Inteligência Computacional

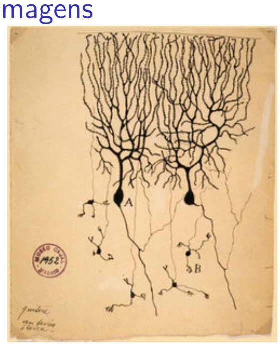
Luís A. Alexandre

UBI

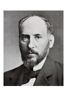
Ano lectivo 2023-24

Conteúdo

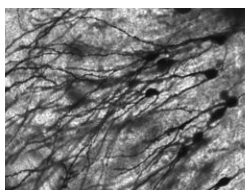
Imagens



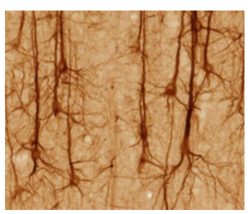
Cerebelo do pombo



Ramón y Cajal



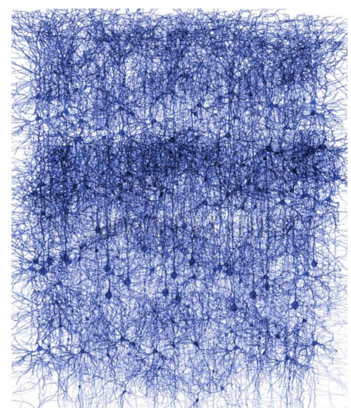
Hipocampo humano



Cortex humano

Imagens da wikipedia: `http://en.wikipedia.org/wiki/Neuron`

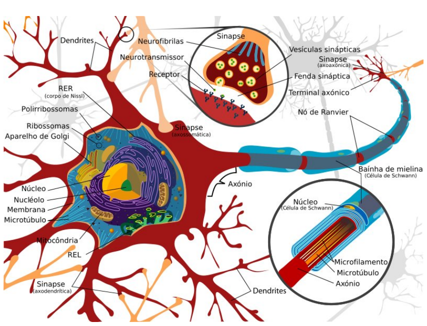
Imagens



Parte do neocortex duma ratazana

Imagem do BlueBrainProject: `http://mediatheque.epfl.ch/sv/Blue_brain`

Composição



- O neurónio é composto por:
  - núcleo
  - corpo celular
  - dendrites
  - axónio
  - sinapses

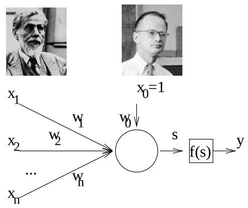
Imagem da wikipedia: `http://pt.wikipedia.org/wiki/Imagem:Complete_neuron_cell_diagram_pt.svg`

Alguns factos

- O número de neurónios no cérebro varia muito com a espécie.
- Estima-se que o cérebro humano (adulto) tenha 10^{11} neurónios e 10^{15} sinapses (uma criança de 3 anos tem 10 vezes mais).
- O cérebro de um nemátodo (um verme) (Caenorhabditis elegans) tem apenas 302 neurónios tornando-o num objecto ideal de estudo: os cientistas conseguiram mapear todos os seus neurónios.
- A mosca da fruta (Drosophila melanogaster) tem cerca de 300 mil neurónios, o que já permite que exiba alguns comportamentos complexos.

Modelo do neurónio artificial

- Modelo proposto por McCulloch e Pitts em 1942:



- De facto não é mais que uma função $g : \mathbb{R}^n \to \mathbb{R}$ que a cada vetor x de entrada, de dimensão n, faz corresponder um real y. Esta função $g(\cdot)$ depende ainda do vetor de pesos $[w_0 w_1 \dots w_n]$ e da função de ativação escolhida $f(\cdot)$.
- Tem uma entrada 'especial' chamada de viés (bias em inglês) que permite deslocar a função de ativação. Exemplo: se o viés for negativo, a soma pesada das entradas tem de superar o seu valor para que o neurónio produza um valor positivo na saída.

Cálculo da saída do neurónio

- Os valores presentes as entradas do neurónio são alvo de uma soma pesada:

$$s = \sum_{i=0}^{n} x_i w_i$$

sendo que $x_0 = 1$.
- De seguida, este valor passa por uma função (normalmente) não-linear, chamada função de ativação, produzindo a saída do neurónio: $y = f(s)$
- Em geral as funções de ativação são monótonas crescentes e verifica-se que (com a exceção da função de ativação linear e da ReLU):

