WANNA GET RICH?

hypothetically…

Carleton

Department of Statistics

Predicting stock prices using social network sentiment: An evaluative analysis of neural networks and time series models

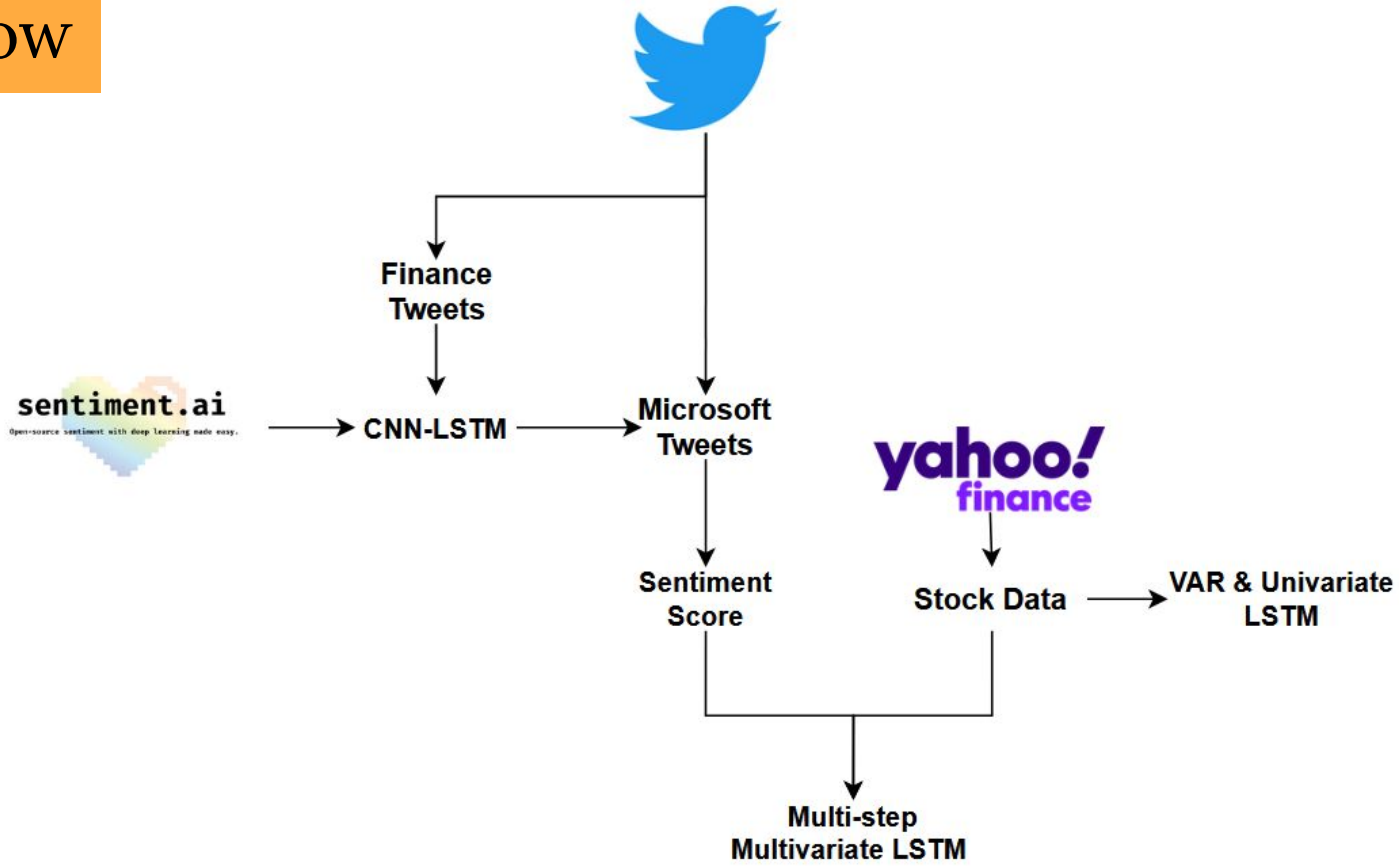
Kevin Chen, David Chu, Carl Zhang, Jeanny Zhang
Advised by Professor Deepak Bastola

(this is neither a CS nor Econ comps)

Project Ideas

- Predict the market sentiments from Twitter using contextual language models
- Incorporate sentiment features to enhance predicting stock prices from their previous values
- Compare the results from neural networks and traditional time series models

Workflow


Finance Tweets
sentiment.ai
Open-source sentiment with deep learning made easy.
CNN-LSTM
Microsoft Tweets
yahoo! finance
Sentiment Score
Stock Data
VAR & Univariate LSTM
Multi-step Multivariate LSTM

Why stocks prices?

Available

Longitudinal

Complex

Great implications

Financial Data

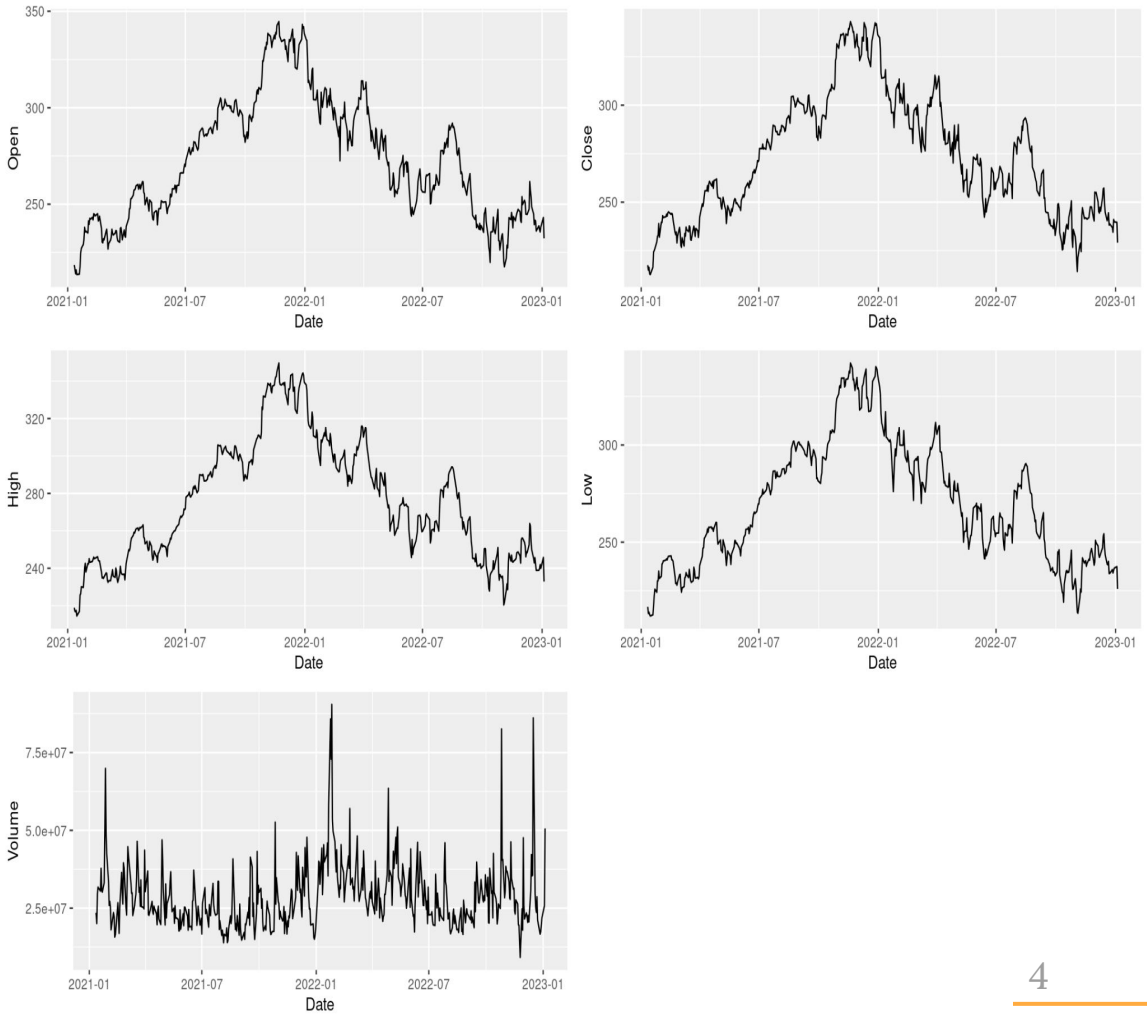
Open: Stock price when trading began that day.

Close: Stock price when trading closed that day.

High: Maximum stock price that day.

