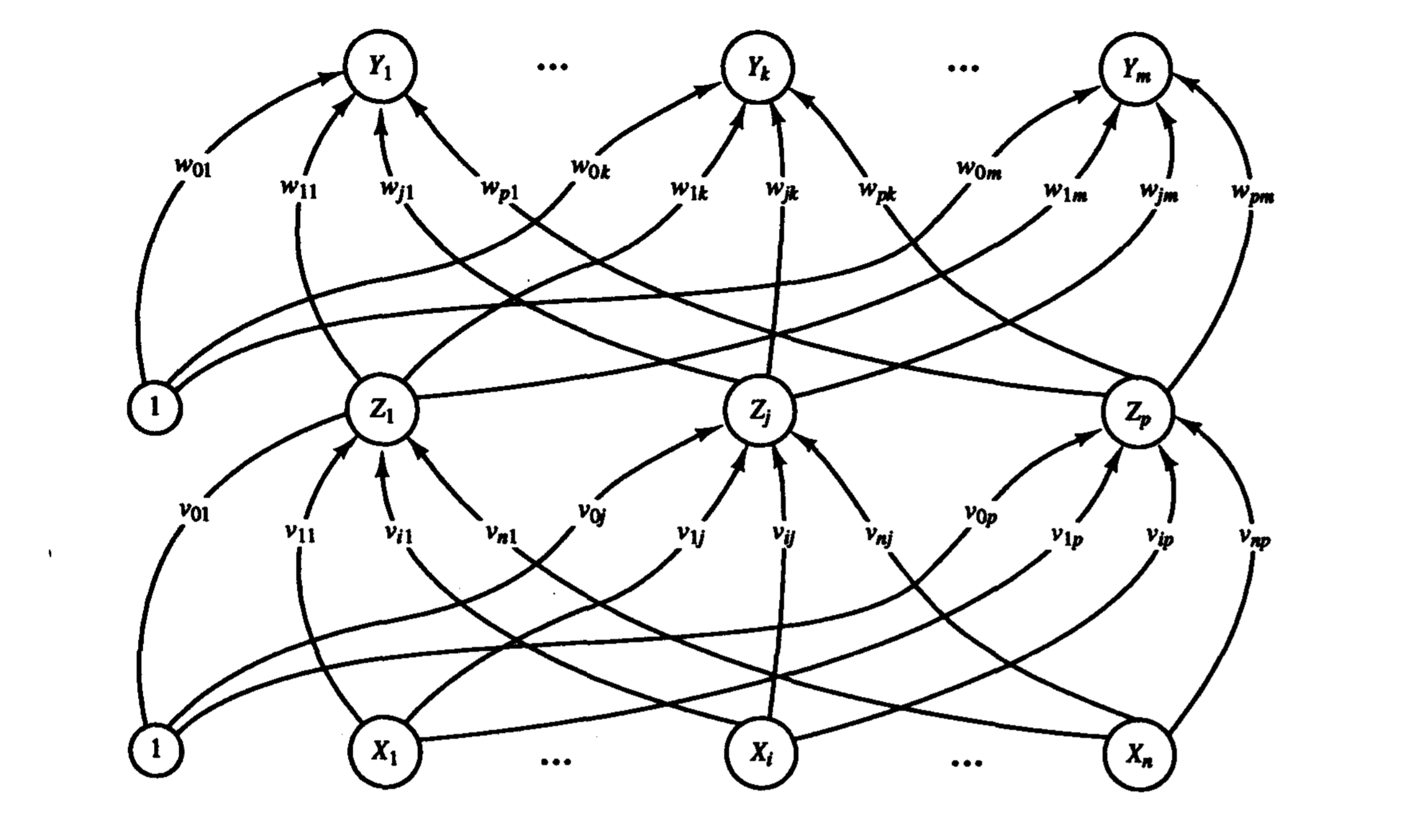
**Authors: Snir Shaharabani Lioz Elmalem and Gal Caspi**

**Backpropagation**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), **backpropagation** (**backprop**, **BP**) is a widely used [algorithm](https://en.wikipedia.org/wiki/Algorithm) in training [feedforward neural networks](https://en.wikipedia.org/wiki/Feedforward_neural_network) for [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning).



**Report**-**Backpropagation:**

Constants: all weights: random [-0.2,0.2]

bias: 1.0

alpha: 0.1

momentum: 0.1

\*Momentum- One method that has been proposed is a slight modification of the backpropagation algorithm so that it includes a momentum term. Applied to backpropagation, the concept of momentum is that previous changes in the weights should influence the current direction of movement in weight space.

