

Workshop

REDES NEURONAIS ARTIFICIAIS

aplicações e potencialidades



REDES NEURONAIS Conceitos

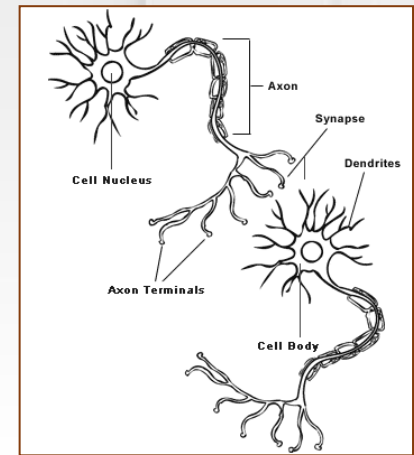
Jorge M. Santos
jms@isep.ipp.pt

Definição e enquadramento histórico

A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use [Simon Haykin].

It resembles the brain in two respects:

- 1. Knowledge is acquired by the network from its environment through a learning process.*
- 2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.*



Marcos históricos

McCulloch e Pits (1943) – Introdução do primeiro modelo matemático do neurónio artificial.

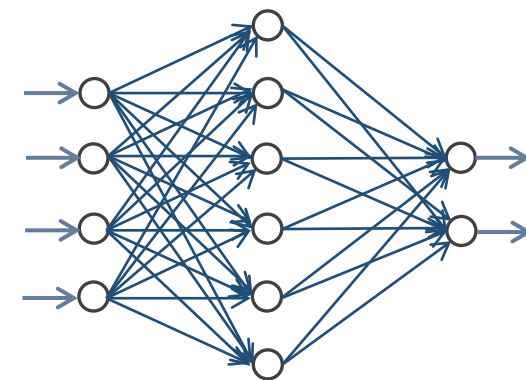
Rosenblatt (1958) – Perceptrão

Minsky e Papert (1969) – Limitação das redes neronais.

Paul Werbos (1981) – Algoritmo de retro propagação (*backpropagation*) para redes neuronais

...

Deep learning in neural networks: An overview, Jürgen Schmidhuber, 2015.



Rede Neuronal

Uma rede neuronal é um mapeador. Mapeia uma variável de entrada x , numa variável de saída y .

A variável de entrada corresponde aos atributos ou características do nosso problema.

