Multi-Task Learning for Stock Selection- Critical Analysis

Kunj Golwala
golwala.1@iitj.ac.in

Indian Institute of Technology Jodhpur,
Jodhpur,
Rajasthan,India

Critical Analysis

Multi Task Learning (MTL) is a training model which uses multiple neural networks simultaneously.The main aim of MTL is to obtain useful information in multiple related tasks which helps in improving the generalization performance of all the tasks. The idea explored in this paper is to apply Multi Task learning for the purpose of stock selection so as to yield better returns than the already existing models and benchmarks.

Parameter Sharing and training algorithm - The data is given to the learner in the form of (x,y) pairs where x is the input and y is the desired value. A training criterion C is defined, which is a function of the inputs x and outputs of the learner. This criterion is to be minimized. The function f is parameterized by the parameters of the ANNs and belongs to a set of Hypotheses H. The generalization error due to variance can be controlled by choosing a small hypothesis space H. In this model, we learn all the tasks parallely, to improve generalization. A number of neural networks share some or all of their hidden layers/parameters. The hidden layers serve as sources of inductive bias.

Application of MTL to stock selection - MTL is applied for the task of stock selection and portfolio management. For our experiment, we considered 35 risky assets(35 large capitalization companies of TSE) and 1 risk-free asset(90-days Canadian treasury bills). Each month one can buy or sell some of these assets so as to redistribute the worth. Here, 5 variables - 2 macroeconomic variables and 3 microeconomic variables. Monthly data is recorded for a span of 8 years. Multiple training experiments were performed on different training windows and different initial weights to ensure that the results obtained were not by chance. The training algorithm for these experiments is based on the optimization of parameters with respect to a financial criterion and hence maximizing profits. Outputs of the networks feed a trading module. The outputs of the neural networks and the weights of the assets act as inputs for the trading module. The weights in turn depend on the previous portfolio weights and relative change in value of each asset. The outputs of the trading module give the new sets of weights for the portfolios. Based on the difference in the old set of weights and the new set of weights, transactions are done.To obtain gradients of this criterion, we have to backpropagate gradients backwards, and we obtain a differentiable function of the inputs.This procedure is found to yield larger profits with risks comparable to training the neural network by minimizing the mean square error.

Results - Four sets of experiments with different types of parameter sharing were performed on two different architectures for the neural network, 5-3-1 (5 inputs, a hidden layer of 3 units, and 1 output), 5-3-2-1 (5 inputs, 3 units in the first hidden layer, 2 units in the second hidden layer, and 1 output). Four types of parameter sharing were:sharing everything, sharing only the parameters of the first hidden layers, sharing only the output layer parameters, and not sharing anything.

Conclusions - From the data obtained we can conclude the following - 1. Sharing some parameters yields more consistent results across architectures. 2. Improvement is observed in the results when some parameters are not shared. 3. The performance obtained in this way yields much better yearly returns when compared to benchmarks.

Pros - The biggest advantage of the method used in this paper is that MTL helps in improving generalization performance with the help of sharing of parameters. Since the parameters and hidden layers are shared between neural networks, the total number of unique parameters required for training the model also decreases. This in turn helps in reducing overfitting of the model on the training data. Since the networks are working parallely, the learning speed is faster by leveraging auxiliary information. The experiment is carried out for a variety of different initial weights and different training windows to ensure that the results obtained are not by chance. Also, a timespan of 8 years is a sufficiently long time period to validate the results of the model.

Cons - The models need to be tested on different data sets and companies from different stock exchanges across the world for it to be labeled as reliable for the purpose of stock selection and increasing profits.In order to have an unbiased learner, the hypothesis space H must contain all possible functions which can be obtained by optimization, but to improve generalization performance, we make several assumptions and make the hypothesis space H smaller. Therefore, in this case the learner is not truly unbiased. Imposing too many assumptions may also lead to an incorrect model and undesired performance.

Technical error - In our experiment, we do not allow borrowing or short-selling. This means that the weights of different assets are all non-negative and the sum of these weights will always be equal to 1.This is done to reduce the complexity of the model but in real life, wholesale traders commonly use short selling and borrow/lend shares to traders. These features should be accommodated in our model so that the model can help in predicting the value of stocks in real life more accurately.

Technical Suggestions - The data that is fed into the network as inputs,should be preprocessed.Data should be scaled/normalized and outliers should be dealt with appropriately. The results of the network should also be cross-validated to check for overfitting. To deal with the short-selling and borrowing problem mentioned above, we can select a trading module which can return negative weights as well.

Future Work - The results obtained in this paper can be extended in many ways. Firstly, the above experiments should be performed on different stocks from different stock exchanges. Secondly, we can look to improve the generalization by allowing for more freedom in the way different tasks influence each other.