Introduction to Financial Timeseries and Recent Advances in Supervised Learning for Financial Trading

Dr Paraskevi Nousi

DeepFinance

European Union
EPAnEK 2014-2020
OPERATIONAL PROGRAMME
COMPETITIVENESS
ENTREPRENEURSHIP
INNOVATION
ΕΣΠΑ
2014-2020

ARISTOTLE
UNIVERSITY
OF THESSALONIKI
SPEEDLAB
DATASCOUTING

Structure

- Time series and financial time series

- Supervised Learning
    - Classification & Regression

    - DL Architectures
        - MLP
        - CNN
        - LSTM

    - Recent Advances in Supervised Learning for Financial Trading

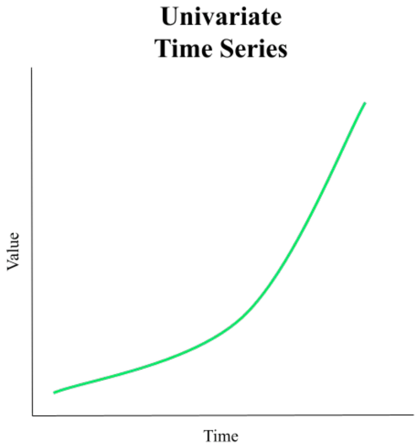
Financial Timeseries

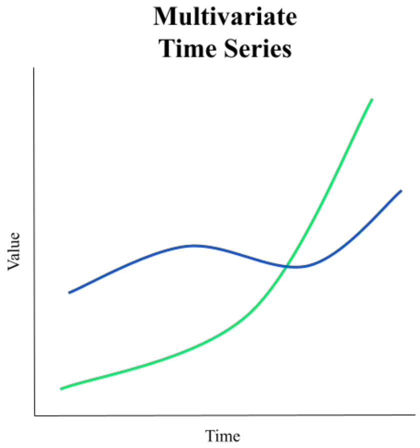
Timeseries

- A series of data points indexed in time order
- Commonly taken at a predefined frequency, i.e., equally spaced points in time
  - E.g. at second, minute, hourly intervals
- Tasks involving timeseries data:
  - Exploratory analysis, e.g., correlation to examine dependence, spectral analysis to examine frequency content and periodicities, decomposition
  - Curve fitting, i.e., finding a curve that best fits the data, such as exponential, linear etc.
  - Forecasting
  - Classification
- Stationarity is an important aspect in timeseries analysis
  - Do the statistics of a timeseries remain more or less the same throughout its duration?
  - This simplifies most aspects of the analysis

Timeseries

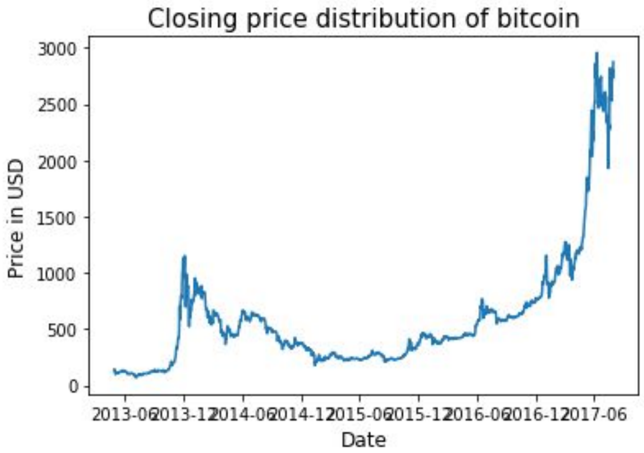
- Univariate
  - Single variable - 1 feature in terms of NNs
- Multivariate
  - Multiple variables - multiple features at each time step

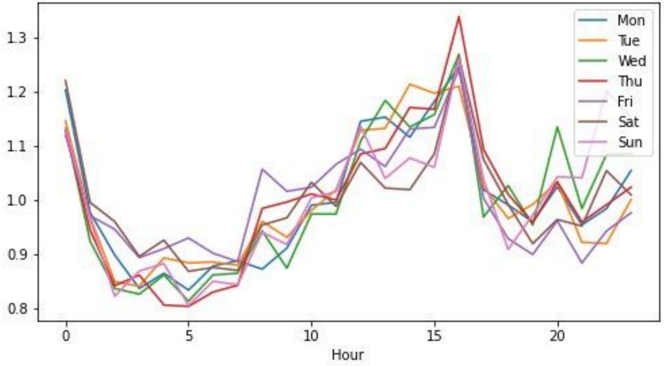




Financial Timeseries

- What's special about financial timeseries
  - Mostly non-stationary
  - Wide variation in the prices of different assets
- Predicting trends in financial markets allows for making profitable trades





Financial Timeseries

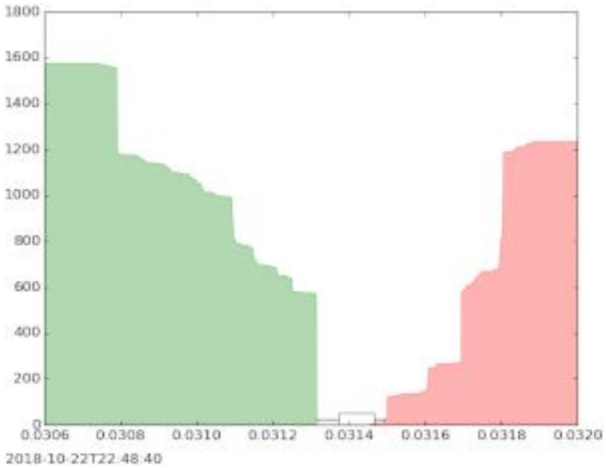
Limit Order Book Data

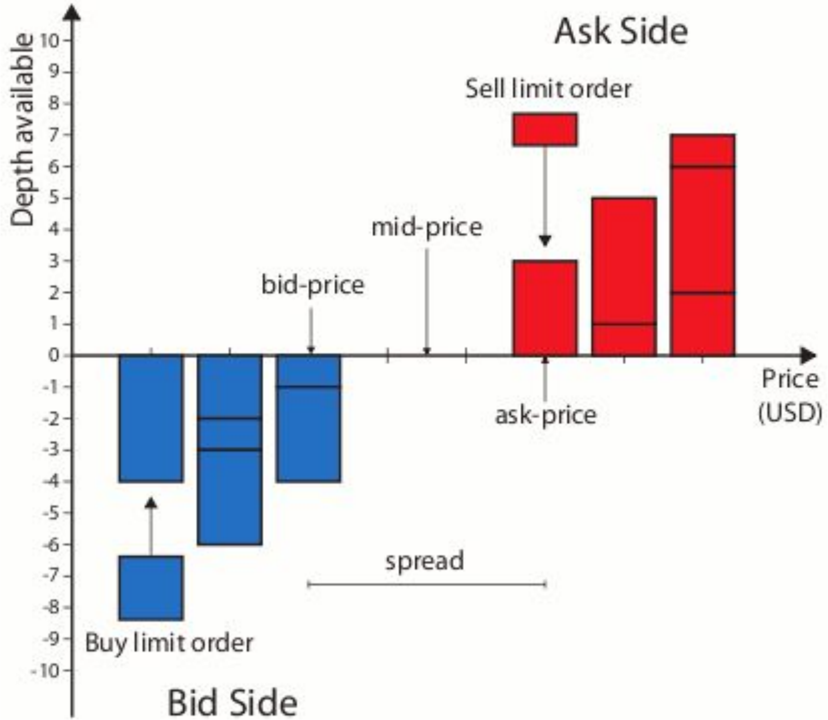
- A limit order is a type of order to buy or sell a specific number of shares within a set price
  - E.g., a sell limit order (ask) of $10 with volume 100 indicates that the seller wishes to sell the 100 shares for no less than $10 a piece
  - respectively , a buy limit order (bid) of $10 means that the buyer wishes to buy a specified amount of shares for no more than $10 each
- Two sides: ask and bid
  - Multivariate timeseries for each side: one for price, one for volume
  - Bid prices are sorted in descending order
  - Ask prices are sorted in ascending order
- Whenever a bid order price exceeds an ask order price an order is executed
- Time steps can be uneven

Financial Timeseries

Limit Order Book Data

- Tasks involving LOB data
  - Prediction of price trend
  - Regression of the future value of a metric
    - E.g., volatility
  - Anomaly detection, that can cause price jumps





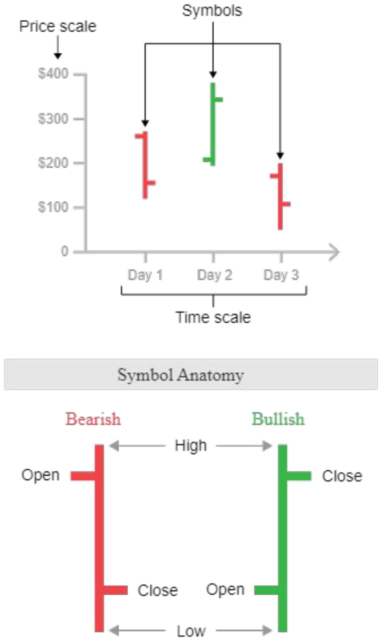
Financial Timeseries

OHLC Candle data

- Open-High-Low-Close candles are a subsampling technique
  - For a predefined duration (e.g., 1 minute or hour) gather price values and keep the following four:
    - Open: the first (time-wise) price
    - High: the highest price within this duration
    - Low: the lowest price within this duration
    - Close: the last (time-wise) price
  - Can be extracted from LOB data as well
- Preserves trend features
- Removes microstructure information

