University of Auckland
Department of Accounting & Finance

Neural Networks: A New Frontier for Asset Pricing

A research essay presented in part fulfilment of the requirements for the degree of Bachelor of Commerce (Honours) in the Department of Accounting and Finance at The University of Auckland

Author: Connor McDowall
Supervisor: Dr Paul Geertsema

December 31, 2021

Contents

List of Figures

List of Tables

List of Equations

1 Acknowledgements

Paul Geertsema

2 Abstract

3 Introduction

4 Research Question(s)

5 Motivation(s)

6 Literature Review

Overview of literature in asset pricing (761/751), ML application, factor pricing - very brief, 12pt, double spaced,

6.1 Asset Pricing

6.2 Machine Learning

Convexity is an important concept in optimisation Monotonic ranking

6.3 Machine Learning

A couple of recent publications highlight the increased application of machine learning algorithms in financial contexts. **corporate-culture** Gu et al (**eapvml**) explore the comparative use of machine learning in empirical asset pricing.

6.4 Data

7 Theory

7.1 Modelling, Loss, and Optimisation

We summarize the theory surrounding predictive modelling, loss functions, and optimisation algorithms. These functions train models by comparing predictions to realized observations using optimisation algorithms to minimize the loss function. We examine a linear model as our predictive model (**??**). Mean square error (**??**) and Gradient Descent

(GD) are basic examples of a loss function and optimisation algorithm, respectively.

$$\hat{y} = mx_i + b \tag{1}$$

$$f(y, (mx_i + b)) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))^2 \tag{2}$$

Firstly, gradient descent takes the partial derivatives of the loss function, with the respect to the parameters in our predictive model. In our example, equations **??** and **??** are the partial derivatives for the mean square error loss function.

$$\frac{\partial f(y, (mx_i + b)}{\partial m} = \frac{1}{n}\sum_{i=1}^{n} -2x_i(y_i - (mx_i + b))^2 \tag{3}$$

$$\frac{\partial f(y, (mx_i + b)}{\partial b} = \frac{1}{n}\sum_{i=1}^{n} -2(y_i - (mx_i + b))^2 \tag{4}$$

Secondly, the algorithm explores epochs, using a learning rate to update parameters to move in the opposite directions of the partial derivatives until settling in a local minima. This extrema is the optimisation of the loss function, quantifying the accuracy of the predicative model. Ordinary Least Squares (OLS) regressions is an extension of the linear model prevalent in asset pricing.

7.2 Ordinary Least Squares (OLS)

The OLS regression is the most prominent statistical model in asset pricing theory. Rosenfeld (**olsmf**) summarises OLS. The composition of the true OLS (**??**) model includes four components. Firstly, $\mathbf{X}$, an n x k matrix of k independent variables for n observations. Secondly, $\mathbf{y}$, an n x 1 vector of observation on the dependent variable. Thirdly, ϵ, an n x 1 vector of unexplained error. Lastly, θ, a k x 1 vector of parameters to be estimated.

$$y = X\theta + \epsilon \tag{5}$$

7.2.1 Estimation Criteria

The criteria to obtain the parameter estimate ($\hat{\theta}$) relies on the minimisation of the sum of squared residuals (**??**). We highlight the observed residuals (e) are distinct from unexplained disturbances (ϵ). Equation **??** derives residuals by taking the difference between observations based on parameter estimates.

$$\sum e_i^2 \tag{6}$$

$$e = y - X\hat{\theta} \tag{7}$$

Expanding the quadratic $e^T e$ after substituting in equation **??** leads to the alternative expression of the sum of squared residuals in equation **??**. Minimizing the sum of square residuals requires taking the partial derivative of equation **??** with respect to the estimated parameters (equation) using matrix differentiation (**??**). It is imperative X has full rank where all vectors in the matrix are linearly independent, validating both the presence of a positive definite matrix and minimum.

$$e^T e = y^T y - 2\hat{\theta}^T X^T y + \hat{\theta}^T X^T \hat{\theta} X \tag{8}$$

$$\frac{\partial e^T e}{\partial \hat{\theta}} = -2X^T y + 2X^T X\hat{\theta} = 0 \tag{9}$$

We find the expression for the Ordinary Least Squares (OLS) estimator (**??**) after rearranging equation **??** to normal form, utilizing inverse matrices to form identity matrices, and simplifying.

$$\begin{aligned} 2X^T X\hat{\theta} &= 2X^T y \\ (X^T X)^{-1}(X^T X)\hat{\theta} &= (X^T X)^{-1} X^T y \\ I\hat{\theta} &= (X^T X)^{-1} X^T y \\ \hat{\theta} &= (X^T X)^{-1}(X^T y) \end{aligned} \tag{10}$$

Therefore, we can use the OLS estimator to make predictions with OLS (**??**).

$$\hat{y} = X^T \hat{\theta} \tag{11}$$

7.2.2 Properties of OLS Estimators

There are six key properties in addition to the satisfaction in minimizing the summation of squared residuals.

1. The residuals are uncorrelated with the observed values of X i.e., $X^T e = 0$.
2. The sum of the residuals is zero i.e., $\sum e_i = 0$.
3. The sample mean of the residuals is zero i.e., $\bar{e} = \frac{\sum e_i}{n} = 0$.
4. The regression hyperplane passes through the means of observed values i.e., $\frac{e}{=} \frac{y - X\theta}{n} = 0$. Since $\bar{e} = 0$ assumed, it is implied $\bar{y} = \bar{x}\bar{\theta}$.
5. The residuals are uncorrelated with the predicted y i.e., $\hat{y} = X\hat{\theta}$, $\hat{y}^T e = (X\hat{\beta})^T e = b^T X^T e = 0$
6. The mean of $\hat{y}$ for the sample will equal the mean of the y.

7.2.3 The Gauss-Markov Theorem

However, OLS makes Gauss-Markov assumptions about the true model to make inferences regarding β from $\hat{\beta}$. The intention of the Gauss-Markov Theorem, conditional on the below assumptions, states the OLS estimator is the best linear, unbiased, and efficient

estimator:

$$y = x\beta + \epsilon$$

$$E[\epsilon|X] = 0 \tag{12}$$

$$E(\epsilon\epsilon^T|X) = \Omega = \sigma^2 I \tag{13}$$

$$\epsilon|X \ N[0, \sigma^T I] \text{ (hypothesis testing)}$$

- X is an n x k matrix of full rank
- X must be generated randomly, or fixed, by a mechanism uncorrelated to disturbances.

Equation **??** implies $E(y) = X\beta$ as no observations of the independent variables convey any information about the expected values of the disturbances. Equation **??** captures homoskedasticity and no autocorrelation assumptions. Additionally, The theory underlying Ordinary Least Squares informs the common practice in minimising of the sum of least squares when evaluating prediction performance. The mathematical tractability, in accordance with the aforementioned assumption, frame our thinking surrounding the derivation of custom loss functions.

7.3 Artificial Neural Networks (ANN)

7.3.1 Multi Layer Perceptron (MLP)

Artificial Neural Nets (ANN) are versatile, powerful, and scalable. They sit at the heart of deep learning as frequently outperform other machine learning algorithms on large and complex problems. This research proposal suggests implementing two sets of multi-layer perceptron, a form of ANN, to predict investment decisions and exit opportunities from investment criteria. A linear threshold unit (LTU) feeds the weighted sum of input values ($z = \mathbf{w}^T \cdot \mathbf{x}$) into a step function ($h_w(\mathbf{x}) = step(z)$). A perceptron is a single layer of LTUs where each LTU is connected to every input. Perceptrons are suitable for classification as output the positive investment decision or exit opportunity if a threshold is

met. Perceptrons utilize a training algorithm assessing the strength of connections between perceptrons while considering errors. A perceptron is fed one training instance at a time, making predictions for each instance. For every output LTU that produced a wrong prediction, it re-enforces the connection weights using the perception learning rule (**??**) from the inputs that would have contributed to the right prediction. A Multi Layer Perceptron is composed of one LTU input layer, multiple LTU hidden layers and an output LTU layer. The step functions in each LTU are replaced by a logistic or ReLU function ($\sigma(z) = \frac{1}{1+exp(-z)}$ or $ReLU(z) = max(0, z)$ respectively) to enable gradient descent for optimisation. A shared softmax function replaces the individual activation functions in the output layer to enable exclusive classification. In this instance, the classification of investment decisions, or exit opportunties, from investment criteria. Tensorflow's DNNClassifier function facilitates the implementation of MLPs in this proposal.

$$w_{i,j}^{\text{next step}} = w_{i,j} + \eta(\hat{y}_j - y_j)x_i \tag{14}$$

Where

- $w_{i,j}$ is the connection weights between the ith input neuron and the jth output neuron.
- x_i is the ith input value of the current training instance.
- $\hat{y}_j$ is the output of the jth output neuron for the current training instance.
- y_j is the output of the jth output neuron for the current training instance.
- η is the rate.

7.4 Hypothesis

Include examples on the minimisation of sum of the square errors does not contribute to maximising returns

8 Model

Insert model configuration Example of the whole set

8.1 Configuration

9 Methodology

9.1 Data

Hou et al., (**hou2020replicating**) use an extensive data library to assess 452 anomalies across anomalies literature. Their analysis informs which abnormalities drive the cross section of expected returns. Most abnormalities fail under current standards of empirical finance when using a single hurdle test of absolute t-stat greater or equal to 1.96. Firstly, the paper finds economic fundamentals take precedence over trading frictions in explanatory power, statistical and economic significance. Secondly, micro-caps account for anomalies disproportionately, leading to NYSE breakpoints, value-weighted returns in both portfolio sorts and cross-sectional regressions with weighted least squares. Lastly, arguments in improving anomalies literature credibility follow a closer alignment to economic theory as the field persists to be statistical in nature. Overall, capital market efficiency is higher than expected. Jensen et al., **jensen2021there** use the above dataset to explore hierarchial bayesian models of alphas emphasising the joint behaviours of factors, and provide an alternative multiple testing adjustment, more powerful than common methods. Jensen et al., adapt the global dataset to focus only on one-month holding periods for all factors, only include most recent accounting data (quarterly or annually) and

add 15 new factors. The exhaustive nature and accessibility of the global dataset makes it well-suited for exploring optimisation functions in neural-network construction.

9.2 Limitations

9.3 Summary Statistics

10 Methodology

10.1 Target Variable

10.2 Google Cloud Platform

10.2.1 Configuration

10.2.2 Limitations

10.3 Tensorflow

10.3.1 Automatic Differientiation

10.4 Loss Functions & Performance Metrics

Table **??** emphasises the separation between training and validation datasets.

Variable	Description	$MSE(y, \hat{y})$	$HP(y, \hat{y})$
θ	Estimation Training	$\hat{\theta}_{MSE}$	$\hat{\theta}_{HP}$
λ	Validation	$\hat{\lambda}_{MSE}$	$\hat{\lambda}_{HP}$

Table 1: Objective (MSE: Mean Square Error, HP: Hedge Portfolio)

10.4.1 Mean Square Error (MSE)

Section **??** outlines advantages to Ordinary Least Squares. Subsequently, MSE serves as a baseline for loss function and performance metric comparisons. The following function (**??**) and partial derivative (**??**) describe Tensorflows's Mean Square Error implementation, both from in-built and custom contexts. Python classes describe equation **??** to enable Tensorflow's automatic differentiation capabilities, approximating the partial derivatives of the loss function (**??**) with numerical methods. Please note the use of Hadamard exponentiation ($x^{\circ n}$) as an element-wise operation.

$$f(y, X^T\hat{\theta}) = \frac{\vec{1}}{\vec{1}^T\vec{1}}(y - X^T\hat{\theta})^{\circ 2} \tag{15}$$

$$\frac{\partial f(y, X^T\hat{\theta})}{\partial \hat{\theta}} = \frac{\vec{1}}{\vec{1}^T\vec{1}}(-2(y - XT\hat{\theta})^{\circ 1}) \tag{16}$$

10.4.2 Hedge Portfolio

Hedge portfolios rely on monotonic ranking functions for optimisation as their monotonic nature preserves or reverses a given ordered set. The analysis cross-section of one-month lead portfolio excess returns using monotonic functions

$$R(y_{i,t}) \tag{17}$$

The ranking function (R($y_{i,t}$)) and thresholds (u,v) form subsets of long and short portfolios. Long (L) or Short (S) sets include excess returns conditioned on the associated monotonic ranking given a threshold, bound by the cardinality of the excess return vector ($|y|$). The subsequent truth sets mathematically express aforementioned time-series

hedge portfolios.

$$L = \{y_{i,t} | R(y_{i,t}) \leq u\}$$

$$S = \{y_{i,t} | R(y_{i,t}) \geq v\}$$

$$0 < u \leq |y|$$

$$0 < v \leq |y|$$

$$u < v$$

Equation **??** describes hedge portfolio lead excess returns (H_t) at a given time (t).

$$H_t = \frac{1}{|L|} \sum_{i \epsilon L} y_{i,t} - \frac{1}{|S|} \sum_{i \epsilon S} y_{i,t} \tag{18}$$

Figure **??**) illustrates an approximate linear monotonic ranking function with a sample of 100 uniformly distributed excess returns between -10% and 10%. Boundary conditions u and v are set to 20 and 80 , respectively. Subsequently, excess returns above (below) the green (blue) dotted line belong to the long (L) (short (S)) set.

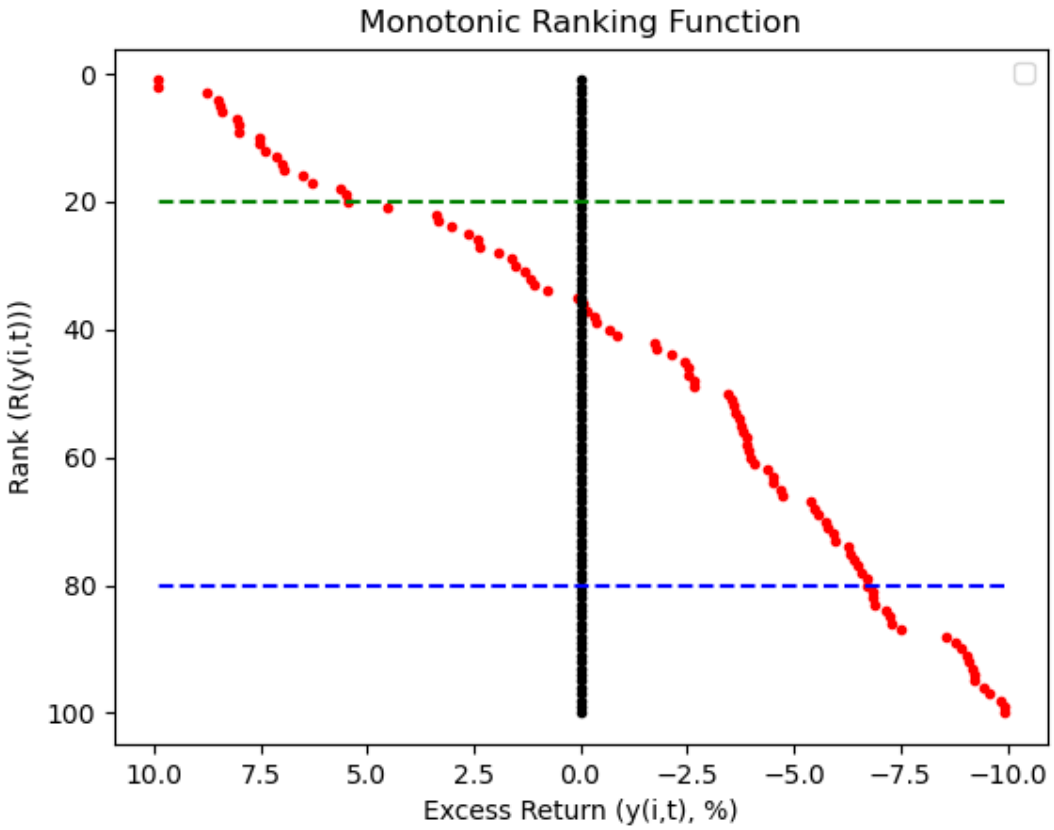

Figure 1: Approximate Linear Monotonic Ranking Function

The permutations in monotonic ranking functions, and subsequent hedge portfolios, are endless. This research essay develops a monotonic ranking function proportionally weighting one month lead excess returns (**??**). Therefore, equation **??** defines the loss function.

$$R(\hat{y}) = W \tag{19}$$

$$W := \frac{\hat{y}}{\vec{\mathbf{1}}\hat{y}}$$

$$\hat{y} = X^T\hat{\theta}$$

$$f_{\hat{\theta}}(X) = (\frac{X^T\hat{\theta}}{\vec{\mathbf{1}}X^T\hat{\theta}})^\top X^T\hat{\theta} \tag{20}$$

The above loss function is differentiable using symbolic mathematic as shown in equation

??.

$$\frac{\partial f_{\hat{\theta}}(X)}{\partial \hat{\theta}} = \frac{\partial((\frac{X^T\hat{\theta}}{\vec{1}X^T\hat{\theta}})^\top X^T\hat{\theta})}{\partial \hat{\theta}}$$
$$\frac{\partial(f_{\hat{\theta}}(X))}{\partial \hat{\theta}} = \frac{1}{(\hat{\theta}^\top X\vec{1})}XX^\top\hat{\theta} + \frac{1}{\vec{1}X^\top\hat{\theta}}XX^\top\hat{\theta} - \frac{1}{(\hat{\theta}^\top X\vec{1})^2}\hat{\theta}^\top XX^\top\hat{\theta}X\vec{1} \tag{21}$$

Our research Subsection **??** explains the theory supporting loss minimisation. Applying gradient descent methods to the product of the loss function and scaler of -1 transforms the minimisation to maximisation. This transformation leads to finding the argmax of maximisation function with respect to $\hat{\theta}$ (**??**). The aforementioned transformation is simply and suitable for exploration in the context of the research intent. More sophisticated methods exist for maximisation such as reinforcement learning (**??**).

$$\underset{\hat{\theta}}{\text{argmax}} : (\frac{X^T\hat{\theta}}{\vec{1}X^T\hat{\theta}})^\top X^T\hat{\theta} \tag{22}$$

Conventional asset pricing methodologies persist in academic literature. Subsequently, this research essay uses the Hedge Portfolio Mean (**??**)), Capital Asset Pricing Model (**??**), Fama-French Three Factor Model (**??**), and Fama-French Five Factor Model (**??**) as performance metrics for each loss function.

$$H_\mu = 1 \tag{23}$$
$$H_{CAPM} = 1 \tag{24}$$
$$H_{FF3} = 1 \tag{25}$$
$$H_{FF5} = 1 \tag{26}$$

The use of the Capital Asset Pricing Model (CAPM) persists, regardless of the identifiable shortcomings in market proxies and empirical failings invalidating use (**fama2004capital**). Nonetheless, this research essay uses the model as a performance metric for comparative

purposes.

$$R_{i,t} - R_{f,t} = /alpha_{i,t} +$$

E. Fama and K. French (**eugene1992cross**) validate the explanatory power of size and value (book-to-market) factors in their ability to capture the cross-sectional variation in average stock returns, in association with market risk, size, leverage, book-to-market, and earnings-price ratios. This combination of factors is known as the Fama-French Three Factor Model (FF3). E. Fama and J. MacBeth developed a two-step portfolio modelling approach

Fama-French Five Factor Model (FF5) continues to inform asset pricing E. Fama & K. French produce **fama2004capital**

10.4.3 Sharpe Ratio

Nobel Laurette William F. Sharpe (**sharpe1994sharpe**) introduced the Sharpe Ratio (**??**) as a measure for risk-adjusted returns where $\mathbb{E}[R_a - R_f]$ is the expectation for excess returns and $\sigma(R_a)$ is the standard deviation of excess returns.

$$SR = \mathbb{E}\,\frac{[R_a - R_f]}{\sigma(R_a)} \tag{27}$$
$$\sigma = \sqrt{\sum_{i=1}^{n} \frac{(R_a) - \bar{R_a})^2}{n}}$$

The matrix notation for the Sharpe Ratio loss function () follows:

$$SR = 1 \partial SR \tag{28}$$

10.4.4 Information Ratio

The Information Ratio (IR) is an extension

10.5 Reinforcement learning

10.5.1 Dynamic Programming

10.5.2 Bellman's Algorithm

10.5.3 Q-Learning

11 Results

12 Discussion

13 Contributions

14 Conclusion

15 Appendix

15.1 Tables and Charts

15.2 Technical Details

15.2.1 Organisation

This research essay uses data science best practise (**J:10**). Data and results saved regularly and reproducible. Data retention in all forms receives high levels of attention. Project files synchonises continuously to Google Drive (**Google˙Drive**). Git (**Git**) manages version control protocols for source code, data, documents, and results. Git stores a complete history of versions using Git hashes. These hashes are strings unique to each state of the publicly available finance-honours repository[1]. Git hashes enable discretisation of finance-honours development, enabling the accessibility and recollection of all previous states given a unique git hash. This functionality enables reproducibility, error correction, and the ability to revert to previous models.

15.2.2 Version Control

Git, hosted by GitHub, provides a comprehensive set of version control technologies and range of benefits. Firstly, Git enables collaborative functionalities. The master version of a project is accessible for all who have access to the repository. Each contributor can create custom copies of branches through pull requests on the master branch. Contributors can commit changes to custom branches and push these changes to the master branch through push requests. Product managers can review push requests, approving valid requests for integrating changes to the master branch. Collaborative efforts are possible with commit messages describing contributions from each contributor. This research essay has only one contributor, rendering collaborative functionalities redundant in this instance. Git ensures the storage of code, work, and author histories. The descriptive nature of commit logs ensures journal accuracy.

15.2.3 Directories

This research essay follows directory structure recommendations from Wilson et al (**J:10**). Organisation is crucial as the modelling of artificial neural networks involves integrating

[1] https://github.com/CMCD1996/finance-honours

a range of optimisation models, data files and documents. Directory management is most efficient and comprehensive. **finance-honours** is the root directory containing the following sub directories: bin, data, doc, src, and results. The **bin** sub directory contains external scripts and compiled programmes. The **data** sub directory contains all raw data associated with the project. The **doc** sub directory stores user guides, academic resources, research reports and project deliverables. The **results** sub directory contains the outputs from project analysis. The **src** sub directory stores the source code for preparing datasets, partitioning sets of geographies with varying granularities. All files were continuously backed up using Google Drive and Git.

15.2.4 Python

Python 3.9.7 is the primary programming language for this research essay. The language is omnipresent, widespread in software development. Python's language design makes the language highly productive and simple to use. Python can hand off computationally straining tasks to C/C++ using supporting first-class integration capabilities. The language also has a very active and supportive community. Python is the most popular coding language on the planet defined by the PYPL PopularitY of Programming Language Index. As at December 2021, Python has 30.21% of all language tutorial search instances on Google (**PYPL˙Pop**). Python's dynamic, low cost, and open source nature makes programming quick.

15.2.5 Package Management

The Anaconda package management platform for Python (**Anaconda**) is the chosen coding environment. Anaconda is a well defined, free platform, with known versions of python packages such as matplotlib, numpy, and pip. The use of this environment ensures reproducibility and consistency across infrastructure. Pip is the default package manager for Python, included in the Anaconda package. Pip manages package installation and updates.

15.2.6 Code Style

The PEP8 style for Python Code is formatting style for development code **PEP8**. Yapf, a formatter maintained by Google, manages formatting. Standardised formatting is important as makes supports readability, optimisation, and consistency. Docstrings and rigourous commenting are important in documentation. A docstring is a Python inline comment describing function use, inputs, and outputs. An unique docstring belongs to each Python class and function. The Google style docstring is most appropriate because of it's readability, writing ease, and consistency with Google's Style Guide. The parsing of yapf docstrings enables automated documentation generators to create docstring documents describing functions and classes.

15.2.7 Infrastructure

This research essay deploys variations in artificial neural networks of changing size and complexity. Analysis either took place locally, or remotely, depending on the computational requirements for the particular analysis. An Apple MacBook Pro 13 Inch 2019 with 8 GB 2133 MHz LPDDR3 memory and 1.4 GHz Quad-Core Intel Core i5 processor handles simple tasks locally. A Virtual Machine Instance on the Google Cloud Platform **Insert specification before submission** handles complex tasks remotely.

15.2.8 Documentation

The research essay documentation keeps an accurate record of key design decisions. Commit histories (**??**) is the most important form of documentation. Application of auxiliary documentation methods are supplementary.

15.3 Code

All files, resources, and code is available for download from Github. The document listing function and class docstring is available for download here. Furthermore, the coding listings for this research essay follow.