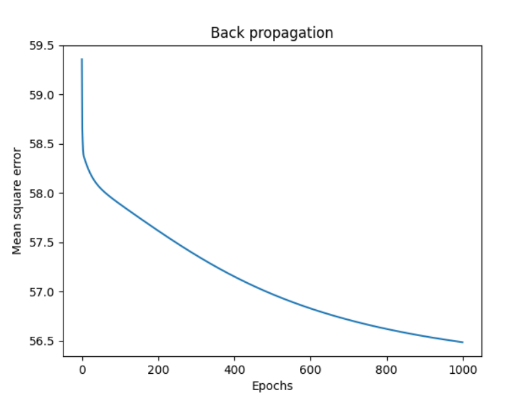
iterations: 1000

1. test percent: 25%

train percent: 75%

accuracy: 44%

time: 5.39 sec

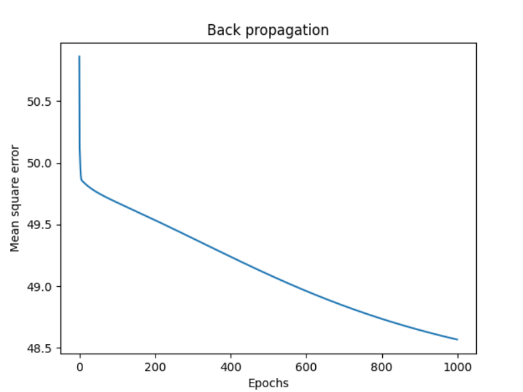


1. test percent : 33%

train percent : 67%

accuracy: 47.05%

time: 5.4 sec

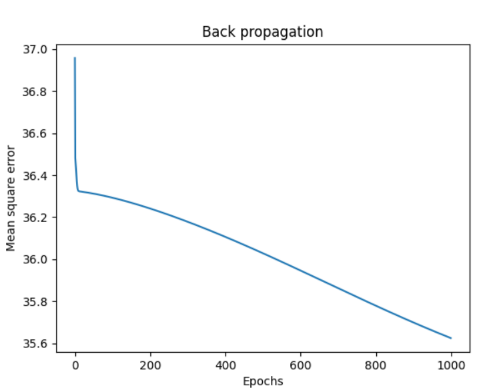


1. test percent : 50%

train percent : 50%

accuracy: 75.75%

time: 3.79 sec



\*As we can see from the above graphs, the higher the test percentage, the higher the accuracy.

\*Average of cross validation: 55.6%

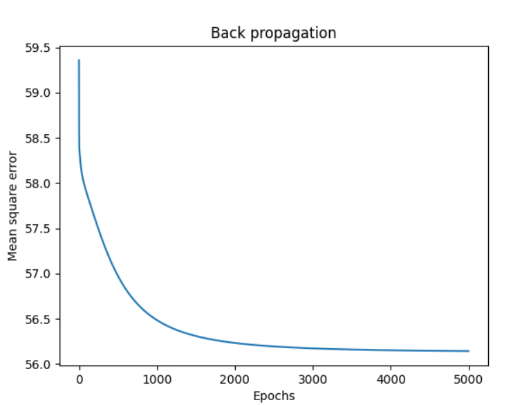
iterations: 5000

1. test percent: 25%

train percent: 75%

accuracy: 50%

time: 16.94sec

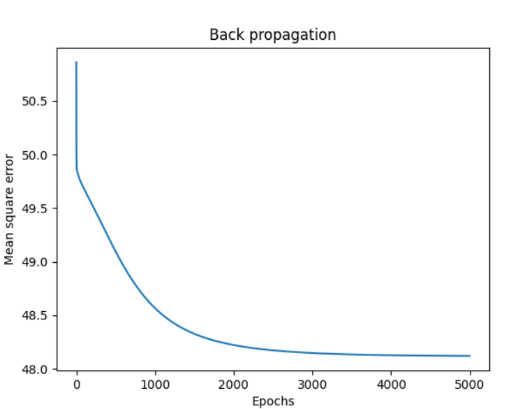


1. test percent: 33%

train percent: 67%

accuracy: 52.94%

time: 15.36 sec

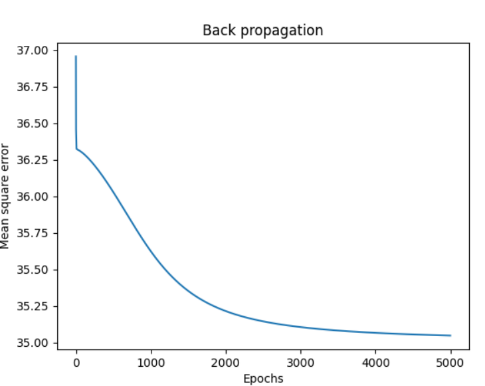


1. test percent : 50%

train percent : 50%

accuracy: 72.72%

time: 11.92 sec



\*As in previous iterations, we can see from the above graphs, the higher the percentages of the test, the higher the accuracy.

\*Average of cross validation: 58.55%

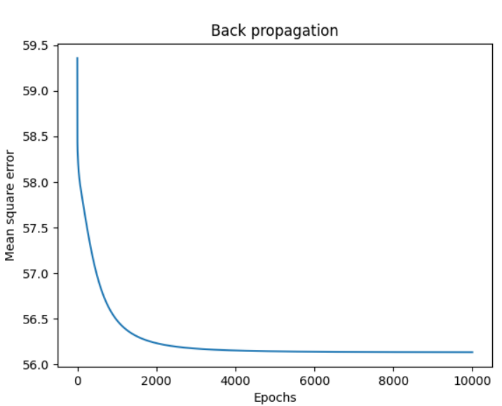
iterations: 10000

1. test percent : 25%

train percent : 75%

accuracy: 50%

time: 29.53 sec

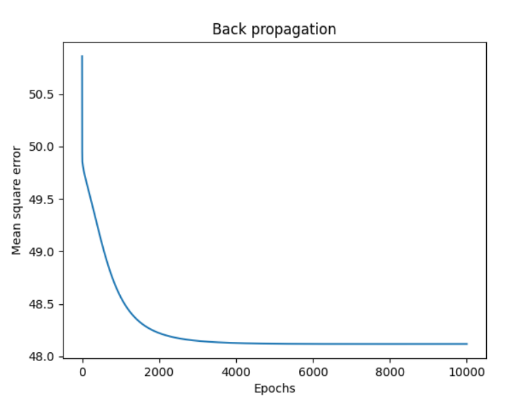


1. test percent : 33%

train percent : 67%

accuracy: 54.41%

time: 26.07 sec

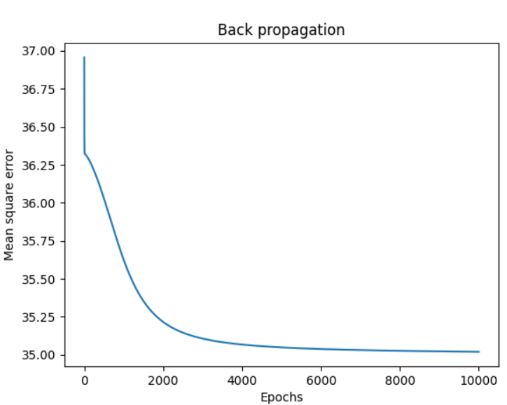


1. test percent : 50%

train percent : 50%

accuracy: 73.73%

time: 20.21 sec



\*As in previous iterations, we can see from the above graphs, the higher the percentages of the test, the higher the accuracy.

\*Average of cross validation: 59.38%

All parameters can be changed simply.

**Conclusion:**

We started the project from collecting information about the Backpropagation algorithm. Once we understood how the algorithm works, we started writing code in Python.   
The given examples are only a small but representative selection of all test runs and measurements performed for this project and illustrate the advantages and disadvantages of genetic methods for the training of neuronal networks. Theoretic analysis and the optioned practical results suggest the following conclusions:

From the data we received, it was very difficult to get past the 76% accuracy (we explained in summary about the Adaline above).

In a test with 33% and 67% training, as the number of iterations increases, so does the accuracy. Because with many iterations, the algorithm learns from errors of previous iterations.

When the test percentages are 25% and training 75%, between 5000 and 10000 iterations, the accuracy remains the same.

When testing percentages are 50% and training 50%, between 1000 and 5000 iterations, the accuracy level drops.

In summary, like Adaline, the optimal percentage of accuracy in Backpropagation is 33% testing and 67%.

\*Link: We took code from [https://github.com/jgabriellima/backpropagatio](https://github.com/jgabriellima/backpropagatio%20) and we changed it