

BIG DATA

MACHINE LEARNING

MULTILAYER PERCEPTRON
SELF ORGANIZING MAP
REDES DE PROBABILIDADE

Humberto Sandmann
humberto.sandmann@gmail.com

Roteiro do curso

Aula 1: Introdução

Conceitos, mercado, tendências e arquitetura

Aula 2: Ferramentas

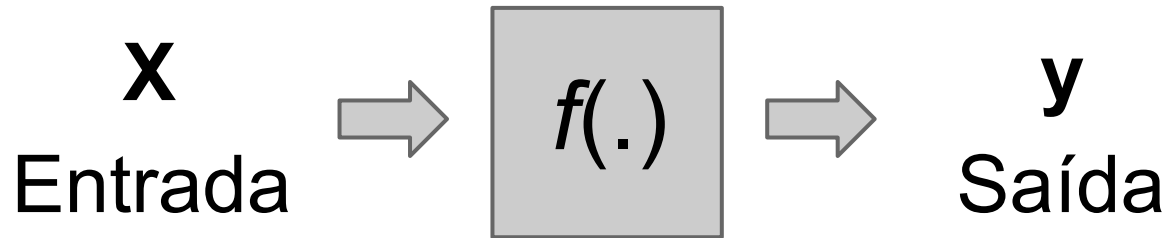
Hadoop, MongoDB, Neo4j e Solr

Aula 3: HBase, Hive e tratamento de dados

Aula 4: Machine learning

Perceptron - MLP, DeepLearning, SOM,
Redes probabilística (Redes Bayesianas)

Aprendizado de Máquina



Tipos de Aprendizado

Supervisionado

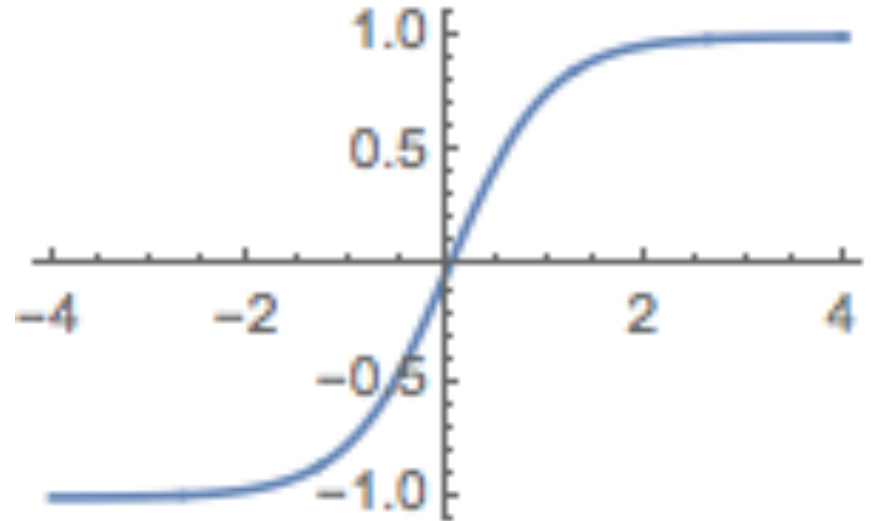
Não supervisionado

Entrada de dados (normalização)

Domínio da função

Ex:

$$x \in [-1;1]$$



Técnicas Abordadas - Conexionistas

Multilayer perceptron (MLP)

Self Organizing Map (SOM)

Bayesian Decision Tree

Multilayer Perceptron (MLP)

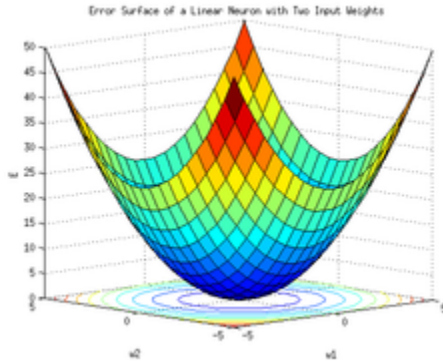
Aprendizado supervisionado

Aplicação:

Classificação

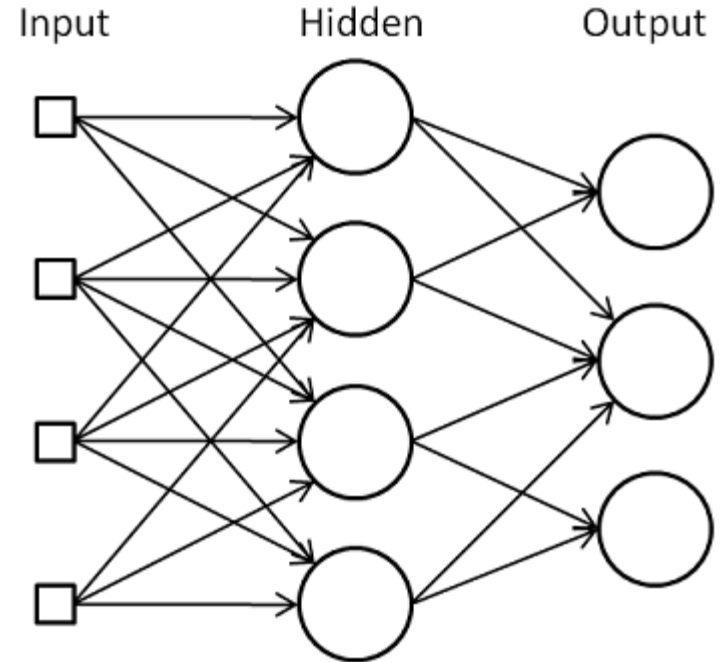
Predição temporal

$$o_j = \varphi(\text{net}_j) = \varphi\left(\sum_{k=1}^n w_{kj}x_k\right)$$



$$\mathcal{E}(n) = \frac{1}{2} \sum_j e_j^2(n)$$

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n)$$



Self Organizing Map (SOM)

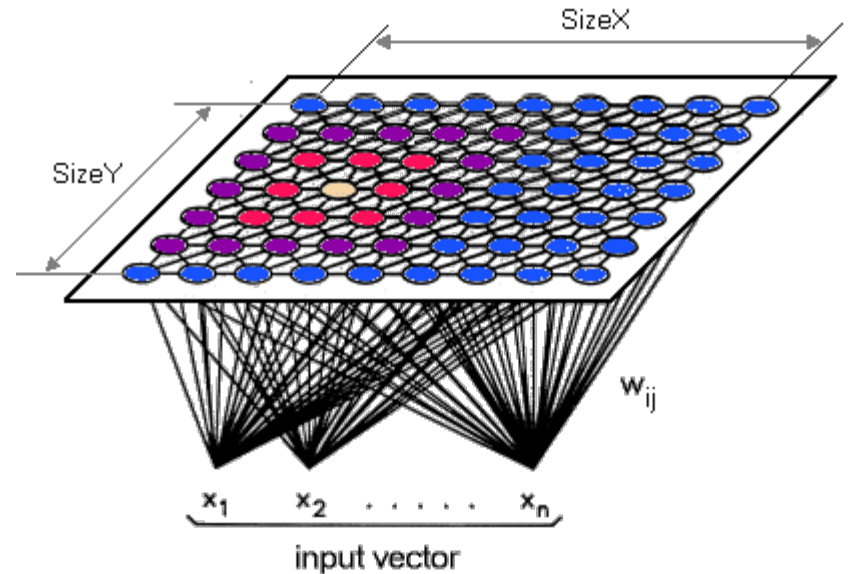
Aprendizado não-supervisionado

Função compacta $f(\mathbf{x}) \rightarrow g(\mathbf{x})$

$$i(\mathbf{x}) = \arg \min_j \|(\mathbf{x}(n)) - (\mathbf{w}_j)\|, j = 1, 2, \dots, l$$

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n)h_{j,i(\mathbf{x})}(n)(\mathbf{x}(n) - \mathbf{w}_j(n))$$

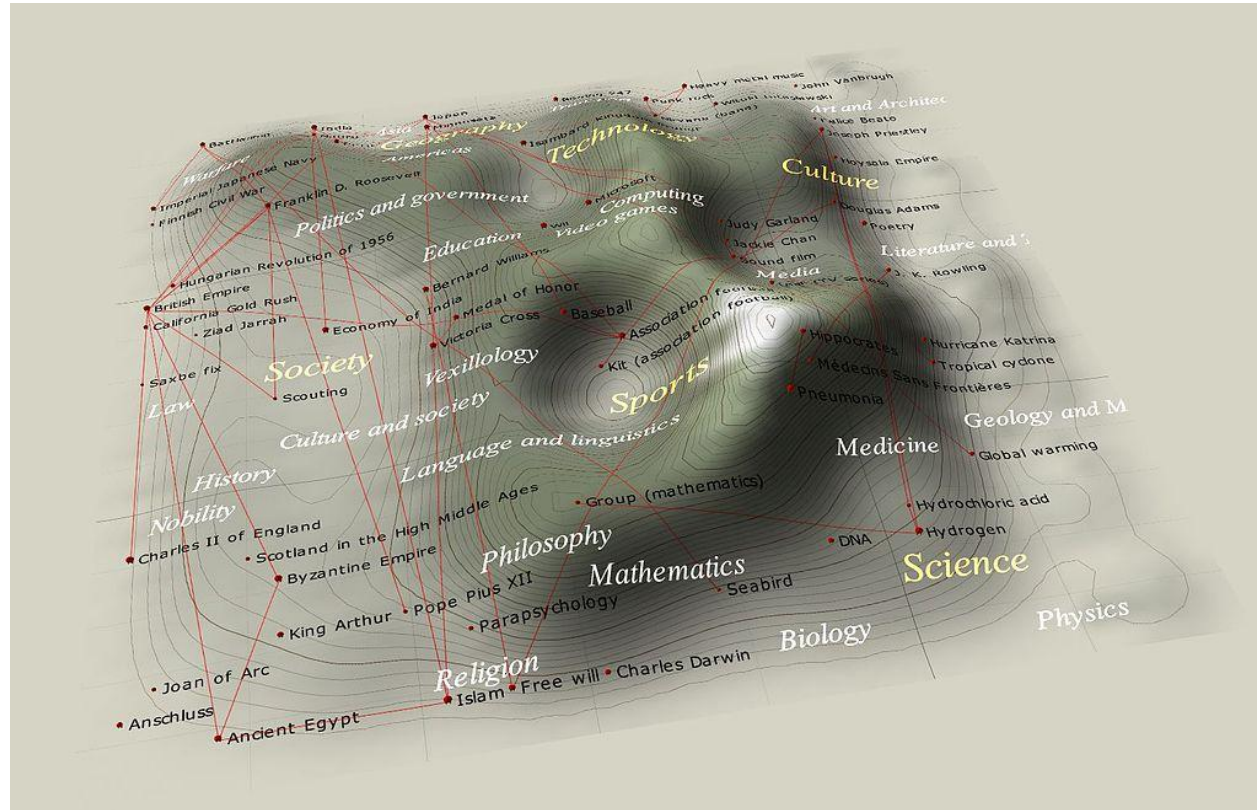
Representação dos dados em um mapa
U-Map



Self Organizing Map (SOM)