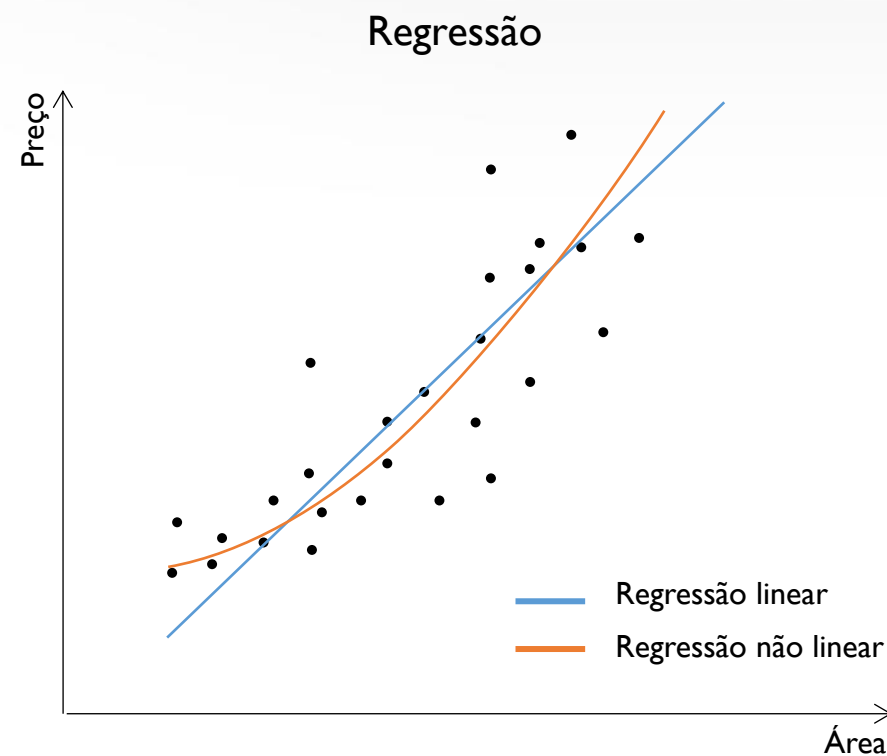
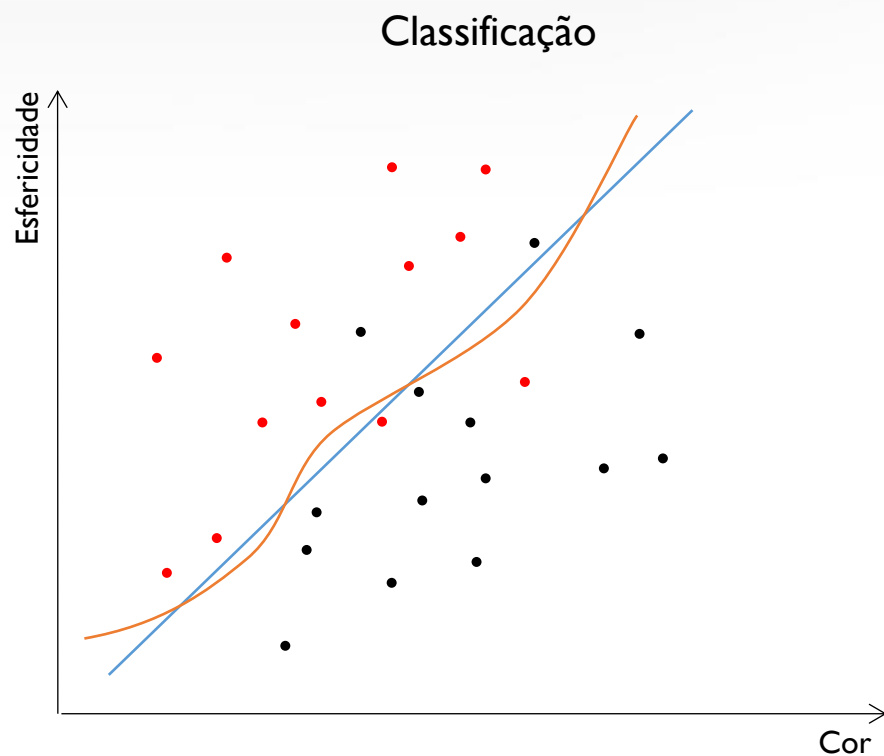
A variável de saída corresponde ao que se pretende atingir. Pode ser discreta (classificação) ou contínua (regressão).