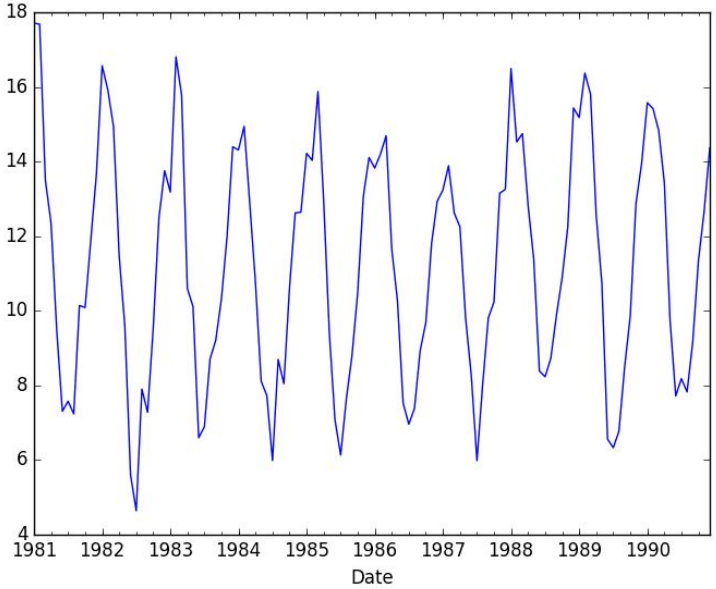
Low: Minimum stock price that day.

Volume: The amount of shares traded on that day.



Autoregressive Model (AR)

AR(p) Model: $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \epsilon_t$



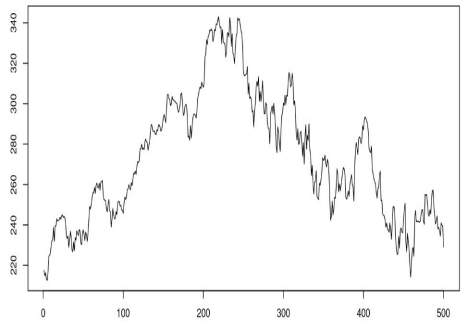
Vector Autoregressive Model (VAR)

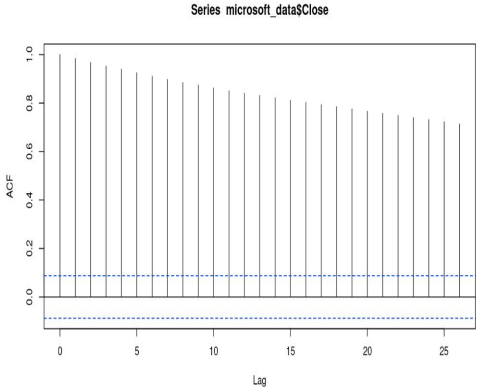
VAR(2):

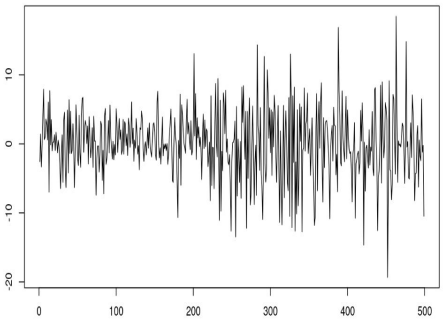
$$
\begin{aligned}
Y_{1,t} &= \alpha_1 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \beta_{13,1} Y_{3,t-1} + \beta_{11,2} Y_{1,t-2} + \beta_{12,2} Y_{2,t-2} + \beta_{13,2} Y_{3,t-2} + \epsilon_{1,t} \\
Y_{2,t} &= \alpha_2 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \beta_{23,1} Y_{3,t-1} + \beta_{21,2} Y_{1,t-2} + \beta_{22,2} Y_{2,t-2} + \beta_{23,2} Y_{3,t-2} + \epsilon_{2,t} \\
Y_{3,t} &= \alpha_3 + \beta_{31,1} Y_{1,t-1} + \beta_{32,1} Y_{2,t-1} + \beta_{33,1} Y_{3,t-1} + \beta_{31,2} Y_{1,t-2} + \beta_{32,2} Y_{2,t-2} + \beta_{33,2} Y_{3,t-2} + \epsilon_{3,t}
\end{aligned}
$$

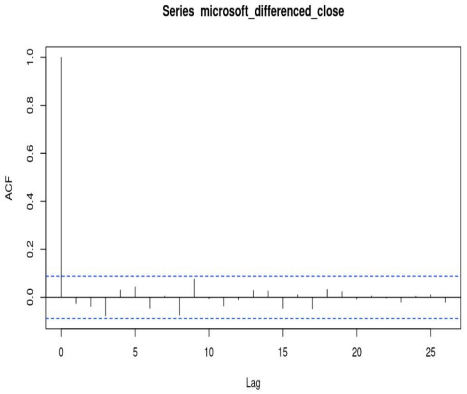
Bayesian Information Criterion: $\mathrm{BIC} = n \ln\left(\frac{\mathrm{RSS}}{n}\right) + k \ln(n).$

Stationarity and Differencing

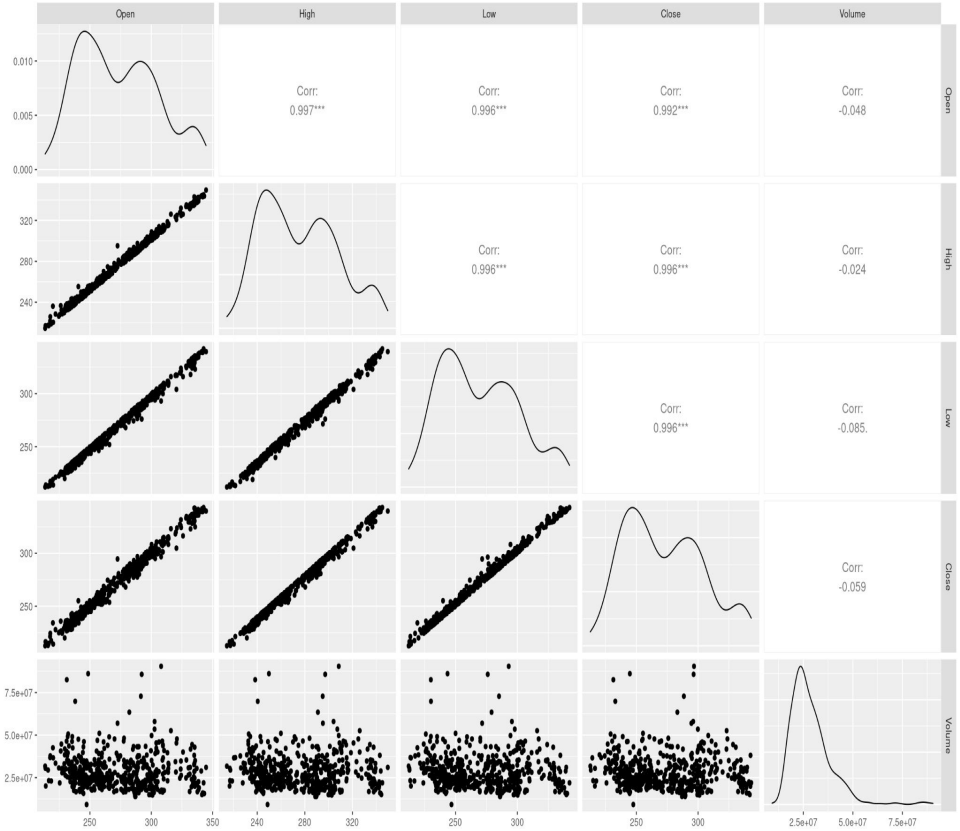


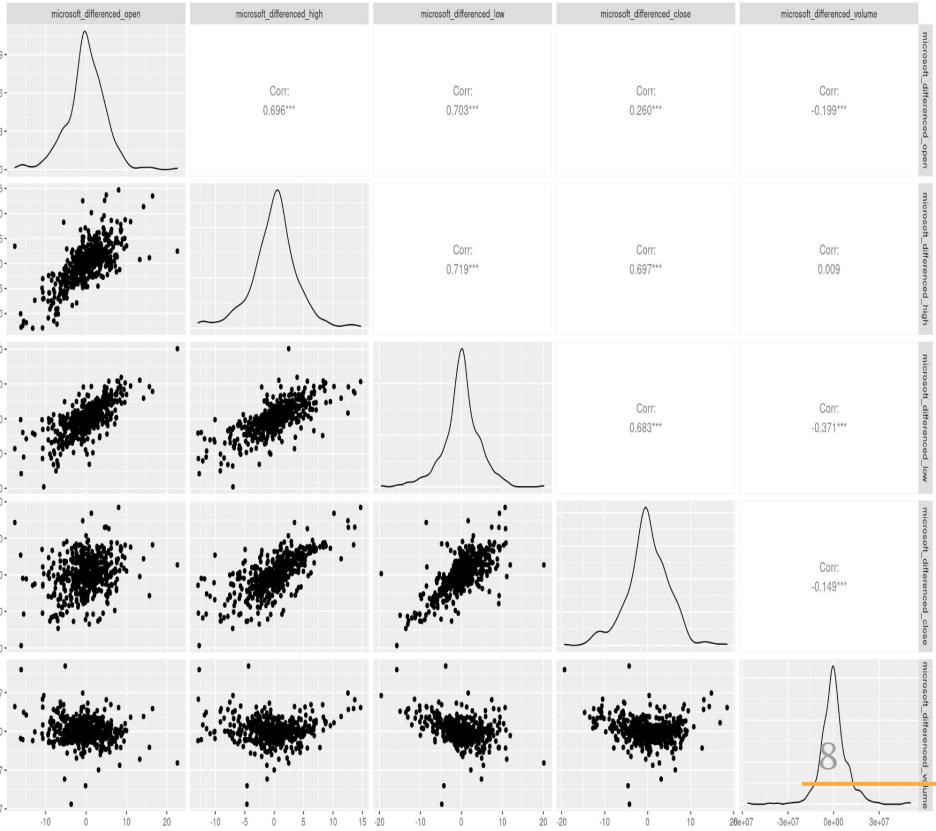






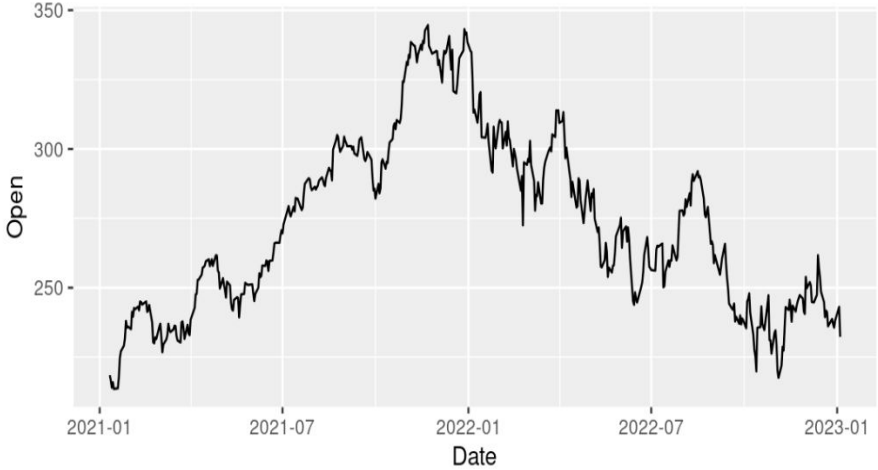
Multicollinearity





Nonlinear Dependencies

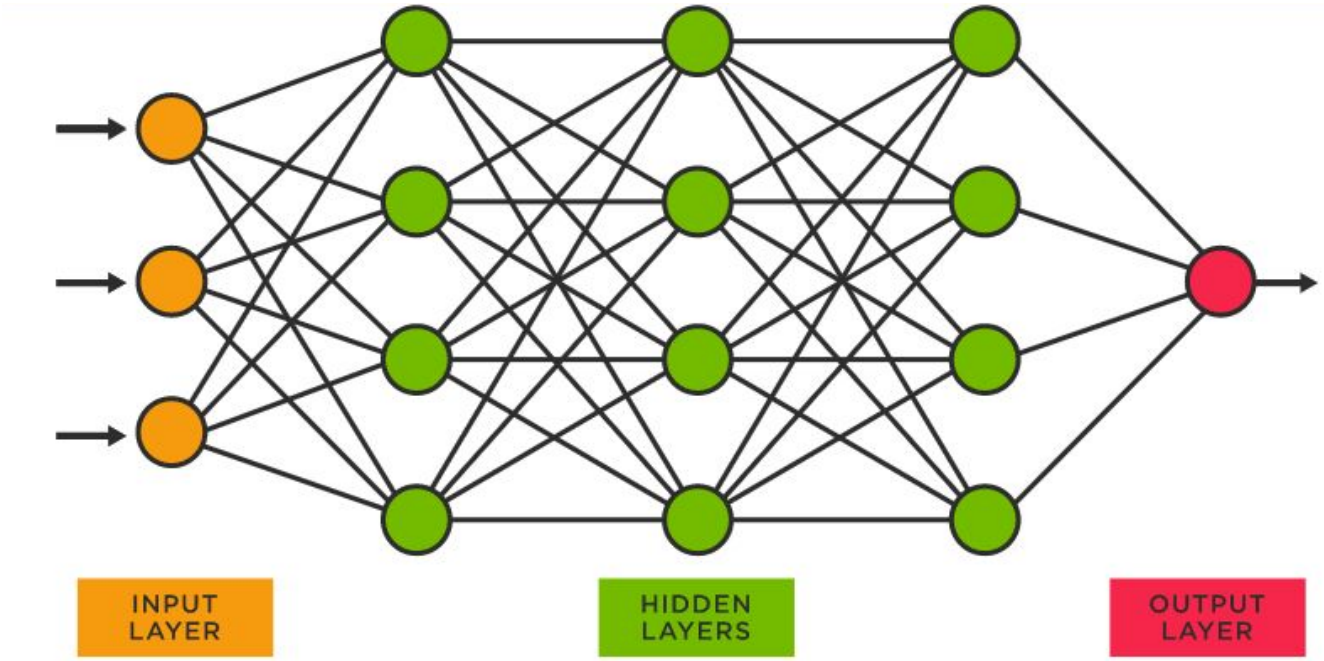
$$Y_{1,t} = \alpha_1 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \beta_{13,1} Y_{3,t-1} + \beta_{11,2} Y_{1,t-2} + \beta_{12,2} Y_{2,t-2} + \beta_{13,2} Y_{3,t-2} + \epsilon_{1,t}$$
$$Y_{2,t} = \alpha_2 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \beta_{23,1} Y_{3,t-1} + \beta_{21,2} Y_{1,t-2} + \beta_{22,2} Y_{2,t-2} + \beta_{23,2} Y_{3,t-2} + \epsilon_{2,t}$$
$$Y_{3,t} = \alpha_3 + \beta_{31,1} Y_{1,t-1} + \beta_{32,1} Y_{2,t-1} + \beta_{33,1} Y_{3,t-1} + \beta_{31,2} Y_{1,t-2} + \beta_{32,2} Y_{2,t-2} + \beta_{33,2} Y_{3,t-2} + \epsilon_{3,t}$$



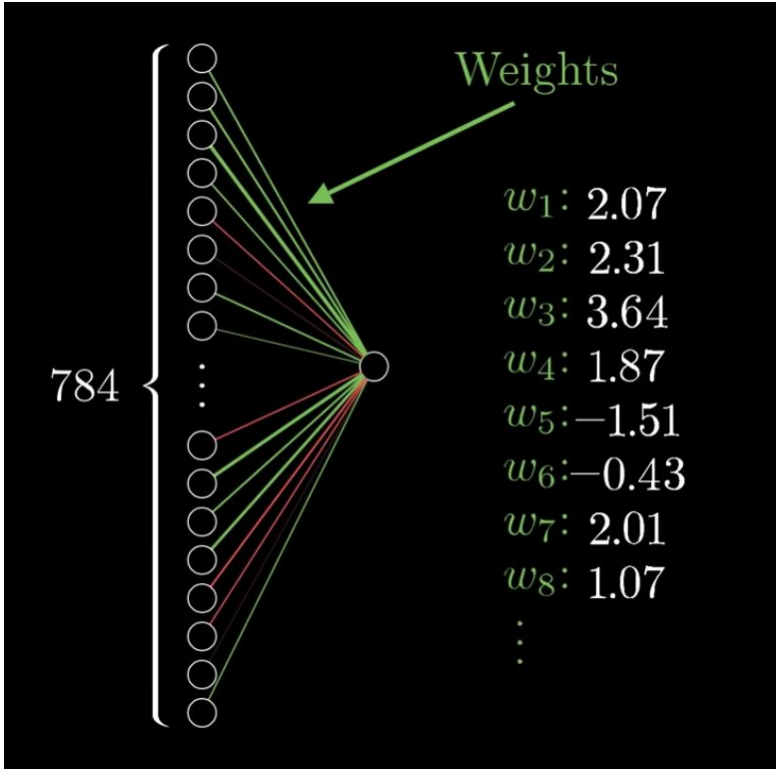
Nonlinear Dependencies

Neural Networks

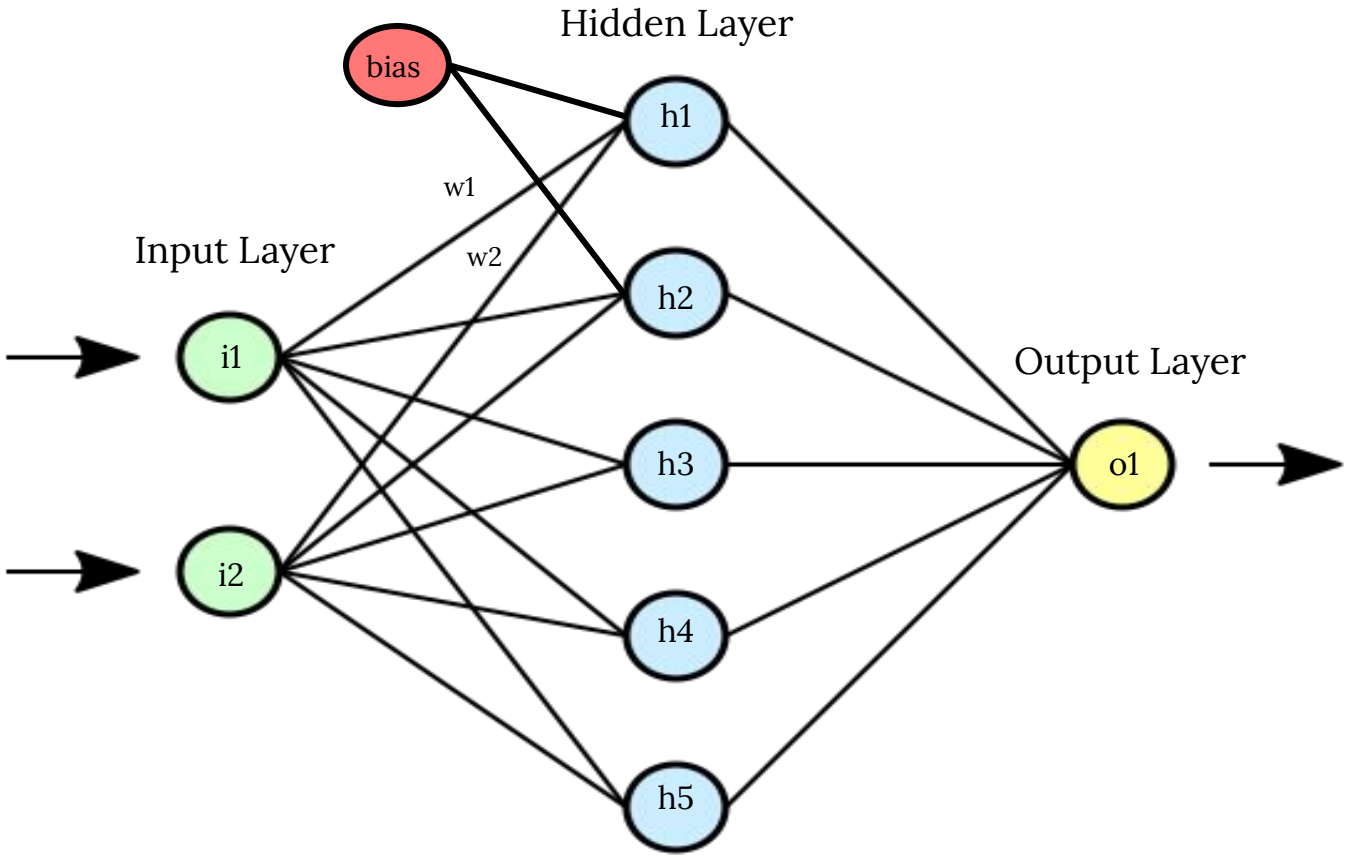
Neural Networks



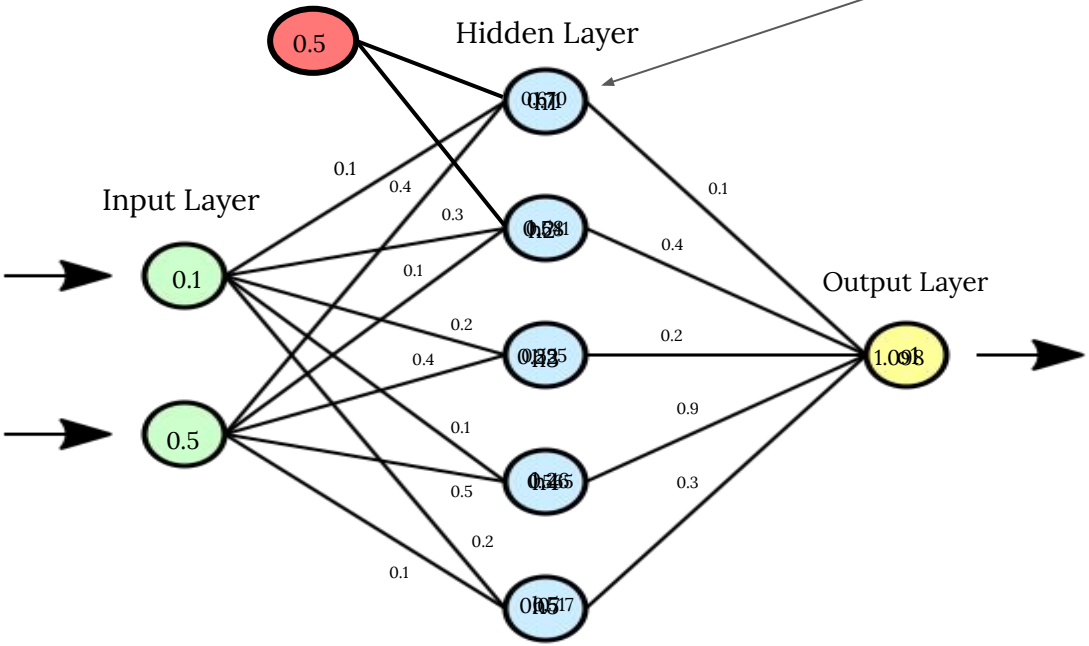
Neural Networks



Neural Network Demonstration



Forward Propagation



h1 = w1 * i1 + w2 * i2 + bias

h1 = 0.1 * 0.1 + 0.4 * 0.5 + 0.5 = 0.71

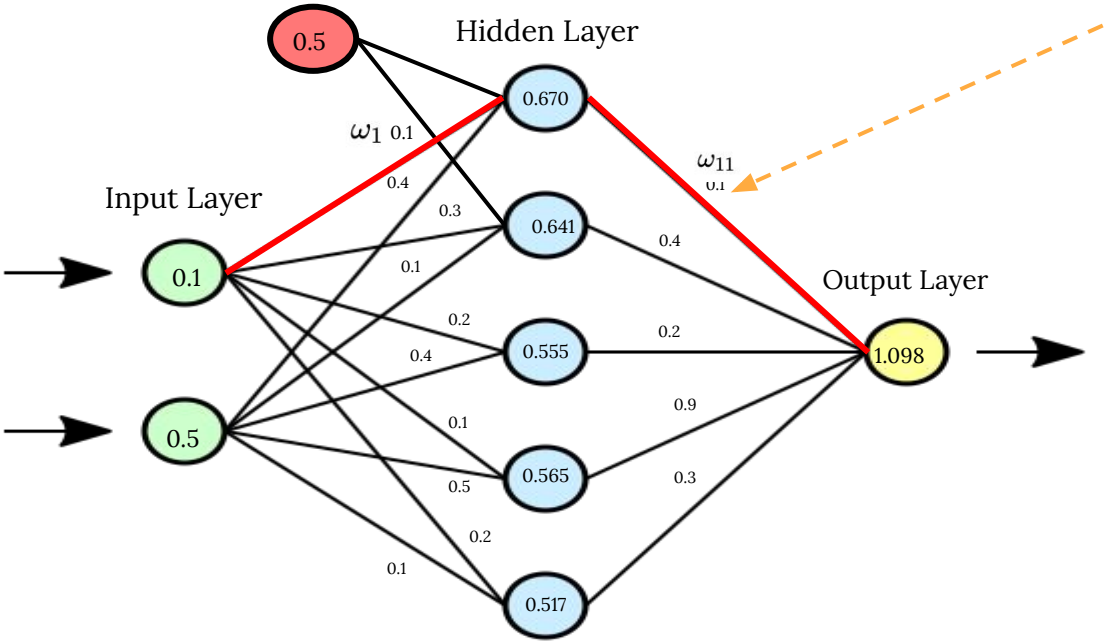
h2 = 0.3 * 0.1 + 0.1 * 0.5 + 0.5 = 0.58
h3 = 0.3 * 0.1 + 0.1 * 0.5 = 0.22
h4 = 0.3 * 0.1 + 0.1 * 0.5 = 0.26
h5 = 0.3 * 0.1 + 0.1 * 0.5 = 0.07

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma(\mathrm{h1}) = 0.6704012$$
$$\sigma(\mathrm{h2}) = 0.6410674$$
$$\sigma(\mathrm{h3}) = 0.5547792$$
$$\sigma(\mathrm{h4}) = 0.5646363$$
$$\sigma(\mathrm{h5}) = 0.5174929$$

o1 = 0.1 * 0.6704012 +
0.4 * 0.6410674 +
0.2 * 0.5547792 +
0.9 * 0.5646363 +
0.3 * 0.5174929 = 1.097843