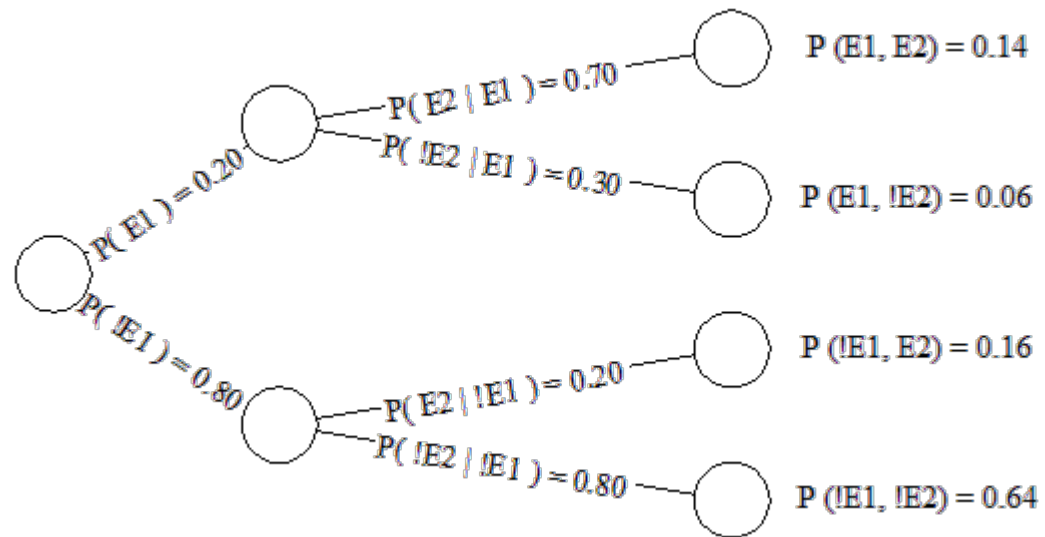
Cartographical representation of a self-organizing map (U-Matrix) based on Wikipedia featured article data (word frequency). Distance is inversely proportional to similarity. The "mountains" are edges between clusters. The red lines are links between articles.

from Wikipedia



Bayesian Decision Tree

$$P(H | E) = \frac{P(E | H) \cdot P(H)}{P(E)}$$



Linguagem R

Linguagem de programação

Ambiente interativo de comando

Plataforma gráfica

Open source



Exemplo MLP sobre o R

```
library(neuralnet)
set.seed(1) # make results reproducible
N=500
x <- cbind(runif(N, min=1, max=500), runif(N, min=1, max=500), runif(N, min=1, max=500), runif(N, min=1,
max=500), runif(N, min=1, max=500))
y <- ifelse(x[,1] + 2*x[,1] + 3*x[,1] + 4*x[,1] + 5*x[,1] > 3750, 1, 0)
trainSMALL <- data.frame(x[1:(N/10),], y=y[1:(N/10)])
trainALL <- data.frame(x, y)
n <- names(trainSMALL)
f <- as.formula(paste('y ~', paste(n[!n %in% 'y'], collapse = ' + ')))
netSMALL <- neuralnet(f, trainSMALL, hidden = c(5,5), threshold = 0.01)
netALL <- neuralnet(f, trainALL, hidden = c(5,5), threshold = 0.01)
print(netSMALL) # error 4.117671763
print(netALL) # error 0.00959846187! # get a sense of accuracy w.r.t small training set (in-sample)
cbind(y, compute(netSMALL,x)$net.result)[1:10,]
# get a sense of accuracy w.r.t full training set (in-sample)
cbind(y, compute(netALL,x)$net.result)[1:10,]
plot(netALL) # show the network
```

<http://stats.stackexchange.com/questions/65292/r-neuralnet-compute-give-a-constant-answer>

Exemplo MLP

Objetivo: construir uma máquina que identifique o idioma de um texto

Passos:

1. Dividir as amostras em conjunto de aprendizado e conjunto de testes
2. Ler do HDFS um arquivo texto de aprendizado em um determinado idioma
3. Normalizar o texto para a entrada da rede neural
4. Fazer um processo de aprendizado
5. Voltar ao passo 2 até que o **critério de parada** seja obedecido
6. Testar e validar a rede através de um conjunto de testes

Hands on

Bibliografia

S.Haykin, Redes Neurais: princípios e prática, 2001