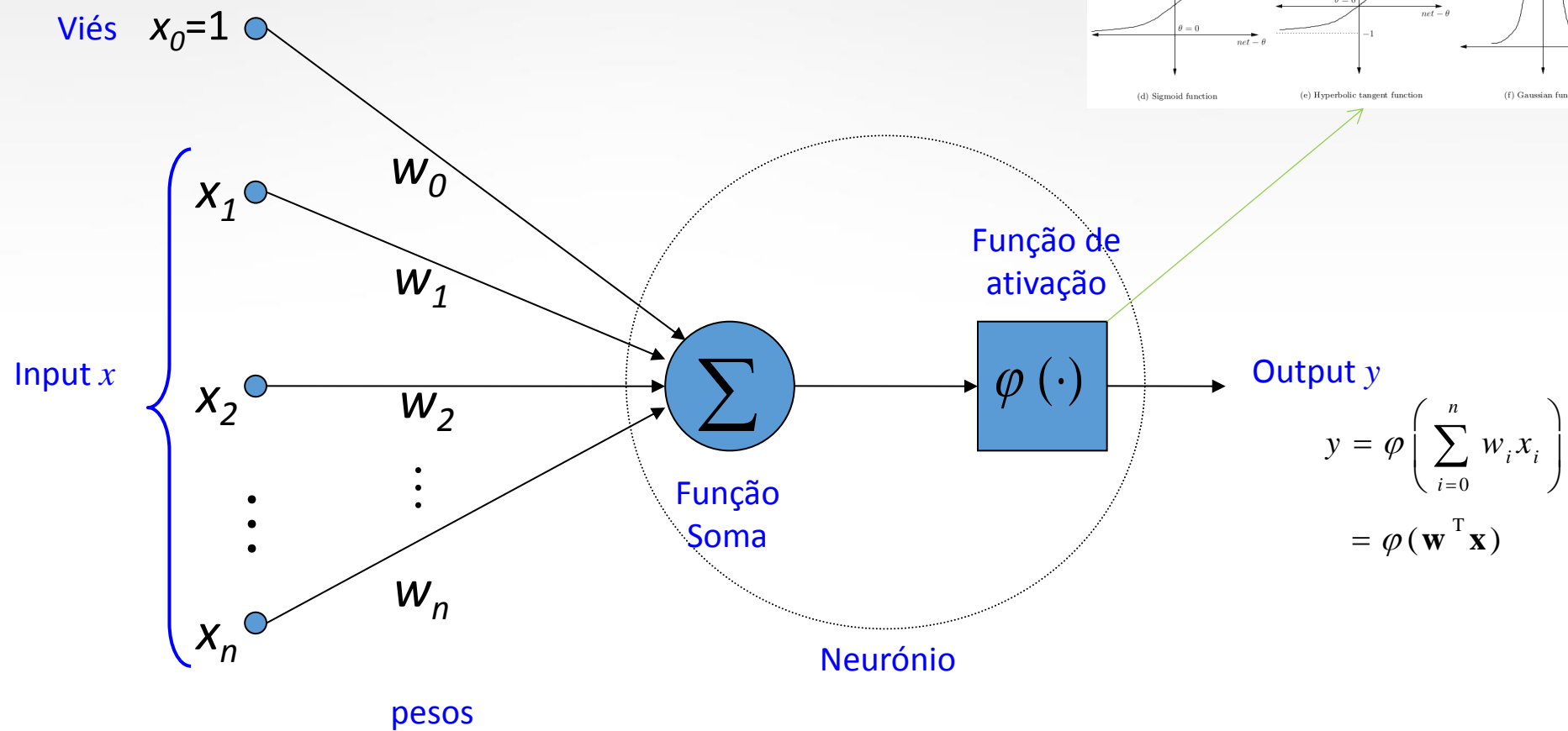
Classificação versus Regressão

Num problemas de classificação o objetivo é encontrar a fronteira de decisão que melhor permita distinguir as classes.

Num problema de regressão pretende-se encontrar a equação que descreva a relação entre duas ou mais variáveis.



