Predicting Service Levels Using Neural Networks

Russell Ainslie[1[0000-0001-5398-1269]], John McCall[1[0000-0003-1738-7056]], Sid Shakya[2], and Gilbert Owusu[3]

[1] Robert Gordon University, Aberdeen AB10 7QB, UK
[2] EBTIC, Khalifa University, Abu Dhabi
[3] British Telecom, Ipswich IP5 3RE, UK
russell.ainslie@bt.com

Abstract. In this paper we present a method to predict service levels in utility companies, giving them advanced visibility of expected service outcomes and helping them to ensure adherence to service level agreements made to their clients. Service level adherence is one of the key targets during the service chain planning process in service industries, such as telecoms or utility companies. These specify a time limit for successful completion of a certain percentage of tasks on that service level agreement. With the increasing use of automation within the planning process, the requirement for a method to evaluate the current plan decisions effects on service level outcomes has surfaced.

We build neural network models to predict using the current state of the capacity plan, investigating the accuracy when predicting both daily and weekly service level outcomes. It is shown that the models produce a high accuracy, particularly in the weekly view. This provides a solution that can be used to both improve the current planning process and also as an evaluator in an automated planning process.

Keywords: Neural Network, NN, Prediction, Service Levels, Early Stopping Strategy, Planning

1 Introduction

A key target during the service chain planning process [1] in service industries, such as telecoms or utility companies, is ensuring adherence to service level agreements [2]. These specify a time limit for successful completion of a certain percentage of tasks on that service level agreement. With the increasing use of automation within the planning process [3], the requirement for a method to evaluate the current planning decisions effect on service level outcomes has surfaced. Further to that, with the criticality of service levels to the company's performance, improvements to current techniques are also paramount.

For this purpose we present a neural network (NN)[4][5][6] model to predict the upcoming service levels based on the current status of the plan and investigate the accuracy this model achieves using some anonymized real world data. The model predicts the service levels daily, but does so in such a way that aggregate predictions can also be produced allowing a weekly prediction to assist the current planning process.

As such the actual volume of tasks successfully completed are predicted and then used to calculate the service level percent, rather than predicting the percentage directly.

In this paper we first define this service level prediction problem in section 2, describing the general problem and introduce our specific real world example. We then outline our NN model we used to solve the problem in section 3. Section 4 contains the results achieved by this solution, looking at both the daily accuracy and the aggregated weekly prediction accuracy, before we conclude in section 5.

2 The Service Level Prediction Problem

2.1 General Problem Definition

The service level prediction problem is that of accurately predicting the percentage of tasks, R_i, by service level agreement, i, which will be successfully completed on time given the current state of the plan. The time allowed varies depending on which service level agreement the task in question is covered by.

The current state of the plan is defined by the current planned completions, C_j by skill, j, the start of day workstack levels, S_j, by skill, j, the intakes I_j(the number of new tasks entering the plan) by skill, j, and the available capacity, V_k, by resource types, k.

$$R_i = f_i(C, S, I, V) \tag{1}$$

Here C, S , I and V define the input set of all completions, workstacks, intakes and capacities respectively, f_i is the function of these inputs to produce the service level of service level agreement i.

In addition, given that service level agreements are a commitment to complete tasks of that type successfully within a certain time frame, historic values of the inputs also need to be considered. For example, if the service level agreement for a particular task is to complete a certain number successfully within two days then the intake from two days ago would contain some tasks that required completion by today. We further define C_{jt} as the planned completions by skill j t days before the prediction date, similarly for S_{jt} and I_{jt} as the workstack and intake for skill j t days prior respectively and V_{kt} as the capacity for resource type k t days prior. For example the input C_{j2} would be the planned completion for skill j two days before the prediction date. Defining T as the maximum number of days and the sets C_T, S_T, I_T and V_T as the sets of all completions, workstacks, intakes and capacities respectively, where $0 <= t <= T$ gives the updated problem definition.

$$R_i = f_i(C_T, S_T, I_T, V_T) \tag{2}$$

2.2 Problem Example

The data used in this paper consisted of 203 data points, each data point representing a day. For each data point (or day) there was the plan state (C, S, I, V) and resulting service levels, R_i, for 56 separate areas. The plan state included the capacity levels for 3 resource types and the completions, intakes and workstacks for 6 skills giving 21 input values for each plan day. The service levels were for tasks on two different service level agreements.