$$f(-\infty) = -1 \text{ ou } f(-\infty) = 0$$

e

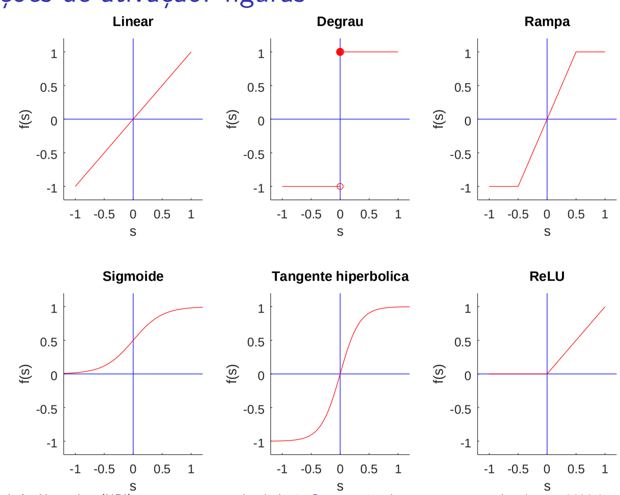
$$f(\infty) = 1$$

Funções de ativação

A função de ativação $f(\cdot)$ pode assumir muitas formas, entre elas:

- linear (saída em $(-\infty, \infty)$): $f(s) = \beta s$
- degrau (step ou Heaviside) (saída em $[\beta_1, \beta_2]$):
$$f(s) = \begin{cases} \beta_2 & , s \geq 0 \\ \beta_1 & , s < 0 \end{cases}$$
Normalmente faz-se $\beta_1 = 0$ e $\beta_2 = 1$, embora também se use $\beta_1 = -1$.
- rampa (saída em $[-\beta_1, \beta_1]$): $f(s) = \begin{cases} \beta_1 & , s \geq \beta \\ s & , |s| < \beta \\ -\beta_1 & , s \leq -\beta \end{cases}$
- sigmóide (saída em $(0, 1)$): $f(s) = \frac{1}{1+\exp(-\lambda s)}$
- tangente hiperbólica (saída em $(-1, 1)$): $f(s) = \frac{\exp(\lambda s)-\exp(-\lambda s)}{\exp(\lambda s)+\exp(-\lambda s)}$
- ReLU (Rectified Linear Unit) (saída em $[0, \infty)$): $f(s) = \max(0, s)$

Funções de ativação: figuras

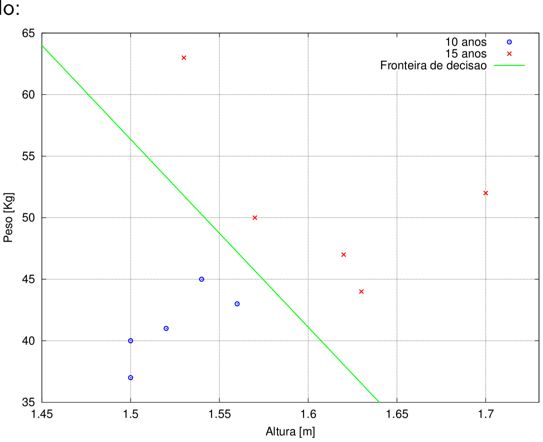


Capacidade de discriminância dum neurónio

- Um neurónio consegue discriminar (distinguir) pontos do espaço de entrada que sejam linearmente separáveis.
- O espaço de entrada corresponde ao espaço onde se encontram os vetores com as medições.
- Um neurónio consegue implementar um hiperplano (chamado fronteira de decisão) separando os pontos em que a sua saída é >= 0 daqueles que produzem um valor na saída <0.

Capacidade de discriminância dum neurónio

- Exemplo:



Capacidade de discriminância dum neurónio: exemplos

- Exemplo de valores para os pesos dum neurónio, que usa a função degrau entre 0 e 1, para que:
  - calcule o OU lógico entre duas variáveis: $w = [-0.25, 0.5, 0.5]$.
  - calcule o E lógico entre duas variáveis: $w = [-0.75, 0.5, 0.5]$.
  - calcule o OU lógico exclusivo entre duas variáveis: $w = [??]$

Aprendizagem

- Tendo ficado claro como é que o neurónio age quando lhe é apresentado um vetor na entrada, fica a faltar perceber como é que se encontram os valores dos pesos w_i.
- Para encontrarmos os pesos precisamos de dados relativos ao problema a resolver.
- Estes dados são conjuntos de pontos (ou vetores) de entrada que representam o problema em questão: por exemplo, os pontos do problema de distinguir através do peso e altura entre crianças de 10 e 15 anos são os seguintes:

```
1.50, 40, 0
1.52, 41, 0
1.56, 43, 0
1.54, 45, 0
1.50, 37, 0
1.62, 47, 1
1.70, 52, 1
1.53, 63, 1
1.63, 44, 1
1.57, 50, 1
```


