In this report, we’ll analyze some basics in using Matlab. Following this idea, we’ll implement a simple sigmoid function, along with its derivative.

*Exp1)* To obtain this, we’ll use 2 for loops to implement the iterations through the elements if the matrix:

we’ll follow the same rules for the derivative and we obtain:



In this part, we’ll perform this task on our matrix M where:



and the result of our sigmoid will be:



*Exp 2) (Neural Network Application)* In this experiment, we define a simple neural network using Matlab’s toolbox to classify the **AND** dataset. To obtain this, we define **Inputs** and **outputs** as:

we then run the toolbox using **nnstart** command from the Matlab’s command line.

This diagram illustrates that training loss is minimized. However, the validation and test loss is not compelling. We can conclude that overfitting has occurred with this model. Playing around with this network won’t improve the final results. **Reason**: there are probably some deficiencies in the model and the dataset is not good as well. We can augment our data in some cases, but it can also result in overfitting again. The important thing is that our dataset should be comprehensive enough.

*Exp 3) (Pattern Recognition Application)* We’ll use the internal glasses dataset which consists of 2 classes.

Confusion Matrix is a useful tool to understand the degree of **sensitivity** and **specificity**, accuracy and other scores of evaluations.

**Note**: a neural network is not an appropriate model for noisy data (not robust enough).