To allow the outputs to be aggregated (e.g. to produce a weekly service level report or an aggregated value for multiple areas) then the number of tasks successfully completed, Y_i, and the number failed, F_i, require predicting separately for each service level i. This way they can be summed to calculate the percentage of tasks successfully completed at any reporting level required. The resulting service level for any aggregated level being calculated using the total success and failure predictions of the individual predictions as follows.

$$R_i = {}^{Y_i}\!/_{Y_i + F_i} \tag{3}$$

Here Y_i and F_i are still a function of the current plan state for the prediction date and the plan state for the past *4* days (chosen after correlation analysis) giving similar equations to 2.

$$Y_i = g_i(C_4, S_4, I_4, V_4) \tag{4}$$

$$F_i = h_i(C_4, S_4, I_4, V_4) \tag{5}$$

3 Neural Network Solution Method

3.1 Model Construction

A feedforward multi-layered perceptron [7] neural network was chosen as they are widely used for forecasting [8]. A single hidden layer, containing 12 nodes, was used to predict each of the outputs, Yi and Fi, individually, giving a single output node. The number of input nodes varied for each output as we dynamically selected the inputs to use from the plan state sets of C4, S4, I4 and V4. Each layer used the sigmoid activation function [9]. The network was implemented using the Encog [10] library in Java.

With such a large number of inputs there is a risk of noise impacting the accuracy of the trained models. As such we also employed a dynamic filtering technique previously used in [11] to use each input's non-linear correlation with the output that the current model is being built for to select which inputs to use. Some initial quick tests showed that choosing a value of 0.5 proved to be a good cutoff point. Thus, during the construction of the model to predict each output in each area only inputs with a correlation value of 0.5 or greater were used.

3.2 Training

The models were trained using the resilient backpropagation algorithm [12] with 25 random restarts. The training data was split into two sets, the training set and the validation set. The models being trained using the training set and having their final accuracies evaluated on the validation set.

Further, to avoid overfitting, the early stopping strategy [13] was used. In the early stopping strategy, after each training iteration, the current accuracy of the model is tested using the validation set. Once the accuracy on the validation set stops improving and starts to decrease then overfitting has started to occur so the training process is halted.

4 Results

4.1 Experimental Method

For each area we split the data into 178 training points and 21 for testing. This data was further clustered by day of the week giving 25 to 26 training points and 3 testing points per cluster. For the training process the training portion was split further, removing 5 points for validation leaving 20-21 points for training.

We then used this data to create and train a model for each of the 4 outputs (Y_1, Y_2, F_1 and F_2) for each of the 56 areas for each day of the week. Each model was trained using the training data points and then tested on the test data. The accuracies of the predictions against the testing data are presented for each experiment.

4.2 Daily Prediction Results

Table 1 shows the accuracy of the daily predictions grouped by day of the week. For example the Monday entry shows the average accuracies for predictions made using the Monday models created for each of the 56 areas. The accuracy is shown for the success prediction model, Y_i, and failure prediction model, F_i, for both of the service levels i, along with a column showing the resulting accuracy when the models are combined to give the service level prediction R_i.

The accuracy for the individual models (success or failure) are fairly low, being about 52.4% accurate on average for the two service levels at predicting failures. Failures are low volume however so this is to be expected.

Table 1. Average accuracy of predictions using the early stopping strategy over all areas by day of the week

Day Of Week	Service Level 1			Service Level 2		
	Y_1	F_1	R_1	Y_2	F_2	R_2
Sat	80.3%	56.4%	89.7%	87.1%	59.4%	92.9%
Sun	61.0%	4.2%	86.1%	52.3%	-4.0%	63.8%
Mon	86.2%	55.0%	93.9%	87.8%	67.3%	93.8%
Tue	84.6%	58.7%	95.3%	74.6%	48.4%	83.2%
Wed	87.8%	62.8%	95.9%	86.7%	64.5%	93.2%
Thu	87.6%	61.8%	93.0%	85.9%	63.3%	93.3%
Fri	89.0%	66.1%	95.0%	88.8%	70.0%	93.0%
Average	**82.4%**	**52.2%**	**92.7%**	**80.5%**	**52.7%**	**87.6%**

This changes when we combine the success and failure model outputs to generate predictions of the resulting service level, R_i. The average accuracy across all days and both service levels in this case is about 90.2% even when including the lower accuracy produced by the reduced volumes on a Sunday. Considering the goal of the problem is to predict the service level this is a much better evaluator and as such we can see this model is useable in practice.

4.3 Weekly Prediction Results

To further evaluate the usefulness of this model to the current planning procedures, whereby senior planners take a weekly view of each areas workstacks to decide where might need additional resourcing, we also analyzed the accuracy of the predictions when aggregated to the weekly level. The success and failure models were summed for each of the three weeks and used to calculate the resulting service level for that week. We then counted the number of times the error was less than a certain percentage in each area and also calculated the average accuracy across all 56 areas for each week.

Table 2 shows the weekly results achieved. The weekly level prediction showing an expected improvement over the daily accuracy, producing an average of 96.7% across all areas across all weeks. We can also see that around 78.6% of the time the error of the prediction in an individual area is less than 5% increasing to 92.9% when extending the range to 8%. This shows that decisions can be taken using this model with reasonable certainty, particularly as some of the higher error values are caused by areas with lower volumes. Something the senior planner would have knowledge of.

Table 2. Weekly accuracy across all areas using early stopping strategy

Week	Service Level 1				Service Level 2			
	< 8%	< 5%	< 2%	Avg Acc	< 8%	< 5%	< 2%	Avg Acc
1	56	49	26	97.4%	46	41	22	96.4%
2	54	49	23	97.2%	52	42	21	96.4%
3	52	44	24	96.8%	52	39	18	96.3%

5 Conclusion

In this paper we identified the requirement for a service level predictor to assist the evaluation of plan solutions generated by automated planning methods as well as the opportunity to improve the current planning process. We defined this service level prediction problem, stating the general problem in equation 2 and introducing a specific real world example. Initial analysis of the data showed the requirement to include 4 days plan states as part of the inputs and we further split the problem into predicting the success (equation 4) and failure (equation 5) separately to allow their combination to produce a service level prediction at any level of aggregation.

We then described the neural network model we used to solve these problems, including a method to dynamically filter the larger number of inputs created by the addition of the past plan states.

The resulting model was then evaluated on the real world data, where we showed that we were producing good daily accuracy to give a reasonable evaluation of plan solutions as well as give current planners indication of days where problems are occurring. In addition we showed that the accuracy at the weekly level, where it would be used within the current planning process, was very good providing a good tool to assist decisions relating to release of additional resources to specific areas.

With the current planners problem solved the next step is now to utilize this solution to evaluate solutions in an automated planning application. Additional work could also be followed in investigating other solution techniques, such as developing the model using support vector machines [14].

References

1. Voudouris, C.: Defining and understanding service chain management. Service Chain Management, pp. 1–17. Springer (2008)
2. Verma, D. Service level agreements on ip networks. *Proceedings of the IEEE*, 92(9):1382–1388, 2004.
3. Owusu, G., O'Brien, P., McCall, J., Doherty, N.F.: Transforming Field and Service Operations. Springer, Stellenbosch (2013)
4. Anderson, J. An Introduction to Neural Networks, MIT Press(1995)
5. Haga, M., Demuth, H., Beale, M., De Jesús, O.:Neural Network Design (Vol 20). PWS publishing company, Boston (1996)
6. Kourentzes, N., Crone, S.: Advances in forecasting with artificial neural networks. Lancaster University (2010).
7. Gardner, M. W., Dorling, S. R.: Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, vol. 32, no. 14, pp. 2627-2636, 1998.
8. Zhang, G., Eddy Patuwo, B., Hu, M. Y.: Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, vol. 14, no. 1, pp. 35-62, 1998
9. Sibi, P., Jones, S., Siddarth, P.: Analysis of different activation functions using back propagation neural networks. Journal of Theoretical and Applied Information Technology, vol. 47, no. 3, pp. 1264-1268, 2013.
10. Heaton, J.: Encog: Library of Interchangeable Machine Learning Models for Java and C#. Journal of Machine Learning Research, pp. 1243-1247. 2015
11. Ainslie, R., McCall, J., Shakya, S., Owusu, G. Predictive planning with neural networks. In Proceedings of the International joint conference on neural networks (2016). doi:10.1109/IJCNN.2016.7727460
12. KISI, Ö., Uncuoglu, E. : Comparison of three back-propagation training algorithms for two case studies. Indian journal of engineering & materials sciences, vol. 12, no. 5, pp. 434-442, 2005.
13. Prechelt, L.: PROBEN1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules. Karlsruhe(1994)
14. Tian, Y., Shi, Y., Liu, X.: Recent advances on support vector machines research. Technological and Economic Development of Economy, pp. 5-33. 2012