Incorporating Prior Knowledge About Financial Markets Through Neural Multitask Learning

Kai Bartlmae[2], Steffen Gutjahr[1], Gholamreza Nakhaeizadeh[2]

[1] *University of Karlsruhe*
Institute of Logic, Complexity and Deduction Systems
Am Fasanengarten 5, 76128 Karlsruhe, Germany
gutjahr@ira.uka.de

[2] *Daimler-Benz AG, Research and Technology FT3/KL*
P.O. Box 23 60, D-89013 Ulm, Germany
{nakhaeizadeh, bartlmae}@dbag.ulm.DaimlerBenz.com

Abstract. We present the systematic method of Multitask Learning for incorporating prior knowledge (hints) into the inductive learning system of neural networks. Multitask Learning is an inductive transfer method which uses domain information about *related tasks* as inductive bias to guide the learning process towards better solutions of the main problem. These tasks are presented to the learning system in a shared representation. This paper argues that there exist many opportunities for Multitask Learning especially in the world of financial modeling: It has been shown, that many interdependencies exist between international financial markets, different market sectors and financial products. Models with an isolated view on a single market or a single product therefore ignore this important source of information. An empirical example of Multitask Learning is presented where learning additional tasks improves the forecasting accuracy of a neural network used to forecast the changes of five major German stocks.

1 Introduction

It has been observed, that the results of inductive methods can be improved by guiding the learning process through auxiliary information about the problem. Abu-Mostafa called this information *hints* about the target function [Abu-Mostafa 1995]. Towell et al. [Towell, Shavlik 1994] showed that incorporating information about the underlying regularities of the domain into a neural network can improve its generalization performance. Especially in the case that only few input-output examples of a complex and non-linear problem are given in a noisy domain, the use of additional information in form of *knowledge-based domain-specific* hints is essential for successful learning.

This paper presents the method of Multitask Learning which incorporates dependencies between tasks into an inductive learning system. A Multitask Learning System uses the input-output examples of related tasks in order to improve the generalization performance of at least one of these tasks. Learning related problems *at the same time* provides an inductive bias towards better representations of the domain's underlying regularities.

Multitask transfer has broad utility especially in the field of financial modeling and forecasting. Here highly complex dependencies between the in-going and out-going variables of models are assumed. Characteristics of financial data are further extreme noisiness and non-stationarity. Because of the non-stationarity past market behavior may not longer hold in the future. Exploitable information is also reduced

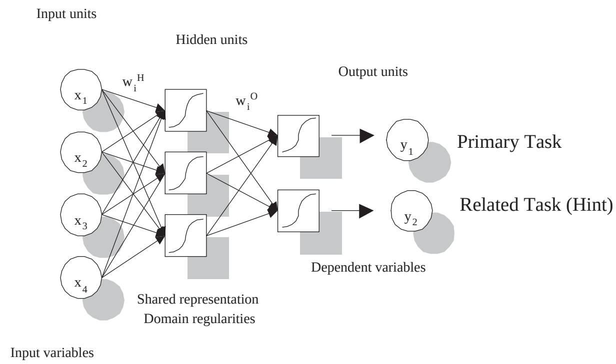

Fig. 1. A Multitask three layer backpropagation network with two related tasks.

by the noisiness of the data.

Rather than giving direct hints about the target function like Abu-Mostafa introduced in [Abu-Mostafa 1995] we extend this idea by showing that giving indirect hints about the underlying regularities of the problem through related tasks improves the generalization performance in a financial forecasting application. These hints can be derived from a large number of theoretically supported dependencies like the integration of global financial markets, market sectors and financial products or from the observation of the correlation between markets and products.

We first introduce Multitask Learning and its realization with neural networks. Then we present an MTL application for five German stocks. Finally we summarize the methodology.

2 Multitask Learning and Hints in Neural Networks

Multitask Learning has a simple realization if used with three layer feedforward neural networks [Caruana 1996]. In the case of a three layer neural network one adds for each additional task a further output unit, which is fully connected to the hidden layer (figure1).

In this configuration a neural MTL network can be thought of as a simultaneous model for different dependent variables. But the opportunities for Multitask Learning go beyond a simple simultaneous estimation of models. MTL offers the opportunity to add additional tasks only in order to improve and stabilize the performance of the main task. The outputs of the related tasks are ignored when the net is used to make predictions. Therefore it is not the major goal to find a very good model over all tasks but to find an excellent model for the main task. This order of importance over the different tasks has to be reflected in the learning algorithm, i.e. in a weighting of the different errors of each problem.

The basic idea underlying neural Multitask Learning is that the related tasks influence the hidden layer weights through the additional gradient information. This is leading to hidden layer representations, which better reflect the domain's regularities: The gradient of a hidden layer weight w_i^H is just a weighted sum of the gradient information of each single task.

Therefore the gradient points into a direction that decreases the error over all tasks. The weight itself represents after the update step a better dependency between the input variable and *all* output variables. This reflects the property of a shared domain's regularity, if a correct additional hint is given.

An improvement of generalization performance has been observed in other domain's applications [Caruana, Baluja, Mitchell 1996]. Different underlying mechanisms have been discussed in [Abu-Mostafa 1995] [Caruana 1995].

3 Application

An interesting field for Multitask Learning and the use of hints is the domain of financial forecasting. Here an application of predicting the direction of five German stocks in a one week horizon was chosen: Allianz, Daimler Benz, Deutsche Bank, Siemens and Veba. Because of its noisy and complex nature this is a well-suited problem for the use of Multitask Learning.

3.1 Hints

In this domain a variety of different dependencies exists that can be exploited as hints to improve the generalization performance of the forecasting task for a single equity.

Market Model: The first hint investigated is derived from *market model* [Steiner, Bruns 1993] applied to equities. It states that the return of stocks depends on overall environmental conditions as well on company specific developments.Therefore the market model assumes that a part of the return of a stock is highly influenced by the environmental behavior of the overall market, which can be observed through a high correlation between a stock index and the return of that stock. The behavior of the stock index DAX was therefore used as a hint for the underlying regularities of a single stock.

Time Horizons: Under the assumption that stock prices are predictable, they can be modeled by a function highly influenced by noise. But high levels of noise can be disastrous for inductive learning, since the information about the true input-output dependency of a training example can be fully blurred. Underlying dependencies can be better identified through the noise if not only the value at time step t is observed, but also the values shortly before and after time step t. Therefore we added the tasks of forecasting the return of two days and two weeks to support the forecast of the one week horizon.

Global Market Integration: In a worldwide integrated market the different national market sectors depend on each other [Gjerde, Saetten 1996]. These interdependencies can be exploited in form of related tasks. Here the Dow-Jones index was used being a leading indicator of the next day's DAX index, thus giving a hint about the return of the equities.

3.2 Results

For each stock models were estimated without hints (STL[3]) and with four hints (MTL). The additional tasks included in the MTL model were the forecasts of the DAX index, the Dow-Jones index and the two time horizons. Each model's input consisted of 20 variables representing historical information about the stocks, stock indices, exchange rates and interest rates. The number of hidden units was set to four and the learning algorithm RProp [Riedmiller, Braun 1993] was used. Each model was trained and validated on data from 1.2.1990 to 24.10.1995 and tested on data from 25.10.1995 to 25.10.1996. Training was continued until the main task reached the best performance on the validation dataset, reflecting the emphasis on the main task. The models were re-estimated 100 times with new initial weight values.

Stock	Without Hints	With Hints	Naive	Buy-and-Hold
Allianz	51,96%	$^{+}$58,70%	$^{+}$54,76%	$^{+}$53,55%
Daimler	60,06%	$^{+}$64,64%	$^{-}$45,59%	61,14%
Dt. Bank	60,83%	59,98%	$^{-}$45,23%	$^{-}$48,47%
Siemens	50,01%	$^{+}$54,95%	$^{+}$52,78%	$^{+}$56,49%
Veba	60,40%	$^{+}$62,02%	$^{-}$58,06%	$^{+}$61,84%
Average	56,65%	60,06%	51,29%	56,28%

Table 1. Classification performance without hints (STL), with four hints (MTL), the naive prediction and the buy-and-hold strategy. $\pm$ shows a statistically significantly better/worse model compared to the model without hints to level 5%

Table 1 shows that in four out of five cases the MTL models were significantly better in prediction rate compared to the models without hints and were never significantly worse. The MTL model dominated on average the buy-and-hold strategy as well as the naive forecast. The buy-and-hold strategy is the strategy that holds the equity until the end of the test interval. The naive prediction states that a change at time step t is the best predictor of the change at time step $t+1$.

In table 2 it can be seen that this is also true for the average annualized return of a simple trading strategy derived from the forecasts of the direction. In this strategy the stock was bought in case of a predicted rise and was sold otherwise. Because of the rising German stock market in 1996 the buy-and-hold strategy represented a benchmark difficult to beat.

Stock	Without Hints	With Hints	Naive	Buy-and-Hold
Average	16,30%	21,19%	3,16%	18,21%

Table 2. Annualized return without hint (STL), with four hints (MTL), the naive prediction and the buy-and-hold strategy

[3] Singletask learning

In further tests with a changing number of given hints it could be observed that in most cases removing tasks from the model MTL resulted in a decrease in performance. Here two smaller Multitask models were estimated. The first model used the DAX index (MTL_{DAX}) as additional task, the second the two time horizons (MTL_{Time}). When the DAX hint was given it resulted in a performance that was better on average and in most cases compared with STL and worse compared to the models with five given tasks (Table 3). When two different time horizons were used as additional tasks, the same could be observed.

Stock	STL	MTL_{DAX}	MTL_{Time}	MTL
Allianz	51,96%	$^+$54,66%	$^+$53,24%	$^+$58,70%
Daimler	60,06%	58,94%	61,07%	$^+$64,64%
Dt. Bank	60,83%	60,35%	$^-$58,42%	59,98%
Siemens	50,01%	50,72%	$^+$52,78%	$^+$54,95%
Veba	60,40%	$^+$63,03%	$^+$62,70%	$^+$62,02%
Average	56,65%	57,54%	57,64%	60,06%

Table 3. Classification performance without hints, with one hint (DAX), with two hints (two time horizons) and with four hints. $\pm$ shows a statistically significantly better/worse model compared to the model without hints to level 5%

Theoretical considerations of good additional tasks are essential in order to gain good results, but are not always sufficient. Multitask Learning is an inductive transfer method that depends on the data of additional tasks. Therefore additional tasks can have no or even a negative influence if the quality is not good enough or if the relationship between the tasks is not present in the sample. In our experiments the MTL approach showed to be robust over many different models and resulted only in a few cases in a decrease of performance.

Two further models were estimated for the Allianz stock to investigate if the MTL effect was really based on the *domain specific* information. Each of the two models was given a different hint: The first was given a noisy hint in form of a randomly generated dataset and the second an additional task in form of the primary task. While these two models showed no impact compared to the model without hints (Table 4), the MTL_{DAX} model resulted in a statistically significant increase of the generalization performance. This shows that the usefulness of a hint clearly depends on its information value.

Equity	Random	Twice	STL	MTL_{DAX}
Allianz	52,74%	51,05%	51,96%	54,66%

Table 4. Classification rate for Allianz models: Model with a randomly generated additional task, model with twice the main task, STL model, MTL_{DAX} model

Notable is that adding further tasks resulted not only in a better performance of the main task, but also of the other additional tasks. In table 5 it can be seen, that an increase was also observed for additional tasks. The observed DAX classification rate of model MTL_{DAX} increased when further tasks were added (MTL). So not only the main task gained from the Multitask approach but also the tasks together. This shows, how the different tasks positively influence each other. It indicates that the relationship between them is successfully used to find a better model that supported more than one task.

Models	MTL	MTL_{DAX}
Allianz	62,79%	58,88%
Daimler	64,37%	52,36%
Dt. Bank	62,08%	51,84%
Siemens	62,90%	49,90%
Veba	63,33%	61,43%

Table 5. Classification rate of the additional task DAX for MTL and MTL_{DAX}

4 Conclusions

We introduced the method of neural Multitask Learning, which incorporates hints about the domain's underlying regularities in form of related learning tasks. Especially in the domain of financial modeling exists a variety of interdependencies which can be expressed as additional tasks making Multitask transfer a valuable source of domain specific information in this field.

We presented the application of MTL to forecast five major German stocks. It could be shown that a single view on one forecasting task ignores information about existing interdependencies on financial markets. Hints based on these interdependencies were successfully incorporated through Multitask Learning to improve the quality of the resulting models.

Several directions of investigation remain open: First of all, methods to identify useful additional tasks and to quantify their information value have to be found. Of further interest are weighting methods between the hints and the main problem. Here techniques like learning schedules [Abu-Mostafa 1995] or a weighting through Bayesian techniques [MacKay 1992] can be used.

References

[Abu-Mostafa 1995] Abu-Mostafa, Y.S.: *Hints*, Neural Computation, 7: 639-671, 1995

[Caruana 1995] Caruana, R.: *Learning many related tasks at the same time with backpropagation*, Advances in Neural Information Processing Systems, 7,656-664, 1995

[Caruana 1996] Caruana, R.: *Algorithms and Applications for Multitask Learning*, The 13th International Conference on Machine Learning, Bari, Italy, 87-95, 1996

[Caruana, Baluja, Mitchell 1996] Caruana, R., Baluja, S., Mitchell, T., *Using the Future to 'Sort Out' the Present: Rankprop and Multitask Learning for Medical Risk Evaluation*, Advances in Neural Information Processing Systems, 8, 1996

[Gjerde, Saetten 1996] Gjerde, O., Saetten, F.: *Linkages among European and world markets*, European Journal of Finance, 2:165-179, 1996

[MacKay 1992] MacKay, D.J.C. : *A practical Bayesian framework for backpropagation networks*, Neural Computation, 4(3): 448-472, 1992

[Riedmiller, Braun 1993] Riedmiller, M. and Braun, H.: *A direct adaptive method for faster backpropagation learning: The RPROP algorithm*, Proceedings of the IEEE International Conference on neural networks, 1993

[Steiner, Bruns 1993] Steiner, M., Bruns, C.: *Wertpapiermanagement*, Schäffer-Poeschel, 1993

[Towell, Shavlik 1994] Towell, G.G., Shavlik, J.W.: *Knowledge-Based Artificial Neural Networks*, Artificial Intelligence, 70: 119-165, 1994