Aprendizagem

- Existem vários tipos de aprendizagem que poderemos usar, mas vamos considerar apenas dois:
  - supervisionada: os pontos contêm informação sobre a que classe pertencem (Se são de crianças de 10 ou de 15 anos)
  - não supervisionada: os pontos não contêm informação relativa à sua classe
- Para o caso em questão vamos ver como efetuar a aprendizagem supervisionada: cada ponto no espaço de entrada do problema é constituído por uma ou mais características e a etiqueta da classe à qual o ponto pertence. Para o exemplo anterior vem:

$$P_1 = (peso_1, altura_1, classe_{P_1})$$

Aprendizagem

- De uma forma mais genérica representamos um ponto do seguinte modo:

$$P_i = (x_{i,1}, \dots, x_{i,n}, d_i)$$

e chamamos aos valores que são colocados na entrada no neurónio, $x_{i,1}, \dots, x_{i,n}$ as características e d_i é a verdadeira classe do ponto i.

Descida do gradiente

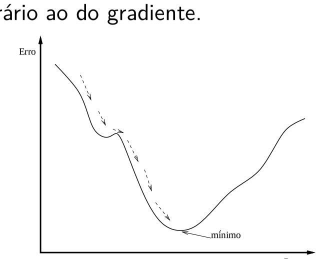
- É a forma de aprendizagem mais usada em redes neuronais.
- É necessária a definição de uma função de custo (em inglês chamada a loss function ou apenas a loss) que permita saber quão próximo da verdadeira classe ficou a saída do neurónio.
- A função de custo mais comum é o quadrado do erro:

$$e_j = (d_j - y_j)^2$$

onde d_j é a classe verdadeira e y_j a saída do neurónio para o ponto j.

Descida do gradiente

- A ideia da descida do gradiente consiste em achar os pesos que minimizam o erro, procurando no espaço dos pesos ao longo do sentido contrário ao do gradiente.



Descida do gradiente

- Dado um ponto j, vejamos como se efetua o ajuste dos pesos.
- Cada peso w_i na iteração t é ajustado de acordo com

$$w_i(t) = w_i(t-1) + \Delta w_i(t)$$

onde

$$\Delta w_i(t) = \eta \left(-\frac{\partial e_j}{\partial w_i} \right)$$

e

$$\frac{\partial e_j}{\partial w_i} = -2(d_j - y_j)\frac{\partial f}{\partial s_j} x_{j,i}$$

e η é a chamada taxa de aprendizagem, $x_{j,i}$ é a característica i do ponto j e f é a função de ativação.

- Para que o termo $\frac{\partial f}{\partial s_j}$ não anule esta derivada e percamos a informação relativa ao gradiente, f não pode ser o degrau.

Descida do gradiente

- Se usarmos o sigmóide como função de ativação vem:

$$\Delta w_i(t) = 2\eta(d_j - y_j)\lambda f(s_j)(1 - f(s_j))x_{j,i} = 2\eta(d_j - y_j)\lambda y_j(1 - y_j)x_{j,i}$$

- Isto porque, se $f(x)$ é o sigmóide, então temos:

$$\frac{df(x)}{dx} = -(1 + \exp(-\lambda x))^{-2}(-\exp(-\lambda x)) =$$

$$\frac{\lambda}{1 + \exp(-\lambda x)} \left(\frac{\exp(-\lambda x)}{1 + \exp(-\lambda x)} \right) = \lambda f(x) \frac{1 + \exp(-\lambda x) - 1}{1 + \exp(-\lambda x)} =$$

$$\lambda f(x) \left(1 - \frac{1}{1 + \exp(-\lambda x)} \right) = \lambda f(x)(1 - f(x))$$

Widrow-Hoff

- A regra de Widrow-Hoff, proposta em 1960, é um caso particular da descida do gradiente em que se considera a função de ativação como a função identidade, $f(s_j) = s_j$, logo vem que

$$\frac{\partial f}{\partial s_j} = 1$$

- Assim, a atualização do valor dos pesos neste caso é feito com

$$\Delta w_i(t) = 2\eta(d_j - y_j)x_{j,i}$$

- Este algoritmo é também chamado de Least Mean Squares ou Delta Rule.

Widrow-Hoff

- O hardware construído por Widrow e Hoff para o implementar eram as Adalines.
- Festejaram-se os 50 anos da construção da primeira Adaline no ano de 2009.

Imagem do paper [1]

Leitura recomendada

- Engelbrecht, cap. 2.
- [1] `http://www-isl.stanford.edu/~widrow/papers/t1960anadaptive.pdf`