CFM Challenge

2018-07-09 Presentation

Cyrille Delabre

Introduction

Considerable dataset with *small* time series.
Ideal for **RNN model**.

- Feature Engineering
- Embeddings for Categorical Variables [(1),(2)]
- LSTM Networks
- ResNet *like* aggregation in Network [(3)]

Public score : 20.8828
Academic score 20.8711

(1) **Entity Embeddings of Categorical Variables (2016):** https://arxiv.org/abs/1604.06737
(2) **Meta-Prod2Vec - Product Embeddings Using Side-Information for Recommendation (2016):** https://arxiv.org/abs/1607.07326
(3) **Deep Residual Learning for Image Recognition (2015)** https://arxiv.org/abs/1512.03385

Feature Engineering

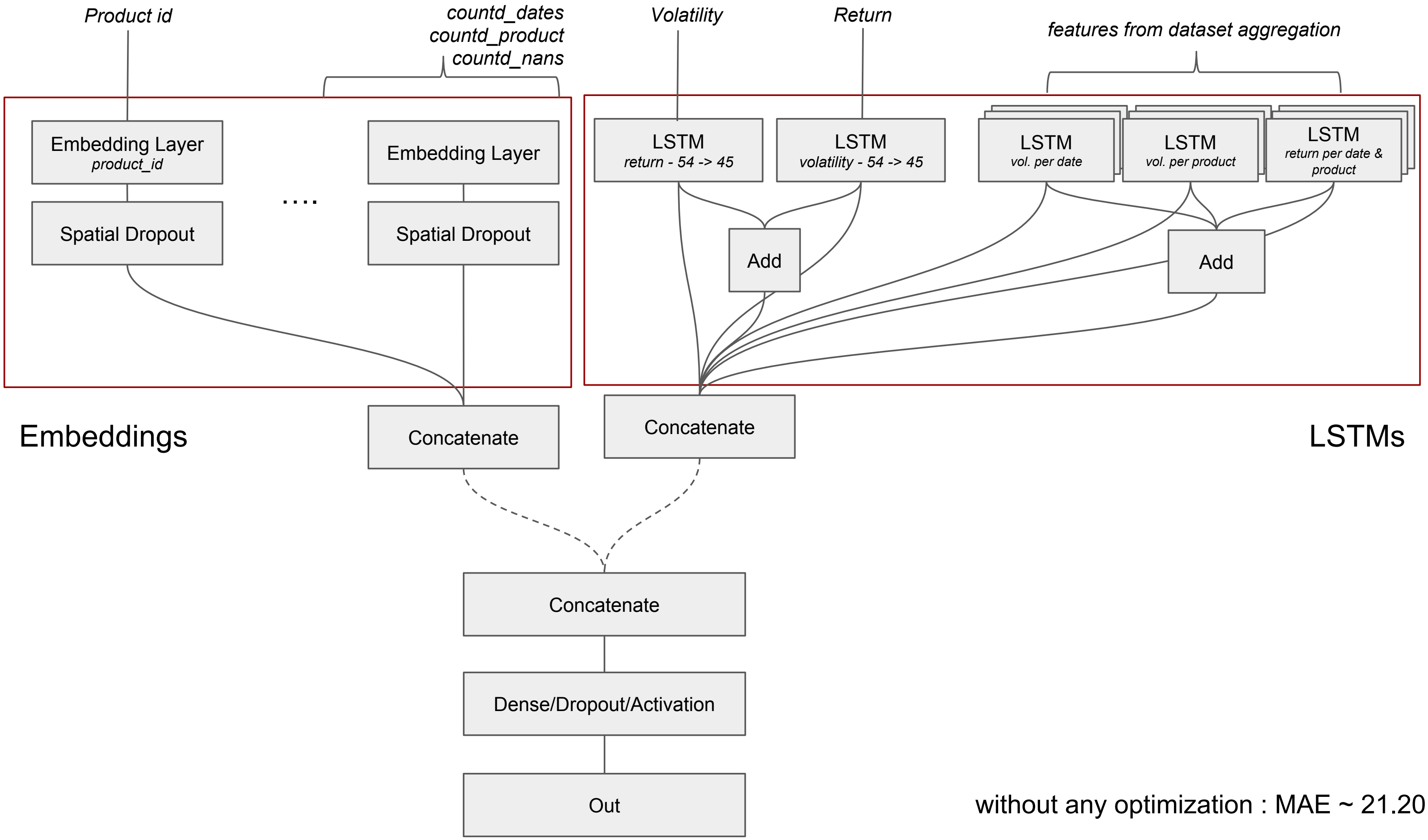
Target values are highly depends on date and product

- Date & Product Id over all dataset
  (aggregation for each col with *mean, std, median, distinct nan, …)*
- *Distinct product ids over date*
- *Distinct dates over product id*

At the end we have :

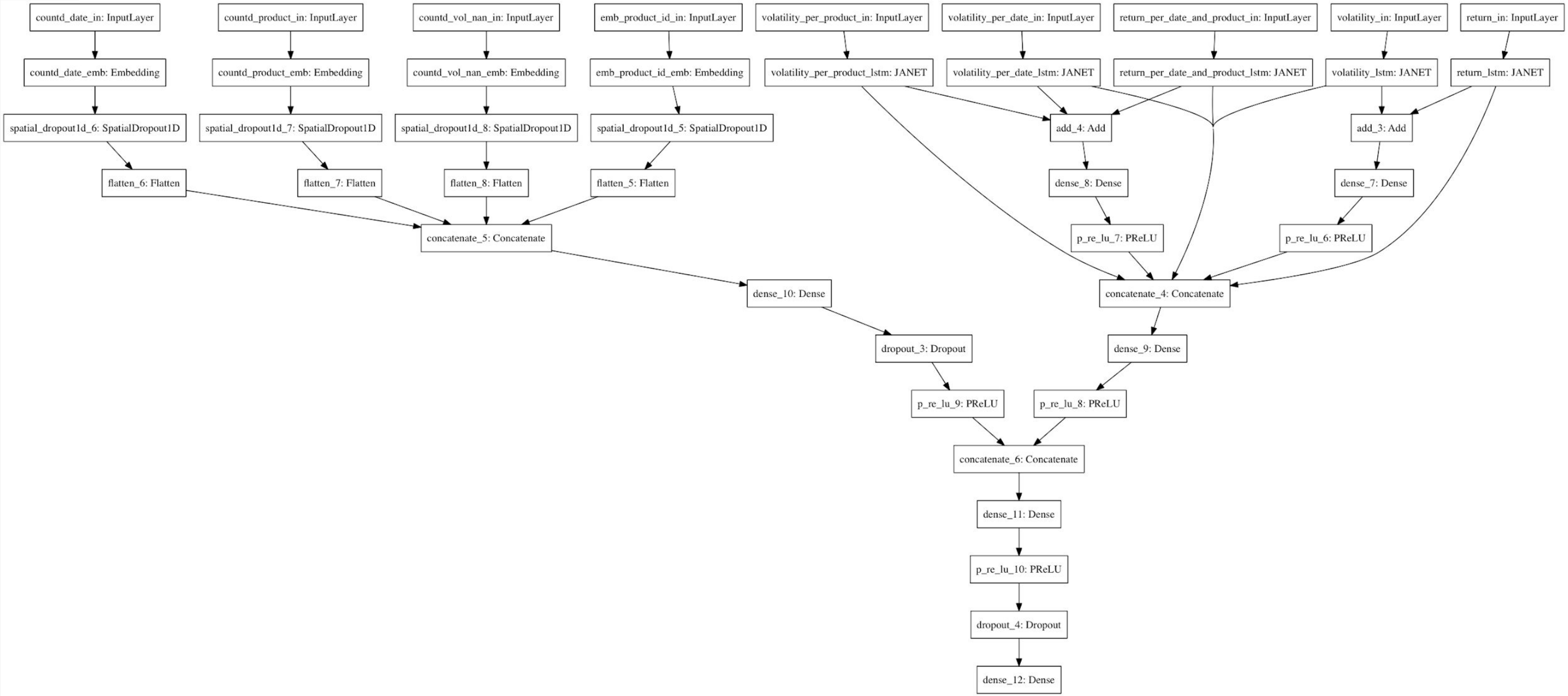
***Time Series Features** (volatility, return + volatility_per_dates_mean, volatility_per_product_mean, ...)*
***Categorical Features** (product_id, countd_product_id , countd_dates, countd_nans,...)*

NN Model



without any optimization : MAE ~ 21.20

Model Details



Architecture Tricks

- **Cyclic Learning Rate** [1]

  Quicker convergence by increasing the LR in a cyclical nature

  Increasing LR is an effective way of “escaping saddle points”

- **Others “regularization” techniques**

  Due to new features created by the aggregation over all the dataset, the network tends to overfit very quickly.

  - KFold with train/valid split per date (valid dates never seen as in test set)

  - Spatial Dropout on Embeddings (helped a lot)

  - Small Neural Net to reduce Overfitting

  - Reduce the size of the layers for engineered features vs input features

- **JANET Network** [2]

  The model uses only forget and context gates out of the 4 gates in a regular LSTM RNN.

  Better performance while using fewer parameters and less complicated gating structure.

- **Average of top Models at the end**

  Averaging top 10 models predictions (~20.95 -> 20.88)

(1) **Cyclical Learning Rates for Training Neural Networks (2015) :** https://arxiv.org/abs/1506.01186

(2) **The unreasonable effectiveness of the forget gate (2018)** : https://arxiv.org/abs/1804.04849

Other Tried Techniques

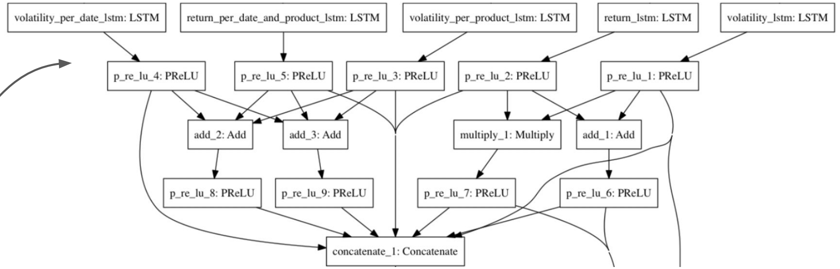
- **Attention Models** [(1)]
  Good results but too much time consuming (x5)

- **Temporal Convolution Nets (~ WaveNets)** [(2)]
  Worse results when tried, seems to be more adapted for longer time series

- **Averaging Weights Leads to Wider Optima and Better Generalization (SWA)**
  Worse results than 5 KFolds average but nice paper idea

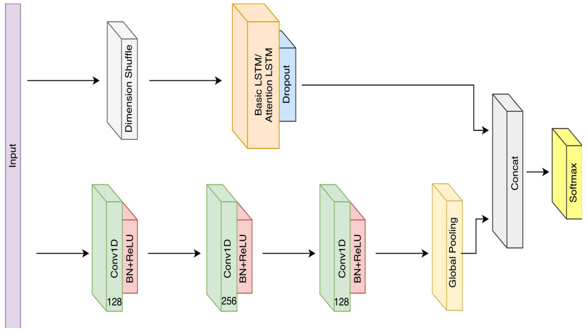
(1) **Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems (2015):** https://arxiv.org/abs/1512.08756
(2) **Temporal Convolutional Networks: A Unified Approach to Action Segmentation (2016):** https://arxiv.org/abs/1608.08242
(3) **SWA** : https://github.com/timgaripov/swa

Further Improvements



- **More ResNet *like* tricks**
  After RNN Net, addition/multiplication layer

- **Try FCN Networks** (1)
  More time consuming, but Conv. Layer should be able to get more insights about interaction between variables (eq. *average_volatility_per_date, current_volatility, current_return, …*)

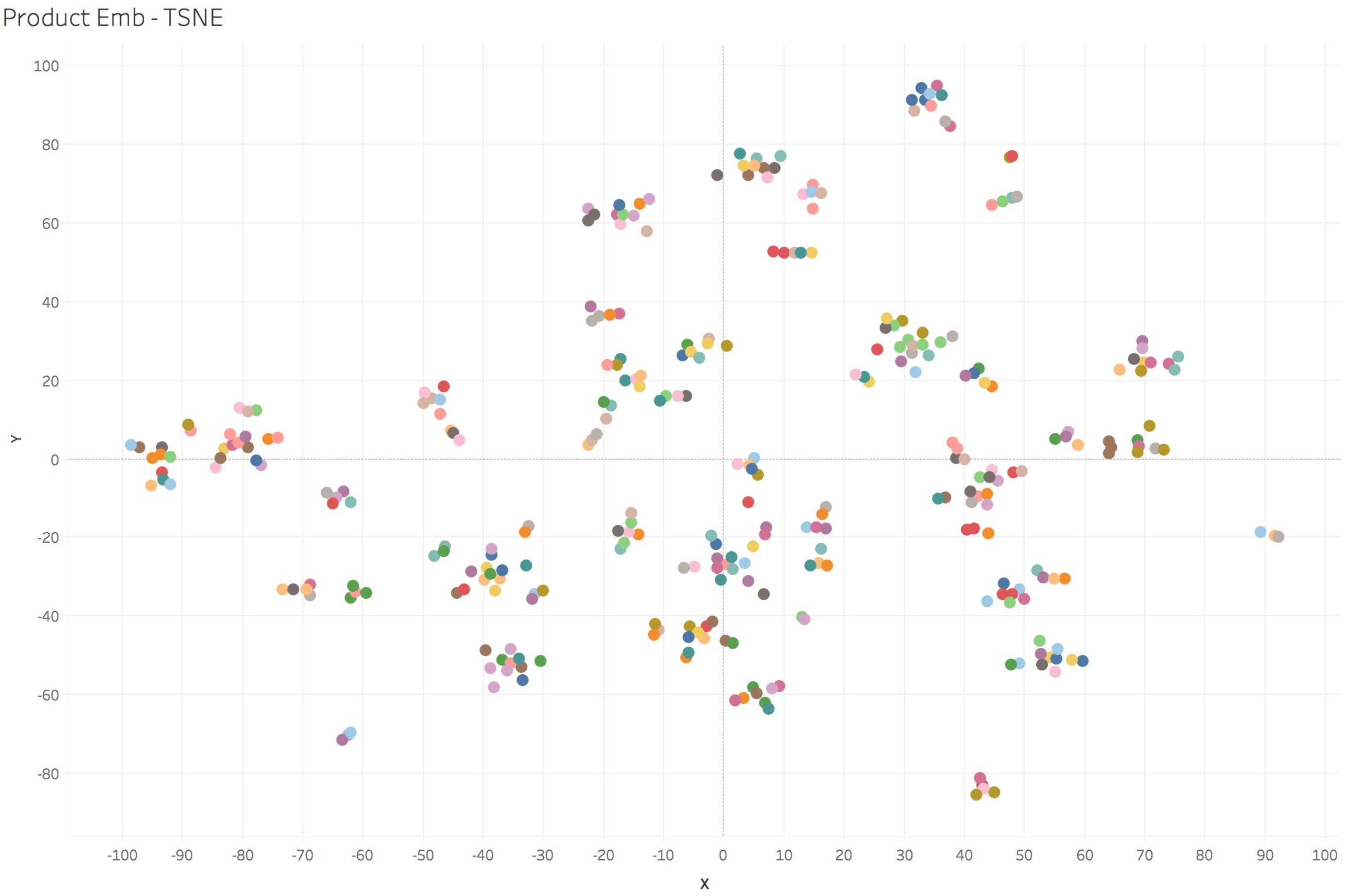


- **Time-series Extreme Event Forecasting with Neural Networks at Uber** (2)
  LSTM Autoencoder approach to create features from time serie.

- **Other Ideas**
  XGboost Models with Trained Product Id Embeddings
  More stacking techniques with distinct models

(1) **LSTM Fully Convolutional Networks for Time Series Classification (2017) :** https://arxiv.org/abs/1709.05206
(2) http://roseyu.com/time-series-workshop/submissions/TSW2017_paper_3.pdf

TSNE on Trained Product ID Embeddings



Thank You !