Web service workload prediction using deep learning - Report 3

Stefan Sebastian

April 20, 2020

1 State of the art

TODO

2 Approach

The main goal is to find a performant model for web application workload prediction, which can be later used by a proactive microservice scaler. The methods used are different architectures of deep learning models: MLP, CNN, CNN-LSTM hybrid. The main contribution of this research is the application of deep learning to this specific problem and the comparison with a classic timeseries approach (ARIMA).

The problem design has been influenced by the goal of integrating this model into a proactive microservice scaler. First of all, the choice of the workload measure is number of requests. The idea is that the scaling prediction should not influence the predicted value, as would be the case with CPU or memory usage. Also this is in line with research done by Jindal et al [3] who propose a metric for measuring microservice performance based on number of satisfied requests. Another consideration is the prediction interval. Taking into account the experience of Netflix [2], who run a microservice architecture in production, the time window should be in the order of minutes, so you can predict spikes and have time to deploy new service instances.

A realistic workload has been used for this experiment, a wikipedia trace for 12 days in september 2007. From this a subset of requests was extracted (all requests for Japanese wikipedia). The subset was selected in order to compare results with Kim et al. [4] which used the same dataset. In order to turn a web request log file into a supervised dataset the following steps were taken: create buckets which contain the number of requests in a time interval, iterate over the buckets using the sliding window technique [1]. Basically we generate training instances with input (t, t-1, ... t-n) and output (t+2). The predicted value is t+2 instead of t+1 because a scaler using this model would need to have a buffer window during which to deploy the services.

After preparing the dataset two baseline models were prepared: the naive approach (predict traffic in window t+2 to be traffic in t) and a classic timeseries model (ARIMA). The next step was to experiment with different deep learning architectures and measure the results.

3 Evaluation of the approach

TODO

References

[1] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. *Machine Learning Strategies for Time Series Forecasting*, pages 62–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[2] Neeraj Joshi Daniel Jacobson, Danny Yuan. Scryer: Netflix's predictive auto scaling engine, November 2013. [Online; posted 05-November-2013].

[3] Anshul Jindal, Vladimir Podolskiy, and Michael Gerndt. Performance modeling for cloud microservice applications. pages 25–32, 04 2019.

[4] I. K. Kim, W. Wang, Y. Qi, and M. Humphrey. Cloudinsight: Utilizing a council of experts to predict future cloud application workloads. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 41–48, 2018.