Gradient descent and Chain Rule Calculations (Back Propagation)



$$\omega_{11} = \omega_{11} - \text{learning rate} \times \text{gradient}$$

$$\text{learning rate} = 0.001$$

$$\text{gradient} = \frac{\partial(\text{Loss})}{\partial \omega_{11}}$$

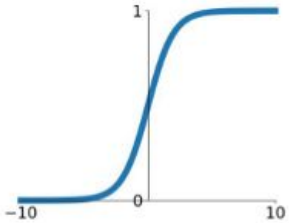
$$\omega_1 = \omega_1 - \text{learning rate} \times \text{gradient}$$

$$\text{gradient} = \frac{\partial(\text{Loss})}{\partial \omega_1} = \frac{\partial(\text{Loss})}{\partial \sigma(\text{h1})} \times \frac{\partial \sigma(\text{h1})}{\partial \omega_1}$$

Activation Functions

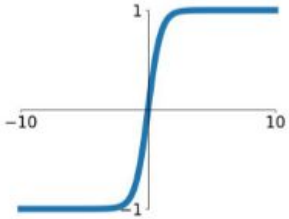
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



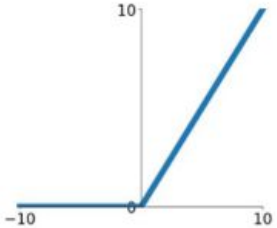
tanh

$$\tanh(x)$$

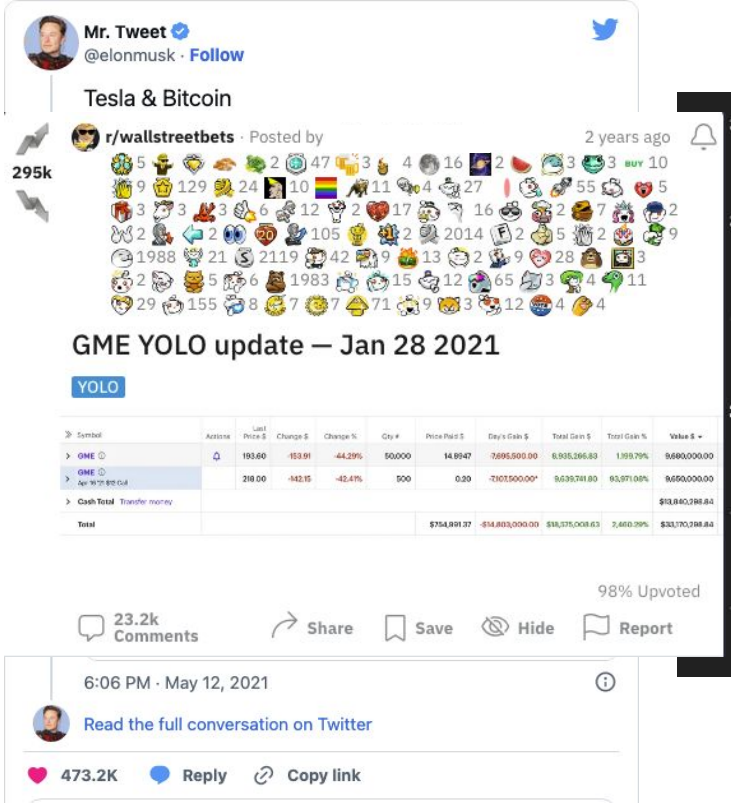


ReLU

$$\max(0, x)$$



Why market sentiment?



Why Microsoft?

Microsoft:

Tesla:

Why Microsoft?

Falcodrin Stock Alerts @falcodrin · 48s

🔥 **AMAZON** : XFX Radeon RX 7900 XT Speedster MERC319- $879.99

Link: stockl.ink/jRffQPCn

ATC: stockl.ink/WUdAc6pw

#amazon | #xfx | #rx7900xts | #ad

leeown / Leon #TeamSeas @e71fs · 23h

The Tri-Strike **meta** is crazy #Splatoon3 #NintendoSwitch

Dare Obasanjo @Carnage4Life · 10h

Microsoft's embrace of ChatGPT with Bing is a great example of counter-positioning.

This is a feature that Google would never ship on its own because it undermines search ads revenue to move to an answers based model.

A masterstroke by Satya.

Text preprocessing

over years should be a billion annual revenue business with a add operating margin with the potential to double that figure in the following years msft