Financial Timeseries

OHLC Candle data



Financial Timeseries

Normalization

- Normalize data to be fed to neural networks
- Z-score normalization

$$X = \frac{x - \mu}{\sigma}$$

- Min-max normalization

$$X = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- But! Data is non-stationary

Advances in Supervised Learning

Regression & Classification

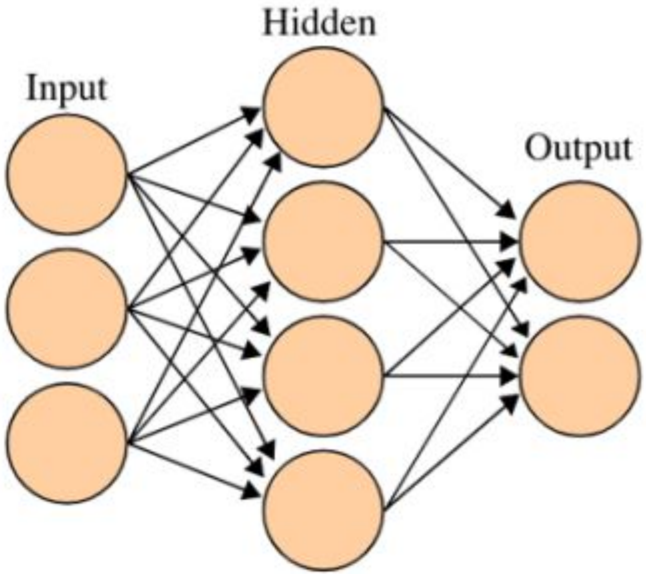
- Regression: predicting real-valued targets
  - Such as the price of an asset in the following time steps
    - Difficult to solve, models tend to degrade to a moving average filter

- Classification: categorization into predefined labels
  - Such as price trend, i.e., upwards, downwards or no change
  - Requires quantization of the real-valued target (i.e., thresholding)
  - Is easier to solve than actual price regression
  - But quantization:
    - Is handcrafted (typically)
    - Leads to imbalanced training sets

DL Architectures

Multilayer Perceptron

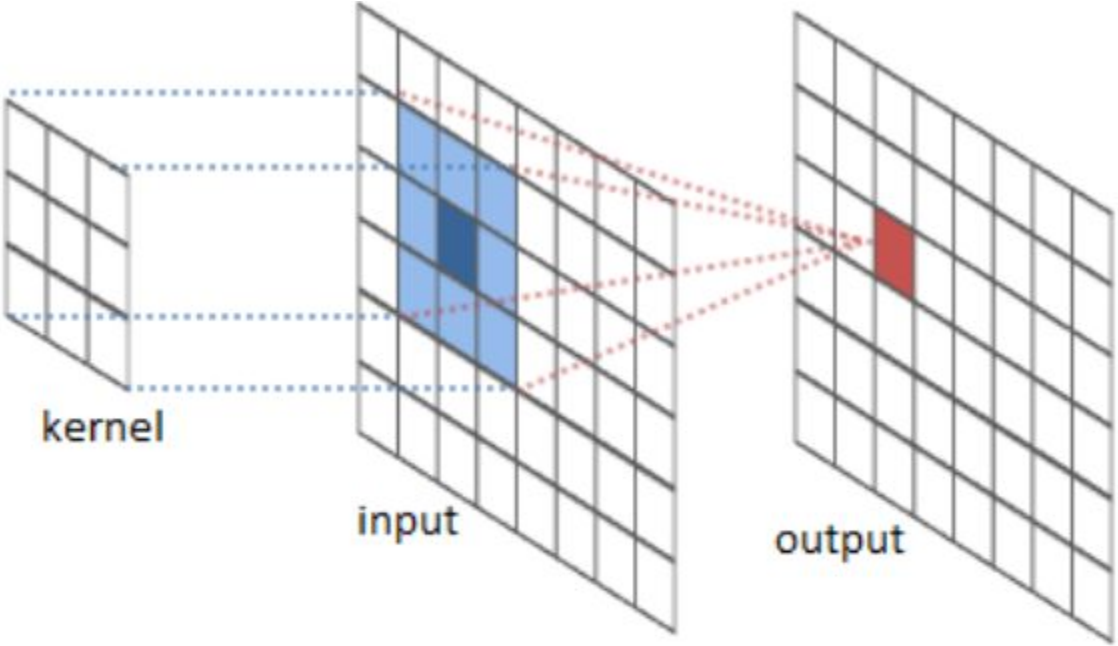
- The most basic form of neural network that allows to utilize all the inputs
  - Can overfit easily



DL Architectures

Convolutional Neural Networks

- Local connections, shared weights, based on the concept of convolution

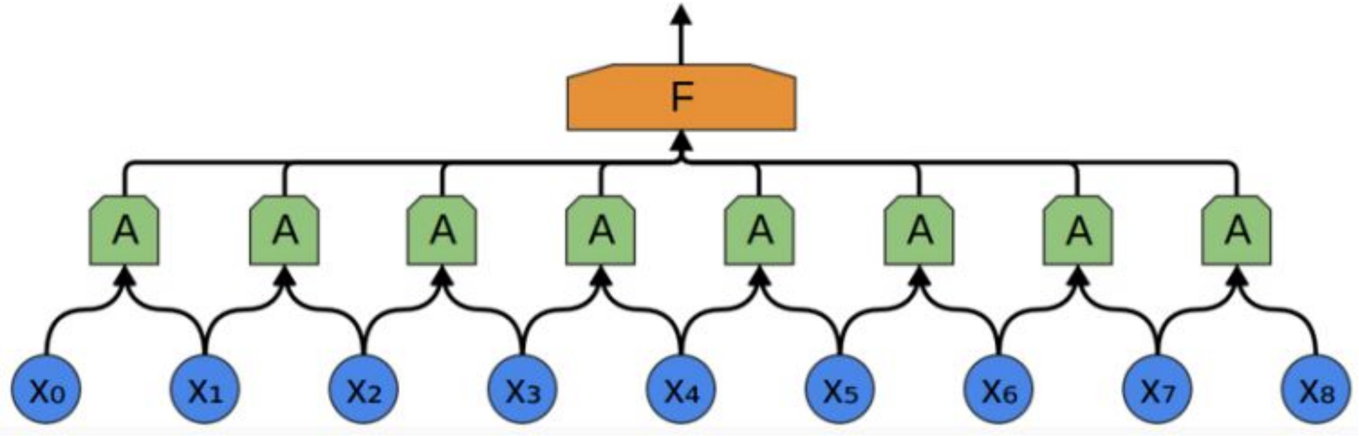


DL Architectures

Convolutional Neural Networks

- Input: univariate timeseries with window size 8

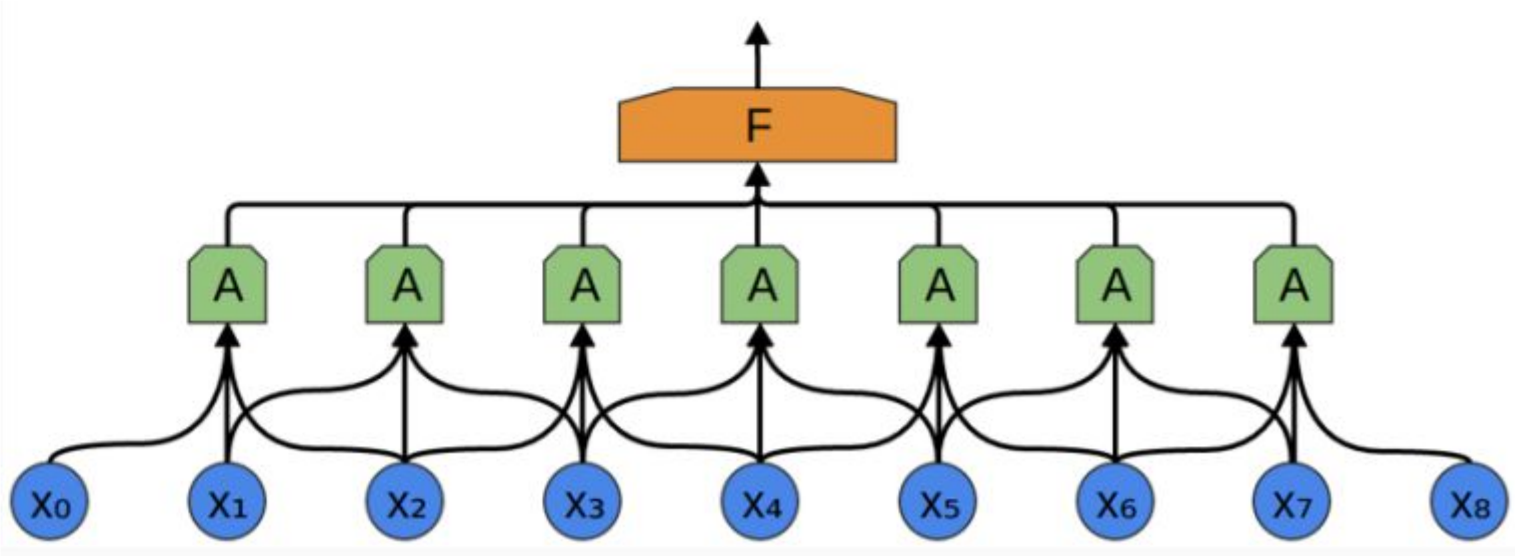
- Convolutional layer with k=2, s=2



DL Architectures

Convolutional Neural Networks

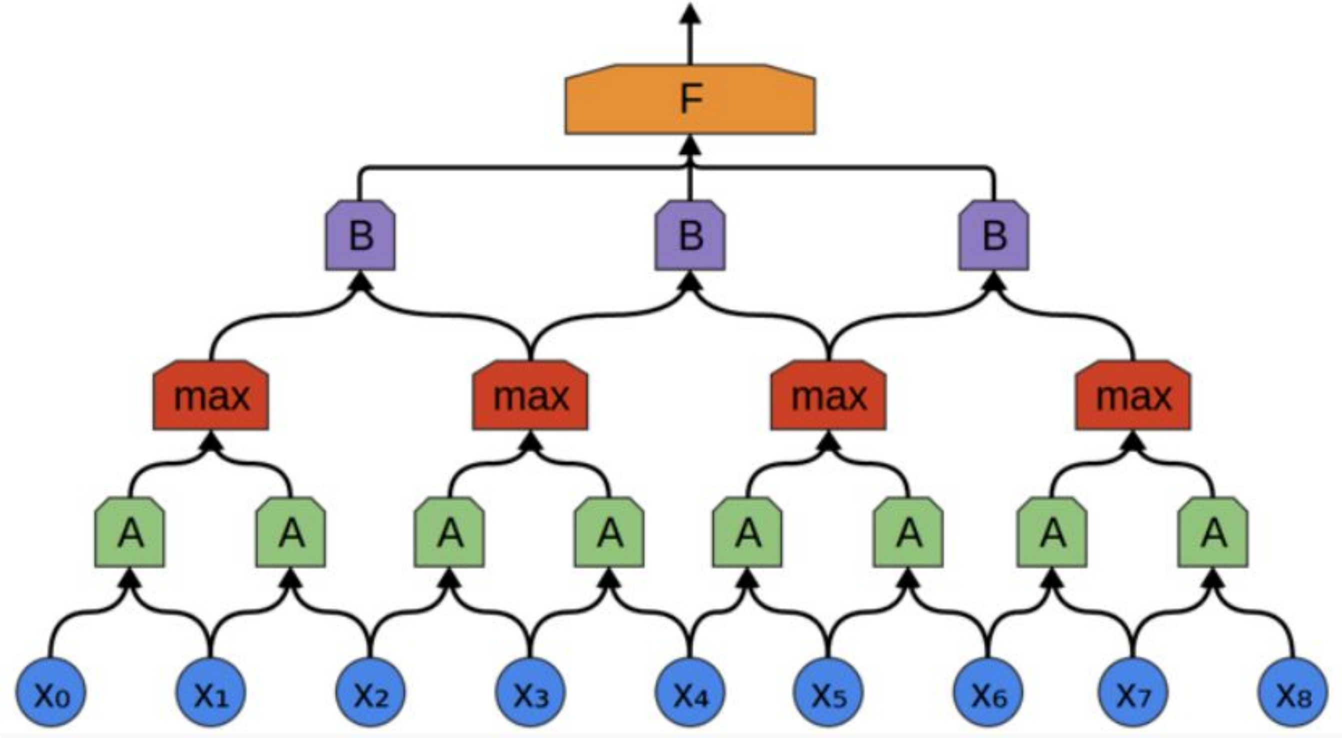
- Convolutional layer with k=3, s=2



DL Architectures

Convolutional Neural Networks

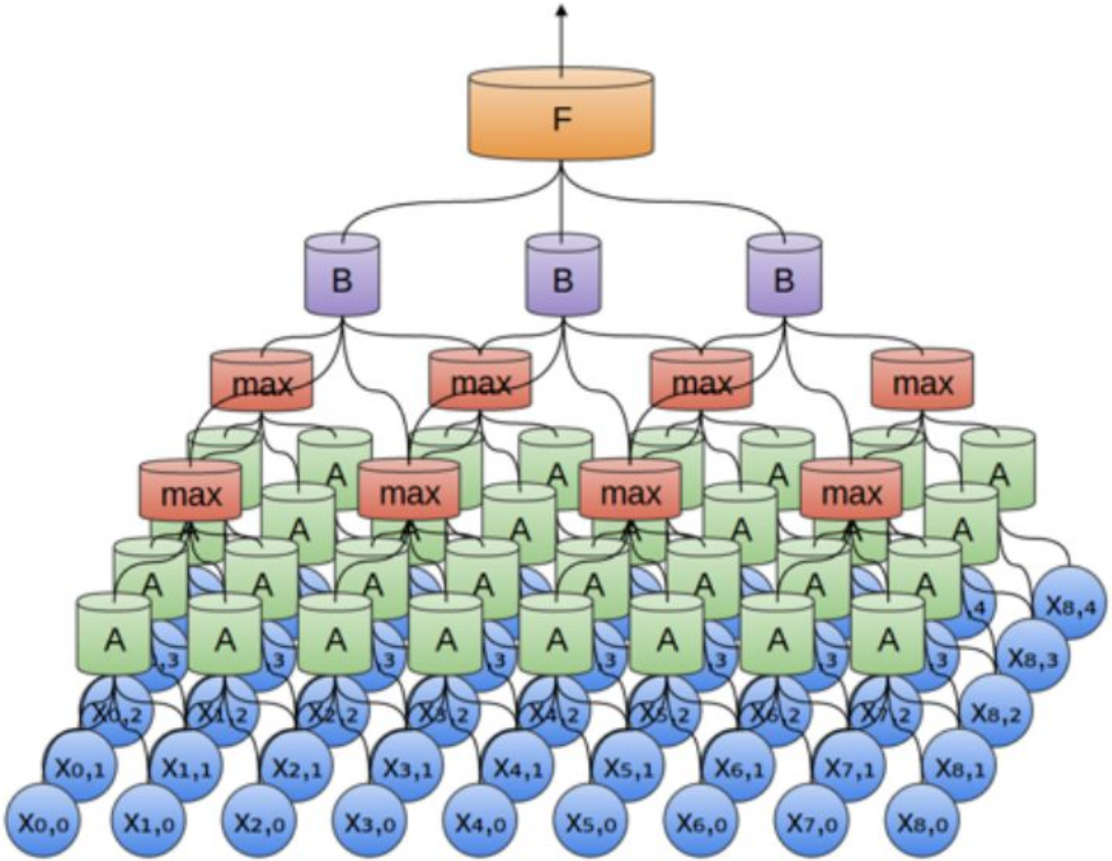
- Convolutional + max pooling layer with k=2, s=2



DL Architectures

Convolutional Neural Networks

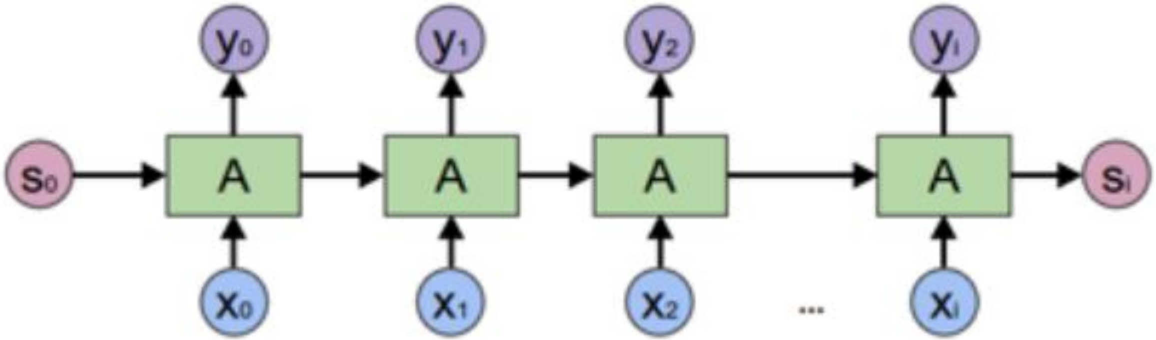
- Multivariate input
- Conv with k=2, s=1, d=4
- Max with k=2, s=2
- Conv with k=2, s=1

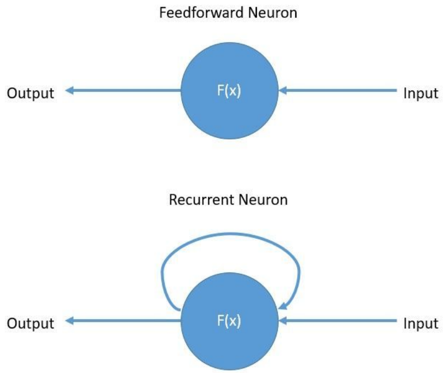


DL Architectures

Long Short-Term Memory Networks

- Recurrent neural networks specialize in analyzing time-series/temporal data
- Embedded memory to augment the ability to remember past events that have an effect on the future
- Prone to vanishing and exploding gradients

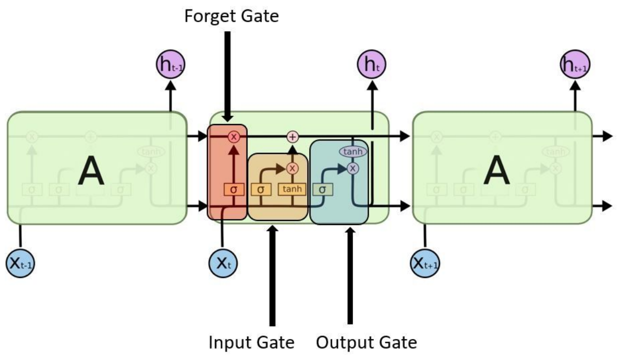




DL Architectures

Long Short-Term Memory Networks

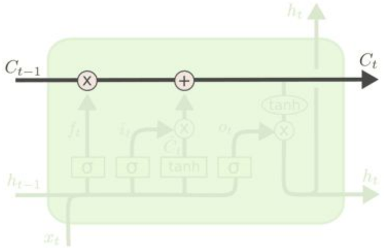
- LSTM: Architectural improvements upon the simple RNN to solve learning problems
- A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate
  - The cell remembers values over arbitrary time intervals
  - The three gates regulate the flow of information in and out of the cell



DL Architectures

Long Short-Term Memory Networks

- In the first step the forget gate looks at h_{t-1} and x_t to compute the output f_t which is a number between 0 and 1
- This is multiplied by the cell state C_{t-1} and yield the cell to either forget everything or keep the information
    - E.g., a value of 0.5 means that the cell forgets 50% of its information
- In the next step the input gate is computing the update for the cell by first multiplying the outputs i_t and $\tilde{C}_t$ and then adding this output to the input $C_{t-1}*f_t$
- Finally the output value is computed by multiplying o_t with the tanh of the result of the previous step



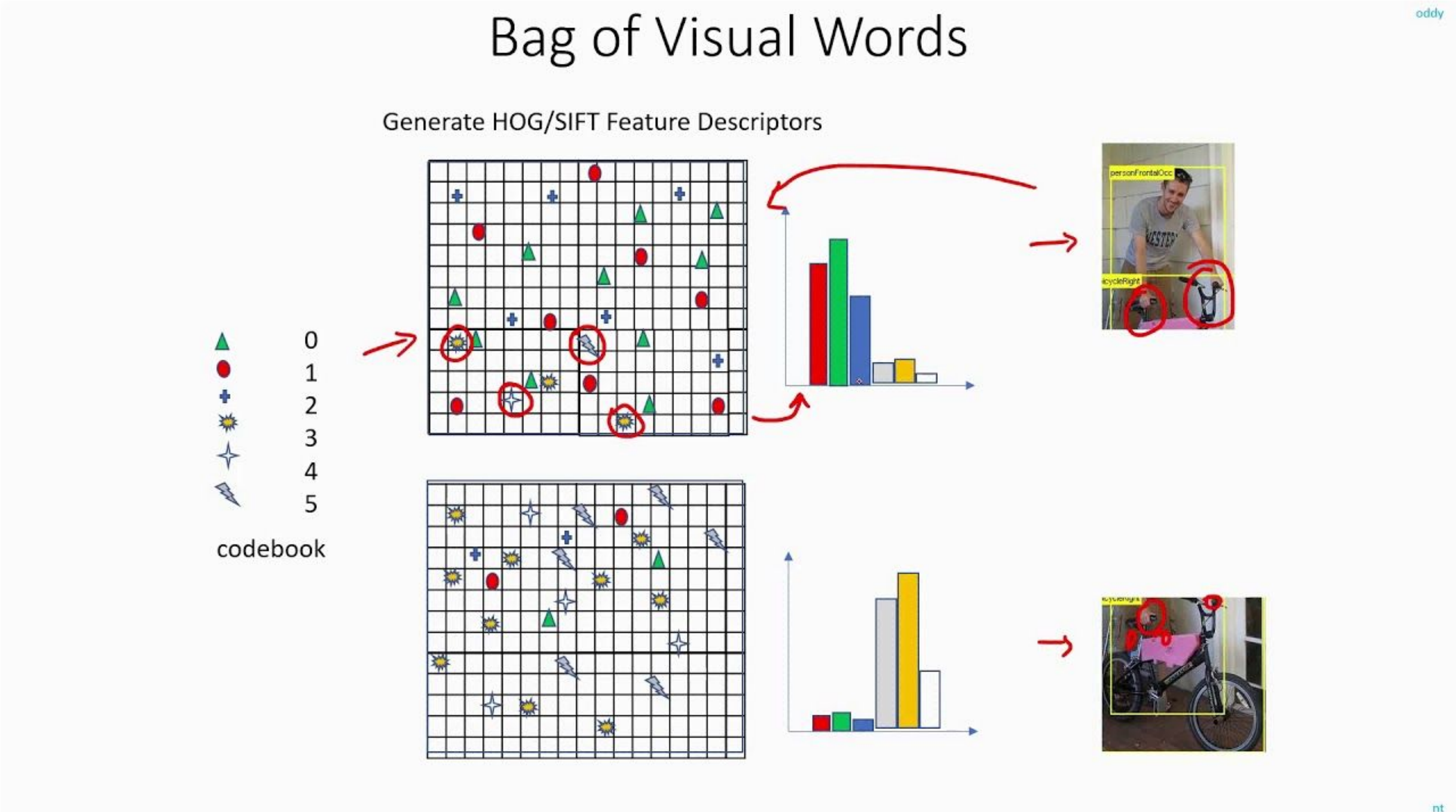
Recent Advances

Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data

- Financial data tend to be high-dimensionality, velocity and variety
- The Bag-of-Features model tackles these issues
  - Extracting histograms of data based on learned codewords
- Novel differentiable Temporal BoF formulation that can be used in combination with any NN
  - Adaptive scaling mechanism
  - Logistic kernel instead of Gaussian-based density estimation

Recent Advances

Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data



Recent Advances

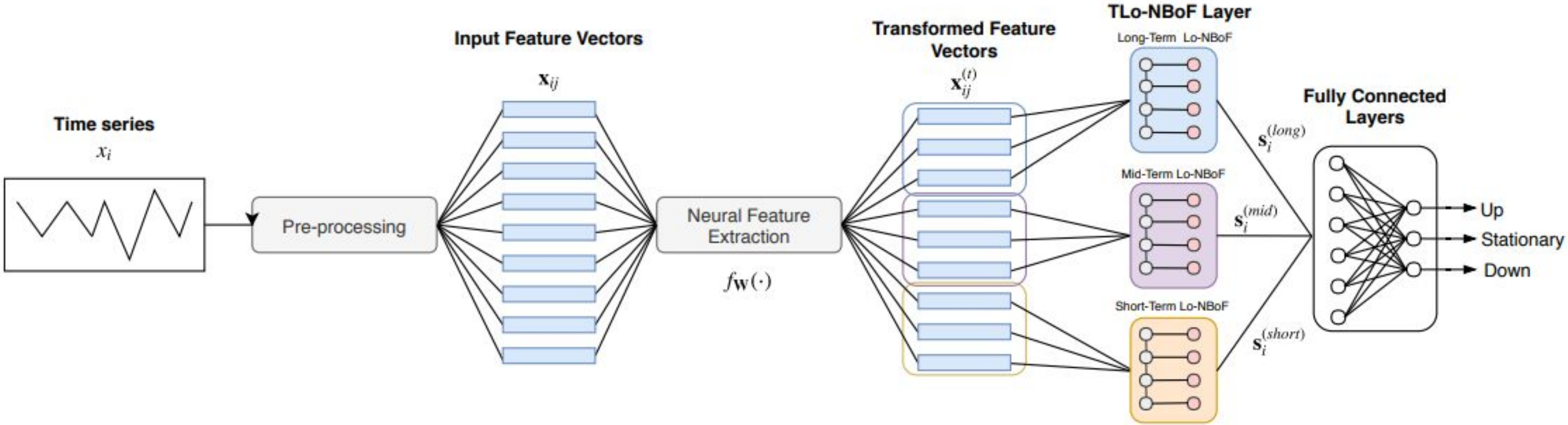
Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data

- Step 0 is to define codewords
  - Typically k-means or selection
  - In this case: convolutional weights that are learnable
- Step 1 is to extract high level features for each time step
  - Using for example a CNN
- Step 2 is to compute similarities between features and codewords
  - Using a sigmoid kernel, fully differentiable
- Step 3 is to normalize similarities
- Step 4 is to aggregate to extract histograms

Recent Advances

Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data

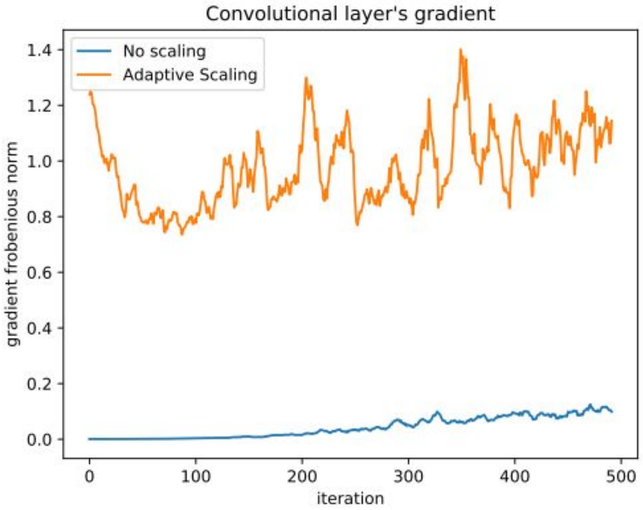
- Three temporal horizons, long-, mid-, short- term

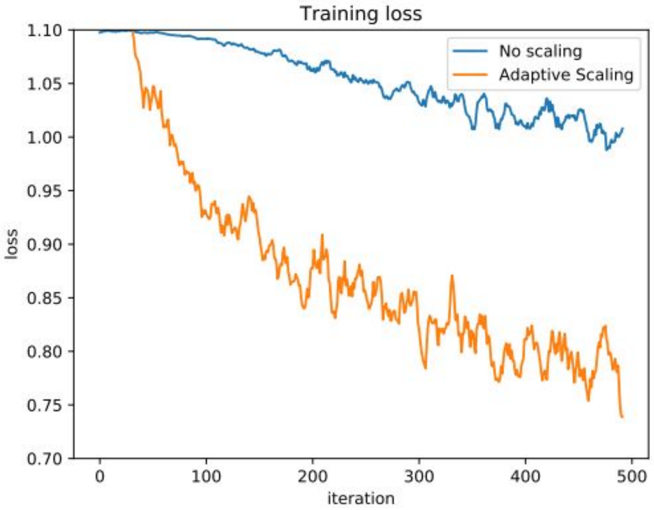


Recent Advances

Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data

- The normalization in steps 3, 4 can prohibit the smooth flow of information both in the forward and backward pass
- Adaptive scaling of learned similarities and histograms leads to faster convergence





Recent Advances

Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data

- Applied to Limit Order Book data

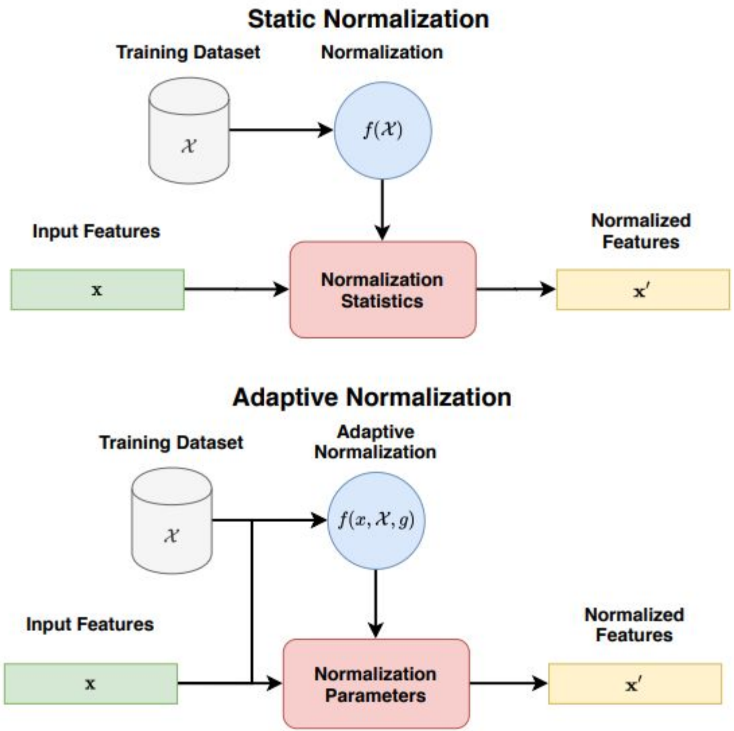
Deep Features	Temp. Model.	Kernel Param.	Adaptive Scaling	Macro-F1	Cohen's κ
-	-	-	✓*	42.66 ± 0.28	0.1847 ± 0.0026
✓	-	-	✓*	46.77 ± 1.53	0.2219 ± 0.0230
-	✓	-	✓*	50.14 ± 1.36	0.2686 ± 0.0179
✓	✓	-	✓*	51.65 ± 0.99	0.2783 ± 0.0109
✓	✓	✓	-	50.65 ± 0.71	0.2603 ± 0.0119
✓	✓	✓	✓*	53.48 ± 0.45	0.3013 ± 0.0075
✓	✓	✓	✓	$\mathbf{53.54 \pm 0.24}$	$\mathbf{0.3031 \pm 0.0066}$

(✓* refers to using the scaling parameters $c_s = N_K$ and $c_u = E[N_i]$, but not adjusting them during the training process)

Recent Advances

Forecasting financial time series using robust deep adaptive input normalization

- Learnable adaptive normalization method
  - Can learn to identify the distribution from which the input data were generated and then apply the most appropriate normalization scheme
  - Operates on a sliding window over the time series allowing for overcoming nonstationary issues



Recent Advances

Forecasting financial time series using robust deep adaptive input normalization

- First shifting and then scaling data $\mathbf{x}_s = (\mathbf{x} - \boldsymbol{\alpha}) \odot \boldsymbol{\beta}$
- Equivalent to static sample-based z-score normalization when

$$\boldsymbol{\alpha} = E[x] \text{ and } \boldsymbol{\beta} = E[(x - E[x])^2]^{-1}$$

- Since data is very non-stationary using static values is not optimal
- Instead we learn shift and scale values based on our input data

$$\mathbf{x}' = (\mathbf{x} - \boldsymbol{\alpha}(x)) \odot \boldsymbol{\beta}(x)$$

- Compute summary representation $\mathbf{s}_\alpha(x) = \frac{1}{L}\sum_{i=1}^{L} \mathbf{x}_i \in \mathbb{R}^d$ then shift by $\boldsymbol{\alpha}(x) = \mathbf{W}_\alpha \mathbf{s}_\alpha(x) + \mathbf{b}_\alpha \in \mathbb{R}^d$
- Compute standard deviation of features and then scale by $\boldsymbol{\beta}(x) = (\mathbf{W}_\beta \mathbf{s}_\beta(x) + \mathbf{b}_\beta)^{-1} \in \mathbb{R}^d$

- Finally attention-like gating:

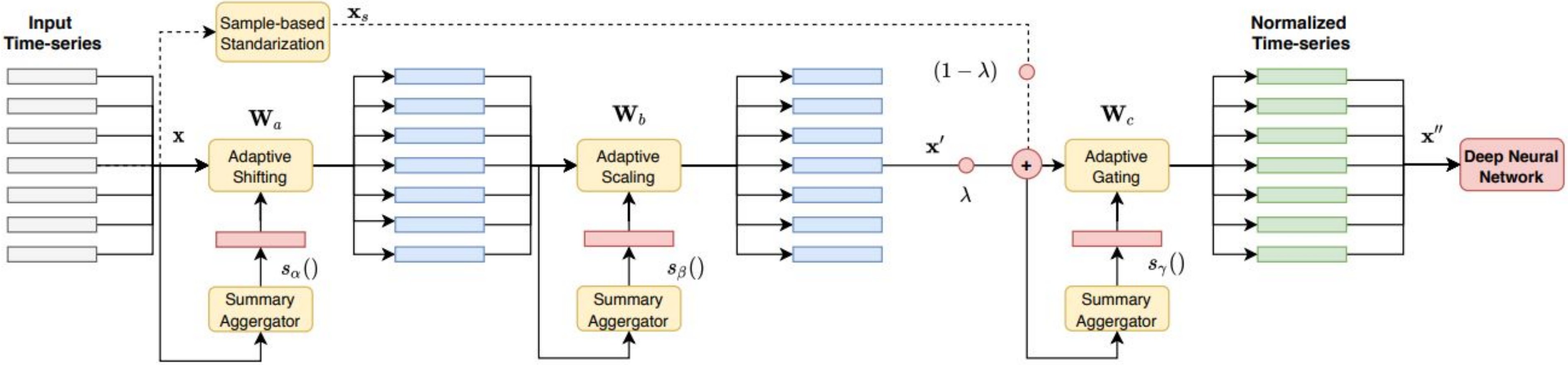
$$\mathbf{s}_\gamma(x) = \frac{1}{L}\sum_{i=1}^{L} \mathbf{x}'_i \in \mathbb{R}^d \qquad \boldsymbol{\gamma}(x) = \mathrm{sigm}(\mathbf{W}_\gamma \mathbf{s}_\gamma(x) + \mathbf{b}_\gamma) \in \mathbb{R}^d$$

$$\mathbf{x}'' = ((\mathbf{x} - \boldsymbol{\alpha}(x) \odot \boldsymbol{\beta}(x)) \odot \boldsymbol{\gamma}(x)$$

Recent Advances

Forecasting financial time series using robust deep adaptive input normalization

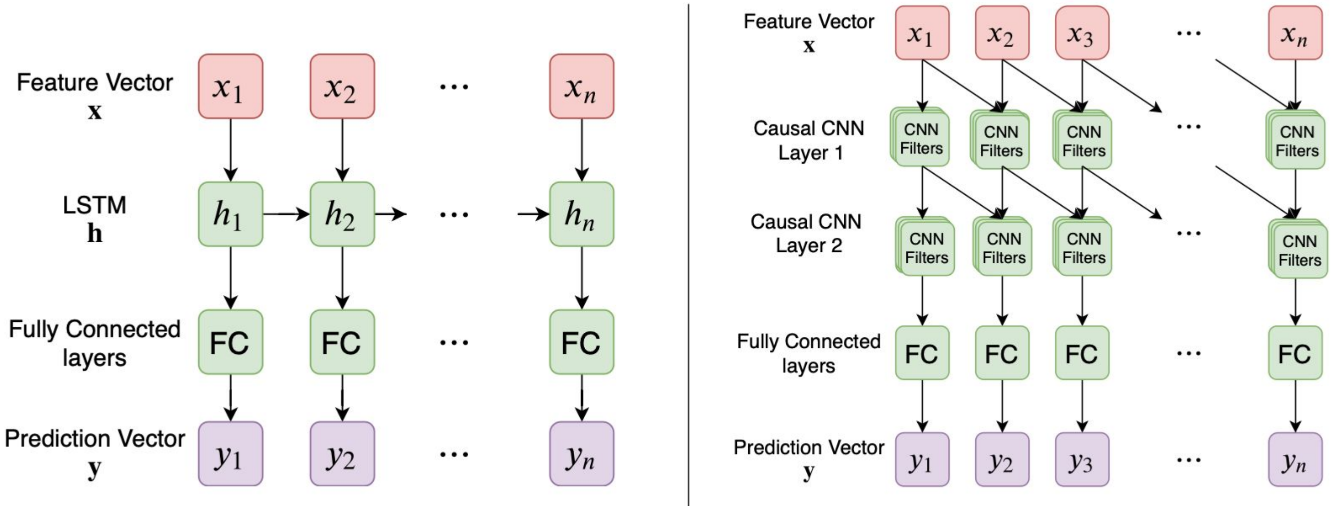
- First step before any NN architecture



Recent Advances

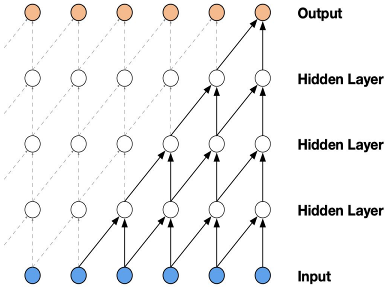
Transferring trading strategy knowledge to deep learning models

- Trading strategies are used to manage investments in the market
- Sometimes based on human emotions/intuition/decisions
- Can DL models learn to - not just blindly mimic - but extract knowledge about the strategy without knowing what it is?



(a) LSTM

(b) Causal CNN



Causal Convolution

Recent Advances

Transferring trading strategy knowledge to deep learning models

- Proposed features from OHLC data
- Handcrafted features that are stationary

1. $p_c(t) - p_c(t-1)$
2. $p_h(t) - p_h(t-1)$
3. $p_l(t) - p_l(t-1)$
4. $\log(p_h(t)) - \log(p_c(t))$
5. $\log(p_c(t)) - \log(p_l(t))$
6. $\log(p_c(t)) - \log(p_c(t-1))$

Strategy	Input Features	Model	Train				Test			
			Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
A	Proposed	LSTM	**0.998**	**0.997**	**0.998**	**0.997**	**0.995**	**0.977**	**0.976**	**0.976**
		CNN	0.994	0.988	0.990	0.989	0.988	0.943	0.932	0.937
	OHLC	LSTM	0.989	0.967	0.965	0.966	0.989	0.962	0.955	0.958
		CNN	0.958	0.885	0.747	0.754	0.952	0.828	0.736	0.743
B	Proposed	LSTM	**1.000**	**0.999**	**0.999**	**0.999**	**0.987**	**0.924**	**0.928**	**0.926**
		CNN	0.977	0.901	0.827	0.859	0.968	0.844	0.767	0.799
	OHLC	LSTM	0.959	0.812	0.644	0.700	0.957	0.810	0.629	0.684
		CNN	0.941	0.350	0.335	0.326	0.941	0.346	0.334	0.325
C	Proposed	LSTM	**0.998**	**0.982**	**0.984**	**0.983**	**0.994**	**0.953**	**0.952**	**0.952**
		CNN	0.996	0.963	0.983	0.973	0.990	0.917	0.940	0.928
	OHLC	LSTM	0.985	0.902	0.860	0.878	0.983	0.885	0.846	0.864
		CNN	0.971	0.808	0.726	0.760	0.967	0.769	0.675	0.706
D	Proposed	LSTM	**0.996**	**0.993**	**0.995**	**0.994**	**0.991**	**0.988**	**0.988**	**0.988**
		CNN	0.987	0.984	0.983	0.983	0.981	0.976	0.973	0.974
	OHLC	LSTM	0.910	0.881	0.834	0.847	0.902	0.866	0.834	0.842
		CNN	0.856	0.830	0.767	0.781	0.852	0.847	0.747	0.764
E	Proposed	LSTM	**1.000**	**1.000**	**1.000**	**1.000**	**0.997**	**0.902**	**0.891**	**0.896**
		CNN	0.999	0.968	0.952	0.959	0.989	0.656	0.534	0.579
	OHLC	LSTM	0.994	0.855	0.726	0.776	0.992	0.789	0.664	0.713
		CNN	0.989	0.544	0.334	0.334	0.989	0.334	0.333	0.332

Recent Advances

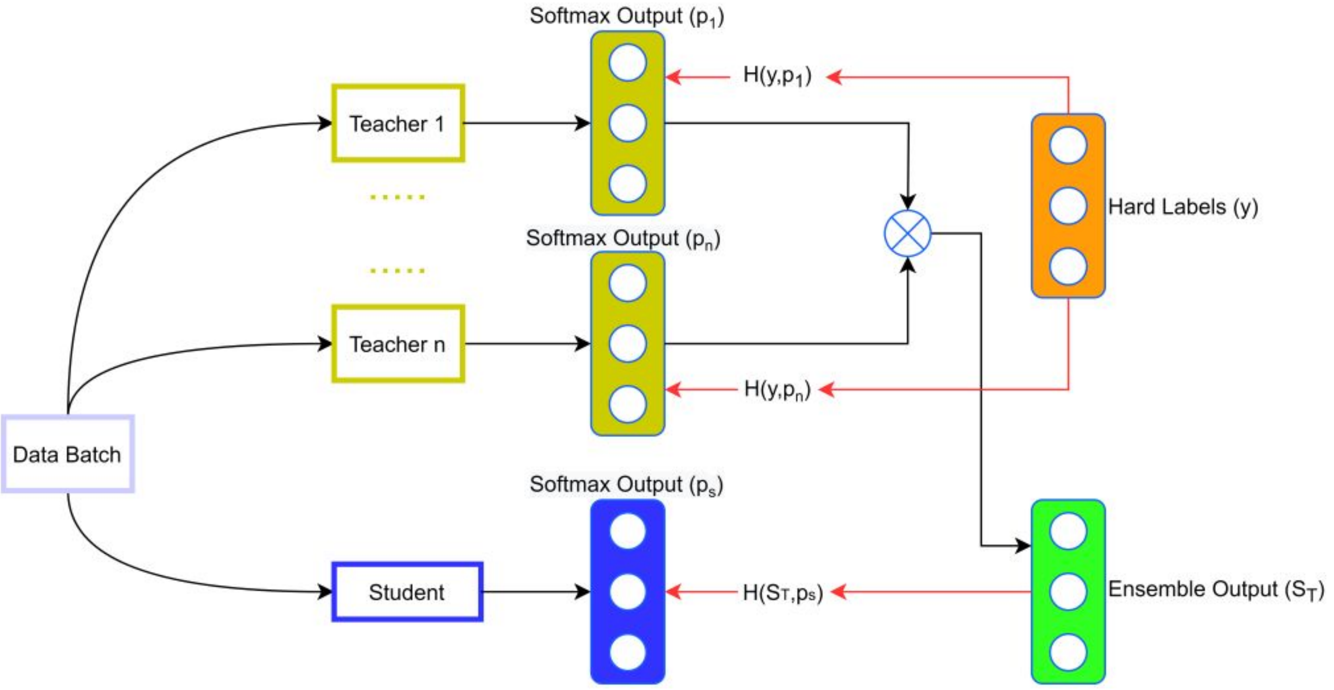
Online Knowledge Distillation for Financial Timeseries Forecasting

- Noisy nature of the data can often cause considerably different behaviors between DL models
    - despite following the same training process, model architecture, and hyper-parameters
- Methods proposed for generating the training labels can sometimes further reinforce such issues
- Ensemble-based online distillation method that can significantly reduce training instability
- In contrast to offline distillation approaches, the proposed method works in a single step

