# Exercises #10 - Introduction to ANNs

Thales Menezes de Oliveira

Brazilian Center for Physics Research (CBPF)

April 1, 2022

2. Why is it generally preferable to use a Logistic Regression classifier rather than a classical Perceptron (i.e., a single layer of linear threshold units trained using the Perceptron training algorithm) How can you tweak a Perceptron to make it equivalent to a Logistic Regression classifier?

A: The classical Perceptron is only suited when the dataset is linearly separable, which isn't usually the case. On the other hand, the Logistic Regression is able to operate in non-linearly separable dataset, and can also estimate the class probabilities. Starting with the Perceptron, if we use a logistic activation function, or the softmax if we are dealing with more than one class, we will be able to estimate class probabilities. Training this modified Perceptron with a optimization algorithm that minimizes the cost function, this Perceptron will be equivalent to the Logistic Regression Classifier.

3. Why was the logistic activation function a key ingredient in training the first MLPs?

A: A MLP only will be able to learn if the derivative of the activation functions is nonzero. So the logistic activation function was crucial to training the MLPs since its derivative is always nonzero, so the SGD is able to update the weights of the NN and progress the learning process. For the Heavside step function, for example, where its derivatives is always zero, the SGD isn't able to update the weights of NN, impossibilitating it of learn.

4. Name three popular activation functions. Can you draw them?

A: The Rectified LInear Units and its variants, which is great to deal with vanishing gradient problems since it doesn't saturate. The sigmoid activation functions can be used in the output layer to allow the NN to estimate class probabilites for each instance. The hyperboloic tangent can be used in the hidden layers as well, but it saturates for very large inputs.

5. Suppose you have an MLP composed of one input layer with 10 passthrough neurons, followed by one hidden layer with 50 artificial neurons, and finally one output layer with 3 artificial neurons. All artificial neurons use the ReLU activation functions

- What is the shape of the input matrix $\mathbf{X}$?
- What about the shape of the hidden layer's weight $w_h$, and the shape of its bias vector $b_h$?
- What is the shape of the output layer's weight $w_0$, and its bias vector $b_0$?
- What is the shape of the network's output matrix $\mathbf{Y}$?
- Write the equation that computes the network's output matrix $\mathbf{Y}$ as a function of $\mathbf{X}$, $w_h$, $b_h$, $w_0$, $b_0$.

A:

6. How many neurons do you need in the output layter if you want to classify email into spam or ham? What activation function should you use in the output layer? If instead you want

to tackle MNIST, how many neurons do you need in the output layer, using what activation functions? Answer the same questions for getting your network to predict housing prices as in Chapter 2.

A: To make the binary classification it will be needed two neurons in the output layer, alternatively, one neuron can be used to estimate the probability of the e-mail to be spam. For tackle the MNIST dataset, 10 neurons at the output layer will be necessary since, the dataset is composed by intances distributed in 10 different classes. Also it will be needed to replace the sigmoid function to the softmax, to deal with the multiclassification task, which can output the probability for each class. For regression and prediction tasks, no activation function is necessary in the output layer, as well only one neuron in that layer, since it will be used to predict the housing prices.

7. What is backpropagation and how does it work?

A:

8. Can you list all the hyperparameters you can tweak in an MLP? If the MLP overfits the training data, how could you tweak these hyperparameters to try to solve the problem?

A: