The Impact of Hidden Layer Complexity on Overfitting of Recurrent Neural Networks for Prediction of Stock Returns

Emma Munter
Jheronimus Academy of Data Science (JADS)
munter.emma@gmail.com

Abstract

Stock returns are a source of complex time-series data whose prediction is highly valuable. An essential part of improving the reliability of neural networks for this purpose is selecting the most effective hyperparameters. This paper presents an evaluation of the accuracy of recurrent neural networks with autoregressive moving references (AR-MRNN). A comparison is made between multiple versions of the network, each with a different number of nodes in the hidden layer, seeking to answer the question: What is the effect of hidden layer complexity on overfitting of AR-MRNN models on stock return data?

1 Introduction

The prediction of movements of the financial markets have long been a source of research material. Due to the high complexity of the driving forces behind this movements, neural networks have emerged as the most commonly used prediction method. This is due to three main reasons: they have proven to result in high prediction accuracies in empirical research, they do not require samples to be normally distributed, and they can deal with non-linear problems [Chen *et al.*, 2013]. However, the exact characteristics of different neural network architectures when applied for this purpose is still subject to current research.

The neural network architecture used in this paper is the *autoregressive moving reference recurrent neural network* (AR-MRNN), as introduced by Rather *et al.* [2015] (not to be confused with the *autoregressive moving reference neural network* it is based on, as introduced by Freitas *et al.* [2006]). Through the results to the experiments laid out in this paper, an attempt is made at shedding light on the applicability of AR-MRNN models to the prediction of future movements in the financial markets

2 Method

Several versions of the AR-MRNN are created, each differing only in the number of nodes contained in the hidden layer. Based on the most commonly used parameters for autoregressive moving reference predictors, a regression order of 4 is chosen, and no delay is added to the moving reference [Rather *et al.*, 2015; Freitas *et al.*, 2009]. The resulting model is detailed in the section below.

2.1 The Models

Suppose there is time-series data R of length T on past returns of a stock, as follows:

$$R = (r_{t-(T-1)}, ..., r_{t-1}, r_t) \tag{1}$$

The neural network has inputs $(r_t - z, r_{t-1} - z, r_{t-2} - z, r_{t-3} - z)$, and output $r_{\widehat{t+1}} - z$, where the reference value z is defined as r_{t-4}. The reference value is subtracted from all inputs, and added to the output in order to obtain the final prediction.

The networks consist of 4 layers: an input layer consisting of 4 nodes, a long-term memory layer also consisting of 4 nodes, a hidden layer consisting of n nodes, and an output layer consisting of a single node. All layers are connected sequentially, with additionally the input layer also being directly connected to the hidden layer. A graphical representation can be seen in Fig.1, for a network with $n = 16$. Both the hidden layer and the output layer use a sigmoidal activation function.

Because the research by Rather *et al.* [2015] uses a model with 16 nodes in the hidden layer, all experiments performed in the context of this paper focus on comparing models with anywhere between 1 and 31 nodes.

3 Experimental set-up

3.1 Data

The results shown in this paper are from models trained on the daily close value of the NASDAQ 100 index, between 2018-05-10 and 2019-05-10. Due to the 5-day time span of the input, and due to weekends and holidays, this resulted in a 128 training cycles per stock per epoch, of which 102 were used for training and 26 were used for validation. The same models were also validated on the data from the same period of the Dow Jones Industrial Average index and the S&P 500 index. These experiments yielded similar results, so, for clarity and brevity, all data shown pertains to the NASDAQ 100 index only.[1]

[1] The code and data used to perform this research can be found at https://github.com/EmmaMunter/AR-MRNN-for-Stock-Prediction

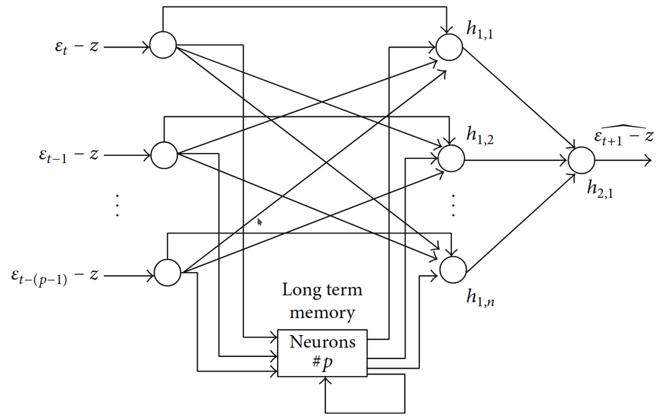

Figure 1: Graphical representation of the architecture with 16 hidden nodes, as used by Rather *et al.* [2015].

3.2 Parameters

Like in the previously established research [Rather *et al.*, 2015; Freitas *et al.*, 2009], the size of input to the network is set to 4.

3.3 Experiements

Each of the models are trained with the objective of minimizing the mean absolute error (MAE) of the resulting predictions.

Experiment 1: Training overview

An initial experiment was conducted to get an overview of the effect of training over time, from which we can deduce the extent to which overfitting affects the models.

Experiment 2: Fitting speed

An experiment was conducted to evaluate the effect that hidden layer complexity has on the number of epochs required to fit to the data. This time required to fit to the data measured by training the model until it did not achieve any improvement over a period of 1000 epochs. The total number of epochs trained by this point was taken as a measure of fitting speed.

Experiment 3: Overfitting speed

An experiment was conducted to evaluate the effect that hidden layer complexity has on the severity of overfitting by when being trained for a certain period of time. The amount of overfitting is measured simply by training the model for a set number of epochs (10,000) and then evaluating the model fit. Naturally, a requirement for this measure to represent overfitting and not underfitting is that the fitting speed measured in the previous experiment is lower than the number of epochs trained during this experiment.

Experiment 4: Optimal accuracy

Through conducting this experiment we can arrive at some more practical insights into the performance of the model. What is the accuracy that can be reacht

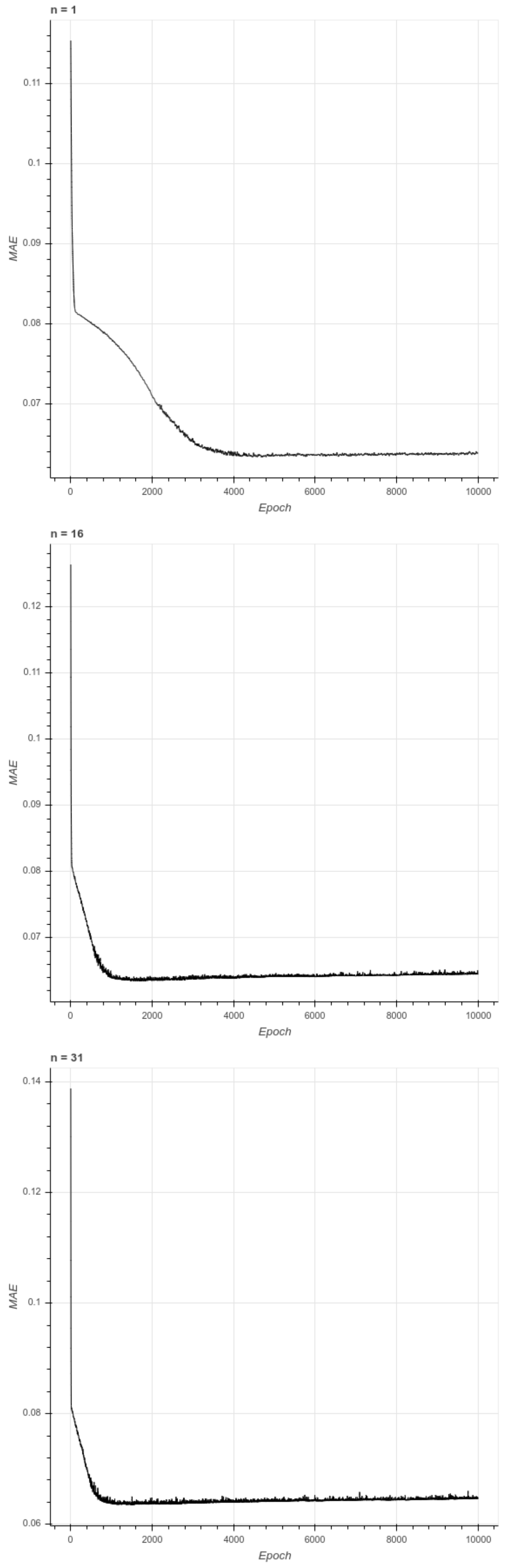

Figure 2: Accuracy over time during training for the models with 1, 16, and 31 nodes in the hidden layer.

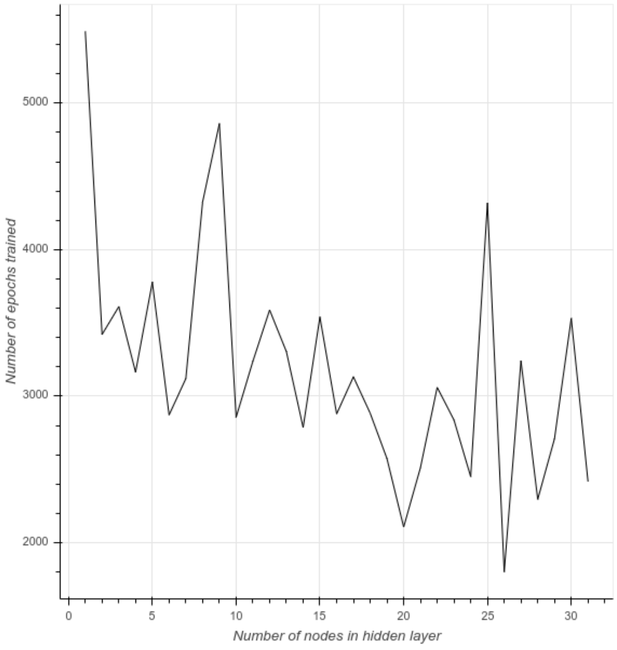

Figure 3: The value of the training speed measure laid out in section 3.3, for each model.

4 Results

Experiment 1: Training overview

The effect on the MAE of the predictions over time while training several models for 10,000 epochs can be seen in Figure 2. It can be seen that at first, the model's accuracy improves quickly, with those improvements slowly coming to a halt later on, and the accuracy eventually decreasing as the model start to overfit.

Experiment 2: Fitting speed

The effect of model complexity on the fitting speed measure put forward in section 3.3 can be seen in Figure 3. It can be seen that high model complexity is associated with fewer epochs required to fit the model.

Experiment 3: Overfitting speed

The effect of model complexity on the amount of overfitting after a 10,000 training epochs can be seen in Figure 4. There is a clear relation between high model complexity and low overfitting after the set number of epochs.

Experiment 4: Optimal Accuracy

The effect of model complexity on the optimal accuracy achieved before significant overfitting occurs can be seen in Figure 5. There is clearly a relation between high model complexity and high accuracy.
Percentage-wise there is not much difference between any of the models, and sometimes extra nodes actually decrease the accuracy. This seems to imply that the model architecture can only discern smaller, less significant patterns, and is unable to find patterns that can explain a large part of the variance. However, that is to be expected, considering the unpredictability of financial markets.

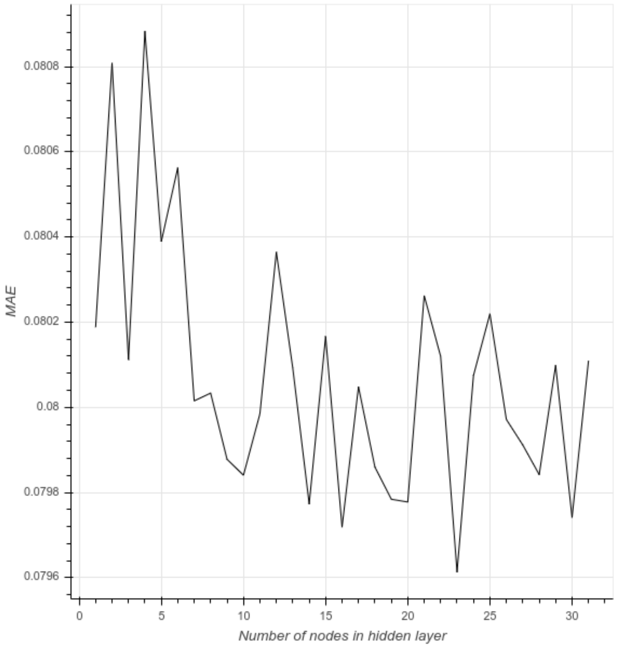

Figure 4: The accuracy achieved after 10,000 epochs, for each model.

