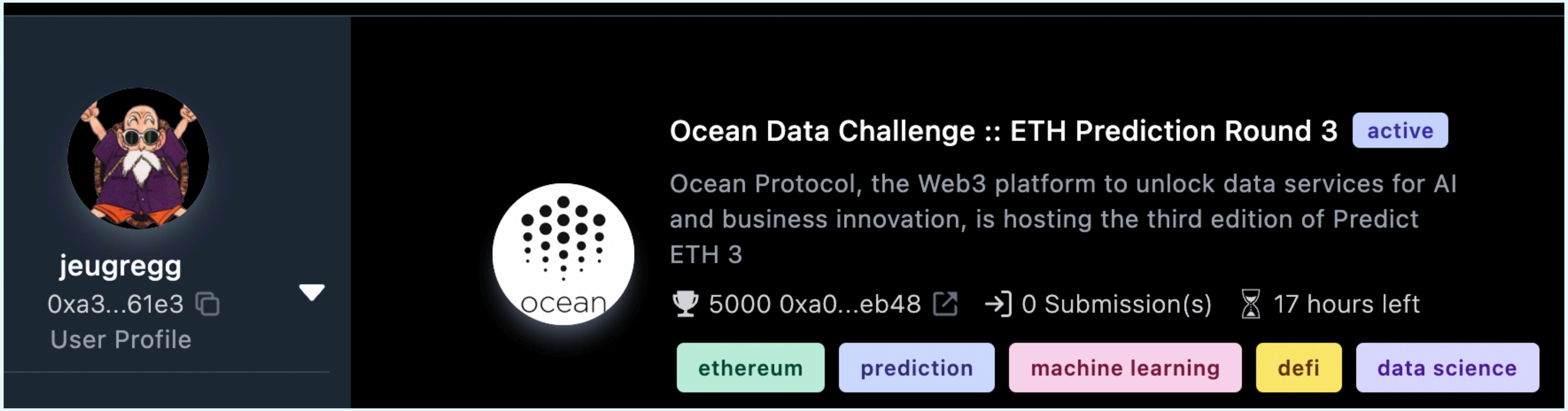
jeugregg - 2023-02-19

Machine Learning Scientist



PREDICT ETH PRICE WITH DIMENSIONALITY REDUCTION USING AUTOENCODER

OCEAN DATA CHALLENGE : ETH PREDICTION ROUND 3

CONTENTS

CONTEXT

- Ocean Data Challenge
    - predict ETH Close price from Binance exchange
        - Submission deadline: Sun Feb 19, 2023 at 23:59 UTC
        - Prediction at times: Mon Feb 20, 2023 at 1:00 UTC, 2:00, ..., 12:00
            - 12 predictions total
- Using all data we need like:
    - exchange price data
    - deFi data
    - on chain data
    - traditional economy data

THE QUESTION AND ITS INTERPRETATION

- Many data are available
  - additional contraints (on my side) : take only free data
  - use library : cctx, yfinance, openbb, …
- Many technics are possible
  - RNN model with all features available (> 100)
    - a lot correlation
  - Try to reduce dimension of data with autoencoder
    - useful without data exploration to select specific features
- Comparaison with classical LSTM

EXPLORATION : THE DATA

- CCTX data : for Price & Volume : OHLCV
  - for ETH, BTC & BNB
  - hourly data
  - Binance (exchange used by Judges!)
  - Kucoin to correct gap (missing data)
- Calculate several indicators with several time range 1h, 1day, 1week
  - Ichimoku (all indicators except lagging span)
  - VWAP (+ extra periods : 1 month, 3 months, 6 months, 1 year, all)
  - Higher high & Lower low
  - Chop & RSI 14 periods
- Other indicator (economy) (hourly) (yfinance)
  - DXY
  - US GOVERNMENT BONDS 5 YR YIELD
  - SP500

PROPORTIONAL / DIRECTLY DEPENDENT TO TOKEN PRICES (EXCEPT CHOP & RSI)

BUT FREE PAST DATA LIMITED TO LAST 2 YEARS

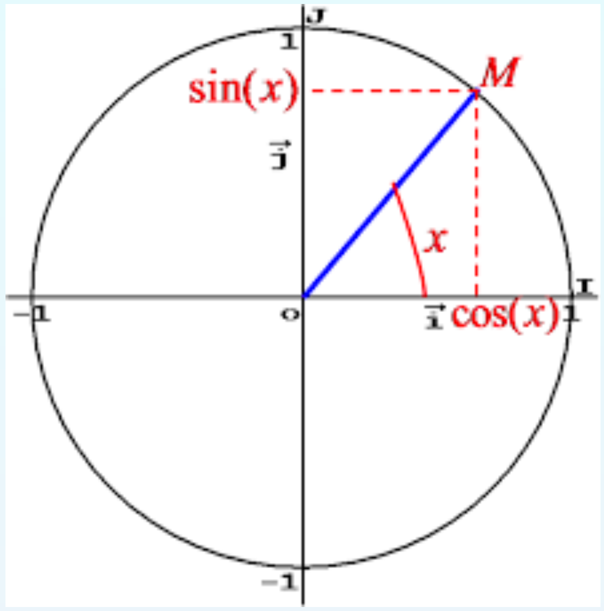
EXPLORATION

- On chain data (daily) with openbb (Glassnode & Messari Free API)
    - Circulating supply
    - Market Cap
    - Number of active wallets on BTC chain and ETH Main net
    - Approximation with cumulative volumes from exchange:
        - MVRV : Market Value / Realized Value and z-score
        - NUPL : Net Unrealized Profit/Loss
- Crypto Fear and Greed index (from alternative.me) (daily)
- Economy Calendar with important US events (daily)
        - inflation (PPI, CPI), Fed Interest Rate
        - estimate sentiment positive if
            - 1st flag : Consensus > Previous
            - 2nd flag : Consensus > Actual
            - Day off

EXPLORATION

EACH FEATURE TRANSFORMED IN 2 FEATURES COS, SIN

- Add temporal data
  - Hour of the Day : 0 -> 24
  - Day of the Week : 0 -> 6
  - Day of the Month : 0->28-31
  - Month of the year : 0->11



TO TAKE THE PERIODIC EFFECT INTO ACCOUNT

EXPLORATION

- Correlation matrix

STRATEGY & DATA PREPARATION

- Lags
  - Each lags = 1h
  - Number of past lags used for prediction : 48 (2 days : 48 hours)
    - to limit training CPU time

TO BE ABLE TO DO A LAST TRAIN AT 23:00 UTC

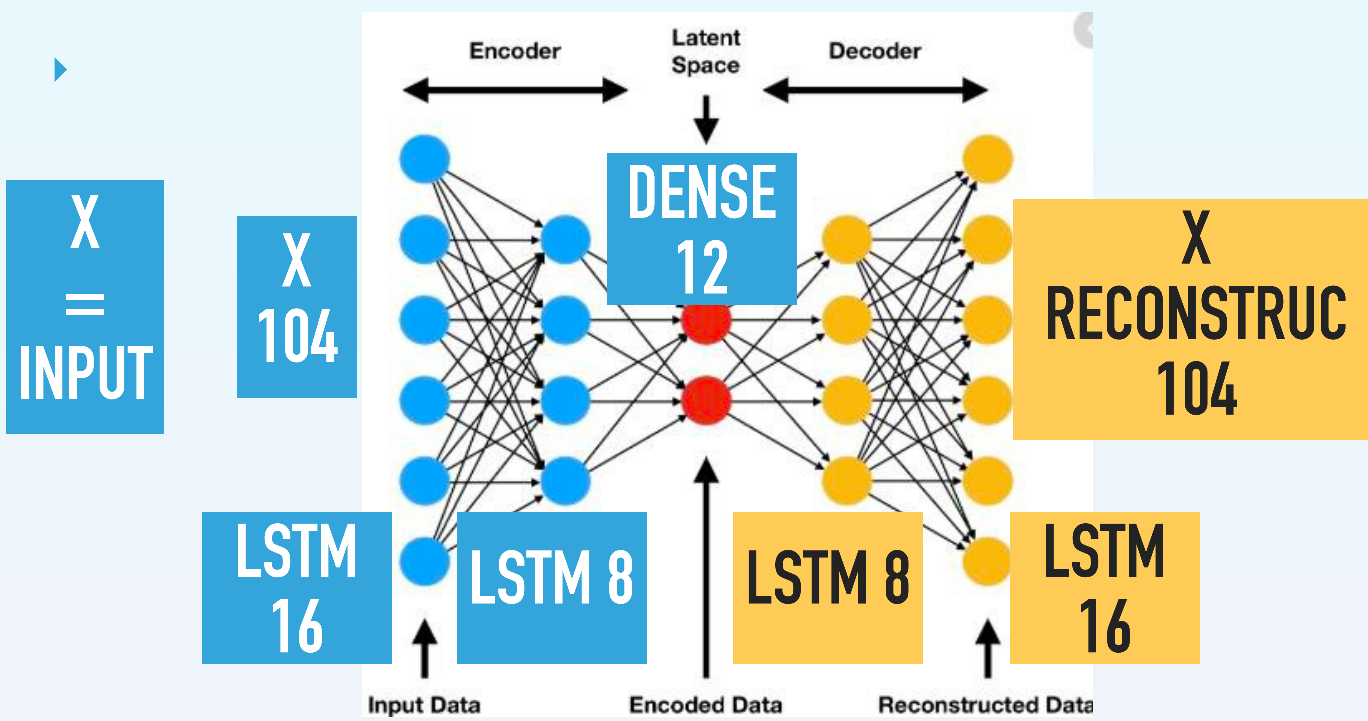
  - Number of future **Lags** to be predicted : 12 +1 = **13**
    - to be able to do a last training + prediction with data at 23:00 UTC
      - to predict 01:00…12:00 UTC
- First Normalisation
  - divide by ETH Close price

TO PREVENT BIG IMPACT OF GLOBAL TREND

    - at last lag before first prediction
      - example : t-n_lags … t-0   t +1…t+12
    - All features proportional to a price
- And apply for all features a classical StandardScaler from scikitlearn

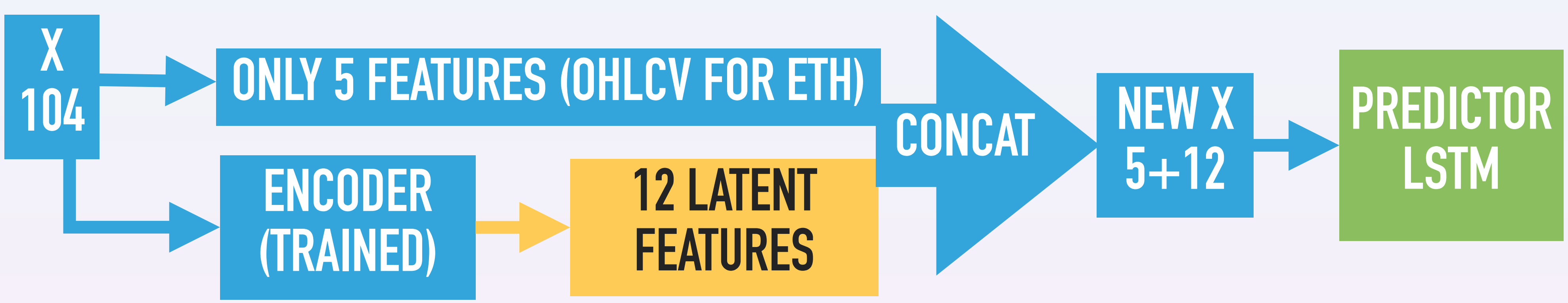
DATA REDUCTION WITH AUTO ENCODER

- A total of 104 Features **MODEL SIZE IS A PROBLEM**
  - Try to reduce the features dimension : 12 latent variables
    - auto encoder can do that !



1) TRAIN AUTOENCODER

2) TRAIN PREDICTOR WITH NEW X

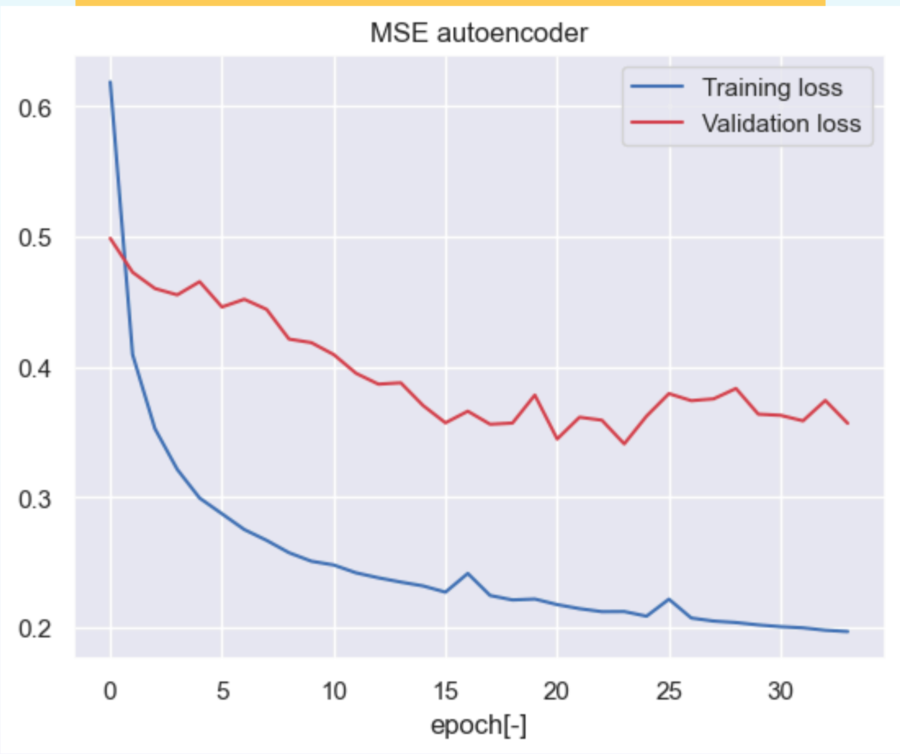


DATA REDUCTION WITH AUTO ENCODER : RESULTS

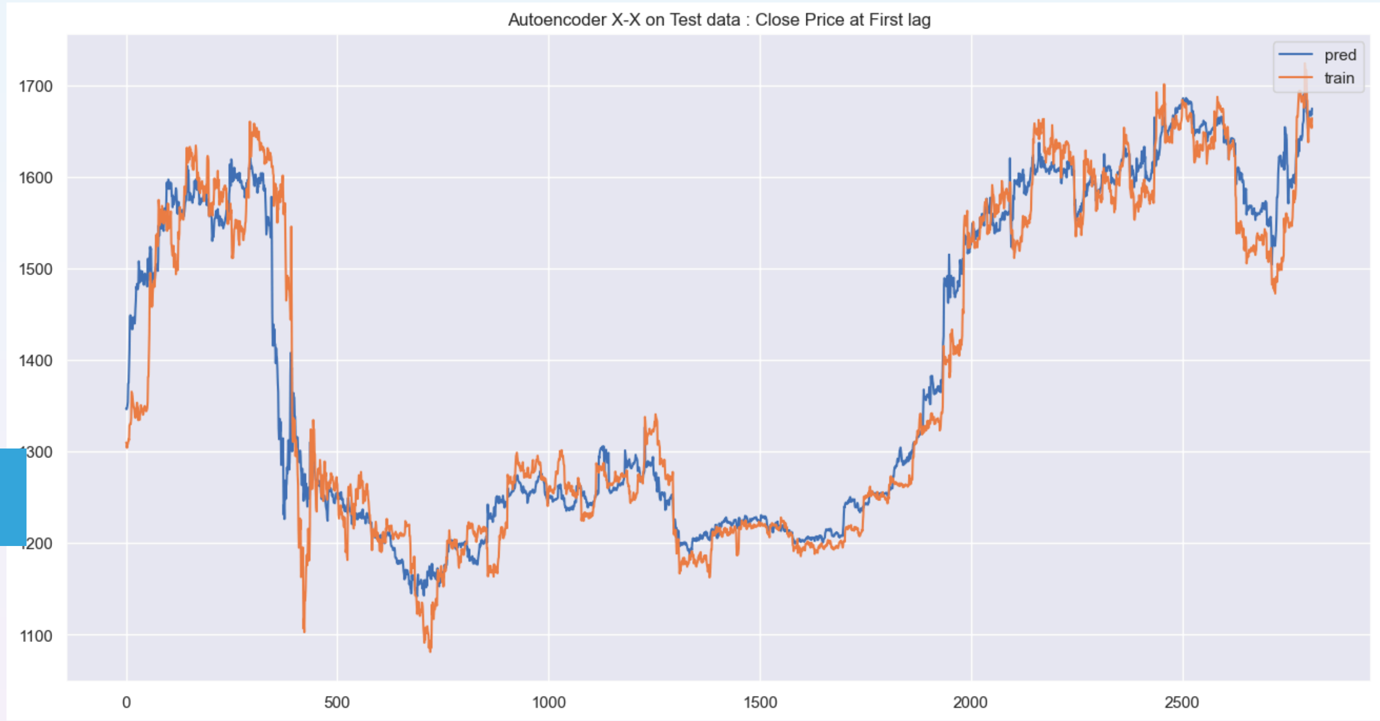
- A total of 104 Features reduce to 12 latent variables

MSE ERROR

NEED TO BE TUNED LONGER…



RECONSTRUCTION : AUTOENCODER(X) = X ?