Replace non ASCII characters

Replace URLs

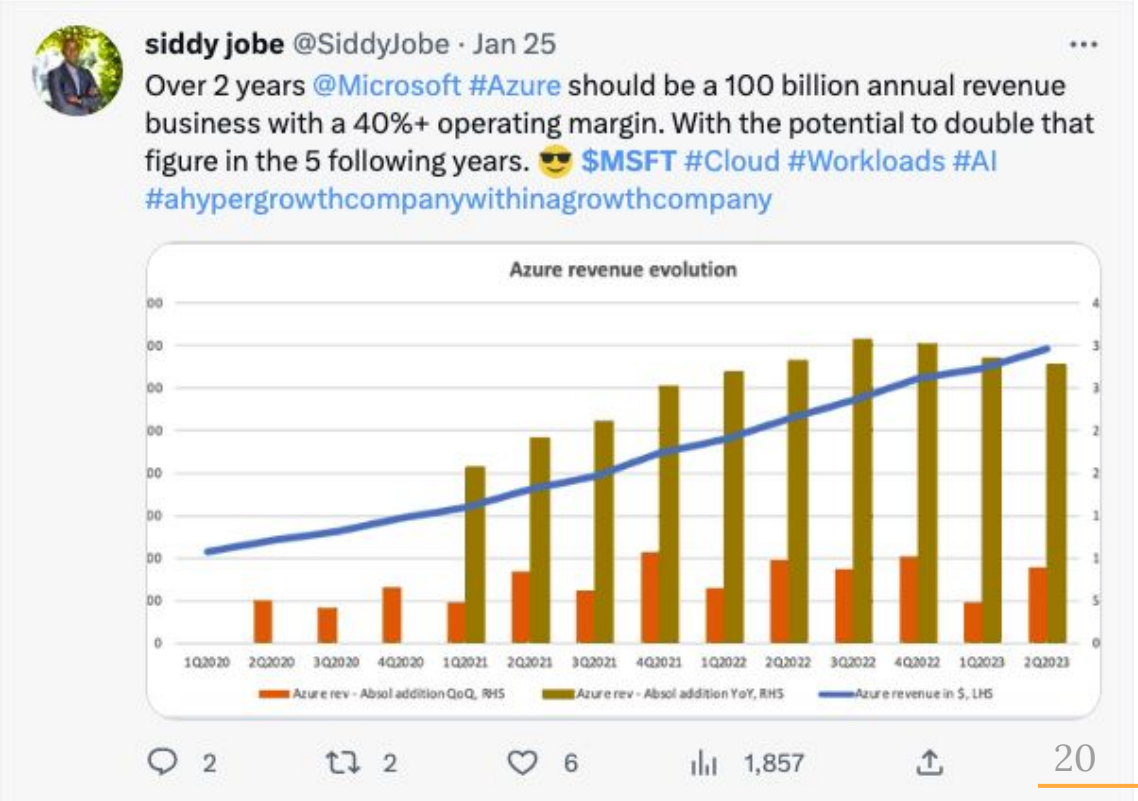
Replace emojis

Replace punctuations

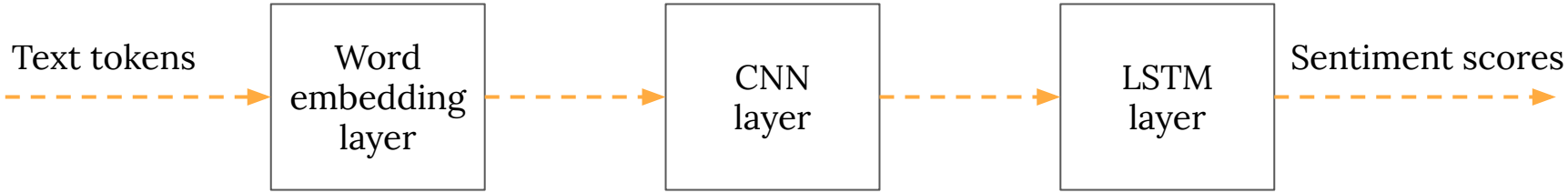
Replace hashtags and tags

Transform to lowercase

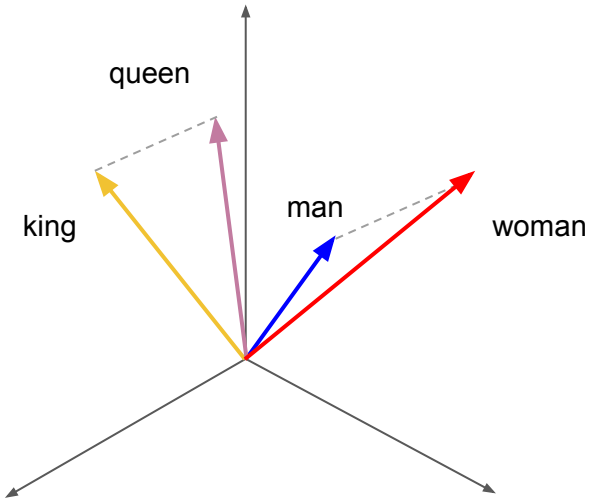
Replace RT or Retweet



Sentiment Analysis Model Architecture



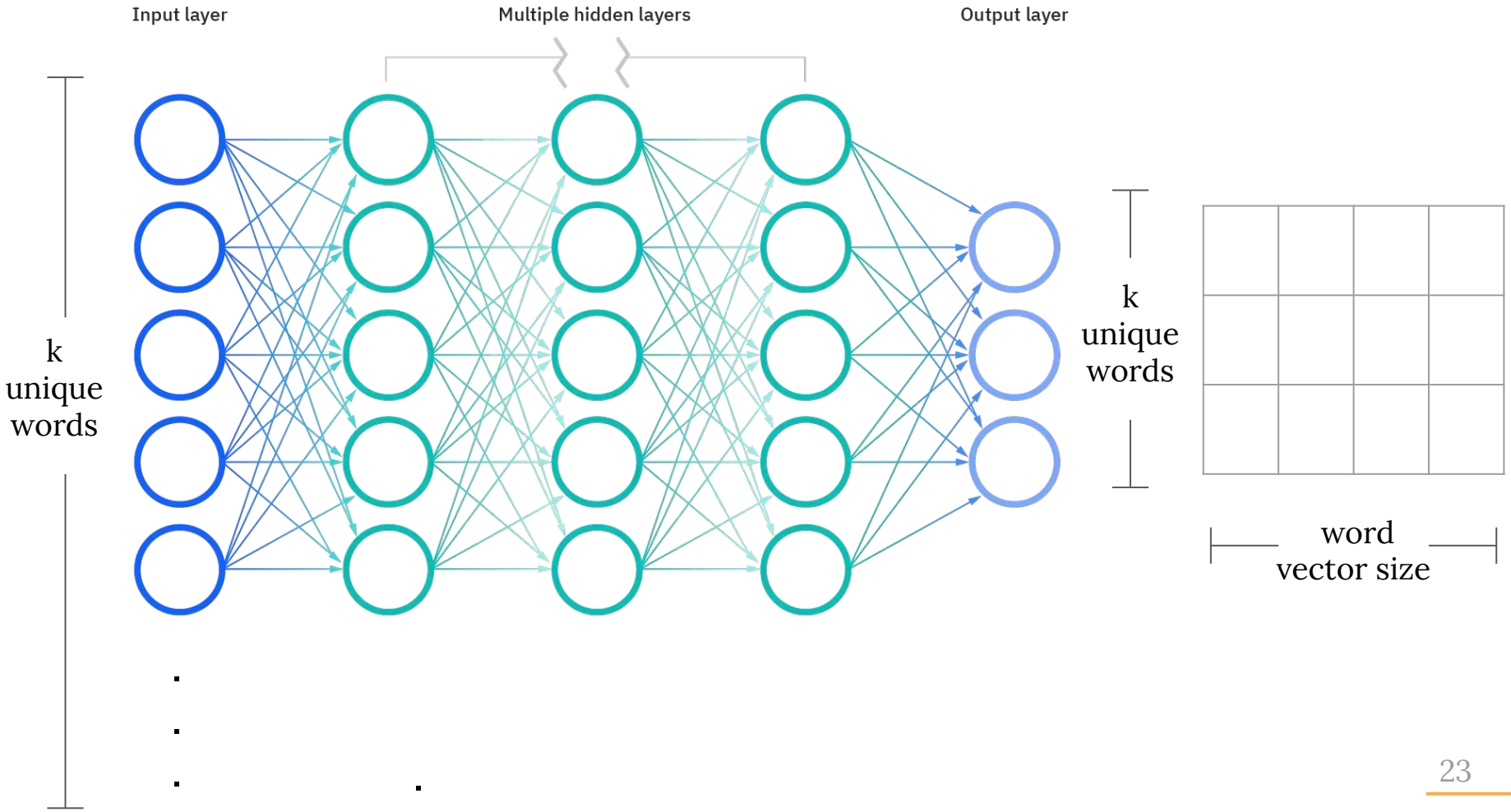
Word Vectors



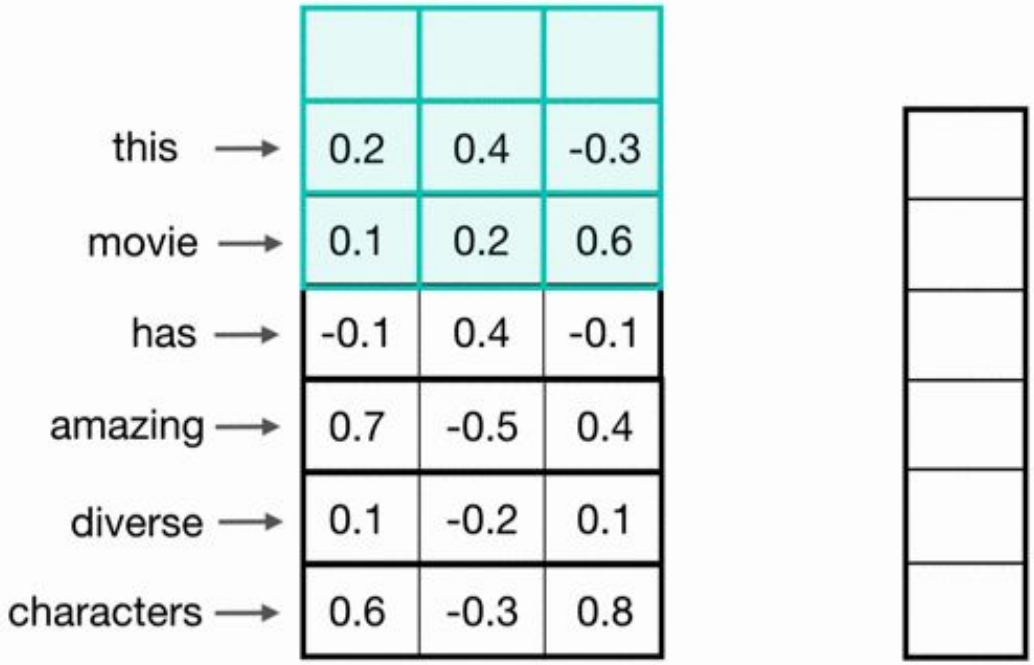
$$\overrightarrow{\text{king}} - \overrightarrow{\text{man}} + \overrightarrow{\text{woman}} = \overrightarrow{\text{queen}}$$

	king	man	woman	queen
	3	2	1	2
	5	3	6	8
Gender	**2**	**2**	**4**	**4**

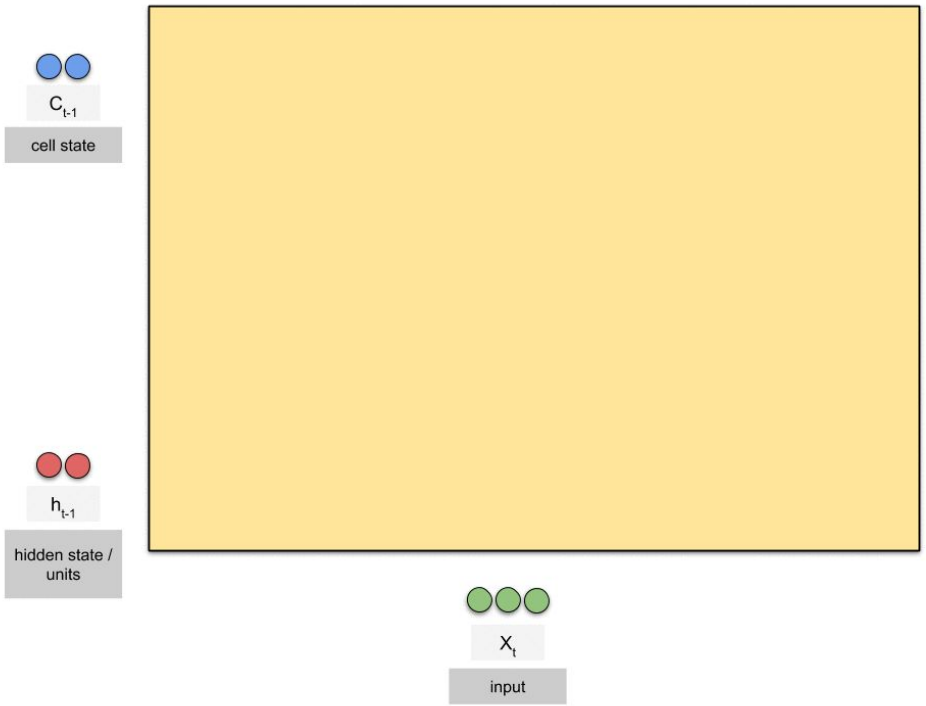
$$\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$


Input layer
Multiple hidden layers
Output layer
k
unique
words
k
unique
words
word
vector size

Convolutional Neural Network



Long-Short Term Memory (LSTM)



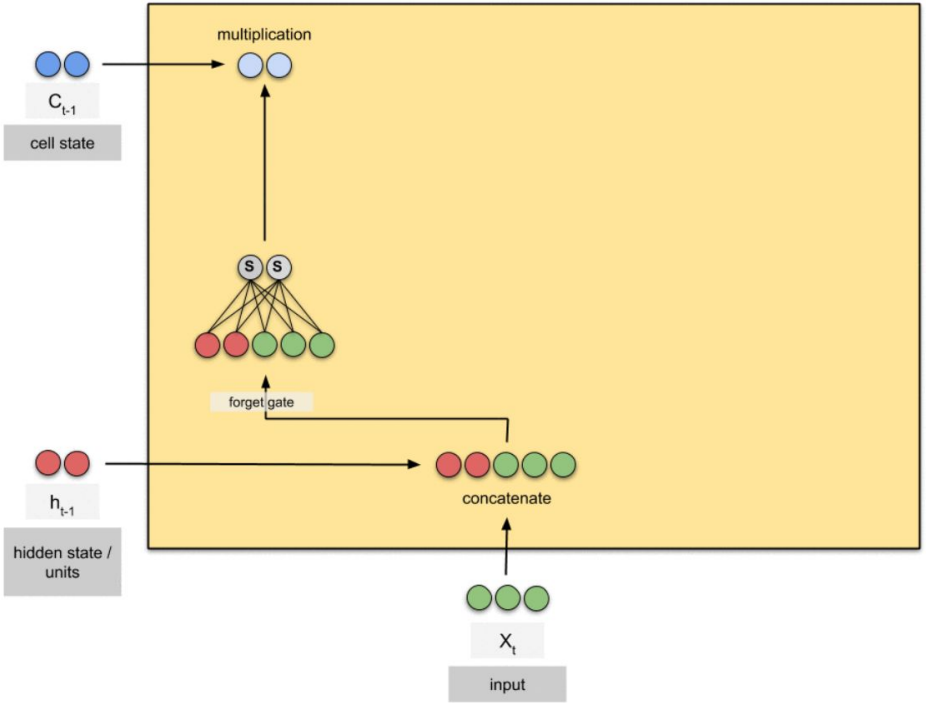
1. Forget gate

2. Input gates

3. Output gate

Karim, R. (2018, December 14). *Animated RNN, LSTM and GRU*. Medium. https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45

Long-Short Term Memory (LSTM)



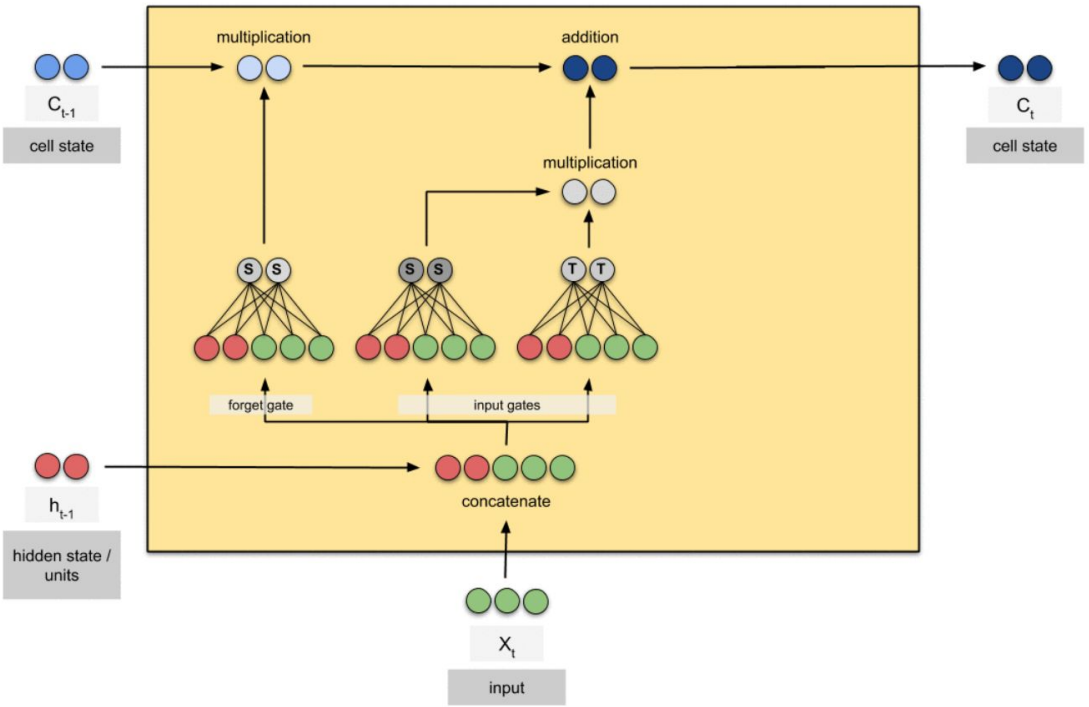
1. Forget gate
   - Sigmoid function
   - Determine what to forget in the current cell state.

2. Input gates

3. Output gate

Karim, R. (2018, December 14). *Animated RNN, LSTM and GRU*. Medium. https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45

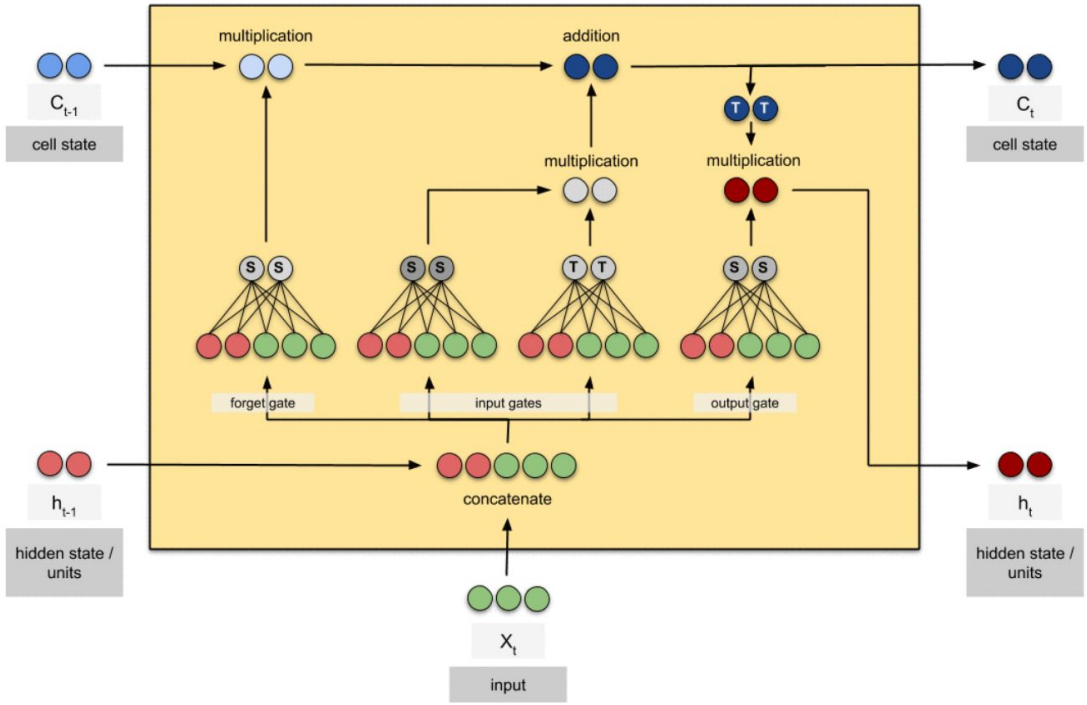
Long-Short Term Memory (LSTM)



1. Forget gate
   - Sigmoid function
   - Determine what to forget in the current cell state.
2. Input gates
   - Sigmoid function
   - Tanh function
   - Output from the forget gate multiplied by the previous cell state and output from the input gate are added together to update cell state.
3. Output gate

Karim, R. (2018, December 14). *Animated RNN, LSTM and GRU*. Medium. https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45

Long-Short Term Memory (LSTM)



1. Forget gate
   - Sigmoid function
   - Determine what to forget in the current cell state.
2. Input gates
   - Sigmoid function
   - Tanh function
   - Output from the forget gate multiplied by the previous cell state and output from the input gate are added together to update cell state.
3. Output gate
   - Sigmoid function
   - Tanh function
   - Regulate hidden state.

Karim, R. (2018, December 14). *Animated RNN, LSTM and GRU*. Medium. https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45

Sentiment Analysis Results

"microsoft about to ruin discord like they ruined skype" → **-0.7915**

"when tasked with the challenge of sourcing talent for a major tech giant we needed a bold leader in place who stays ahead of the game laura belmont our microsoft account director sr engagement manager is collaborative insightful trustworthy a lot of fun" → **0.8163**

Univariate LSTM

A type of LSTM when the data being modeled has a single input variable.