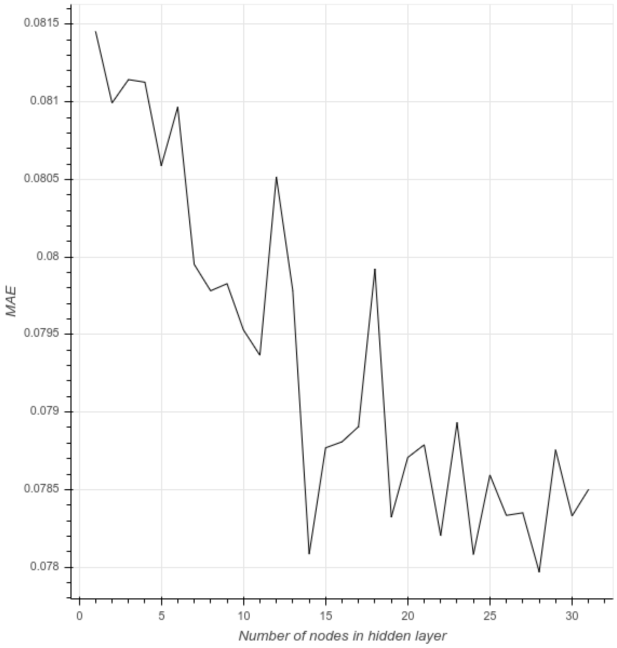

Figure 5: The optimal accuracy achieved, for each model.

5 Discussion

One issue with the results of this research is inherent to research on financial markets: the types of data that are used are not static. Different economic conditions result in different behavior of the financial markets, which can fundamentally alter the effectiveness of the AR-MRNN architecture (as well as any other architecture).

Another issue is that the small difference between the accuracies obtained using different model complexities casts doubts on the general validity of the results. Though there are clear trends related to the complexity of the hidden layer, there are other factors that might be much more significant. This could be the topic of further research, perhaps research that uses a more convoluted neural network architecture. Do note however that this means using a network with a different structure, not using a network with more neurons in a similar architecture, as previous research has shown that high numbers of nodes in neural networks applied to prediction of stock markets tend to significantly increase computational complexity with only negligible improvements to model performance [Chen *et al.*, 2013].

Another possibility for future research would be doing these same experiments on a hybrid model (e.g., the hybrid model described by Rather *et al.*[2015]), and seeing how that changes the results of the experiments.

6 Conclusion

From the results we conclude that high complexity of the hidden layer is associated with low overfitting (experiment 3), and high accuracy (experiment 4), making it models with high complexity superior with regards to model performance after having been trained.

From this we can also conclude that for practical purposes of stock prediction it is better to use the most complex model (with 31 nodes in the hidden layer), because most applications of stock return predictions value accuracy much more than they value low computation requirements, and experiments 1 and 2 showed there to be limited computational requirements associated with its superior accuracy.

References

[Chen *et al.*, 2013] Mu-Yen Chen, Min-Hsuan Fan, Young-Long Chen, and Hui-Mei Wei. Design of experiments on neural network's parameters optimization for time series forecasting in stock markets. *Neural Network World*, 23(4):369, 2013.

[de Freitas *et al.*, 2006] Fábio Daros de Freitas, Alberto Ferreira De Souza, and Ailson Rosetti de Almeida. A prediction-based portfolio optimization model. 2006.

[Freitas *et al.*, 2009] Fabio D. Freitas, Alberto F. De Souza, and Ailson R. de Almeida. Prediction-based portfolio optimization model using neural networks. *Neurocomputing*, 72(10):2155 – 2170, 2009. Lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007).

[Rather *et al.*, 2015] Akhter Mohiuddin Rather, Arun Agarwal, and VN Sastry. Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, 42(6):3234–3241, 2015.