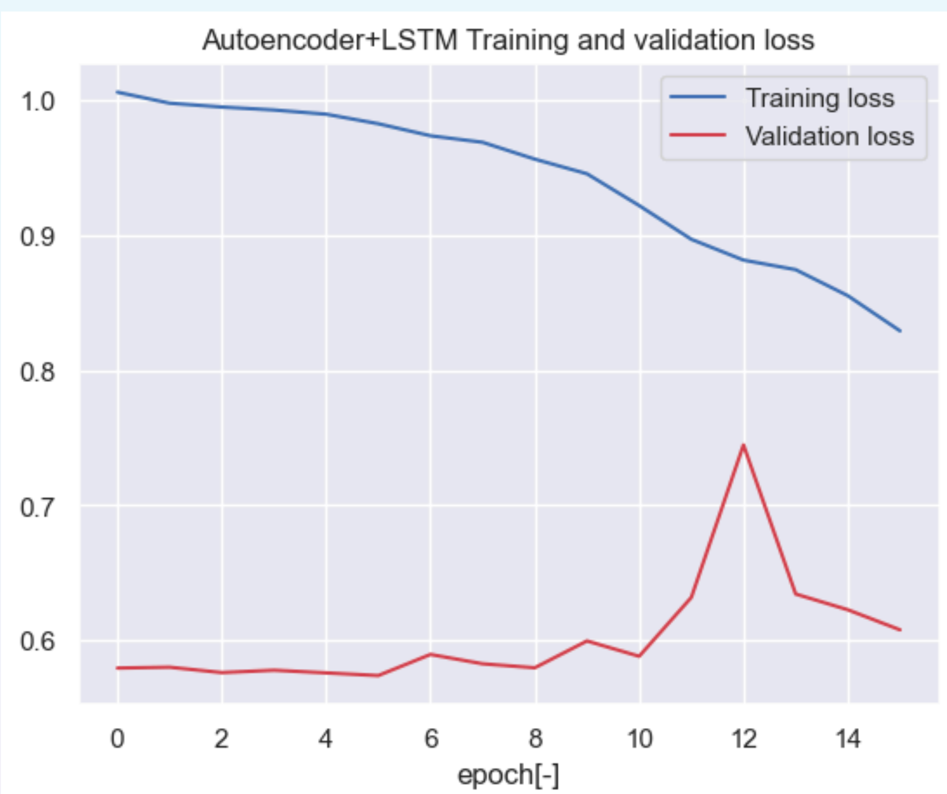
CLASSICAL LSTM : RESULTS

- A total of 104 Features
- Model : X ->

AUTOENCODER+LSTM : RESULTS

- A total of 5 Features [OHLCV] + X encoded (12) = 17

- Model : X[OHLCV]->
  Model : X ->

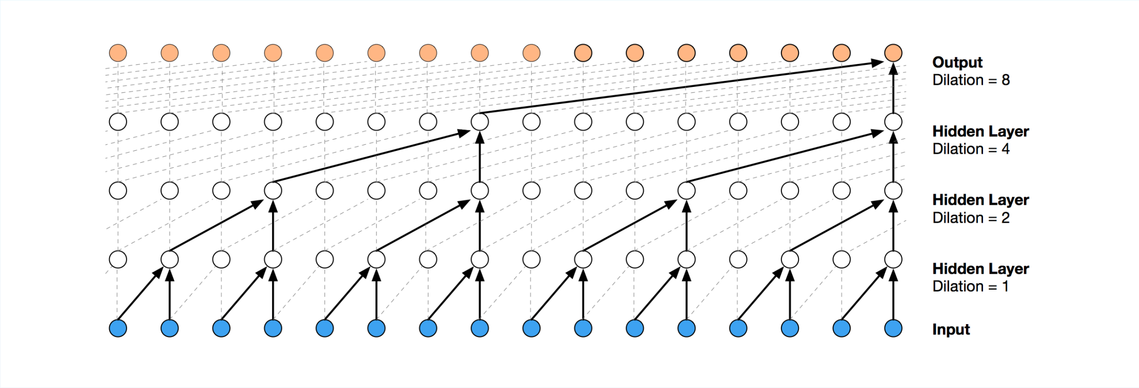


CONCLUSIONS

- Autoencoder useful without in deep exploration of data
  - need to be tuned more
- LSTM model
  - always have good results
  - but not optimal

AXES OF IMPROVEMENT

- More past data
  - limited to 2 years with FREE API
- Try with TCN model (Temporal Convolutional Model)
  - better performance
    - because parallel computing possible
    - compare to RNN model
  - to use more past lags



- Explore data in deep (lack fo time)
  - to find max past lags to use to predict next 12 hours
  - Features importances : find most useful data to reduce input space