Neural Net Final Report

AllFresh
Reporter: o100442 Wei-Yuan Hsu
Advisor: Prof. Hahn-Ming Lee

Outline

Rank

Data

Model

Results

Discussion

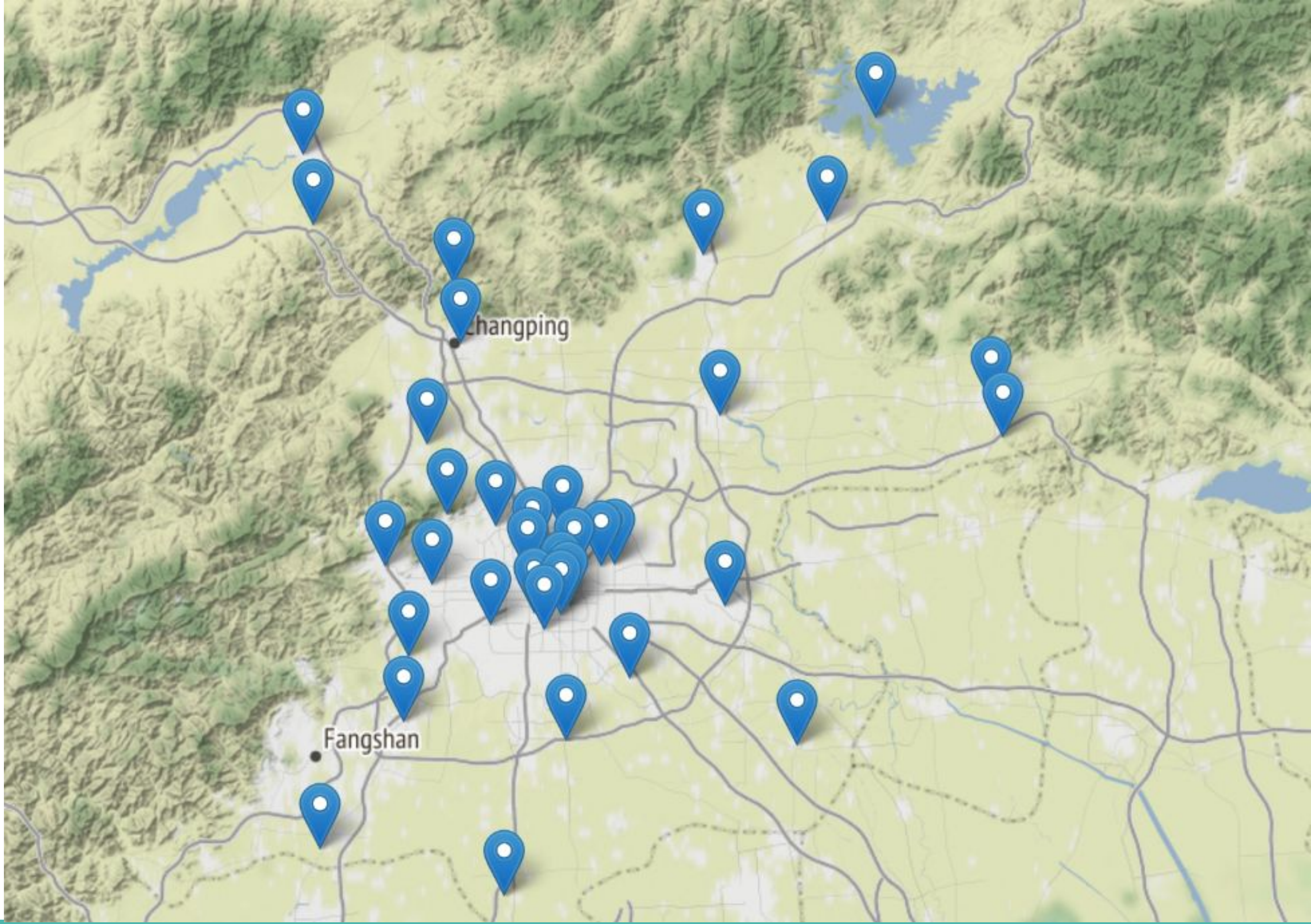
Rank

Rank

Highest rank	200
Public score	294
Specialized Prize for the last 10 days	444
Specialized Prize for the second-day prediction	517
Last rank	523

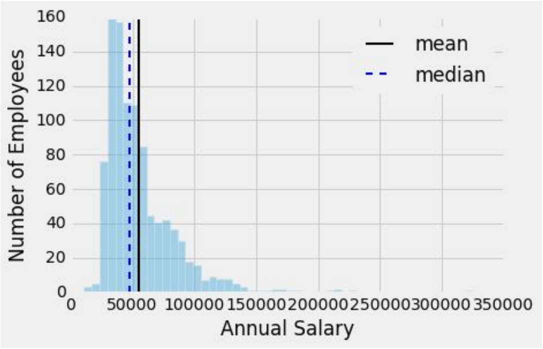
293	—	Jiao_ran	0.62636
294	—	AllFresh	0.62819
295	—	Janicell	0.62991

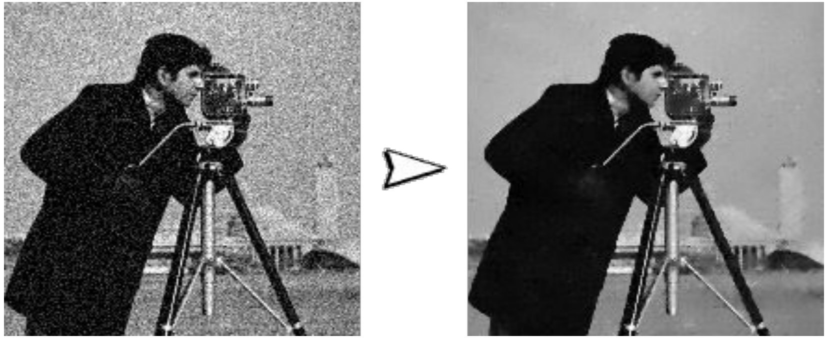
Data

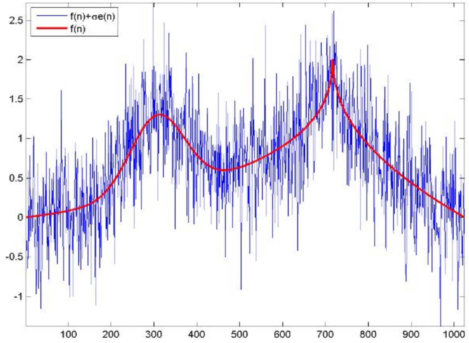


Input Data - Data Preprocessing

- Remove outlier (ex. Temperature, wind direction)
- Denoising: median number





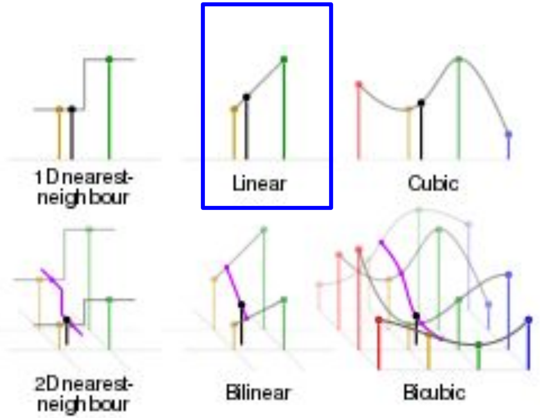


Reference:

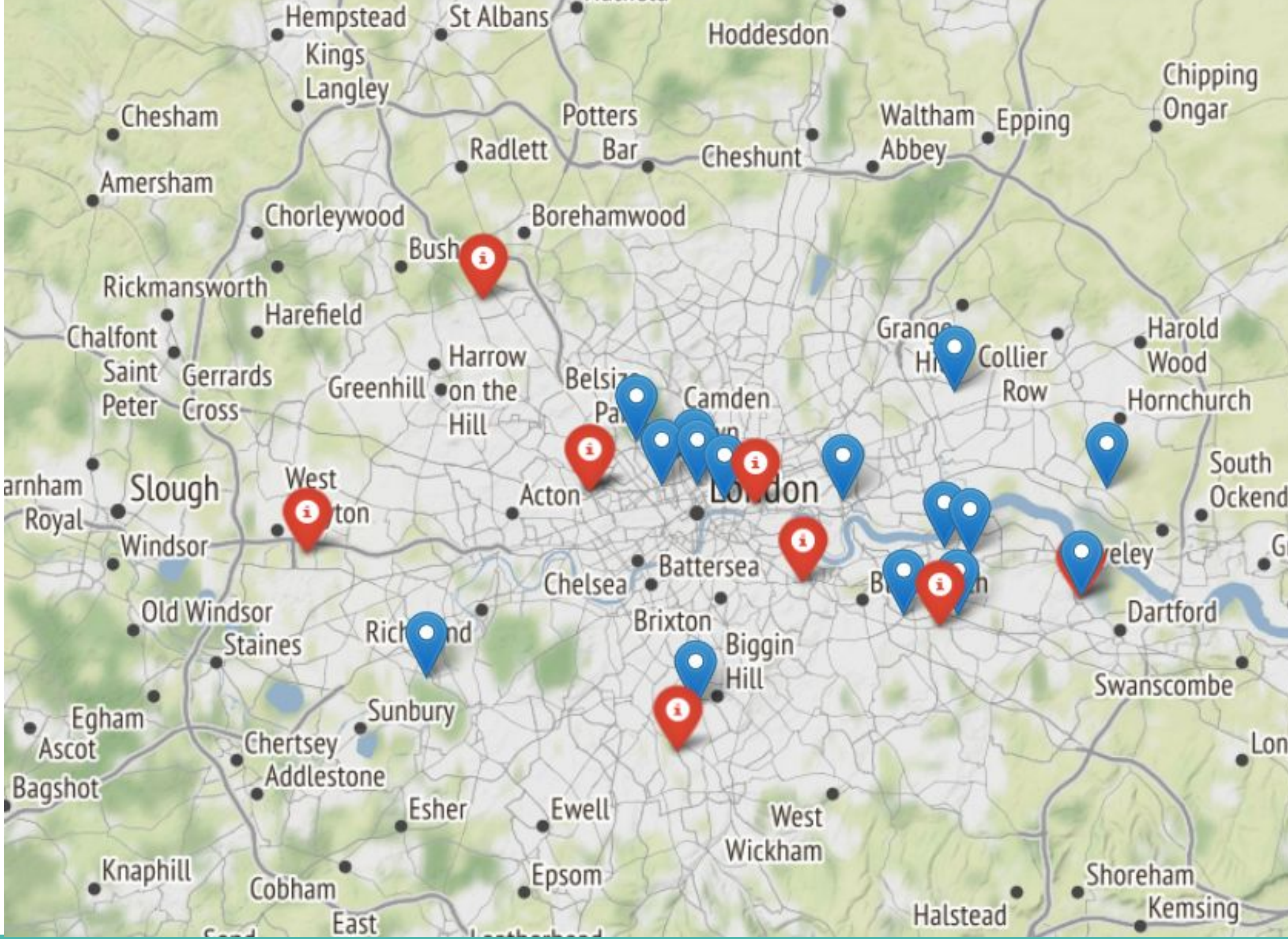
1. Local Approximations in Signal and Image Processing, http://www.cs.tut.fi/~lasip/2D/
2. Wavelet Denoising and Nonparametric Function Estimation, https://goo.gl/HJWVrs

Input Data - Handling missing data

- Missing value
    - missing # <= 23 hrs - pandas interpolation
    - missing 1 day (24 hrs) - average of previous 3 days
- External data
    - beijingair



Feature Engineering



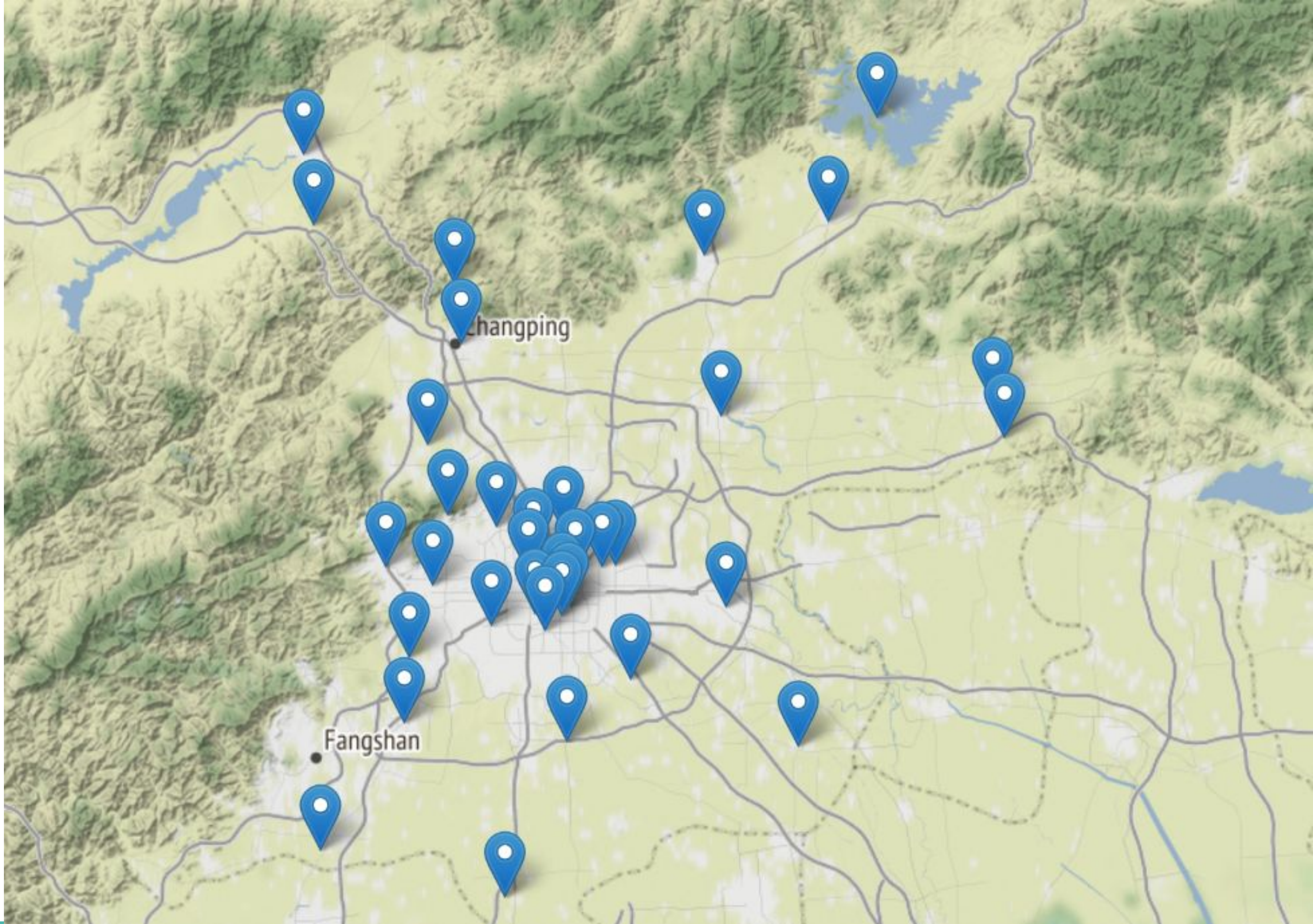
Feature engineering

RNN is powerful enough to learn the pattern behind features.

- **Important feature** (what we want to predict)
  - PM2.5
  - PM10
  - O3
- Other pollution
  - NO2
  - CO
  - SO2
- Weather information
  - temperature
  - pressure
  - humidity
  - wind_direction
  - wind_speed/kph

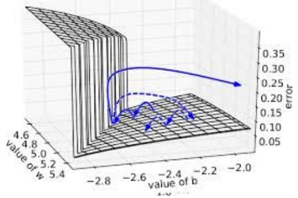
Reference
[1] Arturus/kaggle-web-traffic

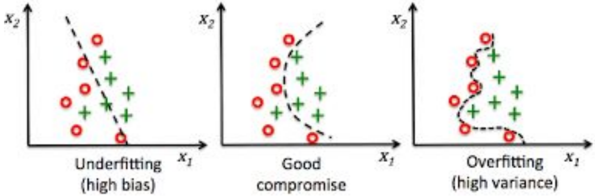
Model

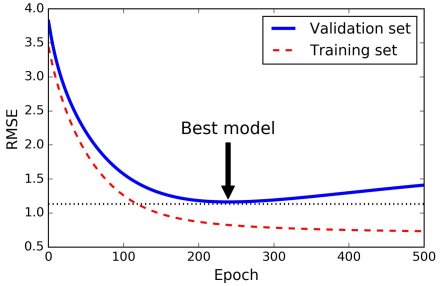


Training and Validation

- SMAPE loss boosting
  - Clipping
- Regularization
  - Overfit -> just fit
  - Neural Network: Dropout (20%)
- Early Stopping
  - Stop the rest training iteration at early stage if the model in case time waste
  - patience: 6





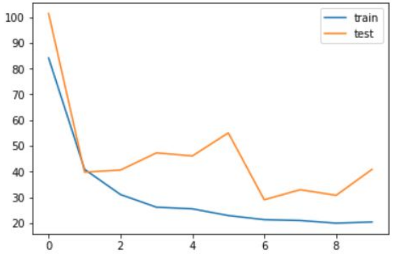


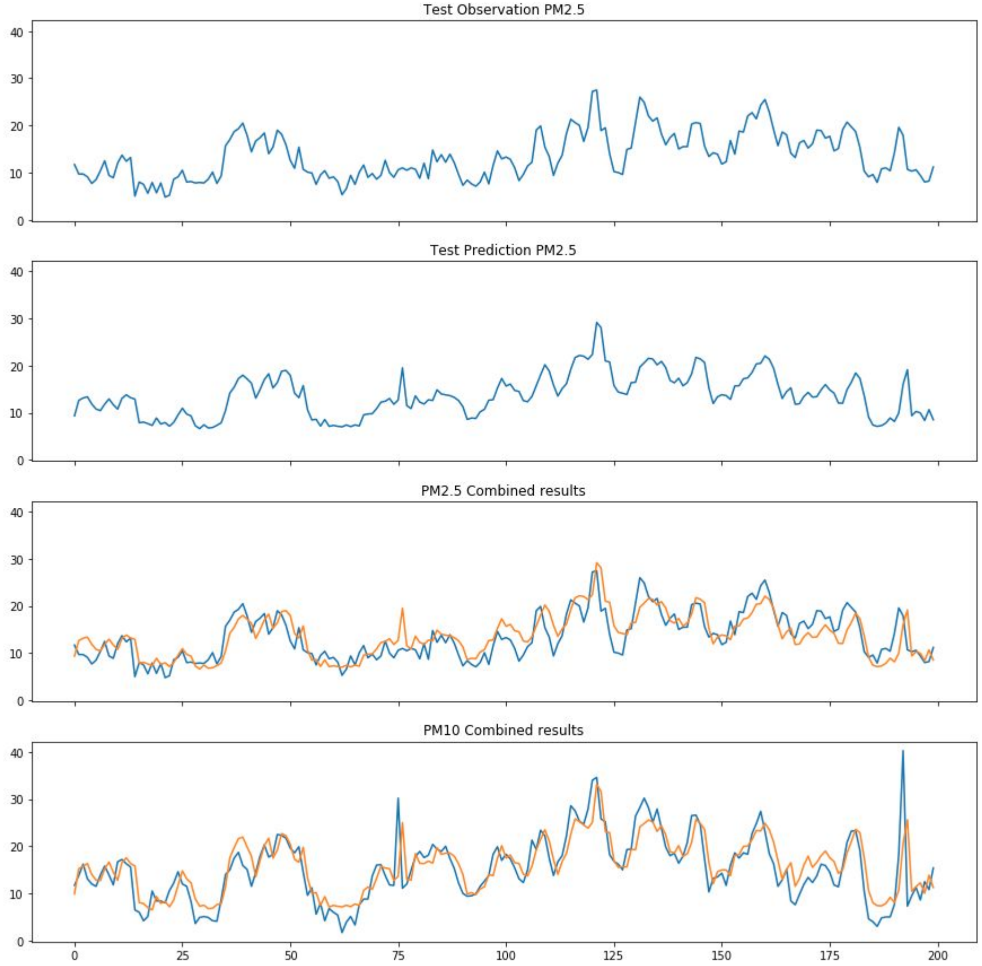
Reference:

1. Hands-On Machine Learning with Scikit-Learn and TensorFlow by Aurélien Géron
2. Why 50% when using dropout?

Phase I - Underfit

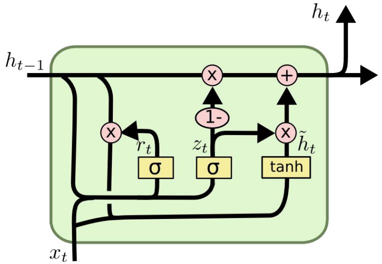
Input layer	(n_features, n_prev)
Hidden layer	LSTM
Hidden layer	LSTM (# of neuron = 300)
Output layer	2 (London) 3 (Beijing)
SMAPE (on original data)	0.65005302971099946





Model Design

- LSTM (GRU is not so good in this competition...)
- Seq2seq



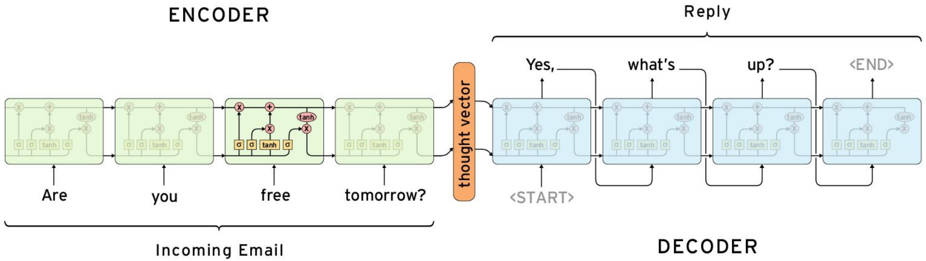
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

LSTM

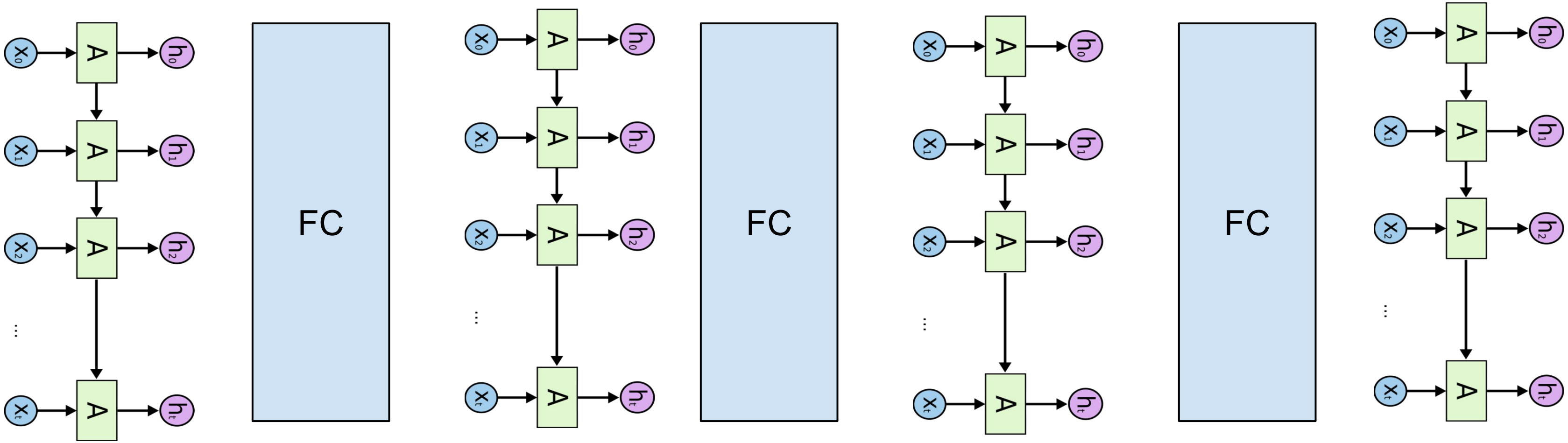


Seq2Seq Model

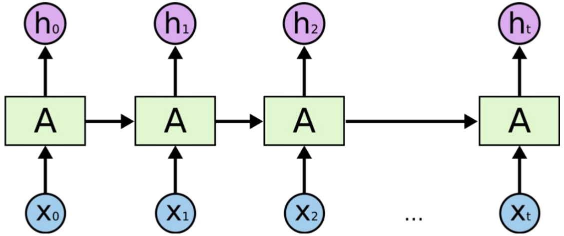
Reference
[1] Understanding LSTM Networks
[2] Generative Model Chatbots
[3] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
[4] Zaytar, Mohamed Akram, and C. E. El Amrani. "Sequence to sequence weather forecasting with long short term memory recurrent neural networks." Int J Comput Appl 143.11 (2016).

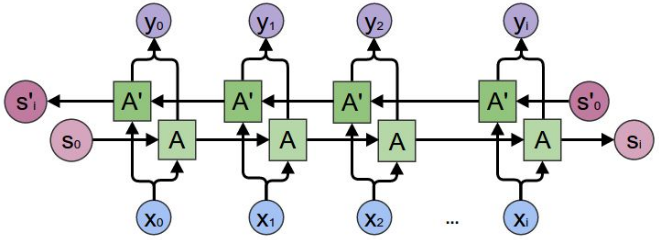
Model Design (cont.)

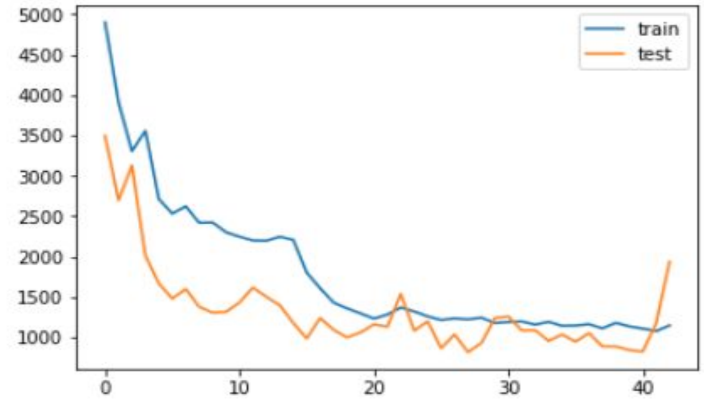
- Model: Seq2Seq Model
- Layers: 4
- Number of LSTM neurons: 1000



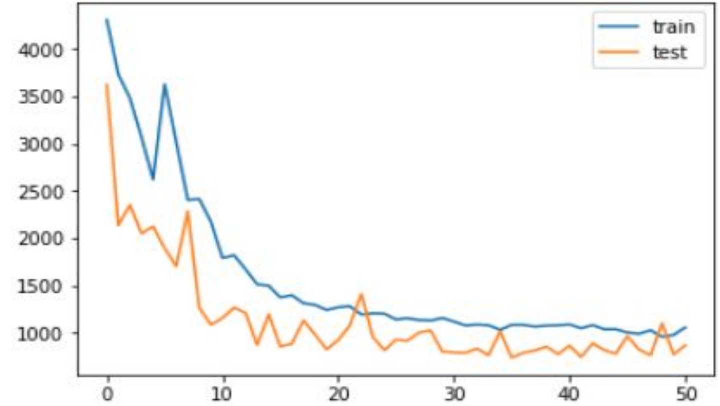
Model Design (cont.) - Bidirectional RNN







SMAPE: 0.5227025467578719

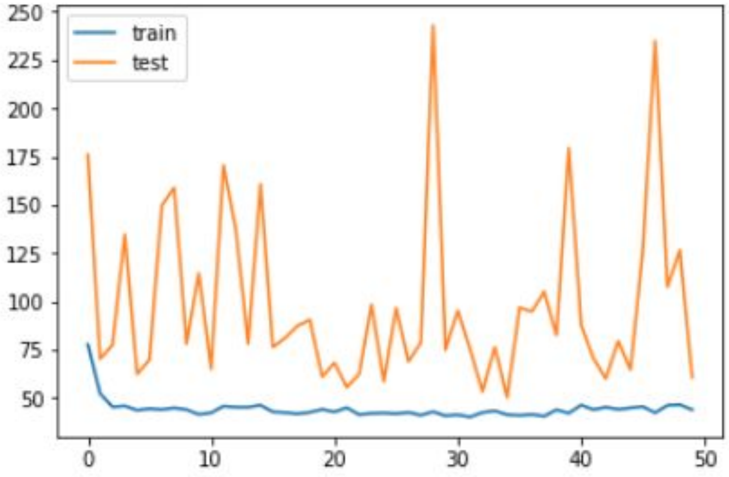


SMAPE: 0.4267148125612628

Model Design (cont.) - Optimizer

Q: Batch Normalization Layer?

Ans: On RNN is a disaster, because the RNN often vary with the length of input sequence. SMAPE got a big escillation in test error

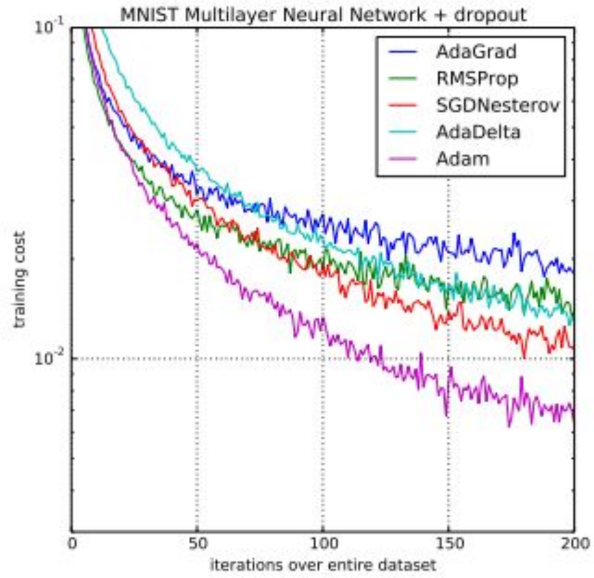


Reference:

1. Quora: Why is it difficult to apply batch-normalization to RNNs?, https://www.quora.com/Why-is-it-difficult-to-apply-batch-normalization-to-RNNs
2. Is it normal to use batch normalization in RNN/lstm RNN?, https://stackoverflow.com/questions/45493384/is-it-normal-to-use-batch-normalization-in-rnn-lstm-rnn

Model Design (cont.)

- Optimizer - Baseline - ADAM
- How's other Optimizer (ex. rmsprop, adagrad etc)?

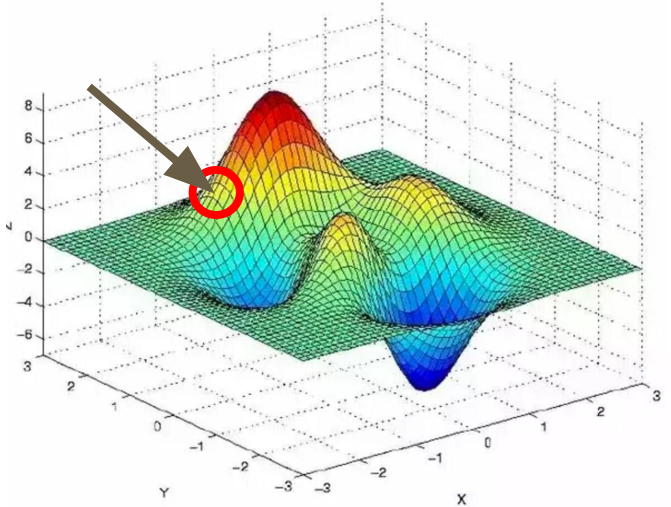


Final Model

```
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_1 (Bidirection (None, 100, 2000)         8080000
_________________________________________________________________
dense_1 (Dense)              (None, 100, 1000)         2001000
_________________________________________________________________
activation_1 (Activation)    (None, 100, 1000)         0
_________________________________________________________________
bidirectional_2 (Bidirection (None, 100, 2000)         16008000
_________________________________________________________________
dense_2 (Dense)              (None, 100, 1000)         2001000
_________________________________________________________________
activation_2 (Activation)    (None, 100, 1000)         0
_________________________________________________________________
dropout_1 (Dropout)          (None, 100, 1000)         0
_________________________________________________________________
bidirectional_3 (Bidirection (None, 100, 2000)         16008000
_________________________________________________________________
dense_3 (Dense)              (None, 100, 1000)         2001000
_________________________________________________________________
activation_3 (Activation)    (None, 100, 1000)         0
_________________________________________________________________
bidirectional_4 (Bidirection (None, 2000)              16008000
_________________________________________________________________
dense_4 (Dense)              (None, 1000)              2001000
_________________________________________________________________
activation_4 (Activation)    (None, 1000)              0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1000)              0
_________________________________________________________________
dense_5 (Dense)              (None, 1000)              1001000
_________________________________________________________________
dense_6 (Dense)              (None, 1000)              1001000
_________________________________________________________________
dense_7 (Dense)              (None, 3)                 3003
=================================================================
Total params: 66,113,003
Trainable params: 66,113,003
Non-trainable params: 0
```

Reproducible Results - Random seed

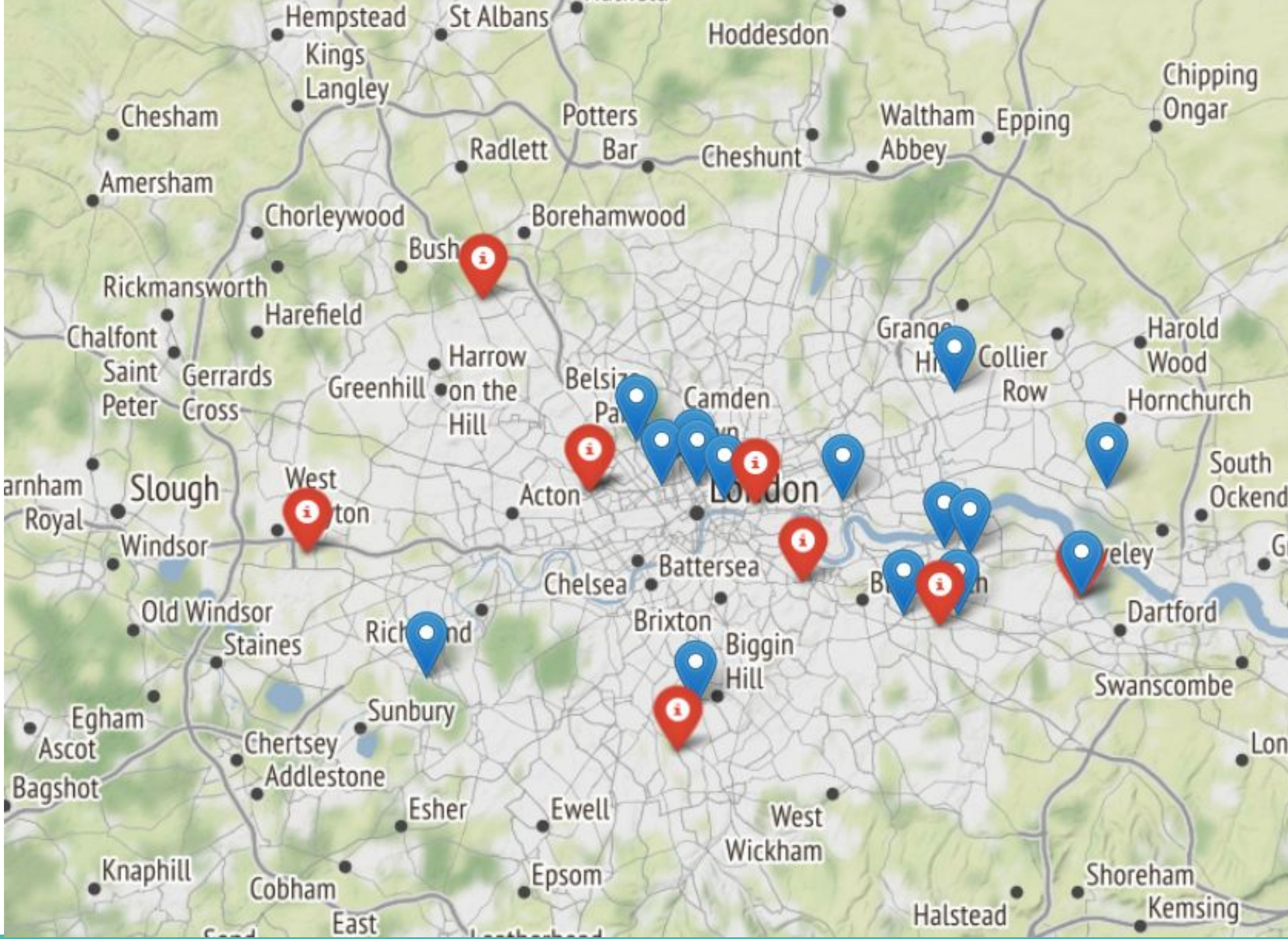
- Initialization random weights of Neural Net Model
- Given same initial to keep same random value (start from same position)



Reference:

1. How to Get Reproducible Results with Keras
2. Why Do I Sometimes Get Better Accuracy With A Higher Learning Rate Gradient Descent?

Results



Hempstead
St Albans
Hoddesdon
Kings Langley
Chesham
Potters Bar
Radlett
Cheshunt
Waltham Abbey
Epping
Chipping Ongar
Amersham
Chorleywood
Borehamwood
Bush
Rickmansworth
Harefield
Chalfont Saint Peter
Gerrards Cross
Greenhill
Harrow on the Hill
Camden
Collier Row
Harold Wood
Hornchurch
Slough
Royal
Windsor
Acton
South Ockend
Chelsea
Battersea
Dartford
Old Windsor
Staines
Brixton
Biggin Hill
Swanscombe
Egham
Ascot
Bagshot
Sunbury
Chertsey
Addlestone
Esher
Ewell
West Wickham
Knaphill
Cobham
Epsom
Shoreham
Kemsing
Halstead
East

Training Time

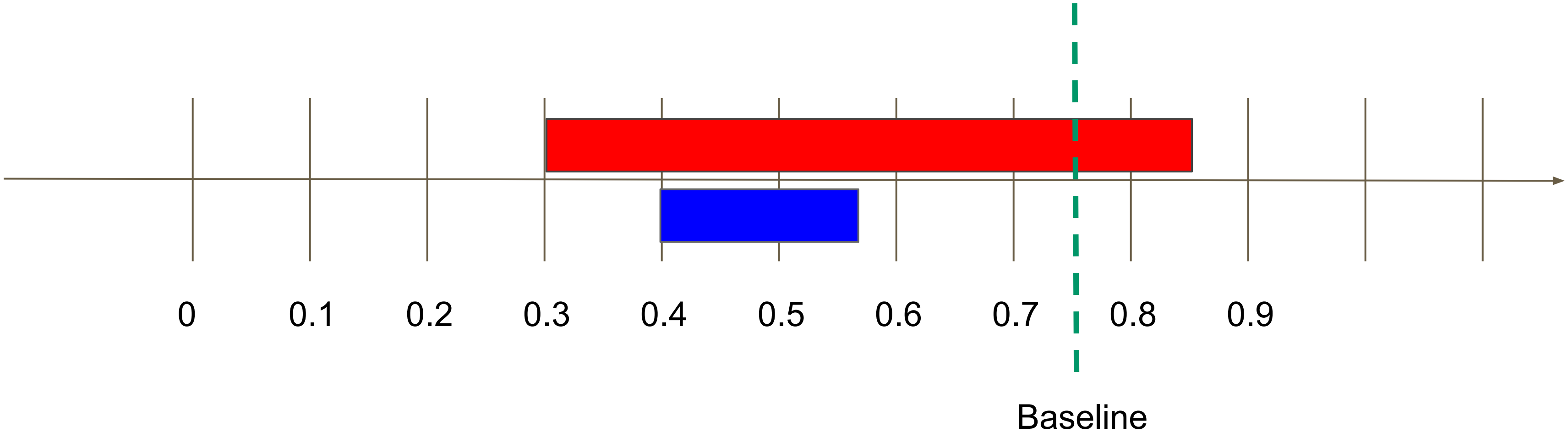
- 48 Models for every station
- Batch size: 32
- Average Time
  - 1 hr ~ 1.5 hr each city
- Total Time
  - around 2 ~ 3 days

Analysis

- RNN Model Underfit Solution
  - Add more layer
  - Add more hidden unit
  - Fully Connected Layer stabilized model when hidden neuron is a large number
    - SMAPE drop from 1.X -> 0.5 ~ 0.8

Analysis (cont.)

- SMAPE baseline is around 0.75 ~ 0.8
- SMAPE of 2 cities:
    - Beijing: [0.3, **0.85**]
    - London: [0.4, 0.57]



Analysis (cont.)

- PM 2.5 - Official data statistics
  - “...北京市全年PM2.5主要来源中本地排放占三分之二，区域 传输占三分之一[1]”
  - Car emissions gas: CO, NO2 [2]

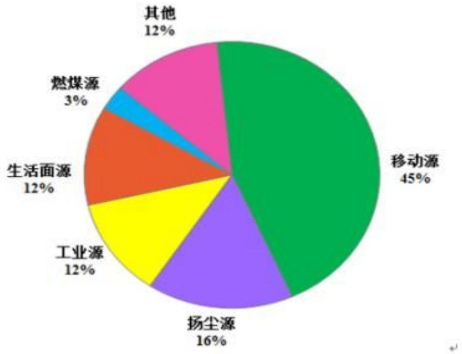


图 2 现阶段北京市大气 $PM_{2.5}$ 本地来源

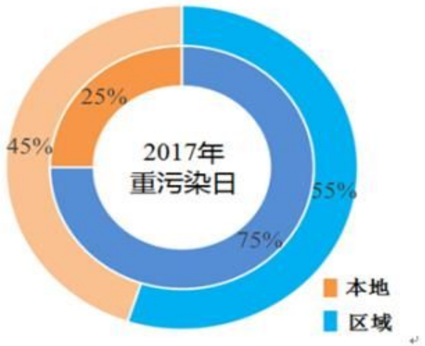


图 1 2017 年北京市重污染日大气 $PM_{2.5}$ 本地和区域贡献

Reference:

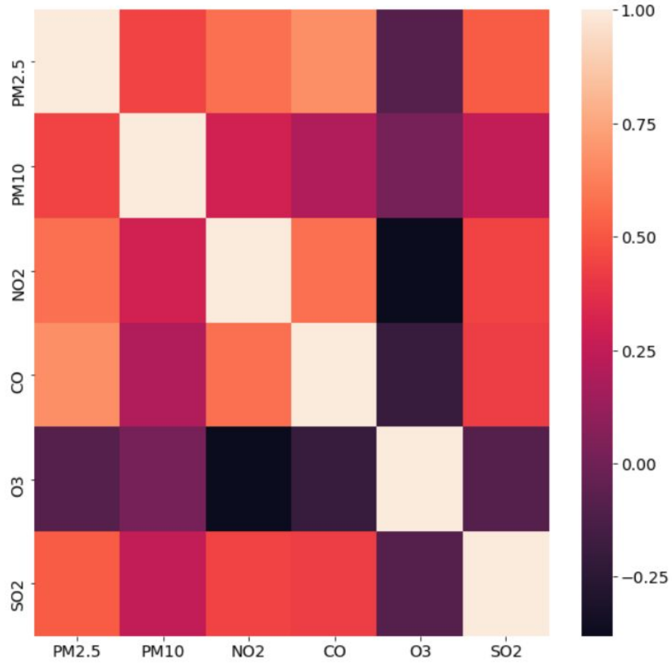
1. 最新科研成果新一轮北京市PM2.5来源解析正式发布, Beijing Environmental Protection Bureau
2. Wiki -- Exhaust gas

Analysis (cont.)

- Correlation: feature importance analysis
- Batch size: 64
- Drop SO2 -> SMAPE no too much difference
  - Save time

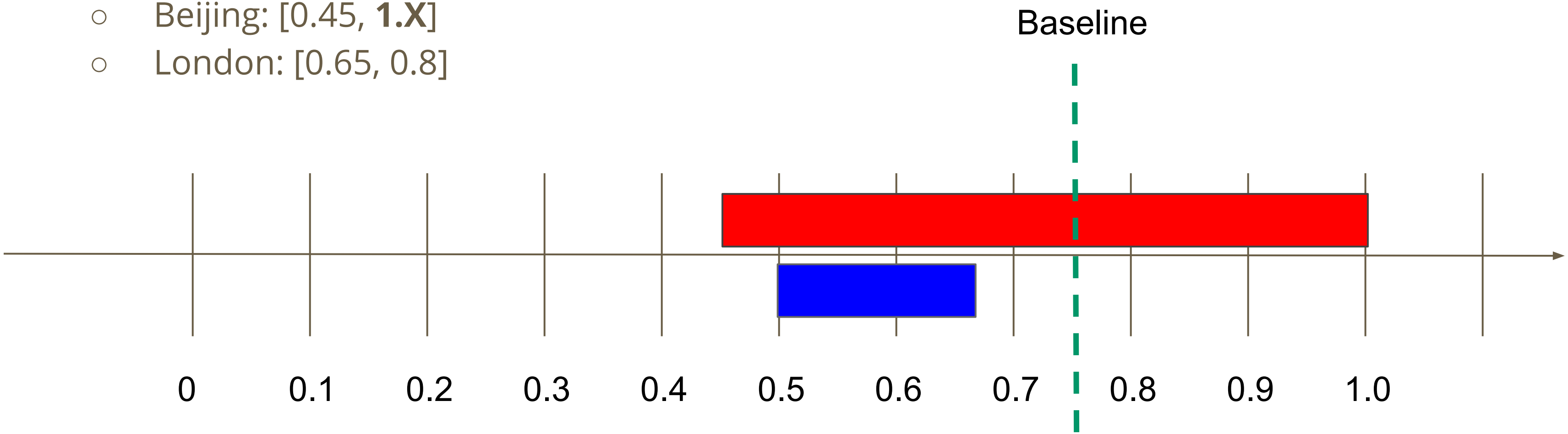
Correlation	Feature 1	Feature 2	
Positive (+)	PM2.5	CO	around 0.75
	PM2.5	PM2.5	around 0.5
X	PM10	O3	
Negative (-)	O3	NO2	around - 0.5

Beijing city feature correlation plot



Analysis (cont.)

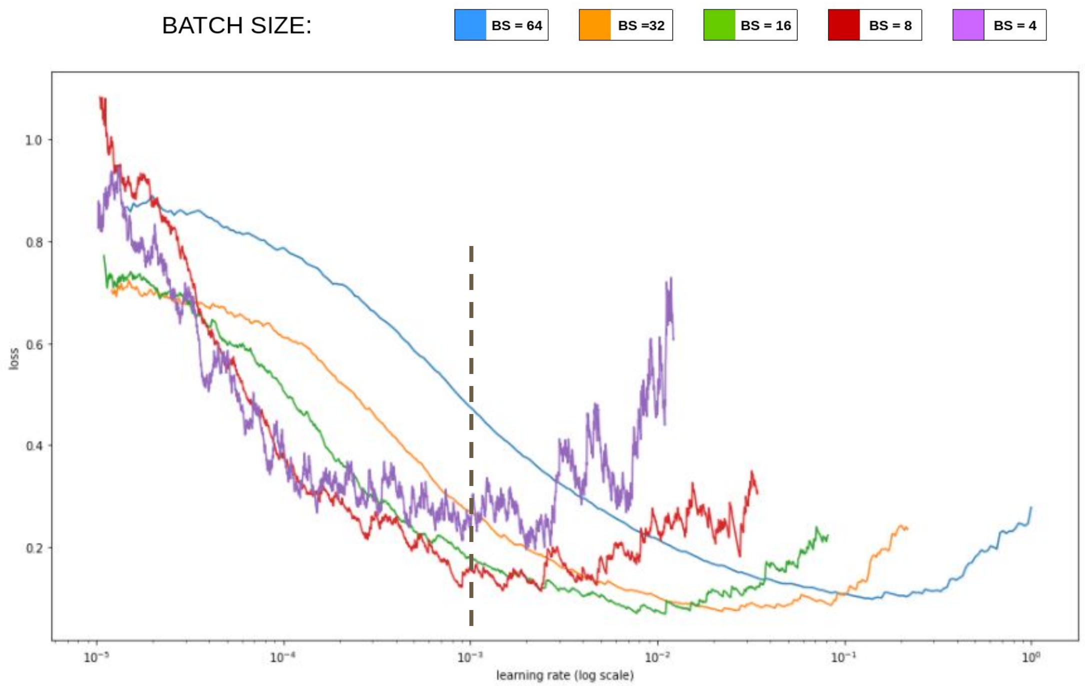
- Batch Size
  - 32
  - How about larger? 64 or 128?
- SMAPE issue
  - batch size (**up**), smape (**down**)
  - Beijing: [0.45, **1.X**]
  - London: [0.65, 0.8]



Analysis (cont.)

LOSS vs. LEARNING RATE FOR DIFFERENT BATCH SIZES

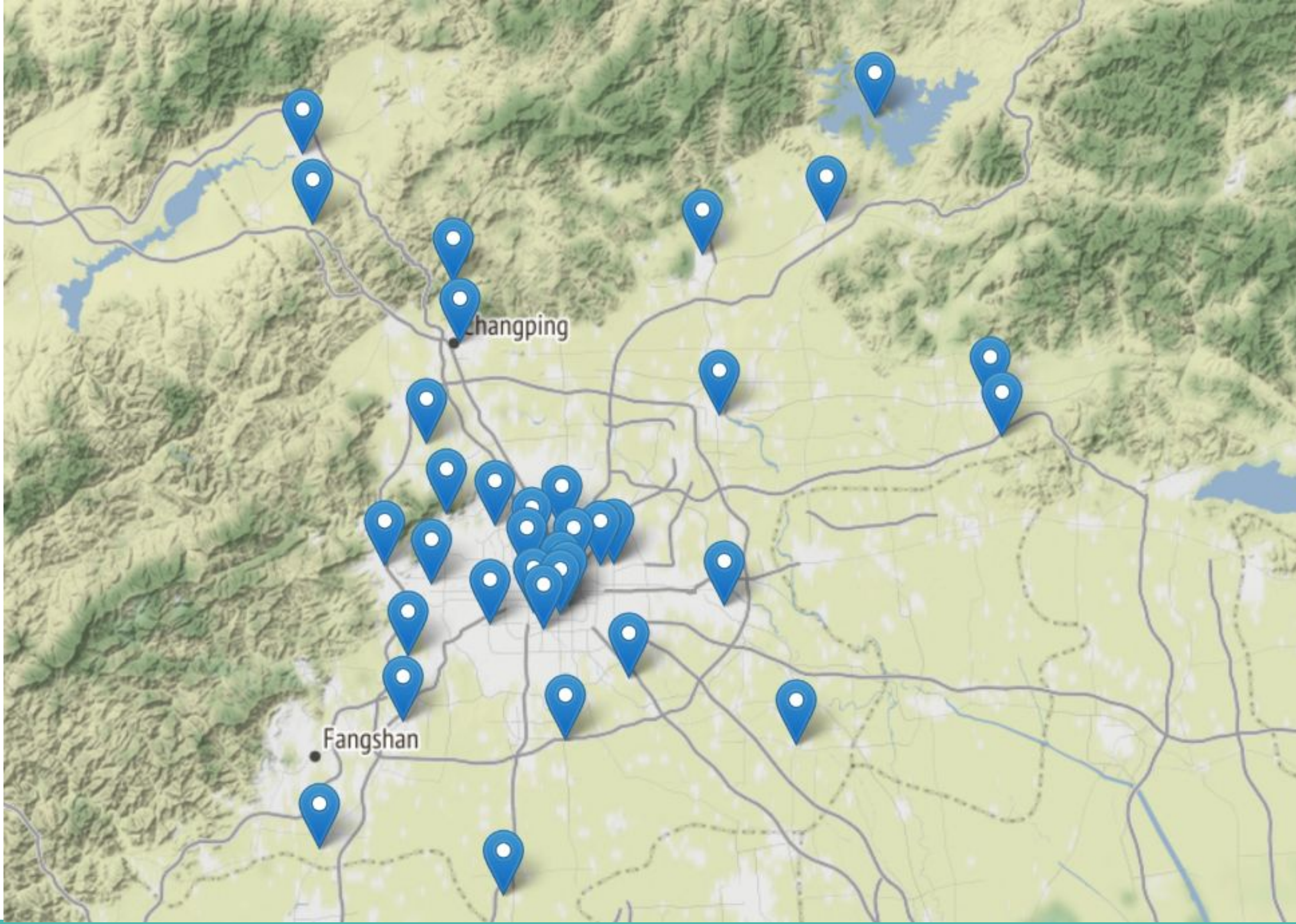
Classification



Reference

[1] Visualizing Learning rate vs Batch size

Discussion



Discussion

- Food(Data) for big guy(RNN) may not enough
  - Seems the data in this competion is a little too few for the models like RNN. [1]
  - Web Traffic Time Series Forecasting - data set range almost **10** years

Reference
[1] Seq2Seq with attention model and not good score, 92th solution

Lesson Learned...

About competition

- Do this as early as possible
- Data Preprocessing & Cleansing consumes up to **60 - 70%** of time
- Fancy is not always good (A non-convergent Model is a nightmare)

Other Resource

1. Kaggle Competition - Web Traffic Time Series Forecasting
2. Arturus/kaggle-web-traffic

Reference

- Géron, Aurélien. Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.", 2017.
- VanderPlas, Jake. Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.", 2016.
- Müller, Andreas C., and Sarah Guido. Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.", 2016.
- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
- Zaytar, Mohamed Akram, and C. E. El Amrani. "Sequence to sequence weather forecasting with long short term memory recurrent neural networks." Int J Comput Appl 143.11 (2016).

Thank You For Your Attention

Q&A