Recent Advances

Online Knowledge Distillation for Financial Timeseries Forecasting

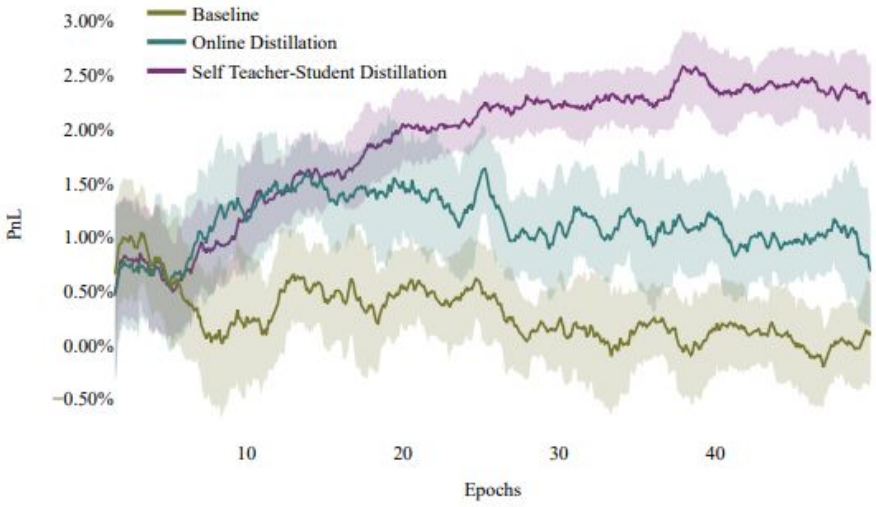
- The same data batch is given to the teachers and students
- The probability distribution of all teachers gets averaged to to form the soft labels of the teachers ensemble



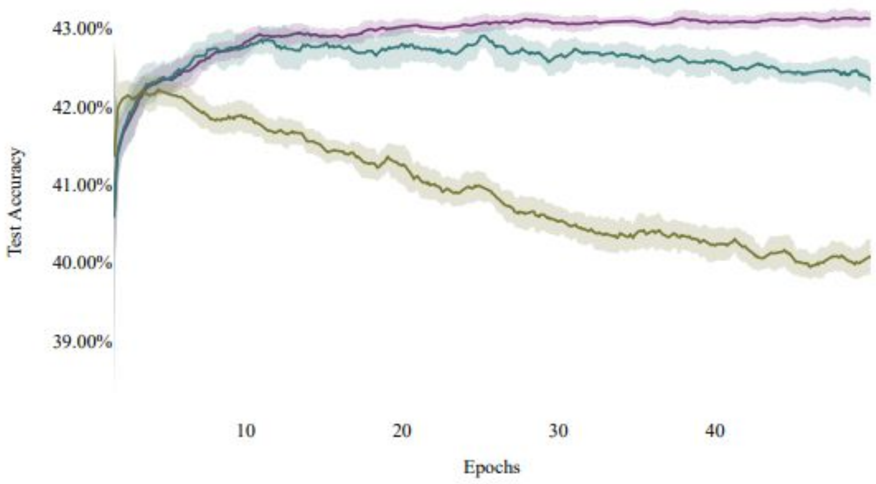
Recent Advances

Online Knowledge Distillation for Financial Timeseries Forecasting

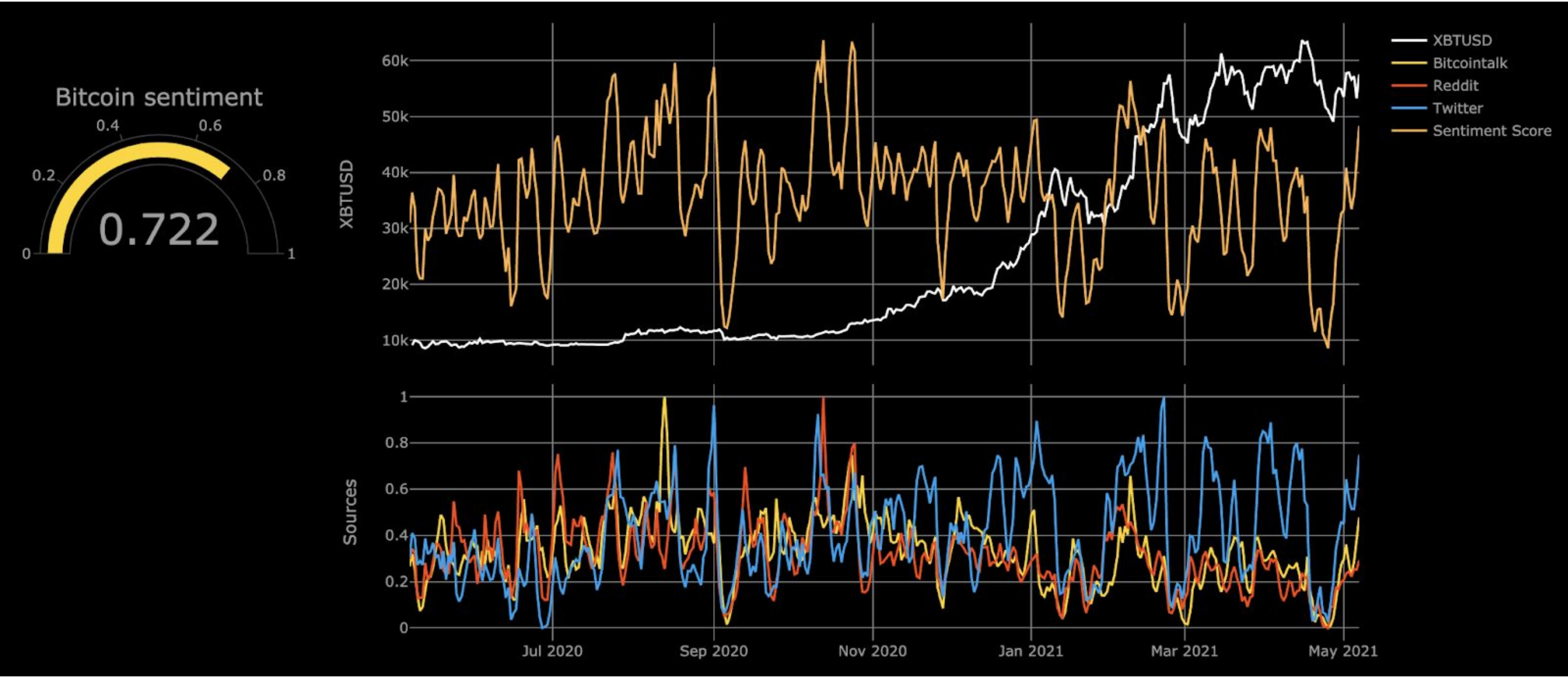
- Faster convergence than baseline (offline) distillation
- And faster and more stable training training



(a) PnL



(b) Accuracy



Adding Sentiment Information to the models

Recent Advances

Sentiment-Aware Distillation for Bitcoin Trend Forecasting Under Partial Observability

- Sentiment harvesting pipelines can be difficult to implement
- Proposed a method that can exploit sentiment information as a source of additional supervision during training
- Allowing the trained agent to operate under partial observability
    - I.e., even when sentiment information isn't available during deployment

Method / Epoch	**50**	**70**	**100**
Price - Baseline	0.198	-0.233	-0.398
Sentiment - Baseline	0.148	-0.348	-0.235
Price - Cross Distillation	0.946	0.483	0.048
Sentiment - Cross Distillation	1.246	1.002	0.923
Price - Proposed	**1.334**	**1.354**	**1.155**

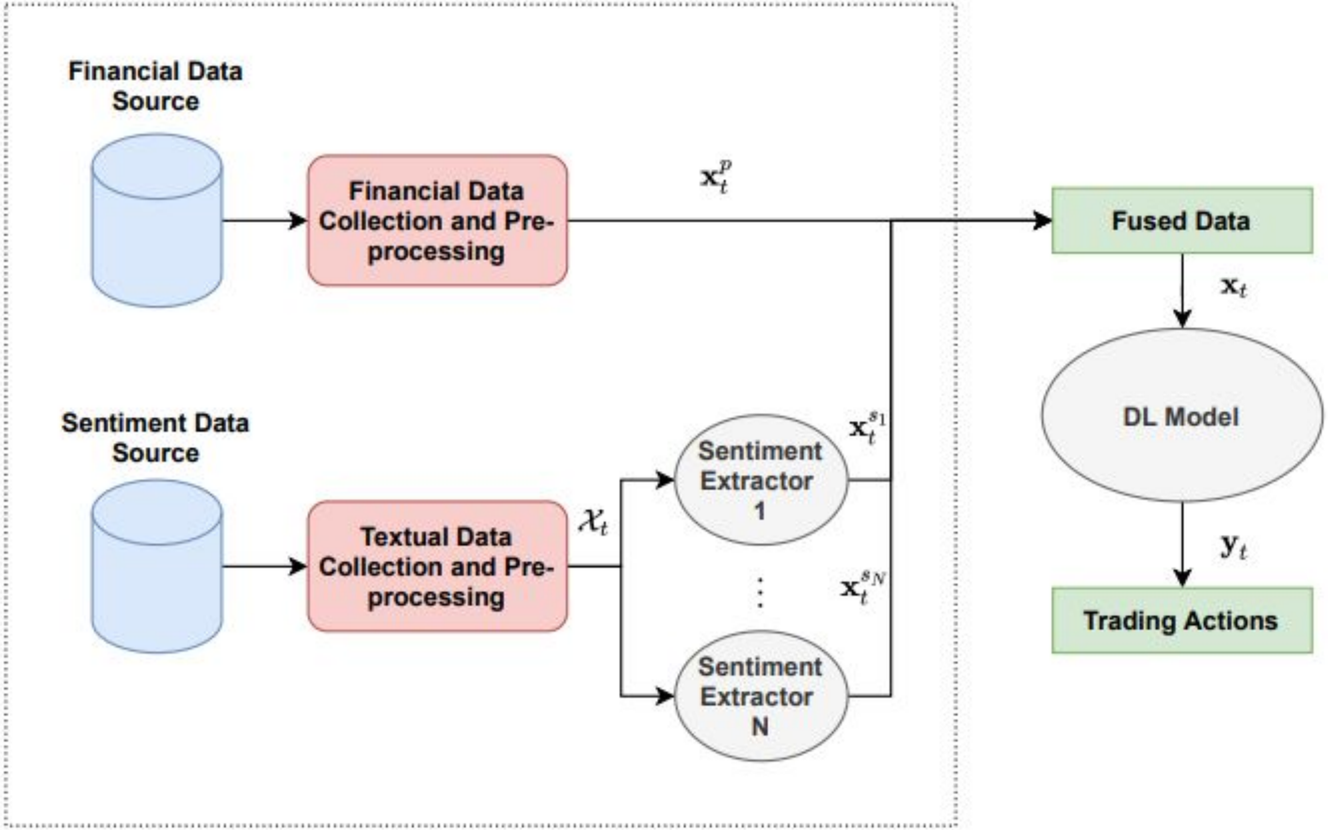
Recent Advances

Multisource financial sentiment analysis for detecting Bitcoin price change indications using deep learning

- Until recently, DL models for financial trading have mostly ignored sentiment-related information
- Examining whether the use of sentiment information, as extracted by various online sources, including news articles, is beneficial when training DL agents for trading
- Proposed a multi-source sentiment fusion approach that can improve the performance over the rest of the evaluated approaches

Recent Advances

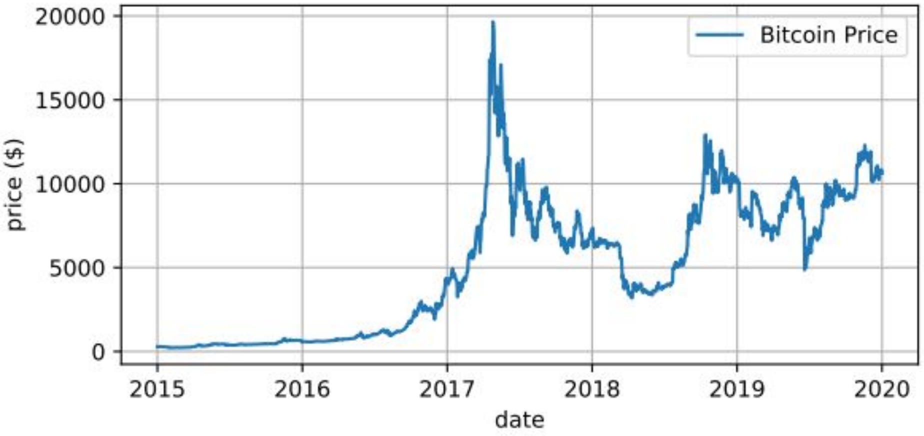
Multisource financial sentiment analysis for detecting Bitcoin price change indications using deep learning

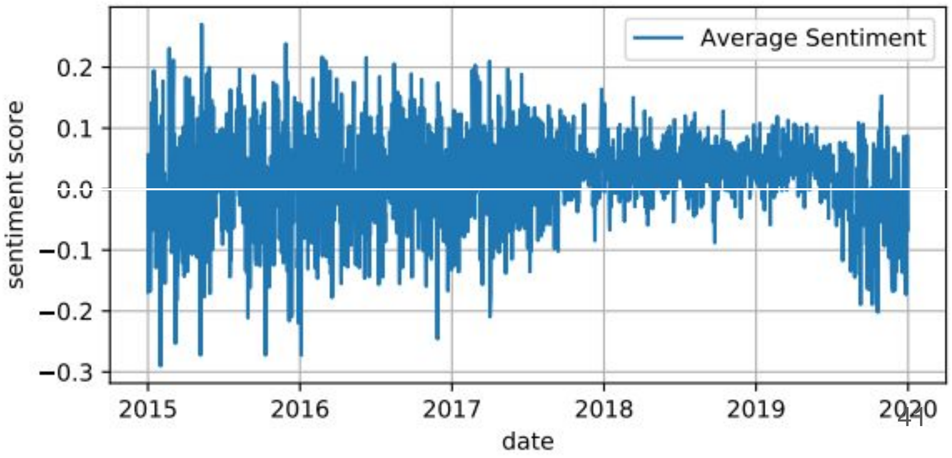


Recent Advances

Learning sentiment-aware trading strategies for bitcoin leveraging deep learning-based financial news analysis

- Extensive evaluation on several different NN models like MLPs, CNNs and RNNs
- Sentiment information might actually be a stronger predictor compared to the information provided by the actual price time-series
    - For BitCoin





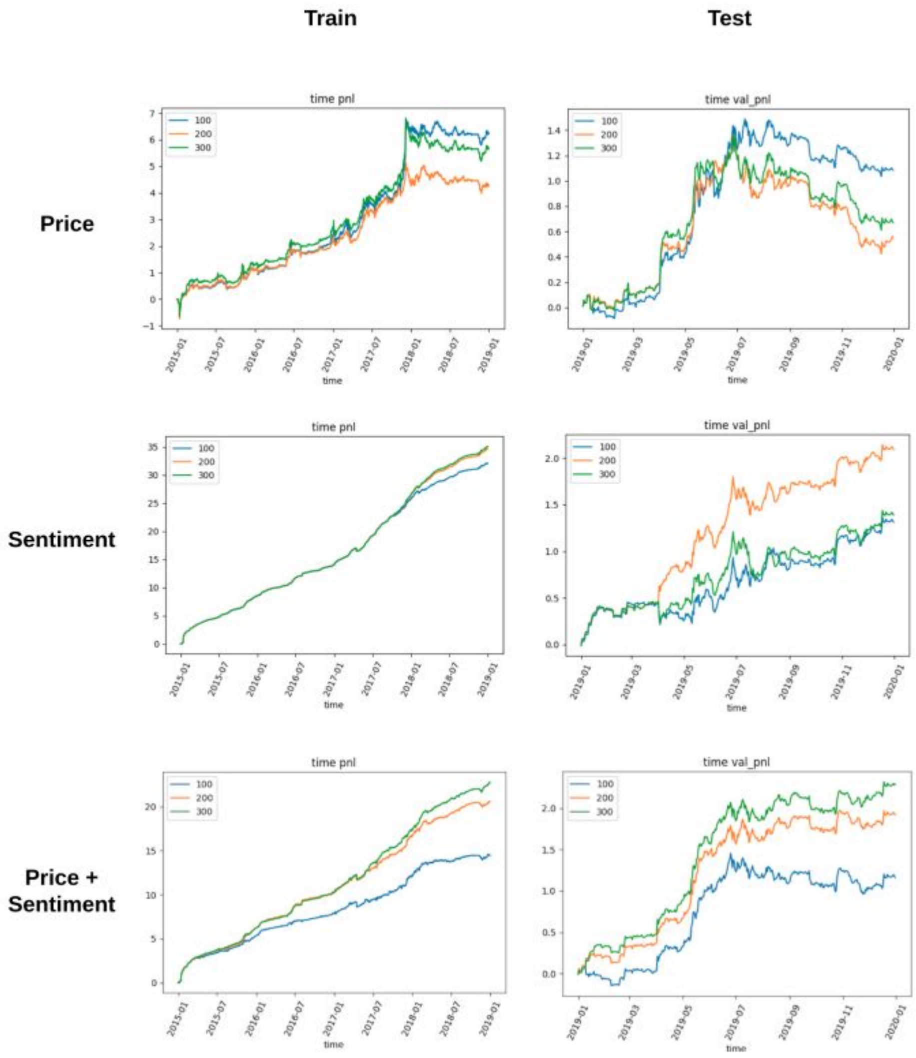
Recent Advances

Learning sentiment-aware trading strategies for bitcoin leveraging deep learning-based financial news analysis

This result demonstrates that sentiment-information for cryptocurrencies, such as Bitcoin, might actually be a stronger predictor of its future behavior compared to the information provided by the price time-series

Recent Ac

Learning sentimen



g-based financial news analysis

But also, 300 neurons leads to overfitting without price information

Thank you!

paranous@csd.auth.gr

DeepFinance

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ

ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ

ΕΣΠΑ
2014-2020
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης