# Researching Stage 9 – Improving Predictions

## Machine Learning

Guresen, Kavakutlu, and Daim (2011) explored comparing the performance of neural networks for predicting daily exchange rate values of the NASDAQ stock exchange index. They compared a traditional multilayer perceptron (MLP), the dynamic artificial neural network (DAN2), and a hybrid neural network which uses generalised autoregressive conditional heteroscedasticity (GARCH). With the correct level of training artificial neural networks generally achieve more accuracy than traditional methods in the majority of scenarios, but they are rarely compared with each other. The conclusion was that the multilayer perceptron was superior at the time, but they predicted that DAN2s may surpass them in performance with improved architecture. The MLP had a difference of just 0.54% between predicted and real values.

Jiang and Liang, (2017) attempted to use reinforcement learning to predict cryptocurrency value changes with a convolutional neural network but concluded that their experiment less successful than previous studies.

Persio and Honchar (2016) compared neural networks as a means to predict S&P 500 historic time series (investing.com UK, 2018) prices based on previous days’ values. They used a multilayer perceptron, a convolutional neural network, and a long short-term memory recurrent neural network. A multilayer perceptron was selected as the most basic neural network implementation appropriate for a continuous function that maps real number inputs to real number outputs (Csáji, 2001). A convolutional neural network was selected as it works similarly to a multilayer perceptron, but with the added benefit of breaking an input vector down into subsets of higher level features. This allows it to learn during training instead of requiring sophisticated pre-processing. Finally, a recurrent neural network (specifically, the long short-term memory architecture) was selected as their neurons are able to send feedback signals to each other to consider sequential information. The results indicated that convolutional neural networks are the superior method for predicting stock market data when comparing these architectures, with the long short-term memory neural network performing marginally worse.

Considering the above research, it would appear that a multilayer perceptron is required if a neural network implementation is to be used to improve the prediction accuracy of the project. Using a convolutional or long short-term memory architecture should further improve the prediction accuracy.

## Researching approaches

As with any project, taking an approach where everything is created from scratch by the development team can be both extremely useful and wasteful simultaneously. Each component developed gives the team a fuller, more insightful understanding of how it works and thus a superior ability to quality assure, expand, or change the product. However, this also means that each complex component will require significantly more implementation time. If this project were to be assigned a greater amount of time, then it would be highly desirable to develop a neural network in Java from scratch. In its current state time is very finite, so the following technologies have been researched.

### TensorFlow

TensorFlow (2018) is an open-source software library released by Google that can be used for machine learning. It is Java compatible and the website has a variety of information available including Maven installation instructions and tutorials for starting both convolutional and recurrent neural networks. It is also used to power a large number of well-known companies’ products, so it is unlikely that it will be discontinued in the near future. Unfortunately, the Java compatible version of the technology does not have backwards compatibility guarantees afforded to the C API.

### DeepLearning4J

Deeplearning4j (2018) is an open-source software library released by Eclipse that can be used for machine learning. It is written specifically for Java and Scala, and the website has a variety of information available including Maven installation instructions. There are also tutorials for implementing long short-term memory, convolution, and recurrent neural networks. There is a more basic deep learning tutorial for beginners which even assists in selecting the correct neural network for a given problem.

### MATLAB

MATLAB (2018) is an IDE and programming language that specialises in the manipulation of matrices and arrays with mathematical equations. It has a toolbox which provides neural networks through a wizard and even prescribes recommendations for which kind to use on a variety of problems. Using MATLAB would be a reasonable decision due to Plymouth University having a license. However, purchasing a commercial license is expensive so it would be prudent to use an open-source solution.

## Selection

If given infinite time to continue the project, it would be preferable to continue by designing and developing a neural network without any library assistance. If given more time to continue the project but less time than required to do this, DeepLearning4J would be the preferable option. This is because its purpose is to be used with Java applications and there is no hefty commercial license fee, although there is for a more powerful version.

# References

Csáji, B. C. (2001) Approximation with Artificial Neural Networks. Eötvös Loránd University, Hungary.

Deeplearning4j. (2018). *Deeplearning4j: Open-source, Distributed Deep Learning for the JVM*. [online] Available at: <https://deeplearning4j.org/> [Accessed 10 May 2018].

Guresen, E., Kayakutlu, G., and Daim, T. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, 38(8), pp.10389-10397.

investing.com UK. (2018). S&P 500 Historical Rates - Investing.com UK. [online] Available at: <https://uk.investing.com/indices/us-spx-500-historical-data> [Accessed 9 May 2018].

Jiang, Z. and Liang, J. (2017). Cryptocurrency portfolio management with deep reinforcement learning. *2017 Intelligent Systems Conference (IntelliSys)*.

MATLAB. (2018). *MATLAB - MathWorks*. [online] Available at: <https://www.mathworks.com/products/matlab.html> [Accessed 10 May 2018].

Persio, L. D. and Honchar, O. (2016). Artificial Neural Networks architectures for stock price prediction: comparisons and applications.

TensorFlow. (2018). *TensorFlow*. [online] Available at: <https://www.tensorflow.org/> [Accessed 10 May 2018].