Dataset:

Japan wiki, has a spike; grouped by time window; 5 – 10 – 15 min ; with 10min used for tuning

(t, t – 1, … t- n) to predict (t + 2)

The prediction is done as we close moment t, therefore if we predict moment t+1 and most requests are in the beginning of the interval the system would not have time to react.

Baseline

1.The naïve approach: predict that the workload won’t change, so nr requests at moment t+2 will be nr requests at moment t.

See reports/baseline – for plots and measurements

2. A classic approach: ARIMA

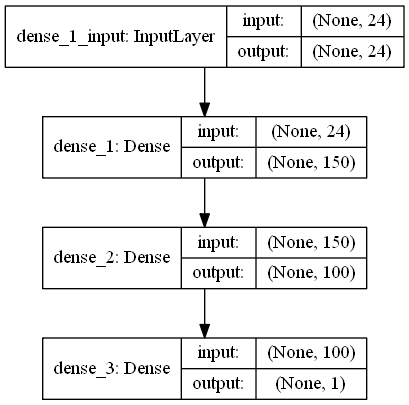
See arima/params for choice of params

See arima/centralized for results

Parameter tuning

1. This is the validation phase ; so we use K-Fold cross validation and average the results; K-Fold splits the dataset into k parts, performs training on k-1 and validation on the one left out; we use 3 folds for this exp
2. Set the baseline mlp model

After manually trying some configurations, set a baseline MLP model, with n = 24



1. Epoch + batch size

Performed a grid search for selecting an optimal epoch no and batch size

Batch size should ideally be a power of 2 (to fit into GPU architectures, some exp were ran on google colab which offers this feature). Lower batch size is more accurate but higher gives more training speed. As expected the best MSE is obtained for the lowest batch\_size(4) however it does not drop significantly at 8, regardless of epochs no. The selection of epoch no is again a tradeoff between speed and acc. We see a smaller no of epochs(50) performs poorly, while the difference between 100 and 250 is not that great, meaning that we can get a good approximation of a model using a batch size of 100.

