Redes Neuronais

Índice

**Não foi encontrada nenhuma entrada de índice.**

# Perceptrão e Outras Unidades

Um perceptrão:

* Tem um conjunto de entradas de valor real (x1 ... xn)
* Calcula uma combinação linear destas entradas, C
* Saida
  + +1, se C > -w0 (threshold)
  + -1, caso contrário

Uma imagem com texto, file, diagrama, Tipo de letra

Descrição gerada automaticamente

Portanto:

Uma imagem com texto, Tipo de letra, captura de ecrã, file

Descrição gerada automaticamente

* Os *wi* designam-se por Coeficientes Sinápticos ou de Ponderação (weighting factors) que determinam a contribuição da entrada *xi* para a adição realizada no seio do perceptrão.
* Se x0 = 1 ( fixo em vez de entrada variável), então -w0 representa o limite que a combinação linear das entradas tem de ultrapassar para a saída ser +1.
  + Seja  o vetor dos coeficientes de ponderação
  + Seja o das entradas
* Então a saída O do perceptrão, função apenas das entradas considerando os pesos invariáveis, é:

Uma imagem com texto, Tipo de letra, file, captura de ecrã

Descrição gerada automaticamente

Tipos de funções de ativação mais comuns:

Uma imagem com diagrama, Esquema, Desenho técnico, file

Descrição gerada automaticamente

O que pode “aprender” ou “representar” um perceptrão?

* Seja um perceptrão de 2 entradas *x1*, *x2* e coeficientes *w1=1*, *w2=1*
* Seja *w0=0*
* A sua saída será
  + *+1,* se *x1.w1 + x2.w2 > 0*
  + *–1,* se *x1.w1 + x2.w2 < 0*
* Como estamos a considerar *wi=1*

Uma imagem com texto, Tipo de letra, branco, caligrafia

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, file, diagrama

Descrição gerada automaticamente

Se os coeficientes sinápticos não fossem 1 teríamos:

* *x1.w1 + x2.w2 = 0*
* *X2 = -w1/w2.x1*
* Ou seja, os valores (relativos) de w1 e w2 regulam o declive da reta

E se w0 não fosse 0, teríamos:

* *x1.w1 + x2.w2 + w0= 0*
* *X2 = -w1/w2.x1- w0/w2*
* Ou seja, o valor de w0 (relativamente a w2) regula a ordenada na origem

Isso possibilita a separação de exemplos + e – em situações diversas.

Por isso a aprendizagem se faz regulando os valores dos w’s de modo a ajustar a reta ao conjunto de exemplos de treino fornecidos, i.e., a separar os + dos -.

Como as duas zonas + e – são separáveis por uma reta, diz-se que um perceptrão pode representar (apenas) Funções Linearmente Separáveis

Uma imagem com file, diagrama, captura de ecrã, Gráfico

Descrição gerada automaticamente

Nesta figura:

* (a) *I1+I2=1.5* Saída + se I1 e I2 forem ambas próximas de 1: AND
* (b) *I1+I2=0.5*  Saída – se I1 e I2 forem ambos próximos de 0: OR
* (c) Representa a função XOR. Esta função NÃO é Linearmente Separável porque nenhuma reta consegue separar os exemplos positivos dos negativos. Um único perceptrão não pode representar um XOR.

A reta que separa os exemplos positivos dos negativos chama-se ***superfície de decisão***.

Para um perceptrão é sempre linear, mesmo que a função de ativação não o seja. Por exemplo, para a unidade sigmoide cuja função de ativação é traduzida por:

Uma imagem com file, diagrama, Gráfico, texto

Descrição gerada automaticamente

A condição de decisão será:



# Treino de Perceptrões e de Outras Unidades

Algoritmos de Treino

1. Perceptrion Traning Rule
2. Gradient Descent
3. Stochastic Approximation to Gradient Descent (Delta Rule)

Em qualquer deles:

* Ao perceptrão ou unidade são aplicadas entradas que constituem exemplos de treino classificados com “positivos” ou “negativos”
* A cada um corresponde um valor alvo (target) que se compara com um resultado: A saída apresentada pelo perceptrão ou unidade linear
* Se este resultado gerar uma classificação errada, reajustam-se os coeficientes sinápticos
* Terminado o treino, em presença de exemplos diferentes dos utilizados, o perceptrão ou unidade deverá responder corretamente (generalização)