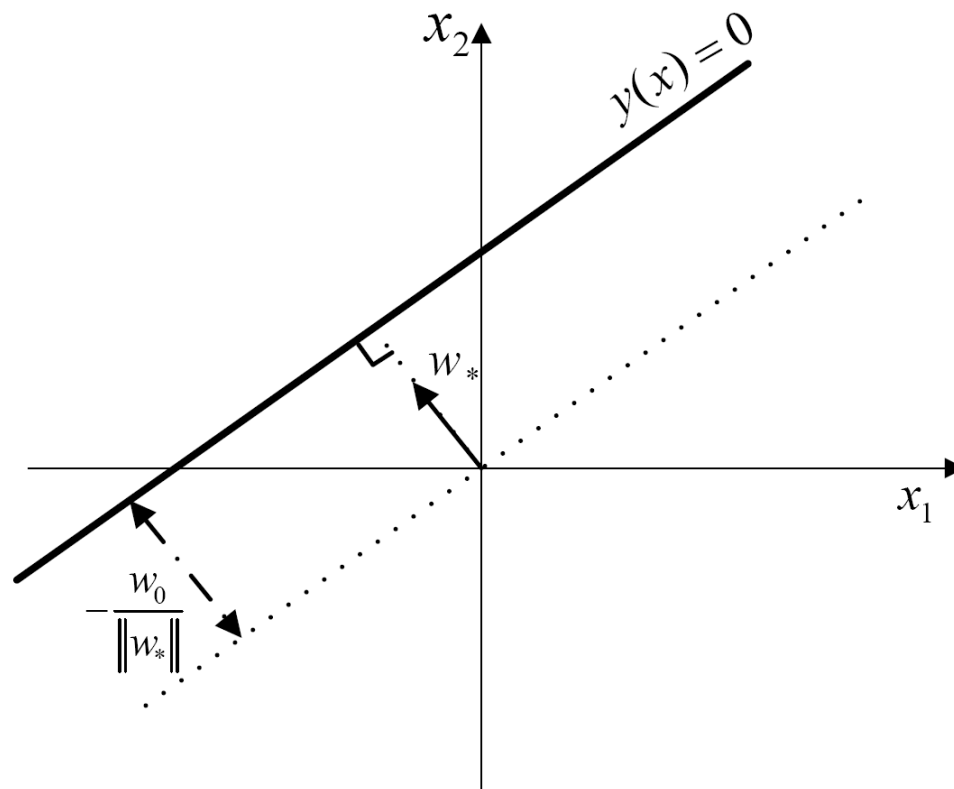
○ Perceptrão



O hiperplano de decisão

Os pesos da rede w_* (sem o peso do viés) definem a orientação (inclinação) do hiperplano e o peso associado ao viés, w_0 , controla a posição do hiperplano em termos da sua distância à origem.

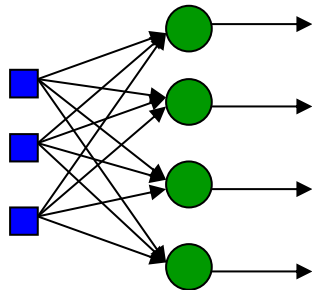
Na figura representa-se um exemplo para um hiperplano de duas dimensões, uma reta, correspondente a um problema com duas características.



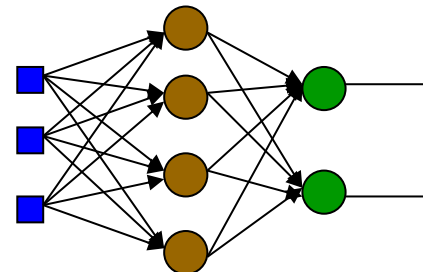
O perceptron multi-camada (MLP)

O perceptron multi-camada é uma rede *feed-forward*: um grafo acíclico, direcionado (geralmente completamente ligado), em que as entradas de cada camada são as saídas da camada anterior.

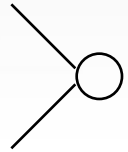
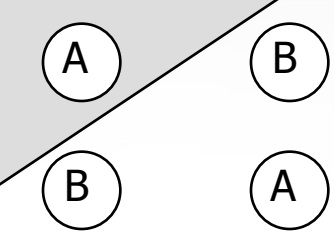
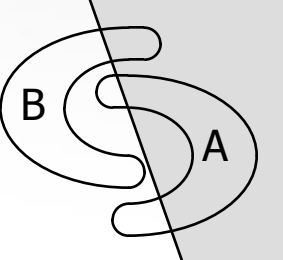
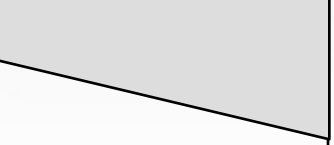
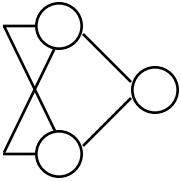
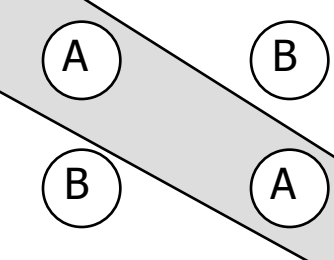
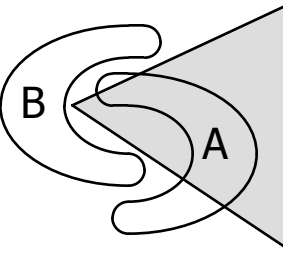
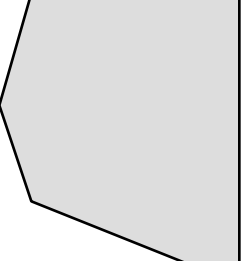
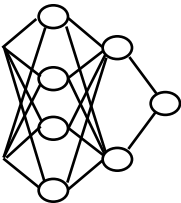
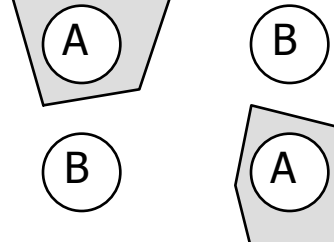
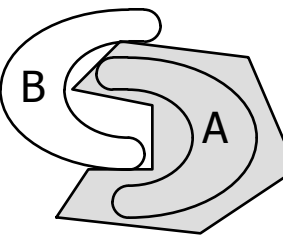
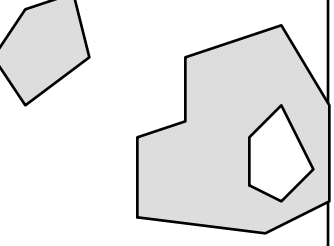
Uma camada



Duas camadas



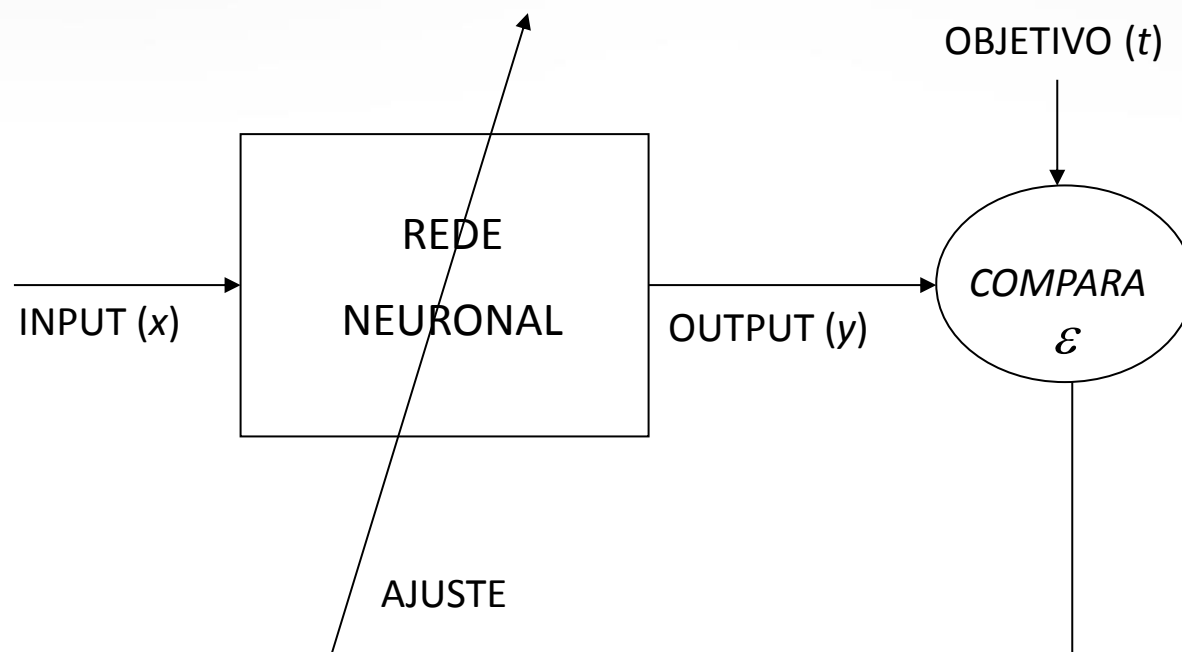
As regiões de decisão num problema de classificação

Structure	Types of Decision Regions	Exclusive-OR Problem	Classes with Meshed regions	Most General Region Shapes
Single-Layer 	Half Plane Bounded By Hyperplane			
Two-Layer 	Convex Open or Closed Regions			
Three-Layer 	Arbitrary (Complexity Limited by Number of Nodes)			

Adaptado de Lippman, 1987

A aprendizagem

A aprendizagem da rede neuronal consiste em ajustar os pesos da rede por forma a minimizar o erro cometido entre a saída da rede e o objetivo pretendido. Usa-se um conjunto de treino e um conjunto de validação para evitar ajuste exagerado ou deficiente dos pesos ao problema e um conjunto de teste para avaliar a performance da rede.



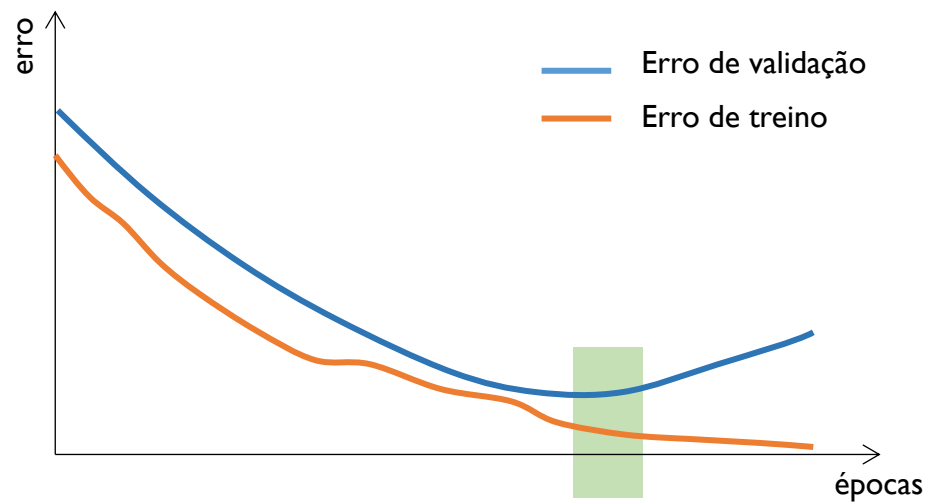
O algoritmo de treino

A atualização dos pesos da rede faz-se através da minimização do erro dado por uma **função de custo**, sendo que as mais comuns são o erro quadrático para a regressão e a *cross-entropy* para classificação.

Os algoritmos baseados na **descida do gradiente** são os mais usados no processo de minimização do erro.

O treino da rede pode ser feito usando a abordagem **Batch** ou a abordagem **Online**. Na primeira, cada atualização dos pesos é feita após a apresentação à rede de todos os casos do conjunto de treino: na segunda, a atualização dos pesos é feita após a apresentação de cada caso. À passagem completa de todos os casos pela rede dá-se o nome de **época**.

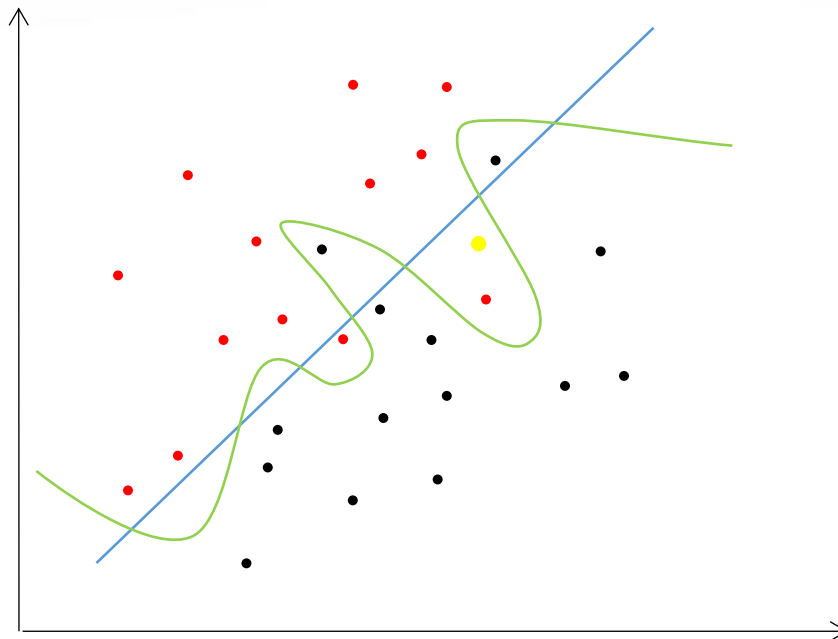
O número de épocas de treino pode ser fixo mas o mais apropriado é que seja orientado pelo erro no conjunto de validação.



