Automated Stock Trading using Neural Networks

Matthew Knowles

Department of Mathematics
University of York

mk1320@york.ac.uk

November 28, 2021

Mathematical Underpinning of Neural Networks

Mathematical Underpinning of Neural Networks

- Layers of nodes (perceptrons). k “hidden” layers sandwiched by an input layer and an output layer.

Mathematical Underpinning of Neural Networks

- Layers of nodes (perceptrons). k "hidden" layers sandwiched by an input layer and an output layer.
- Each node in layer j connects to each node in layer $j + 1$. Connection is defined by a weight and a bias.

Mathematical Underpinning of Neural Networks

- Layers of nodes (perceptrons). k "hidden" layers sandwiched by an input layer and an output layer.
- Each node in layer j connects to each node in layer $j+1$. Connection is defined by a weight and a bias.
- Let the layer k_i have m nodes, and the layer $k_i + 1$ have n nodes. Then the weights between the layers are given by the matrix

Mathematical Underpinning of Neural Networks

- Layers of nodes (perceptrons). k “hidden” layers sandwiched by an input layer and an output layer.
- Each node in layer j connects to each node in layer $j+1$. Connection is defined by a weight and a bias.
- Let the layer k_i have m nodes, and the layer k_i+1 have n nodes. Then the weights between the layers are given by the matrix

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{bmatrix} \tag{1}$$

Mathematical Underpinning of Neural Networks

- Layers of nodes (perceptrons). k "hidden" layers sandwiched by an input layer and an output layer.
- Each node in layer j connects to each node in layer $j+1$. Connection is defined by a weight and a bias.
- Let the layer k_i have m nodes, and the layer $k_i + 1$ have n nodes. Then the weights between the layers are given by the matrix

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{bmatrix} \quad (1)$$

- The values of the nodes in layer $k+1$, denoted $A_{(k+1)}$ is given by the matrix equation $A_{(k+1)} = \sigma(WA_{(k)} + b_{(k)})$. Where b is the vector containing the biases, and $\sigma()$ is the *activation function*.

Backpropogation

Backpropogation

- But how does this network learn? The answer: backpropogation!

Backpropogation

- But how does this network learn? The answer: backpropogation!
- Define an error function $E(X, \theta) = \frac{1}{2N} \sum_{i=1}^{N} (y_i' - y_i)^2$. Where θ incorperates the weights and biases.

Backpropogation

- But how does this network learn? The answer: backpropogation!
- Define an error function $E(X, \theta) = \frac{1}{2N} \sum_{i=1}^{N} (y_i' - y_i)^2$. Where θ incorperates the weights and biases.
- The idea behind backpropogation is to find a local minimum of this function. We then update the weights and biases according to the differential of $E(X, \theta)$ with respect to each weight. For the k^{th} layer, update the weight between node i in k to j in $k+1$, we have:

Backpropogation

- But how does this network learn? The answer: backpropogation!
- Define an error function $E(X, \theta) = \frac{1}{2N} \sum_{i=1}^{N} (y_i' - y_i)^2$. Where θ incorperates the weights and biases.
- The idea behind backpropogation is to find a local minimum of this function. We then update the weights and biases according to the differential of $E(X, \theta)$ with respect to each weight. For the k^{th} layer, update the weight between node i in k to j in $k+1$, we have:

$$\Delta w_{ij}^{k} = -\alpha \frac{\partial E(X, \theta)}{\partial w_{ij}^{k}}. \tag{2}$$

Where α is called the *learning rate*.

Why use Neural Networks for Stock Trading?

- Improvements on historical trading methods. Primarily using a companies "fundamentals".

Why use Neural Networks for Stock Trading?

- Improvements on historical trading methods. Primarily using a companies “fundamentals”.
- Too many of these to identify which ones are important/irerlevant.

Why use Neural Networks for Stock Trading?

- Improvements on historical trading methods. Primarily using a companies "fundamentals".
- Too many of these to identify which ones are important/irerlevant.
- Financial Data is very dense. It is too complex for a human analyst, so NNs can be used to comb through immense amount of data and identify patterns useful for trading.

Why use Neural Networks for Stock Trading?

- Improvements on historical trading methods. Primarily using a companies “fundamentals”.
- Too many of these to identify which ones are important/irerlevant.
- Financial Data is very dense. It is too complex for a human analyst, so NNs can be used to comb through immense amount of data and identify patterns useful for trading.
- Key issue: The networks can’t explain *why* they make the decisions they do. This can make it hard for financial engineers to work out what’s gone wrong if something does.

How do we use Neural Networks for Stock Trading

- Two main ways: “signalling” and “prediction”.

How do we use Neural Networks for Stock Trading

- Two main ways: “signalling” and “prediction”.
- 

Figure: Signalling technique: Black = Buy, Red= Sell

How do we use Neural Networks for Stock Trading

- Two main ways: “signalling” and “prediction”.

Figure: Signalling technique: Black = Buy, Red= Sell

- Can maximise the amount of profit that can be made from one stock rather than buy and hold.

How do we use Neural Networks for Stock Trading

- Two main ways: "signalling" and "prediction".
- 

Figure: Signalling technique: Black = Buy, Red= Sell

- Can maximise the amount of profit that can be made from one stock rather than buy and hold.
- The second method uses NNs to predict a value of the stock at a future time.

How do we use Neural Networks for Stock Trading

- Two main ways: “signalling” and “prediction”.
- 

Figure: Signalling technique: Black = Buy, Red= Sell

- Can maximise the amount of profit that can be made from one stock rather than buy and hold.
- The second method uses NNs to predict a value of the stock at a future time.
- This allows investors to make an informed choice on which stocks to put money in.

Improvements to Automated Trading

- **Chosing which stocks to use these automated methods on**: Genetic Algorithms for selecting a portfolio.

[1] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," Expert systems with Applications, vol. 19, no. 2, pp. 125–132, 2000.

Improvements to Automated Trading

- **Chosing which stocks to use these automated methods on**: Genetic Algorithms for selecting a portfolio.
- **Data**: Every day, more data is added to the pool of training data that these models are trained on.

[1] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," Expert systems with Applications, vol. 19, no. 2, pp. 125–132, 2000.

Improvements to Automated Trading

- **Chosing which stocks to use these automated methods on**: Genetic Algorithms for selecting a portfolio.
- **Data**: Every day, more data is added to the pool of training data that these models are trained on.
- **Improving the networks themselves**: Use of genetic algorithms and improved training algorithms for finding an optimal set of weights. Roughly 10% improvement. (Kim et. al, 2000) [1]

[1] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," Expert systems with Applications, vol. 19, no. 2, pp. 125–132, 2000.

Improvements to Automated Trading

- **Chosing which stocks to use these automated methods on**: Genetic Algorithms for selecting a portfolio.
- **Data**: Every day, more data is added to the pool of training data that these models are trained on.
- **Improving the networks themselves**: Use of genetic algorithms and improved training algorithms for finding an optimal set of weights. Roughly 10% improvement. (Kim et. al, 2000) [1]
- **Reducing Dimensionallity**: Financial data is very complex. Reducing dimensionality to focus on the most important features of the data- reducing training time and memory used by the system.

[1] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," Expert systems with Applications, vol. 19, no. 2, pp. 125–132, 2000.

Conclusion

- These methods are being employed by investment firms and banks daily to turn a profit from the world's stock exchanges.

Conclusion

- These methods are being employed by investment firms and banks daily to turn a profit from the world's stock exchanges.
- Has made a vast improvement on the old fundamental methods that traders used.

Conclusion

- These methods are being employed by investment firms and banks daily to turn a profit from the world's stock exchanges.
- Has made a vast improvement on the old fundamental methods that traders used.
- Provides a wonderful testbed for pattern recognition techniques due to wealth of data.

Conclusion

- These methods are being employed by investment firms and banks daily to turn a profit from the world's stock exchanges.
- Has made a vast improvement on the old fundamental methods that traders used.
- Provides a wonderful testbed for pattern recognition techniques due to wealth of data.
- Tehcniques used here can be applied to other areas related to time-series data. Such as looking at how populations of animals change over time.