        |                             |                        |
| Circularid. | (4.1)  | ×                          | ×                              |                        |                             |                        |
| Desv_Med.   | (4.30) |                            |                                | ×                      | ×                           |                        |
| Desv_Pad.   | (4.31) |                            |                                | ×                      | ×                           |                        |
| Skewness    | (4.32) |                            |                                | ×                      | ×                           |                        |
| Kurtosis    | (4.33) |                            |                                | ×                      | ×                           |                        |
| FNC         | (4.34) |                            |                                | ×                      | ×                           |                        |
| SNC         | (4.35) |                            |                                | ×                      | ×                           |                        |
| $I_{Nu}$    |        |                            |                                | ×                      |                             |                        |
| $I_{Ci}$    |        |                            |                                |                        | ×                           |                        |
| $A_{Nu}$    |        | ×                          |                                |                        |                             |                        |
| $A_{Ci}$    |        |                            | ×                              |                        |                             |                        |
| $P_{Nu}$    |        | ×                          |                                |                        |                             |                        |
| $P_{Ci}$    |        |                            | ×                              |                        |                             |                        |
| $R_I$       | (4.36) | ×                          | ×                              |                        |                             |                        |
| $R_A$       | (4.37) | ×                          | ×                              |                        |                             |                        |
| $R_P$       | (4.38) | ×                          | ×                              |                        |                             |                        |

<sup>17</sup> Estes momentos foram calculados usando os centróides obtidos a partir dos momentos binários. Assim, a independência à escala foi conseguida através da equação (4.8).

Como pode ver-se na tabela anterior, os descritores foram aplicados ao citoplasma, núcleo e a toda a região célula. Note-se ainda que alguns descritores estão associados às componentes binárias do núcleo e citoplasma. Poderíamos também ter associado estes descritores às componentes de cor das referidas regiões, mas pareceu-nos que nesse caso as capacidades de generalização do sistema saíam diminuídas, devido às variações de tonalidades que estas imagens apresentam.

#### 4.4 Segmentação do Núcleo/Citoplasma

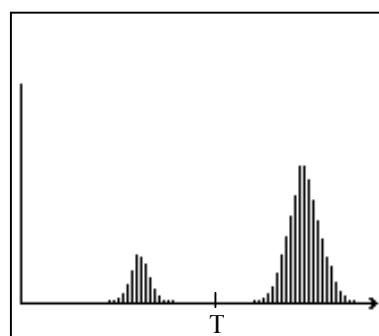
Depois de analisarmos as secções anteriores, torna-se óbvia a necessidade de possuímos um mecanismo que nos permita separar a região núcleo da região citoplasma.

Se repararmos, a luminância dos pixels que constituem o núcleo dos leucócitos é sempre mais baixa que a dos pixels constituintes da região citoplasma. Isto implica que o histograma típico (figura 4.2) de uma região célula, com núcleo e citoplasma, apresente duas zonas mais ou menos bem definidas, dependendo em grande parte da qualidade das imagens.

---

**Figura 4.2** Histograma típico de um leucócito com núcleo e citoplasma.

---



O valor **T** (threshold) deve ser obtido garantindo a separação clara das duas zonas do histograma. No nosso caso **T** é obtido da análise da variação da envolvente do histograma. Com este valor e usando a parametrização seguinte, conseguimos identificar os pixels do núcleo e os pixels do citoplasma. Os pixels assinalados na imagem *reg\_seg* a 1 pertencem ao citoplasma e os que são assinalados a 0 pertencem ao núcleo.

$$reg\_seg[x][y] = \begin{cases} 1 & \text{se } reg[x][y] > T \\ 0 & \text{se } reg[x][y] \leq T \end{cases} \quad (4.39)$$

Existe ainda a situação em que os leucócitos não possuem citoplasma, não podendo a estes, obviamente, aplicar-se a técnica anterior. A técnica que usamos para detectar um leucócito sem citoplasma também recorre ao histograma. Um leucócito que apenas tem núcleo apresenta uma gama dinâmica de histograma, inferior a um leucócito com citoplasma. Se isto acontecer, todo o restante processo de classificação ficará condicionado à não existência de citoplasma.

#### 4.5 Descrição do classificador neuronal

De cada vez que o processo de segmentação assinala um novo leucócito, as intensidades dos pixels que o formam são fornecidas aos vários descritores de acordo com a tabela (4.1). Por sua vez, os valores calculados por meio destes descritores são fornecidos à rede neuronal, as saídas da rede neuronal são então comparadas e dessa comparação resultará a classificação do leucócito (figura 4.3).

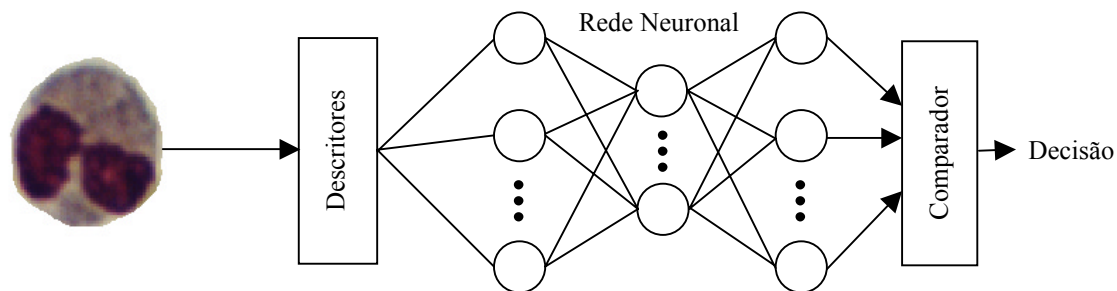
O número de entradas da rede neuronal usada na classificação é igual ao número de descritores usados, enquanto o número de saídas dessa rede neuronal é igual ao número de classes de leucócitos que se pretende que o sistema identifique. Para a realização do nosso

trabalho, as lâminas que nos foram cedidas pelo laboratório de análises não apresentavam classes de leucócitos anormais, o que implica a existência de cinco classes de leucócitos e por conseguinte cinco neurónios na camada de saída da rede neuronal. Na camada escondida foram experimentados diferentes números de neurónios, tendo-se chegado a um valor de 12 neurónios que nos pareceu o mais indicado para conseguir bons resultados.

---

**Figura 4.3** Esta figura esquematiza o nosso classificador neuronal. As características geradas pelo conjunto de pixels que formam o leucócito são fornecidas à rede neuronal através dos descritores geométricos. Cada saída da rede neuronal representará uma classe de leucócito e será comparada com as restantes. A que apresentar um valor mais elevado sugerirá a classe do leucócito.

---



#### 4.5.1 Treino neuronal

O primeiro passo a dar para se poder treinar a rede neuronal é obter os padrões de treino. Cada padrão de treino é obtido sobre uma região leucócito.

Seleccionamos um conjunto de imagens que contêm leucócitos de classes diversas. Submetemos estas imagens ao processo de segmentação. Cada região assinalada é entregue aos descritores. Os valores dos descritores são então guardados em ficheiros. Desta forma, de cada vez que este processo é repetido é acrescentado ao ficheiro um novo padrão de treino de entrada. À medida que são gerados estes padrões de entrada, são também gerados os padrões de saída, embora neste caso seja necessário recorrer a um especialista. Cada leucócito assinalado é identificado pelo especialista, sendo gerado um vector de cinco posições que codifica (tabela 4.2) cada classe de leucócito. Este vector corresponde ao padrão de treino de saída e será posteriormente apresentado aos neurónios de saída da rede neuronal.

Tabela 4.2

|            | V[0] | V[1] | V[2] | V[3] | V[4] |
|------------|------|------|------|------|------|
| Basófilo   | 0.8  | -0.8 | -0.8 | -0.8 | -0.8 |
| Neutrófilo | -0.8 | 0.8  | -0.8 | -0.8 | -0.8 |
| Linfócito  | -0.8 | -0.8 | 0.8  | -0.8 | -0.8 |
| Eosinófilo | -0.8 | -0.8 | -0.8 | 0.8  | -0.8 |
| Monócito   | -0.8 | -0.8 | -0.8 | -0.8 | 0.8  |

##### 4.5.1.1 Validação Cruzada

Para que o treino da rede neuronal seja também conseguido em termos de generalização, foi usada uma técnica de validação cruzada [Haykin, 1999].

A técnica por nós usada obriga a possuir dois conjuntos de padrões distintos. Um, o maior, é usado para treinar a rede neuronal, o outro é usado para validar o referido treino.

Assim de  $n$  padrões de treino em  $n$  padrões de treino é obtido o erro médio relativo aos padrões de validação. No final do treino os pesos a reter são aqueles que apresentam o menor erro para os padrões de validação. Desta maneira evita-se que a rede se especialize em demasia nos padrões de treino, reduzindo a capacidade de generalização. Este fenómeno é também conhecido por *sobre aprendizagem das redes neuronais*.

Lembramos que esta técnica não foi usada no treino da rede de segmentação, porque, nesse caso, o conjunto de padrões é tão extenso e diverso que por si só proporciona a mesma capacidade de generalização que a técnica de validação cruzada.

Porém, no caso da classificação, o conjunto de padrões de treino é muito mais reduzido, podendo esta técnica ajudar a melhorar as capacidades de generalização.

#### 4.5.2 Classificação

Depois da rede treinada, o processo torna-se autónomo. Através dos descritores, cada leucócito encontrado na imagem será avaliado pela rede neuronal. Esta produz um vector na sua saída que, de acordo com a tabela (4.2), identificará o respectivo leucócito. Repare-se que a posição do vector que contém o valor mais elevado (0.8) identifica a classe do leucócito.

Porém, a classificação nem sempre é exequível, pois pode acontecer que os valores contidos no vector existente à saída da rede neuronal se afastem dos valores dos vectores mostrados na tabela 4.2. Resolvemos assim considerar duas situações inclusivas para as quais não é possível identificar com exactidão os leucócitos:

1. A classificação não será feita se a diferença entre os dois valores mais elevados for inferior a um determinado valor  $K$ . Esta condição, quando verdadeira, indica não haver margem suficiente para garantir com precisão a pertença do leucócito à classe assinalada pelo valor mais elevado. É preferível, assim, não classificar. A indicação obtida no nosso protótipo nesta situação é “Não tenho a certeza de que célula se trata”.

2. A classificação também não será feita se o maior elemento contido no vector for inferior a um determinado valor **T**. Esta condição obriga a que a classificação seja feita com base num valor máximo que se aproxima de 0.8. Assim, a probabilidade de o elemento máximo indicar a célula correcta aumenta. Desta forma, o nosso protótipo anuncia: “Não sei qual é a célula”.

O valor de **K** e **T** foram determinados experimentalmente com base num conjunto de células de teste.



## Capítulo 5

### Resultados experimentais

Neste capítulo serão apresentados resultados que irão demonstrar a validade das abordagens descritas nos capítulos anteriores. Na primeira parte deste capítulo, poderemos observar, através de imagens, o comportamento do processo de segmentação. Na segunda parte, iremos apresentar através de uma matriz confusão os resultados obtidos com o processo de classificação.

Para a obtenção dos resultados foram usadas imagens reais que padecem na sua aquisição de qualquer tipo de controlo que assegure a sua estabilidade em termos de cor e brilho. Isto significa que os processos por nós apresentados foram testados até aos seus limites. Estes resultados irão demonstrar isso.

Os resultados aqui apresentados (segmentação e classificação) foram já obtidos depois da integração no nosso protótipo de algumas técnicas que não são descritas nos capítulos anteriores. Por exemplo, o processo de segmentação foi completado com um mecanismo de separação de regiões leucócito conexas. Foram também desenvolvidos algoritmos que se aproximam em eficiência de uma ordem logarítmica e que permitem a aceleração do processo de segmentação, tornando-o temporalmente menos dependente do tamanho das imagens.

Durante a fase de treino foi obtida uma multiplicidade de pesos que proporcionaram diversos resultados. À medida que os conjuntos de pesos iam sendo obtidos, tentava-se por um processo de *trial-and-error* melhorar os resultados, fazendo o ajuste em vários factores. Também as múltiplas e distintas inicializações à rede, que este processo implica, fazem com que as hipóteses de conseguir um conjunto de pesos que se aproximam da situação de mínimo global aumentem.

## 5.2 Resultados da segmentação.

Os modos e parâmetros finais que se revelaram mais apropriados para o mecanismo de segmentação foram:

- Gama de valores de inicialização dos pesos = **[-0.02 , 0.02]**;
- O modo de inicialização dos pesos foi o **Nguyen-Widrow**;
- Função de activação: em todas as camadas foram usadas funções do tipo **sigmóide bipolar**;
- Modos de actualização dos pesos: o **delta-bar-delta**;
- $\alpha = 0.002$ ;
- $\beta = 0.7$ ;
- $\kappa = 0.0001$ ;
- $\gamma = 0.4$ ;
- Número de camadas escondidas = **1**;
- Número de unidades na camada escondida = **2**;
- Número de unidades na camada de entrada = **243**;
- Número de padrões de treino utilizado = **95354**;
- Actualização dos pesos pelo método **Batch**.
- Ordem de apresentação à rede dos padrões de treino: **Sequencial**;
- Tamanho da janela de varrimento: **25 × 25**;
- Arranjo dos pixels da janela de varrimento: **um pixel activo intercalado por dois inactivos**.
- Sistema de cor: **HSL**.

Começamos por analisar imagens (da Figura 5.1 à 5.6) que foram usadas para extrair padrões de treino para a aprendizagem neuronal. A partir da imagem 5.7 inclusive podemos observar os resultados obtidos em imagens que não pertencem ao conjunto das imagens de treino.

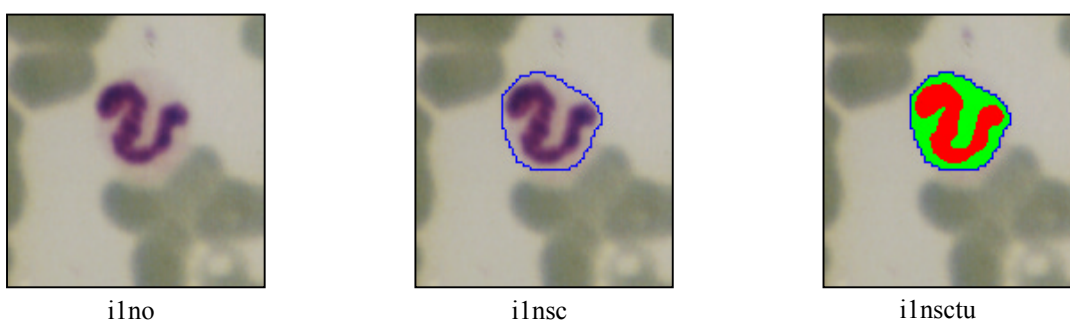
Note-se que associadas às imagens originais temos duas imagens. Numa podemos ver a azul o contorno que delimita a célula; na outra podemos ver, além desse contorno, a segmentação do núcleo (a vermelho) e do citoplasma (a verde).

Refira-se também que serão tecidos alguns comentários sobre os resultados da classificação obtidos para cada imagem, uma vez que estes dependem da eficácia da segmentação.

---

**Figura 5.1** As imagens seguintes mostram um neutrófilo rodeado de alguns eritrócitos. A imagem original (i1no), após ser sujeita ao processo de segmentação, resultou nas imagens seguintes. Podemos ver na imagem (i1nsc) que apesar do ténue tom do citoplasma relativamente ao fundo os limites da célula foram muito bem delimitados. Também a imagem (i1nsctu) mostra um bom resultado no que se refere à separação entre o núcleo e o citoplasma. Por conseguinte, o processo de classificação identificou a célula correctamente.

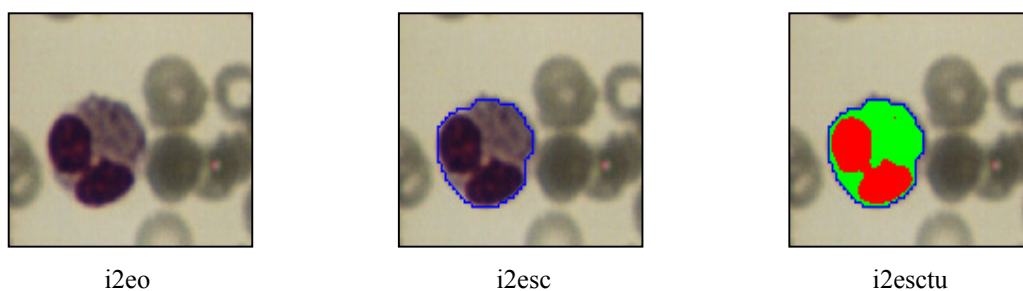
---



---

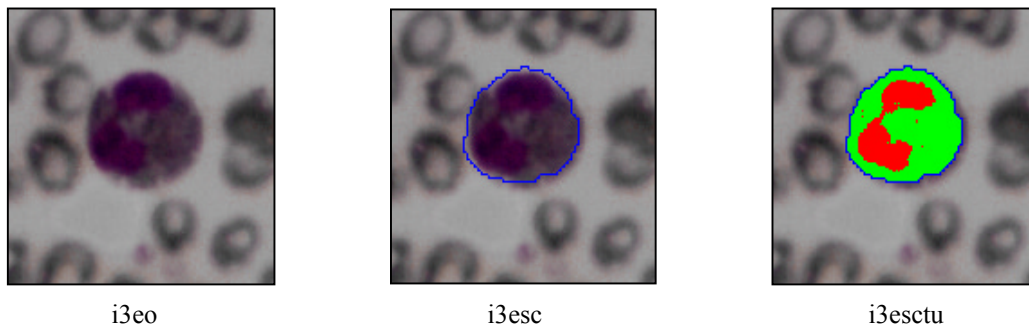
**Figura 5.2** Nestas imagens podemos observar o resultado da segmentação de um eosinófilo. Repara-se na conectividade existente entre o citoplasma do eosinófilo e um glóbulo vermelho. Apesar disso, o contorno delimitador da célula foi obtido com exactidão. Também a separação do núcleo e citoplasma foi perfeita (i2esctu). E o mecanismo de classificação não teve dúvidas ao classificá-lo como um eosinófilo.

---



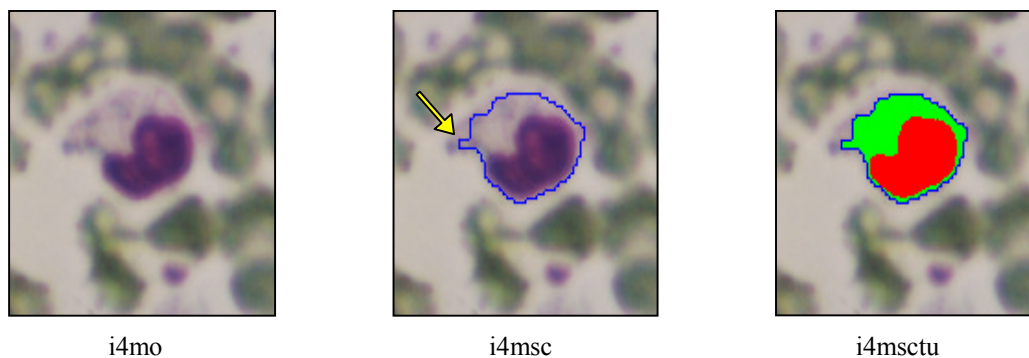
**Figura 5.3** As imagens seguintes apresentam uma tonalidade de fundo bem diferentes das anteriores. Repare-se que, apesar de o contorno encontrado descrever bem a célula, houve alguma imprecisão na segmentação do núcleo (imagem - i3esctu). Isto deve-se à pequena diferença de luminância média que o núcleo apresenta relativamente ao citoplasma. Como vimos, o nosso processo de segmentação -núcleo citoplasma-, baseia-se precisamente na diferença de luminância entre a região núcleo e a região citoplasma. Este leucócito foi correctamente classificado pelo nosso protótipo como sendo um eosinófilo.