Generalização e Complexidade da Rede Neuronal

Um dos aspetos mais importantes das redes neuronais é garantir o seu poder de generalização, isto é, garantir que a performance da rede se mantém para dados não conhecidos. Uma rede mais complexa (mais neurónio/camadas) pode sofrer de *over-training*, ou seja, ser treinada em demasia provocando ajuste exagerado aos dados de treino. Uma rede muito simples (poucos neurónios/camadas) pode não ter complexidade suficiente para modelar um problema mais complexo.

Uma das estratégias para conseguir uma rede equilibrada consiste em efetuar um treino com validação cruzada. Outra forma é usar estratégias de regularização.



Alguns Tipos de Redes Neurais

Multi Layer Perceptron

Radial Basis Function

Kohonen Self-organizing Maps

Recurrent Neural Networks

Modular Neural Networks

Time Delay Neural Networks

Convolutional Neural Networks

Deep Neural Networks





Alguns Conselhos

- Aquisição dos dados que garanta uma amostra representativa.
- Pré-processar os dados (qu岸tos mais melhor!): valores omissos ou repetidos, outliers, normalização. Selecionar atributos ou produzir novos usando os existentes.
- Escolher o tipo de rede neuronal que melhor se adapte ao problema.
- Treinar a rede usando validação cruzada. Efetuar uma busca no espaço dos parâmetros (o mais exaustiva possível).
- Testar em dados não usados para treino.
- Escolher a rede com a melhor performance (atenção ao índice de performance usado!).
- Obter a rede final treinando a rede com a melhor performance usando todos os dados disponíveis.
- Usar a rede final para prever ou classificar dados novos.
- Dados novos podem ser usados para re-treinar e melhorar a rede.



Software com Redes Neurais

Existem vários tipos de software específico para Redes Neurais (RNs) mas, os mais usados na comunidade científica, são os que possuem um conjunto variado de ferramentas que permitem pré-processar os dados, aplicar vários tipos de classificador/regressor e fazer análise estatística dos resultados. Alguns possuem uma interface orientada por módulos o que facilita a implementação de um determinado processo. Alguns exemplos:

MatLab – Comercial. Existe uma Toolbox dedicada exclusivamente às RNs.

Octave – Free. Semelhante ao MatLab.

R – Free. Muito usado na comunidade científica também possui funções específicas para RNs.

Weka – Free. Uma das primeiras multi-ferramentas com vários algoritmos de Machine Learning.

Rapid Miner – Free*. Ferramenta de Data mining mas que também possui módulos dedicados às RNs.

PyBrain – Módulo de Machine Learning para Python que contém RNs.

Bibliografia

Christopher Bishop, Neural Networks for Pattern Recognition, Clarendon Press;, 1 edition, 1996.

Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

Jürgen Schmidhuber, Deep learning in neural networks: An overview, Neural Networks, 61, pp. 85-117, Elsevier, 2015.

R. P. Lippman, An Introduction to Computing with Neural Nets, IEEE Acoustics, Speech and Signal Processing Magazine, 1987.

Sandhya Samarasinghe, Neural Networks for Applied Sciences and Engineering, Auerbach Publications, Taylor and Francis, 2007.

Simon Haykin, Neural Networks and Learning Machines, Prentice Hall, Third Edition, 2008.

Stuart Reid, 10 Misconceptions about Neural Networks, <http://www.turingfinance.com/misconceptions-about-neural-networks/> , 2014

