EP2420 Project 2 - Forecasting Service Metrics

André Silva

December 5, 2020

Task II

In this task we use a Recurrent Neural Network (RNN) [2] to predict, not the current state of a service metric, but the future states of said metric.

More specifically, we will be using a Long Short-Term Memory (LSTM) Network [1]. LSTM is a RNN architecture that mitigates the vanishing gradient problem by introducing 3 gates and a memory cell per unit. This allows each unit to model the remembrance and addition of, respectively, old and new memory. Being a RNN architecture, it also allows us to process, not only but also, sequences of data and build models that take into account the order embedded in sequences.

First, we pre-process the trace, standardizing along each column and removing outliers that deviate more than 40σ. We also apply tree-based feature selection, using 10 estimators, to get the top 16 features.

After this, we split the trace into training and test sets, and then transform them into new datasets that follow the structure $([x^{(t-l)}, ..., x^{(t)}], [y^{(t)}, ..., y^{(t+h)}])$. As we are using analyzing *KV periodic* [3], our step size (i.e. time between each sample of a sequence), is 1 second.

We then perform an experiment to study the impact of l on the accuracy of a model and its predictions of 0...10 steps into the future by building 11 different models with $l \in \{0, ..., 10\}$. In this experiment we build stacked LSTM Networks [4], composed of 3 layers. The first layer is an LSTM layer with 100 units, the second layer is an LSTM layer with 50 units, while the third and final layer is a Dense layer with $h + 1$ units, corresponding to the output. Each model was trained for 100 epochs, using `Adam` as the optimization algorithm and the mean-squared error as the loss function. The default values were utilized for all other parameters.

The results can be found below in Figure 1.

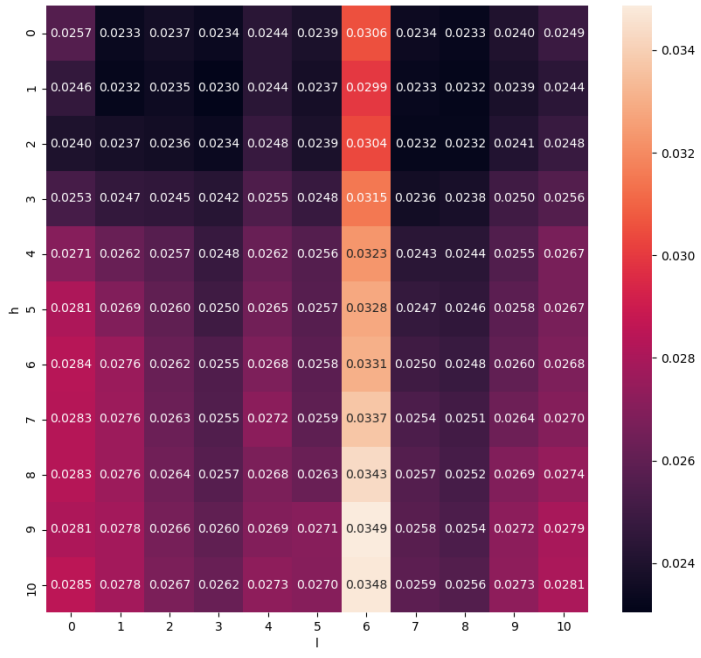

Figure 1: Heat-map of NMAE for each stacked LSTM [4] model with $l \in \{0, ..., 10\}$ when predicting $y^{(t+h)}$

Figure 1 shows us the NMAE of each model when predicting $y^{(t+h)}$. Rows represent the time horizon $h = 0, ..., 10$ and columns represent the lag $l = 0, ..., 10$. The color gradient of the heat-map allows us to visualize and extract conclusions better than if we used a normal table.

When we analyze the results by looking at each row we notice that, in most cases, the NMAE decreases as we increase l. On the other hand, when we look at each column, we see that the NMAE increases as we increase h. Both these events are, intuitively, expected. When we increase l, we are providing more information, or context, to the model, so we expect better predictions. When we increase h, we are asking for a value that is more distant in time, and so, harder to predict.

This being said, we can clearly see that the model for $l = 6$ presents very deviated results. This could suggest that, during the training phase, the model hit a local optima or it is overfit, not allowing it to reach the same level of accuracy as the other models.

We can also see that, as in Task I, the NMAE doesn't monotonically decrease as we increase l from 0, as expected. One possible explanation, other than the two, still applicable, explanations suggested in the previous Task, could be related to the training of each model, seeing that each model converges differently.

When comparing these results with the results of Task I, we notice that, in most cases, our NMAEs are lower, which could be explained by the fact that Neural Networks can, theoretically, approximate any given function, which is not the case for Linear Regression. We can also see that the gradient of the results along the h axis is lower, suggesting that these models can also better forecast than the previous one.

References

[1] *Long Short-Term Memory Networks*. `https://en.wikipedia.org/wiki/Long_short-term_memory`. accessed: 2020-12-03.

[2] *Recurrent Neural Networks*. `https://en.wikipedia.org/wiki/Recurrent_neural_network`. accessed: 2020-12-03.

[3] F. S. Samani, H. Zhang, and R. Stadler. "Efficient Learning on High-dimensional Operational Data". In: *2019 15th International Conference on Network and Service Management (CNSM)*. 2019, pp. 1–9. DOI: `10.23919/CNSM46954.2019.9012741`.

[4] *Stacked Long Short-Term Memory Networks*. `https://machinelearningmastery.com/stacked-long-short-term-memory-networks/`. accessed: 2020-12-03.