---



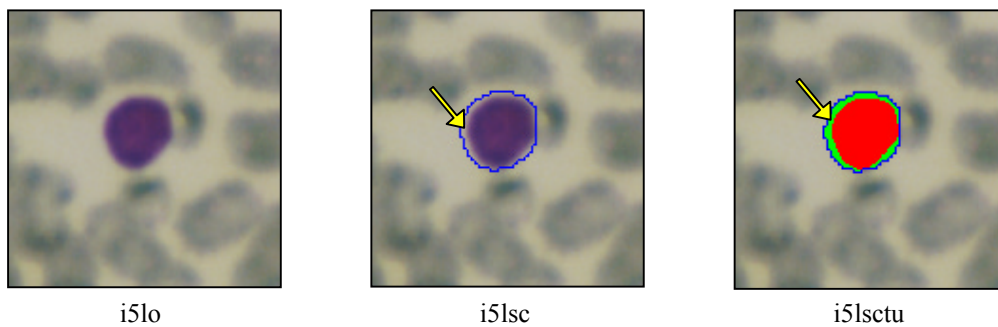
**Figura 5.4** A imagem (i4mo) mostra um monócito com um citoplasma algo disforme. Podemos verificar na imagem (i4msc) que o contorno da célula é assinalado correctamente pelo mecanismo de segmentação, à excepção de uma pequena porção (assinalada por nós com uma seta amarela). Na imagem (i4msctu) verificamos que o núcleo do leucócito foi correctamente assinalado. O processo de classificação identificou a célula como sendo um monócito, o que está correcto.

---



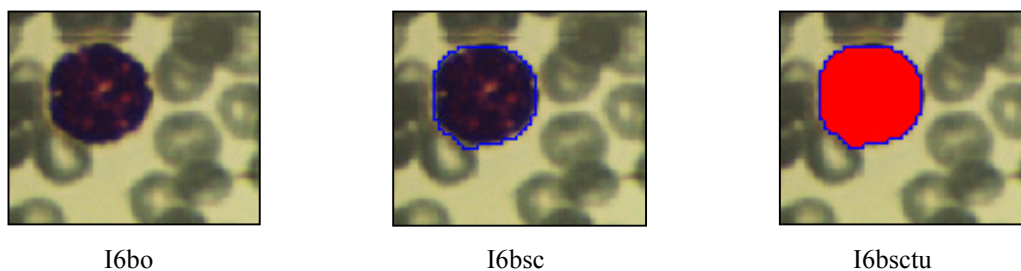
**Figura 5.5** Nestas imagens podemos observar a segmentação de um linfócito. Repara-se que existe uma pequeníssima parcela circular de fundo que foi incluída na região célula. Isto vai provocar um aumento da gama dinâmica do histograma da região célula. Este aumento faz com que o mecanismo de segmentação -núcleo citoplasma- trate a região segmentada como uma região composta por núcleo e citoplasma. Isto obviamente pode produzir inconvenientes ao nível do processo de classificação. No entanto, a célula abaixo foi correctamente classificada. Isto é perceptível se tivermos em conta que os padrões de treino usados para a classificação foram obtidos a partir deste mesmo processo de segmentação.

---



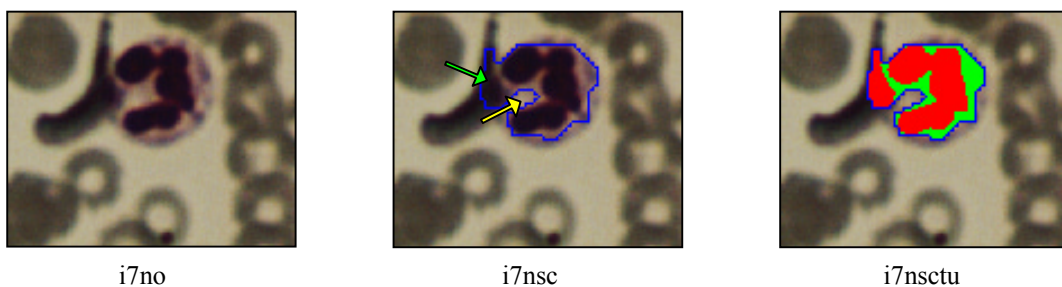
**Figura 5.6** Esta imagem mostra um basófilo. Também nesta célula o citoplasma não existe. A região segmentada apresenta uma gama dinâmica do histograma suficientemente baixa para o processo de segmentação -núcleo citoplasma- considerar a existência apenas do núcleo (i6bsctu). O classificador de leucócitos designou-o correctamente por basófilo.

---



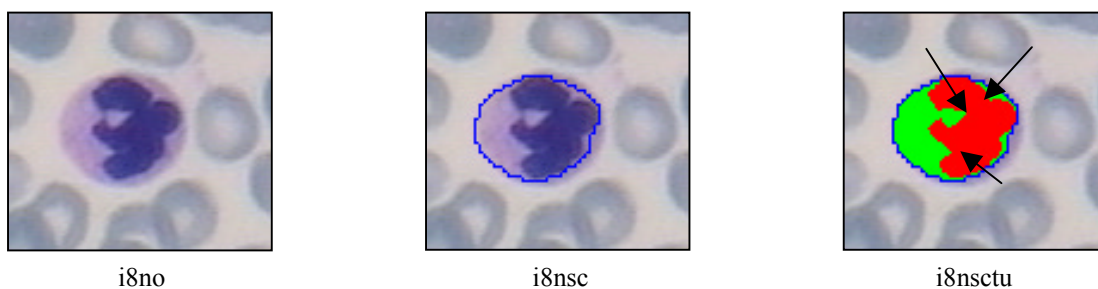
**Figura 5.7** A imagem (i7no) mostra um neutrófilo conexo a um eritrócito disforme. Este eritrócito apresenta em parte da sua superfície uma tonalidade similar à do neutrófilo. Podemos então verificar que uma pequena parte da região eritrócito (seta verde) foi agregada ao leucócito (pelo processo de segmentação) e que existe uma reduzida parcela do leucócito (seta amarela) que não foi integrada na região leucócito (i7nsc). Estas deficiências devem-se fundamentalmente às lacunas existentes no conjunto de padrões de treino. Uma vez que se torna difícil obter um conjunto de padrões de treino representativo de todas as situações particulares, como esta. Estas anomalias da segmentação resultaram numa classificação com a indicação de “Não tenho a certeza de que célula se trata”.

---



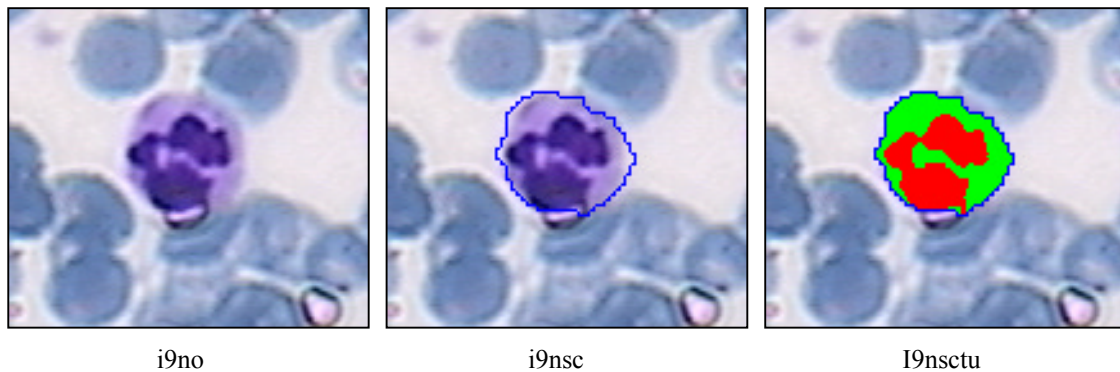
**Figura 5.8** Aqui podemos ver o resultado da segmentação de uma imagem cujas tonalidades diferem significativamente das tonalidades das imagens anteriores. Em adição o núcleo do leucócito tem um aspecto algo peculiar. Apesar de se ter conseguido um bom resultado no que se refere à delimitação do contorno celular (imagem i8nsc), este leucócito foi classificado erradamente como sendo um monócito. A justificação reside essencialmente na pouca eficácia do método de segmentação -núcleo citoplasma-, que mais uma vez perde pormenores dos limites da região núcleo, as quais são decisivas ao adjuvarem na classificação. Comparemos o núcleo do leucócito, visível na imagem (i8no), com a região núcleo assinalada pelo processo de segmentação na imagem (i8nsctu). Observem-se os pontos de perda de particionamento do núcleo sugeridos pelas setas negras na imagem (i8nsctu). De facto o núcleo assim assinalado faz lembrar mais o de um monócito do que o de um neutrófilo.

---



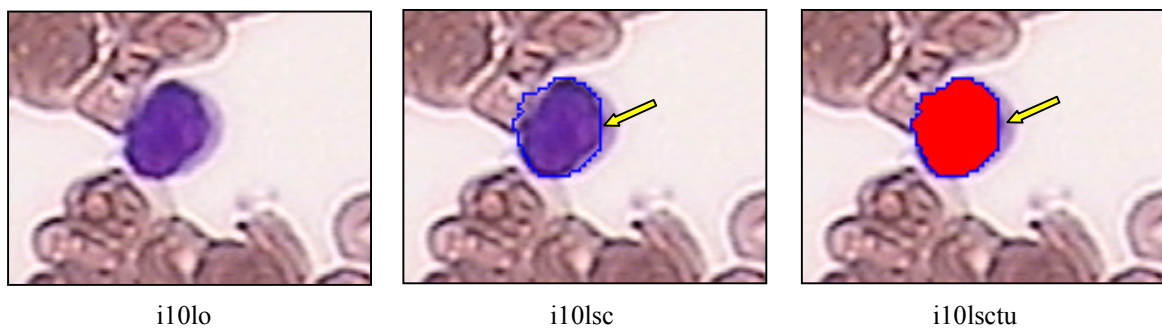
**Figura 5.9** Nestas imagens cujas tonalidades também diferem das imagens anteriores é observável o sucesso da segmentação. Registe-se também a correcta identificação deste neutrófilo por parte do mecanismo de classificação.

---



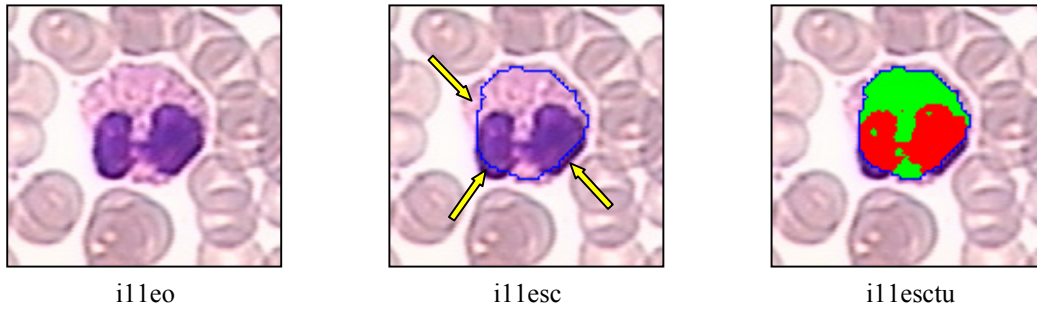
**Figura 5.10** Estas imagens mostram um linfócito rodeado por eritrócitos que possuem tonalidades bem diferentes das tonalidades apresentadas pelos eritrócitos das imagens anteriores. Apesar disso, a segmentação realizou-se razoavelmente bem, perdendo apenas uma pequena porção de citoplasma (seta amarela). Também este leucócito foi bem classificado.

---



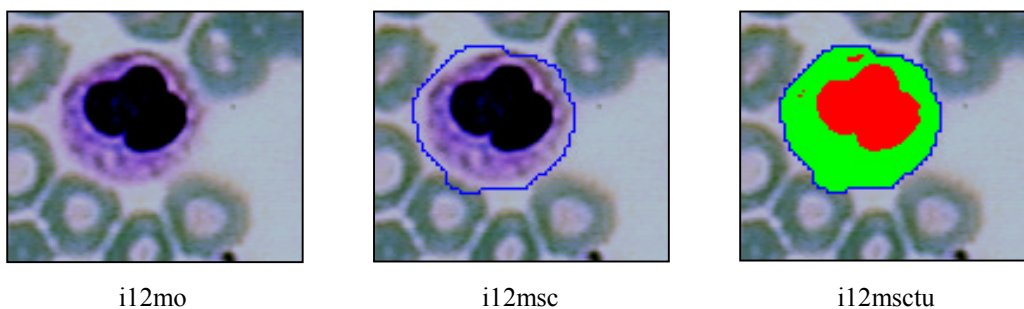
**Figura 5.11** Estas imagens apresentam um nível de luminância relativamente elevado. O que é bem evidente se observarmos a região fundo. Devido a isso, o contorno limítrofe do eosinófilo delimita uma área inferior à efectiva. Todavia, o leucócito foi bem classificado.

---



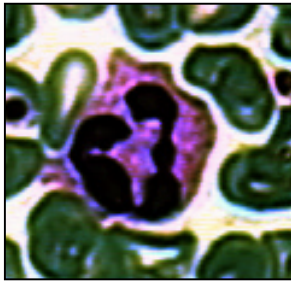
**Figura 5.12** Esta imagem possui um conjunto de tonalidades que se distanciam das existentes nas imagens de treino, fazendo com que o limite celular seja assinalado de forma ligeiramente excedente, em contraste com a imagem anterior. Este leucócito foi identificado como sendo um monócito, embora nós próprios tenhamos alguma dificuldade a classificá-lo.

---

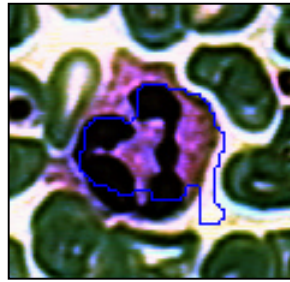


**Figura 5.13** Nesta imagem é visível a imperfeição do contorno, demarcado pelo método da segmentação. Com efeito, as tonalidades que formam esta imagem são mesmo desconformes relativamente às das imagens de treino. No entanto, o mecanismo de classificação anuncia não poder classificar este leucócito devido ao grau de incerteza obtido.

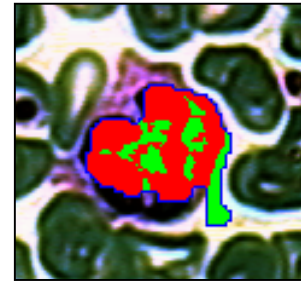
---



i13no



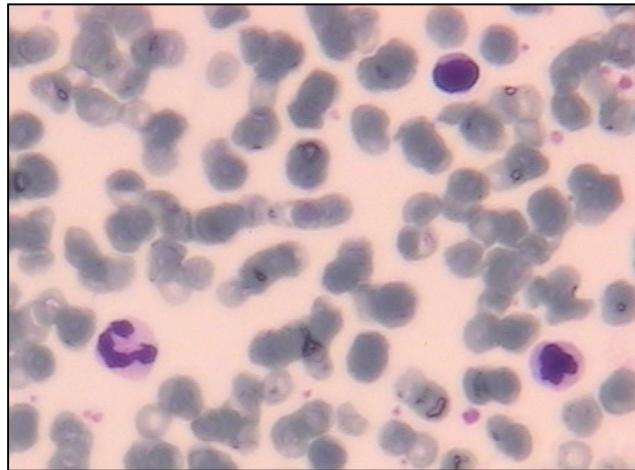
i13nsc



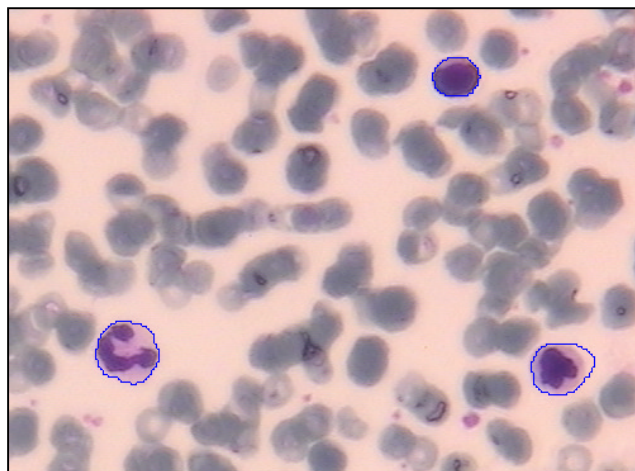
i13nsctu

**Figura 5.14** Estas imagens mostram o resultado da segmentação num quadro em que aparecem três leucócitos. Devido à grande quantidade de eritrócitos visíveis nestas imagens, podemos avaliar também a capacidade de rejeição de regiões eritrócito pelo processo de segmentação.

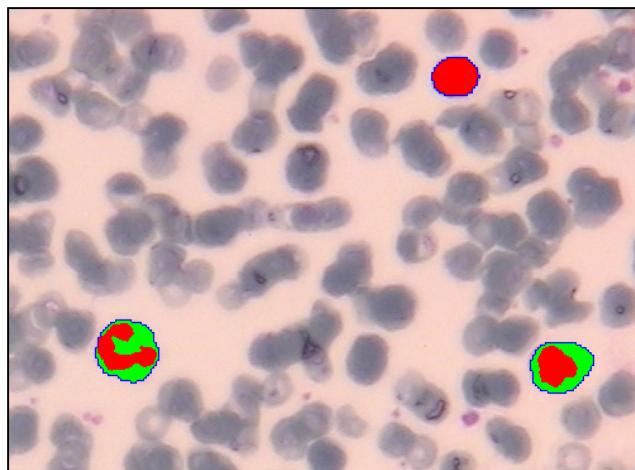
---



i14mixo



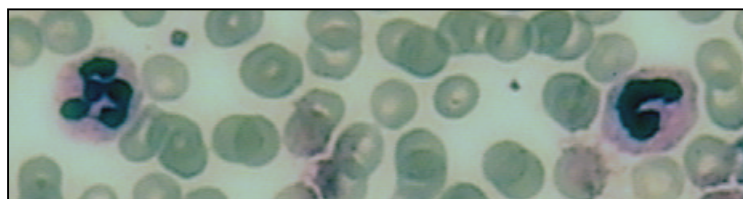
i14mixsc



i14mixsctu

**Figura 5.15** Esta imagem é uma clara demonstração de como os processos de segmentação e de classificação podem falhar, quando a dissemelhança das tonalidades que constituem a imagem a considerar relativamente às tonalidades das imagens de treino é elevada. Neste exemplo nenhum dos leucócitos foi segmentado. Em vez disso, foi-o uma pequena região formada por fundo e por eritrócitos. Também o mecanismo de classificação se mostrou neste exemplo falível, pois identificou essa região como sendo um monócito.

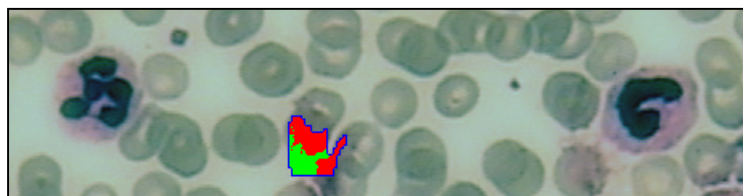
---



i15mix2o



i15mix2sc



i15mix2sctu

### 5.3 Resultados da classificação.

Os modos e parâmetros finais que se demonstraram mais apropriados para o mecanismo de classificação foram:

- Gama de valores de inicialização dos pesos = **[-0.05 , 0.05]**;
- O modo de inicialização dos pesos foi o **Nguyen-Widrow**;
- Função de activação: em todas as camadas foram usadas funções do tipo **sigmóide bipolar**;
- Modos de actualização dos pesos: o **delta-bar-delta**;
- $\alpha = 0.0035$ ;
- $\beta = 0.7$ ;
- $\kappa = 0.0009$ ;
- $\gamma = 0.5$ ;
- Número de camadas escondidas = **1**;
- Número de unidades na camada escondida = **12**;
- Número de unidades na camada de entrada = **46**;
- Número de padrões de treino utilizado = **176**;
- Actualização dos pesos pelo método **Batch**.
- Ordem de apresentação à rede dos padrões de treino: **Sequencial**;
- Periodicidade da validação cruzada = **176**;
- Número de padrões de teste usados na validação cruzada = **48**;
- Os valores fronteira de não classificação: **K = 0.15 e T = -0.3**.

O número de padrões de treino usado para obter estes resultados foi de 176. Este conjunto era constituído por padrões que representavam cada uma das classes de glóbulos brancos. A distribuição numérica de cada uma das classes nesse conjunto de padrões de treino foi a seguinte:

- Basófilos = 5;
- Eosinófilos = 32;
- Linfócitos = 53;
- Neutrófilos = 65;
- Monócitos = 21

A desigualdade numérica assim obtida fica a dever-se à própria distribuição quantitativa natural dos leucócitos no sangue humano (ver Cap.1), que dificulta a aquisição nas amostras sanguíneas de todas as classes equitativamente. No entanto, o facto de se treinar a rede com este desequilíbrio de classes pode trazer vantagens em situações de decisão duvidosa, pois nesses casos a maior probabilidade de acerto recai logicamente sobre a classe que aparece mais vezes. Este condicionalismo faz com que a rede ao longo do treino fique mais *sensível* a essas classes, resultando no referido efeito [Masters, 1993].

Os resultados que vamos apresentar referem-se apenas a células que não apareciam no conjunto de padrões de treino, tanto na segmentação como na classificação, pois os resultados obtidos com as células de treino eram quase de 100% de acertos.

Observando a matriz de confusão seguinte, podemos ver que a cada linha está associada uma classe. Foram então adquiridas imagens de células que foram analisadas por um especialista, que as foi integrando numa das sete classes possíveis (5 classes de leucócitos + 2 classes de não classificação). Posteriormente, as mesmas imagens foram sujeitas à avaliação pelo nosso protótipo e cada leucócito classificado foi comparado com a decisão do especialista. Assim, foi obtida a seguinte *matriz confusão*:

|                            |            | Decisão do classificador neuronal |            |           |          |            |         |         |
|----------------------------|------------|-----------------------------------|------------|-----------|----------|------------|---------|---------|
|                            |            | Basófilo                          | Eosinófilo | Linfócito | Monócito | Neutrófilo | Incerta | Não sei |
| Decisão do<br>especialista | Basófilo   | 4                                 |            |           | 1        |            |         |         |
|                            | Eosinófilo | 1                                 | 37         |           | 4        | 4          | 2       | 2       |
|                            | Linfócito  |                                   |            | 55        | 6        | 1          | 3       |         |
|                            | Monócito   | 1                                 | 1          | 2         | 18       | 3          | 2       | 3       |
|                            | Neutrófilo | 2                                 | 9          |           | 4        | 57         | 8       | 4       |
|                            | Incerta    | 1                                 |            |           | 3        |            | 12      | 2       |
|                            | Não sei    |                                   | 4          | 1         | 3        | 2          | 4       | 22      |

Por exemplo, as células identificadas pelo especialista como pertencentes à classe basófilo foram cinco, como podemos observar na primeira linha da matriz. Porém, um desses leucócitos foi classificado erradamente pelo classificador neuronal, como sendo um monócito. Os restantes quatro foram correctamente classificados.

Analisando a matriz de confusão, facilmente calculamos as seguintes percentagens de acerto para cada uma das classes de leucócitos (não foram consideradas para a obtenção destes cálculos aquelas células que foram classificadas pelo sistema como *Incerta* ou *Não sei*<sup>18</sup>):

- Basófilo: 80.0%
- Eosinófilo: 80.4%
- Linfócito: 88.7%
- Monócito: 72.0%
- Neutrófilo: 79.2%

Repare-se que os maiores erros surgem logicamente entre as classes semelhantes geometricamente. Por exemplo, a percentagem de neutrófilos confundidos com eosinófilos foi de 12.5%, enquanto apenas 2.7% dos neutrófilos foram confundidos com monócitos. A fragilidade do processo de segmentação -núcleo citoplasma- contribui certamente para que isso aconteça.

A classe dos linfócitos, devido à sua simplicidade geométrica e dissemelhança com as outras classes é a que maior percentagens de acertos obtém. Já a classe dos monócitos, que encontra semelhanças com a maior parte das outras classes, obtém a pior percentagem de acertos. Mais uma vez, destes factos não estará também dissociado o baixo número de padrões de treino que foram usados na aprendizagem do processo de classificação.

---

<sup>18</sup> Lembramos que, por se tratar de uma contagem percentual, a não classificação dos casos duvidosos é preferível à classificação errada.

## Capítulo 6

### **Conclusões e trabalho futuro**

Constitui-se como principal objectivo desta tese o desenvolvimento e a implementação de técnicas que permitam a segmentação e a classificação de imagens de leucócitos. Daí que o estudo e implementação que fizemos para solucionar os diversos problemas intrínsecos tanto ao processo de segmentação como ao de classificação tiveram sempre uma orientação principal: a de obter os melhores resultados. Tentámos, por isso, ao longo do nosso trabalho explorar exaustivamente as abordagens por nós adoptadas e desenvolvidas. Contudo, isto não significa que o tenhamos conseguido, nem tão pouco que essas abordagens constituíssem a melhor opção. Os estudos prévios realizados associados a um certo grau de empirismo levaram-nos a optar por soluções que acabaram por se mostrar umas mais capazes outras menos. Se a confinante temporal não existisse sem dúvida que a vontade e a curiosidade ter-nos-iam levado a desenvolver e implementar outras abordagens também interessantes e promitentes. Todavia julgamos que este trabalho permitiu concluir sobre a aptidão das abordagens usadas, suas vantagens e desvantagens.

Como conclusão global podemos dizer que o uso de redes neuronais para solucionar os problemas da segmentação e da classificação de leucócitos é viável desde que se consiga formar conjuntos de padrões de treino bem representativos das características fundamentais das imagens células. Este facto acaba por tornar-se mais significativo na medida em que se torna difícil a construção de um conjunto de treino completo. Porém, este estudo demonstra que, conseguindo minimizar esse problema e utilizando as abordagens desenvolvidas, é possível obter resultados satisfatórios e promissores.

## 6.1 Principais conclusões sobre o processo de segmentação

O uso de uma rede neuronal supervisionada para executar a segmentação dos leucócitos mostrou-se perfeitamente adequada desde que os padrões de treino usados sejam bem representativos do universo de padrões inerentes a este tipo de imagens, desde que a estrutura da rede neuronal seja apropriada e desde que o treino desta seja bem sucedido.

Uma das principais vantagens que o uso dessas redes neuronais traz a este processo é precisamente o de aprender a segmentar os leucócitos a partir de exemplos, evitando assim a difícil tarefa de construir segmentadores por meio, por exemplo, de heurísticas, que dificilmente permitiriam chegar aos mesmos resultados.

Em termos de desvantagens este processo obriga a uma escolha algo cuidada dos padrões de treino de forma a que esses padrões sejam bem representativos dos futuros padrões a classificar, aproveitando-se assim as capacidades de generalização da rede. Ora, tendo em conta a forma como esses padrões são obtidos e tendo em conta também a necessidade de obter uma grande diversidade de padrões para atender às exigências que advêm do problema da variabilidade de tonalidades, isto constitui de facto um trabalho moroso e custoso.

Ficou também patente a dificuldade de fazer convergir a rede neuronal à medida que se lhe aumenta a estrutura, bem como o número de padrões de treino (aliás, o primeiro factor não está de todo dissociado do segundo). Na prática, essa dificuldade é de imediato sentida em termos temporais, uma vez que, se aumentarmos o número de padrões de treino e a estrutura da rede neuronal, o número de cálculos necessários para que a convergência seja possível cresce substancialmente.

Durante o nosso trabalho chegámos a considerar estruturas e padrões de treino que inicialmente se mostravam promissores para a melhoria do desempenho da segmentação, mas que aquando da fase de treino no final de 48 horas tinham apenas possibilitado a execução de 3 iterações de aprendizagem.

Como foi explicado no capítulo 3, o uso do sistema de cor HLS face ao sistema RGB contribuiu para algum melhoramento dos resultados.

## 6.2 Principais conclusões sobre o processo de classificação

Não podemos esquecer que os resultados que obtivemos em termos de classificação estão intimamente ligados aos resultados da segmentação. Daí que o valor médio de acertos que foram obtidos se tenha aproximado dos 80% (valor que fica um pouco aquém do exigido para um sistema comercial.). Este valor reflecte as deficiências existentes ao nível da segmentação da célula, ao nível da segmentação -citoplasma núcleo- e ao nível da própria classificação.

Pela análise que fizemos dos resultados obtidos, podemos concluir que de entre os três processos referidos o que se mostrou mais falível foi o de segmentação -núcleo citoplasma-, que devido à sua simplicidade deforma em muitos casos a verdadeira região núcleo e a verdadeira região citoplasma, enganando posteriormente o classificador neuronal.

A utilização de descritores para a redução da dimensionalidade trazia à partida um risco, relacionado com a perda de informação importante, mas acabou por se mostrar funcional, pois tentou escolher-se minuciosamente esses descritores de forma a minimizar esse risco. De facto, as células que não apresentavam deficiências na segmentação eram quase sempre bem classificadas, fora algumas excepções de células que apresentavam geometrias ambíguas, embora mesmo essas, na maior parte das vezes, fossem “não-classificadas”.

Embora não nos tenha sido possível confirmá-lo, cremos também que o processo de classificação poderia ainda assim ser melhorado se o número de padrões de treino fosse mais vasto do que o usado, pois pensamos que a diversidade de padrões conduziria a uma redução dos problemas trazidos pelas deficiências da segmentação.

Foi interessante verificar que, a certa altura, quando a opinião do especialista apontou para a pertença de uma célula a uma determinada classe que contrariava a decisão do protótipo, o especialista foi posteriormente confrontado com essa situação e veio admitir que se tinha equivocado, alterando a opinião em acordo com a decisão do nosso protótipo.

Ainda que o número de classes de leucócitos a que tivemos acesso se tenha limitado a cinco, estamos certos de que com mais ou menos dificuldades os mesmos resultados poderiam ser obtidos com um maior número de classes.

Resta ainda referir que algumas das conclusões proferidas para o uso da rede neuronal no mecanismo da segmentação são perfeitamente transferíveis para o caso da classificação,

nomeadamente as conclusões que se referem ao treino da rede neuronal e às vantagens e desvantagens da aprendizagem supervisionada.

### 6.3 Trabalho futuro

Pensamos que o trabalho por nós realizado constitui um indicador de que estas abordagens são perfeitamente válidas para a solução do problema. No entanto, está-se ainda longe de se poder integrar essas soluções num equipamento comercial para ser usado num laboratório de análises clínicas.

A nossa proposta dos passos a serem realizados futuramente com vista a melhoria dos resultados seria:

- Adequar a representação do plano Hue às entradas da rede neuronal. Os valores que codificam este plano variam entre 0 e 360, representando circularmente as diferentes frequências de luz. Isto significa que a cor indicada pelo valor zero é muito semelhante à cor indicada pelo valor 360. Ora, esta descontinuidade numérica dificulta o treino da rede neuronal. Uma forma de se resolver o problema consiste na decomposição desse valor em dois outros, através de um seno e de um coseno.
- O desenvolvimento de um outro mecanismo mais complexo e funcional de segmentação -núcleo citoplasma-. Por exemplo, poder-se-ia basear também numa rede neuronal.
- A utilização de outros meios de aprendizagem neuronal que assegurassem mais eficazmente a convergência para o mínimo global; por exemplo através de algoritmos genéticos.

- Um sistema de obtenção de padrões mais simples, que exigisse menos intervenção humana. Por exemplo, um sistema que gerasse automaticamente várias tonalidades diferentes para cada um dos padrões originais, reduzindo assim a dificuldade na obtenção dos múltiplos padrões de treino.
- A implementação física dos sistemas de segmentação e classificação, recorrendo a cartas de aceleração neuronal ou a estruturas de computação paralela. Acreditamos que este ponto traria uma melhoria significativa aos resultados, pois iria permitir o aumento da estrutura neuronal e, por conseguinte, uma aprendizagem neuronal que iria especializar-se mais minuciosamente na distribuição geométrica das tonalidades na janela de varrimento (segmentação). Além disto, o número de padrões de treino poderia também assim ser aumentado na sua diversidade (segmentação e classificação), bem como o tamanho da janela de varrimento, agregando assim ainda mais informação para uma correcta segmentação. Além disso, permitiria o processamento às imagens num espaço de tempo pragmaticamente mais proveitoso.
- O uso de métodos na aquisição das imagens e preparação das lâminas que garantissem uma maior consistência das tonalidades destas. Ex: poder-se-ia recorrer a um método mecânico que garantisse a uniformidade da distribuição do corante na lâmina.
- O uso de algoritmos de eliminação da variabilidade das tonalidades, “Discount Illuminant” [Grossberg & Wyse, 1992]. Estes algoritmos possibilitam a redução dos efeitos causados pela variação das tonalidades que emerge das lâminas.



## Apêndice A

### Implementação da Backpropagation em C++.

Esta aplicação foi desenvolvida para ambiente Microsoft Windows 32 bits. A implementação da rede neuronal *backpropagation* foi feita em C++, recorrendo ao compilador da Microsoft, Visual C++ 6.0 e respectivas MFC (Microsoft Foundation Class). Os recursos gráficos usados permitem simplificar a manipulação da rede neuronal do ponto de vista do utilizador. O motor desta rede neuronal foi construído recorrendo a uma organização de classes, de forma a aproveitar as várias potencialidades oferecidas pela linguagem C++, tais como:

- Herança
- Polimorfismo
- Encapsulamento
- Reutilização.

Como sabemos, uma rede neuronal do tipo *backpropagation* pode ser formada por uma associação de camadas, uma camada de unidades de entrada, uma camada de saída e pode conter ainda várias camadas escondidas.

A ideia foi construir uma classe **abstracta** a que chamamos CLayer para agrupar métodos e dados comuns aos três tipos de camadas neuronais. Seguindo esta ideia, construímos então a classe CInputLayer, uma especialização de CLayer, de forma a representar a camada de entrada, e a classe ChiddenLayer, que implementa a camada

escondida. A camada de saída difere apenas das camadas escondidas numa questão: toda a informação resultante da camada de saída é processada por elementos externos à rede, enquanto a informação que chega às camadas escondidas provém sempre de outras camadas. Daí que a classe que implementa a última camada (camada de saída), `CoutLayer`, seja uma classe derivada da classe `CHiddenLayer` e na qual alguns métodos são definidos como *virtual* (este facto obriga a invocação de métodos nas classes derivadas mesmo que a referência incida sobre a base). É o caso do método *virtual void Set\_Error\_In()*. Este método permite a retro-propagação do erro entre camadas. Como é evidente, na camada de saída, o erro não é propagado por nenhuma outra camada, mas sim pelas diferenças entre o padrão alvo e pelo padrão actual, o que faz com que as especificações deste método para a classe `COutLayer` sejam diferentes.

A classe responsável por coordenar e gerir as respectivas camadas da rede neuronal é a classe `CNet`. Desde a manipulação dos ficheiros de treino, métodos de treino, inicialização dos pesos, iterações de aprendizagem a iterações de decisão, todos os processos neuronais passam pelo objecto desta classe. Nesta classe existe um *array* de apontadores para a classe `CLayer`. Assim cada elemento deste *array* representa uma camada da rede neuronal.

Cada neurónio das camadas escondidas e da camada de saída terá uma função de activação que poderá ser uma qualquer para cada camada. Decidimos criar aqui uma classe para cada função. Desta forma, e usando a capacidade oferecida pelo polimorfismo dinâmico do C++, podemos facilmente evocar as funções de activação pretendidas para uma qualquer camada. Desta maneira, evitamos ter que escrever um método de treino para cada função de activação. As classes que nós construímos para implementação dos objectos das várias funções de activação foram:

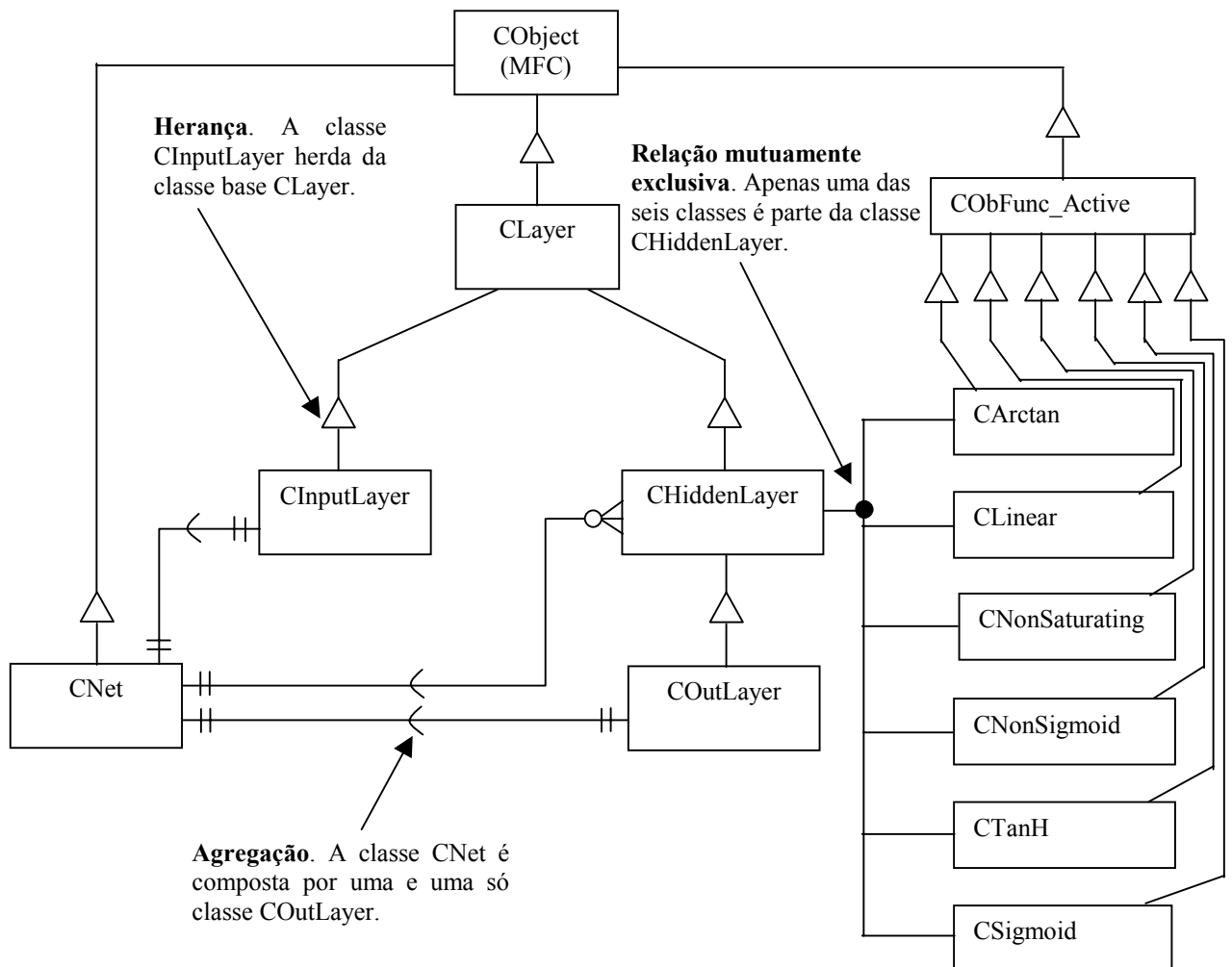
- `CTanH` (Função Tangente Hiperbólica)
- `CArctan` (Função Arco Tangente)
- `CLinear` (Função Identidade)
- `CNonSigmoid` (Função Gaussiana)

- CNonSaturating (Função não saturação)
- CSigmoid (Função Sigmóide)

Estas seis classes têm como classe base uma classe **abstracta** a que chamamos CObFunc\_Active.

Vejamos agora na figura seguinte como se encontram relacionados<sup>19</sup> os objectos que formam o motor da rede neuronal.

**Figura A.1** Relacionamento entre as classes da rede neuronal.

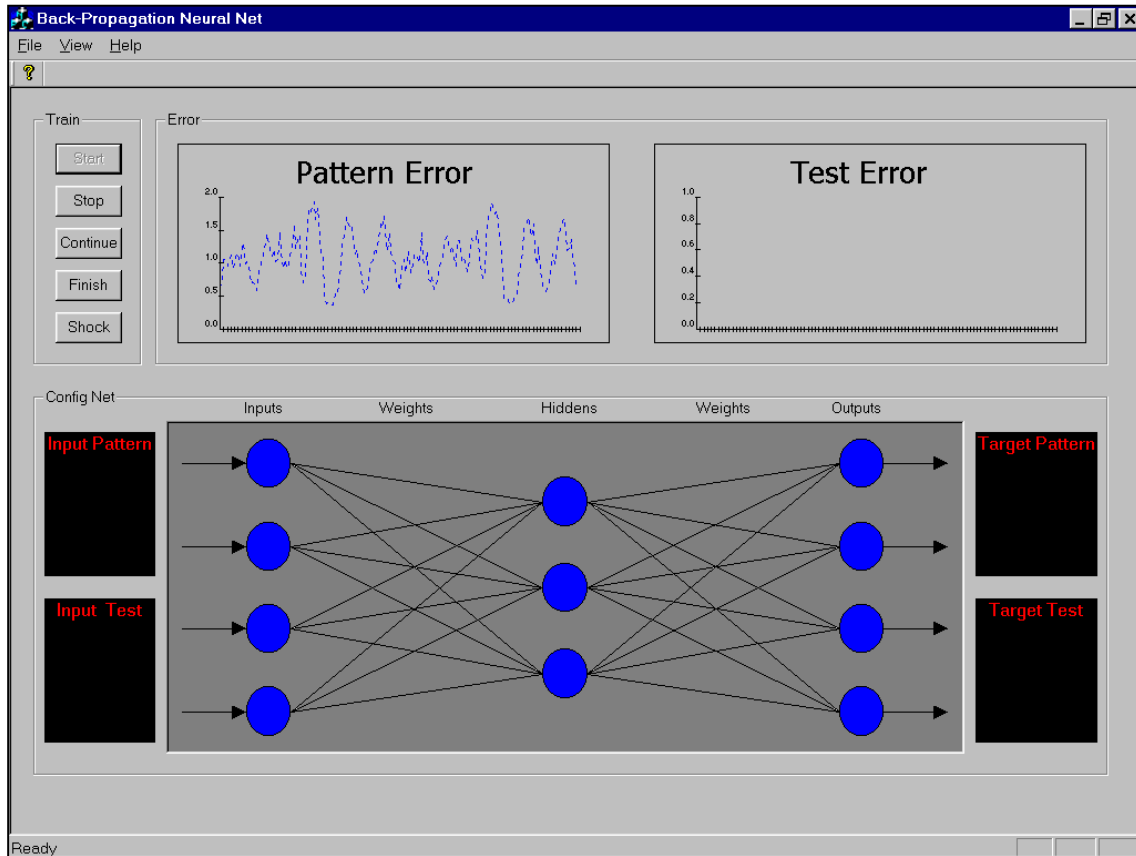


Explicações sobre as metodologias do C++ usadas, podem ser obtidas em [Schildt, 1995].

---

**Figura A.2** Janela principal do simulador neuronal.

---



---

<sup>19</sup> A simbologia usada para representar o relacionamento entre classes baseia-se em [James Martin, 1993].

## Apêndice B

### Derivação da regra delta generalizada.

Neste apêndice são apresentadas as deduções matemáticas que permitem obter as equações gerais da regra delta generalizada [Nauck & Klawonn & Kruse, 1997].

A soma dos quadrados das diferenças entre as saídas actuais e as saídas pretendidas será a função de custo a minimizar (B.1).

$$E = \frac{1}{2} \sum_{k=1}^{m_n} (T_k - Y_{(n,k)})^2 \quad (\text{B.1})$$

O argumento  $v$  de uma função de activação pertencente a um neurónio  $k$  da camada  $i$  é dado por:

$$v_{(i,k)} = \sum_{j=1}^{m_{i-1}} W_{(i,k,j)} Y_{(i-1,j)} + b_{(i,k)} \quad (\text{B.2})$$

A saída de um neurónio de uma qualquer camada com excepção da primeira ( $i \neq 1$ ) é descrita na equação (B.3):

$$Y_{(i,k)} = \varphi_i(v_{(i,k)}) \quad (\text{B.3})$$

O algoritmo da *backpropagation* baseia-se no procedimento da descida segundo a direcção do gradiente. Este processo de aprendizagem pretende minimizar  $E$  para um conjunto de padrões de treino, medindo a contribuição de cada peso para o erro e ajustando através de iterações consecutivas os pesos da rede neuronal da seguinte forma:

$$\Delta W_{(i,k,j)} \propto -\frac{\partial E}{\partial W_{(i,k,j)}} \quad (\text{B.4})$$

Utilizando a regra da cadeia de derivação, podemos desdobrar a derivada parcial de (B.4) num produto de duas derivadas parciais. Assim, o primeiro factor na equação (B.5) reflecte a dependência de  $E$  relativamente ao somatório ( $v$ ) dos sinais que chegam ao neurónio  $k$  da camada  $i$ . O segundo factor reflecte a dependência do somatório dos referidos sinais relativamente ao peso que liga o neurónio  $k$  da camada  $i$  ao neurónio  $j$  da camada  $i-1$ .

$$\frac{\partial E}{\partial W_{(i,k,j)}} = \frac{\partial E}{\partial v_{(i,k)}} \frac{\partial v_{(i,k)}}{\partial W_{(i,k,j)}} \quad (\text{B.5})$$

Substituindo  $v$  no numerador do segundo termo da equação (B.5), então, e de acordo com a equação (B.2), resulta:

$$\frac{\partial v_{(i,k)}}{\partial W_{(i,k,j)}} = \frac{\partial \left( \sum_{j=1}^{m_{i-1}} W_{(i,k,j)} Y_{(i-1,j)} + b_{(i,k)} \right)}{\partial W_{(i,k,j)}} = Y_{(i-1,j)} \quad (\text{B.6})$$

Por uma questão de conveniência vamos definir  $\delta_{(i,k)}$  como sendo:

$$\delta_{(i,k)} = -\frac{\partial E}{\partial v_{(i,k)}} \quad (\text{B.7})$$

Então, tendo em conta a equação (B.7) e a (B.6), podemos escrever:

$$-\frac{\partial E}{\partial W_{(i,k,j)}} = \delta_{(i,k)} Y_{(i-1,j)} \quad (\text{B.8})$$

Assim e de acordo com a equação (B.4), a actualização dos pesos é feita como indica a equação (B.9), em que um coeficiente  $\alpha$  (coeficiente de aprendizagem) controla a magnitude da actualização.

$$\Delta W_{(i,k,j)} = \alpha \delta_{(i,k)} Y_{(i-1,j)} \quad (\text{B.9})$$

Falta-nos agora determinar  $\delta_{(i,k)}$ . Recorrendo novamente à regra da cadeia, vamos decompor a derivada parcial da equação (B.7) num produto de dois factores, no qual o primeiro factor representa a dependência do erro em relação à saída de um neurónio, enquanto o segundo factor reflecte a variação da saída de um neurónio em função do somatório dos sinais que chegam a esse neurónio.

$$\delta_{(i,k)} = -\frac{\partial E}{\partial v_{(i,k)}} = -\frac{\partial E}{\partial Y_{(i,k)}} \frac{\partial Y_{(i,k)}}{\partial v_{(i,k)}} \quad (\text{B.10})$$

O segundo factor pode ser facilmente transformado a partir da equação (B.3).

$$\frac{\partial Y_{(i,k)}}{\partial v_{(i,k)}} = \varphi'_i(v_{(i,k)}) \quad (\text{B.11})$$

Para a determinação do primeiro factor duas situações têm que ser consideradas:

1. A primeira situação refere-se aos neurónios da última camada ( $i=n$ ). Derivando a equação B.1 em ordem à saída de um neurónio da última camada, obtemos (B.12).

$$\frac{\partial E}{\partial Y_{(n,k)}} = -(T_k - Y_{(n,k)}) \quad (\text{B.12})$$

Se substituirmos os dois factores da equação (B.10) de acordo com (B.12) e com (B.11), definimos  $\delta_{(i,k)}$  para neurónios da última camada, da seguinte forma:

$$\delta_{(i,k)} = (T_k - Y_{(n,k)}) \varphi'_i(v_{(i,k)}) \quad (\text{B.13})$$

2. A segunda situação refere-se aos neurónios que se situam nas camadas escondidas da rede ( $2 \leq i < n$ ). Aplicando a regra da cadeia mais uma vez ao primeiro factor da equação (B.10), escrevemos:

$$\begin{aligned}
 \frac{\partial E}{\partial Y_{(i,k)}} &= \sum_{j=1}^{m_{i+1}} \left( \frac{\partial E}{\partial v_{(i+1,j)}} \frac{\partial v_{(i+1,j)}}{\partial Y_{(i,k)}} \right) \\
 &= \sum_{j=1}^{m_{i+1}} \left( \frac{\partial E}{\partial v_{(i+1,j)}} \frac{\partial \left( \sum_{u=1}^{m_i} W_{(i+1,j,u)} Y_{(i,u)} + b_{(i+1,j)} \right)}{\partial Y_{(i,k)}} \right) \tag{B.14} \\
 &= \sum_{j=1}^{m_{i+1}} \frac{\partial E}{\partial v_{(i+1,j)}} W_{(i+1,j,k)} \\
 &= - \sum_{j=1}^{m_{i+1}} \delta_{(i+1,j)} W_{(i+1,j,k)}
 \end{aligned}$$

Se ambos os factores de (B.10) forem substituídos por (B.14) e por (B.11), então  $\delta_{(i,k)}$  de um neurónio  $k$  de uma camada escondida  $i$  será dado por:

$$\delta_{(i,k)} = \varphi'_i(v_{(i,k)}) \sum_{j=1}^{m_{i+1}} \delta_{(i+1,j)} W_{(i+1,j,k)} \tag{B.15}$$

Como podemos verificar, as equações (B.15) e (B.13) definem recursivamente os valores de  $\delta$  para toda a rede. Estes valores permitem através da equação (B.9) a actualização dos pesos, a qual tenderá a diminuir a função de custo  $E$ . Não esqueçamos que todo este procedimento pode ser aplicado também à actualização das *polarizações*.

A este método que acabamos de deduzir chamamos *regra delta generalizada* ou *backpropagation*.



## **Bibliografia**

- Bain, Barbara (1996) – *A Beginner's Guide to Blood Cells*. Oxford, Blackwell Science.
- Barto et alii (1983) – *Neuronlike Adaptative Elements That Can Solve Difficult Learning Control Problems*. IEEE Trans. System, Man & Cybernetics, 13:834-846.
- Bates & Tompkins (1998) – *Using Visual C++ 6*. Indianapolis, QUE.
- Baxes, Gregory (1994) – *Digital Image Processing. Principles and Applications*. New York, John Wiley & Sons, Inc.
- Bessman, David (1988) – *Automated Blood Counts and Differentials. A Practical Guide*. Baltimore, John Hopkins University Press.
- Bishop, Christopher (1995) – *Neural Networks for Pattern Recognition*. Oxford, Clarendon Press.
- Cho, Sung-Bae (1995) – *A Neuro-Fuzzy Architecture for High Performance Classification*. In: Furuhashi, Takeshi (Ed.) (1995) – *Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms*. IEEE/Nagoya – University World Wisepersons Workshop Nagoya, Japan, August 1994. Berlin, Springer, 67-84.
- Damásio, António (1995) – *O Erro de Descartes. Emoção Razão e Cérebro Humano*. Mem Martins, Publicações Europa América.
- Fausett, Laurene (1994) – *Fundamentals of Neural Networks. Architectures, Algorithms and Applications*. New Jersey, Prentice Hall International, Inc.
- Fernández, Eugenio (1993) – *Modelos Conexionistas: conceptos generales, origen y evolución*. In: Olmeda & Barba-Romero (Ed.) (1993) - *Redes Neuronales Artificiales. Fundamentos y Aplicaciones*. Actas de un curso de la Universidad de Alcalá, 28-30 Junio 1993. Alcalá de Henares, Universidad de Alcalá de Henares, 3-21.
- Franich & Souček (1992) – *Learning to See*. In: Souček, Branko (1992) – *Fast Learning and Invariant Object Recognition*. New York, John Wiley & Sons, Inc., 3-8.

Gonzalez & Woods (1992) – *Digital Image Processing*. Massachusetts, Addison-Wesley Publishing Company.

Grossberg & Wyse (1992) – *Figure-Ground Separation of Connected Scenic Figures: Boundaries, Filling-In, and Opponent Processing*. In: Carpenter & Grossberg (Ed.) (1992) – *Neural Networks for Vision and Image Processing*. Massachusetts, The MIT Press, 161-194.

Haykin, Simon (1999) – *Neural Networks. A comprehensive foundation*. New Jersey, Prentice Hall.

Heske & Heske (1996) – *Fuzzy Logic for Real World Design*. San Diego, Annabooks.

Hu, M. K. (1962) – *Visual Pattern Recognition By Moment Invariants*. IRE Transactions on Information Theory, 8:2, 179-187.

Jacobs, R. A. (1998) – *Increased Rates of Convergence Through Learning Rate Adaptation*. Neural Networks, 1(4):295-307.

Jähne, Bernd (1997) – *Digital Image Processing. Concepts, Algorithms and Scientific Applications*. Berlin, Springer.

Kain, Eugène (1998) – *The MFC Answer Book. Solutions for Effective Visual C++ Applications*. Massachusetts, Addison-Wesley.

Kernighan & Ritchie (1988) – *The C Programming Language*. New Jersey, Prentice Hall PTR.

Kohonen, T. (1989) – *Self-organization and Associative Memory*. Berlin, Springer-Verlag.

Kosko, B. (1988) – *Bidirectional associative memories*. IEEE Transactions on Systems, Man, and Cybernetics, 18:49-60.

Kruglinski, David (1997) – *Inside Visual C++*. Washington, Microsoft Press.

le Cun (1985) – *Une procedure d'apprentissage pour reseau a seuil assymetrique*. In: *Proc. Cognitiva '85*. Paris, p. 599-604, June 1985.

Lin & Lee (1995) – *Neural Fuzzy Systems. A Neuro-Fuzzy Synergism to Intelligent Systems*. New Jersey, Prentice Hall PTR.

Lindblad & Kinser (1998) – *Image Processing using Pulse-Coupled Neural Networks*. Berlin, Springer.

Marques, Jorge Salvador (1999) – *Reconhecimento de padrões. Métodos Estatísticos e Neuronais*. Lisboa, IST Press.

Martin, James (1993) – *Principles of Object-Oriented. Analysis and Design*. New Jersey, Prentice Hall, Inc.

Martín, Mariano (1993) – *Redes de Propagación hacia adelante: Retropropagación del Gradiente*. In: Olmeda & Barba-Romero (Ed.) (1993) - *Redes Neuronales Artificiales. Fundamentos y Aplicaciones*. Actas de un curso de la Universidad de Alcalá, 28-30 Junio 1993. Alcalá de Henares, Universidad de Alcalá de Henares, 67-82.

Masters, Timothy (1993) – *Practical Neural Network Recipes in C++*. San Diego, Academic Press.

Masters, Timothy (1994) – *Signal and Image Processing with Neural Networks. A C++ Sourcebook*. New York, John Wiley & Sons, Inc.

Minai & Williams (1992) – *Fast Back Propagation with Adaptive Decoupled Momentum*. In: Souček, Branko (1992) – *Fast Learning and Invariant Object Recognition*. New York, John Wiley & Sons, Inc., 9-26.

Minsky & Papert (1969) – *Perceptrons*. Cambridge. The MIT Press.

Mira, José (1993) – *Fundamentos Biológicos de las Redes de Neuronas Artificiales*. In: Olmeda & Barba-Romero (Ed.) (1993) - *Redes Neuronales Artificiales. Fundamentos y Aplicaciones*. Actas de un curso de la Universidad de Alcalá, 28-30 Junio 1993. Alcalá de Henares, Universidad de Alcalá de Henares, 23-41.

Nauck & Klawonn & Kruse (1997) – *Foundations of Neuro-Fuzzy Systems*. Chichester, John Wiley & Sons.

Nguyen & Widrow (1990) – *Improving the Learning Speed of Two-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights*. International Joint Conference on Neural Networks, San Diego, III:21-26.

Olmeda, Ignacio (1993) – *Aprendizaje y Generalización*. In: Olmeda & Barba-Romero (Ed.) (1993) - *Redes Neuronales Artificiales. Fundamentos y Aplicaciones*. Actas de un curso de la Universidad de Alcalá, 28-30 Junio 1993. Alcalá de Henares, Universidad de Alcalá de Henares, 43-63.

Parker (1985) – *Learning-Logic*. Technical Report TR-47, Cambridge, MIT Cen. Computational Res. Economics Management Sci.

Parker, J. R. (1997) – *Algorithms for Image Processing and Computer Vision*. New York, John Wiley & Sons, Inc.

Pinho, Armando José (1996) – *Estudo sobre a aplicação de redes neuronais à detecção de contornos em imagens*. Dissertação de Doutoramento em Engenharia Electrotécnica

apresentada à Universidade de Aveiro.

Pratt, William (1991) – *Digital Image Processing*. New York, A Wiley-Interscience Publication, John Wiley & Sons, Inc.

Rao & Rao (1995) – *C++ Neural Networks and Fuzzy Logic*. IDG Books Worldwide.

Ripley & Hjort (1995) – *Pattern Recognition and Neural Networks*. Cambridge, Cambridge University Press.

Rogers, Joey (1997) – *Object-Oriented Neural Networks in C++*. San Diego, Academic Press.

Rumelhart & Hinton & Williams, (1986a) – *Learning Representations by Back-propagating error*. *Nature*, 323:533-536.

Rumelhart & McClelland (1986b) – *Parallel distributed processing – explorations in the microstructure of cognition: foundations*. Volume 1. The MIT Press, Cambridge, MA.

Schildt, Herbert (1995) – *C++. The Complete Reference*. (Tradução para o castelhano *C++ Manual de Referencia*. Madrid, Osborne/McGraw-Hill.)

Simon, Richard (1996) – *Windows 95 WIN 32 programming API bible*. Corte Madera, Waite Group Press.

Skapura, David (1996) – *Building Neural Networks*. New York, ACM Press.

Tsoukalas & Uhrig (1997) – *Fuzzy and Neural Approaches in Engineering*. New York, John Wiley & Sons, Inc.

Wessels & Barnard (1992) – *Avoiding false local minima by proper initialization of connections*. *IEEE Transactions on Neural Networks*, vol. 3, no. 6, 899-905.

Winkelstein, Alan et alii (1998) – *White Cell Manual*. Philadelphia, F.A. Davis Company.

Young, Michael (1996) – *Mastering Visual C++ 6*. San Francisco, Sybex.