**Perceptron**

Leitura Principal

<http://angom.myweb.cs.uwindsor.ca/teaching/cs574/le2.pdf>

Tutorial Perceptron em Python

<https://pythonmachinelearning.pro/perceptrons-the-first-neural-networks/>

<https://zahidhasan.github.io/2017-09-10-perceptron/>

Leitura complementar

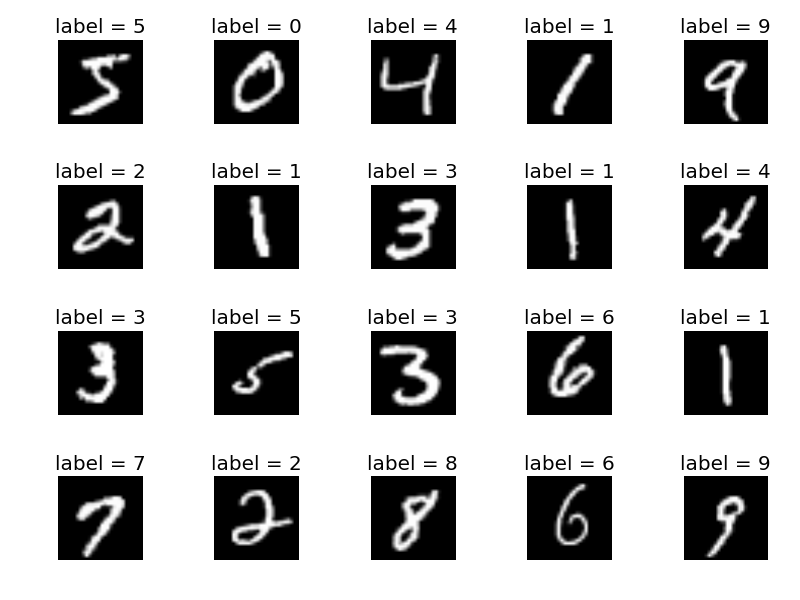
<http://130.243.105.49/~lilien/ml/seminars/2007_02_01b-Janecek-Perceptron.pdf>

<http://www.divms.uiowa.edu/~tinelli/classes/145/Fall05/notes/8c_learning.pdf>

<https://www.dca.ufrn.br/~heitorm/aulasDCA/dca0121/DCA0121-03-RedesNeurais-Perceptron-Adaline_v1.0.pdf>

**Data set – MNIST**

Representação de imagens de números escritos a mão.



https://www.dropbox.com/s/u417rws4uc612om/mnist.zip?dl=0

Depois de descompactar o arquivo vocês terão dois arquivos *mnist\_test.csv*

*mnist\_train.csv*, respectivamente com exemplos de teste e treino já separados.

Cada exemplo/amostra de treinamento é um dígito (0-9) representado por uma matriz de 28x28 pixels. Essa matriz está “rearranjada” no arquivo de treinamento como um “vetor” de 784 posições.

Cada linha dos arquivos representa uma amostra.

A primeira coluna de cada linha dos arquivos contém a classe a qual a amostra pertence.

As outras colunas têm os 784 valores dos pixels que representam uma amostra.

**Tarefa**

Criar um classificador para os dígitos do MNIST baseado em perceptrons de McCulloch-Pitts.

Sugestão, treinem um perceptron para aprender cada dígito. Depois criem uma lógica para decidir, a partir dos dez perceptrons, qual é o dígito correto.

**Artigo da semana**

https://www.eecis.udel.edu/~shatkay/Course/papers/NetworksAndCNNClasifiersIntroVapnik95.pdf

**Dicas de comandos úteis para ipython/jupyter**

<http://pages.physics.cornell.edu/~myers/teaching/ComputationalMethods/python/ipython.html>