Zero bias in training → Overfitting → Bad out-of-sample predictions with large bias & variance

Multi-Step Multivariate LSTM

A type of LSTM when the data being modeled has multiple input variables and the goal is to make predictions for multiple time steps into the future.

- Step-in period
- Step-out period
- Sliding window

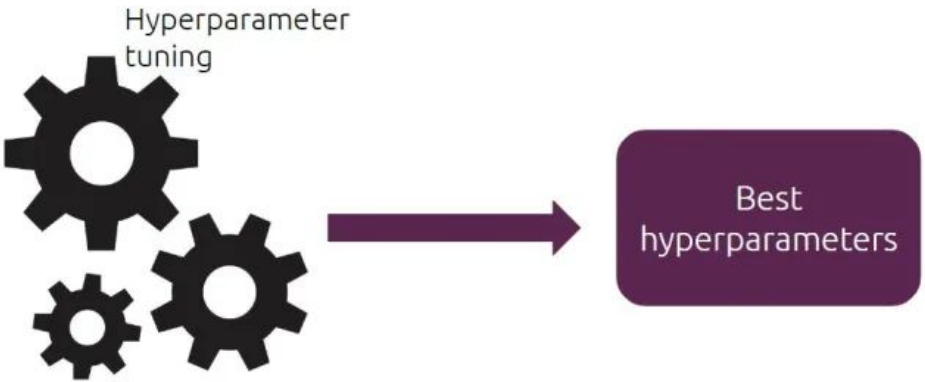
Open	**High**	**Low**	**Close**	**Volume**	**Sentiment Score**
Test	Data	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...

X

Y

Hyperparameter

- Preset before training
- Trial and error
- Search algorithm



Hyperparameter Optimization

	Sentiment Analysis (CNN + LSTM)		Stock Prediction (LSTM)	
Method	Grid search		Random Search	
Hyperparameters	Dense units	16	Dense units	256
	Drop-out rate	0.011	Dense Activation Function	Tanh
	Epoch	4	LSTM units	336, 80
			Drop-out rate	0.1, 0.05

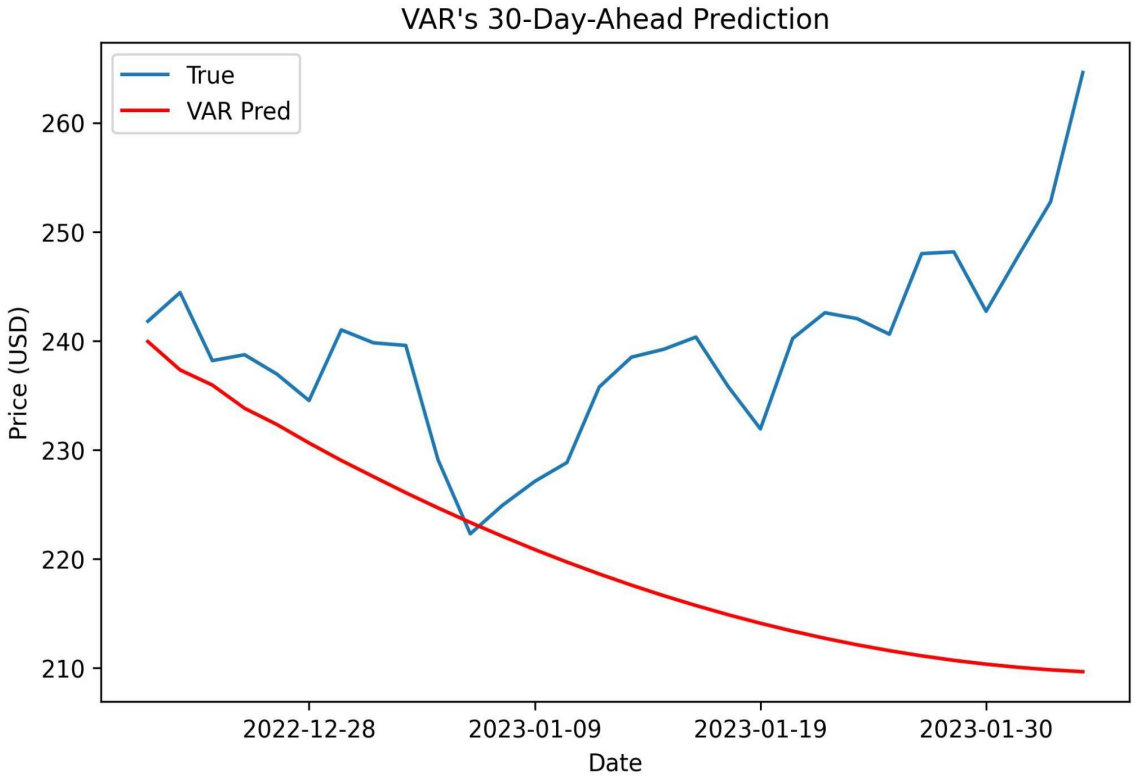
Thormann, M.-L., Farchmin, J., Weisser, C., Kruse, R.-M., Säfken, B., & Silbersdorff, A. (2021). Stock Price Predictions with LSTM Neural Networks and Twitter Sentiment. *Statistics, Optimization & Information Computing, 9(2)*, Article 2. https://doi.org/10.19139/soic-2310-5070-1202

Error metrics

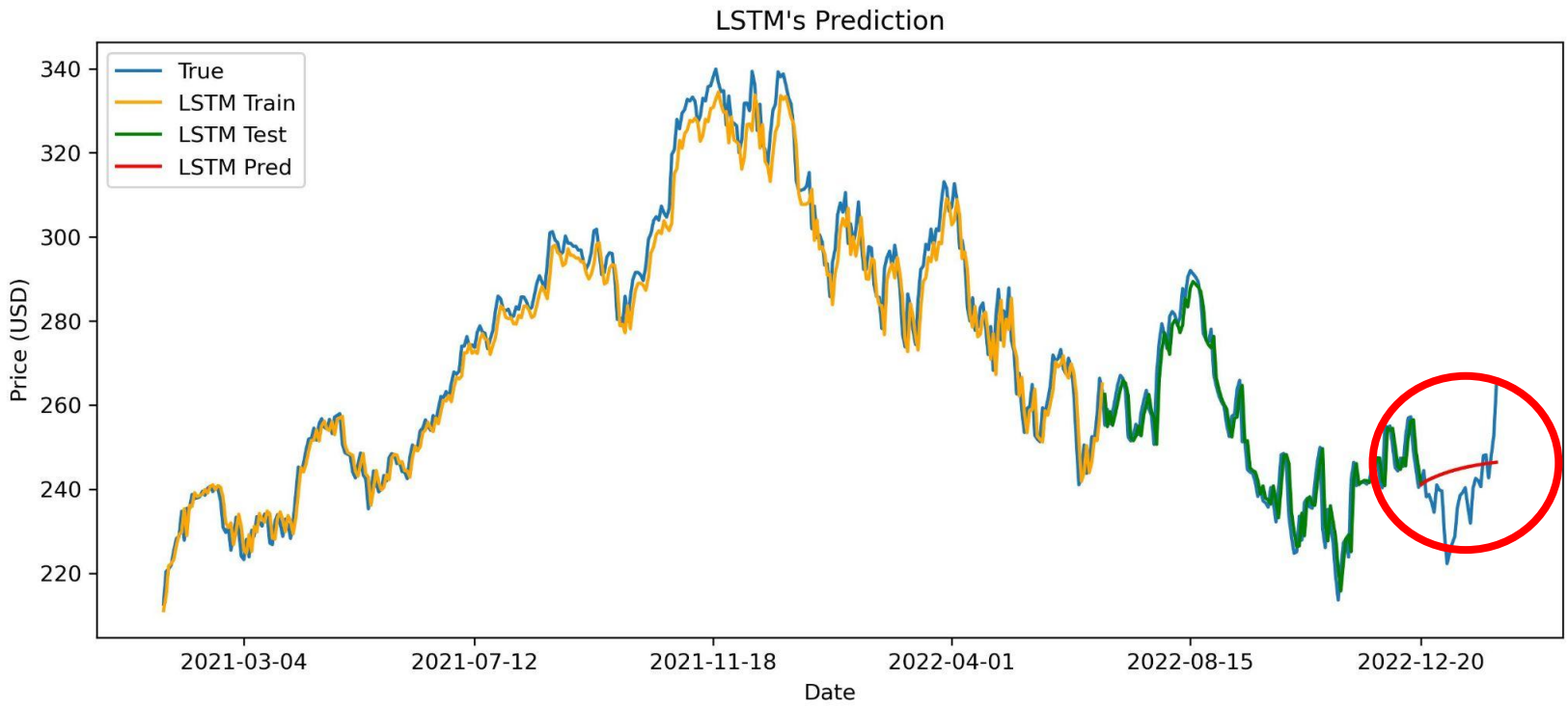
$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i|$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{y_i}$$

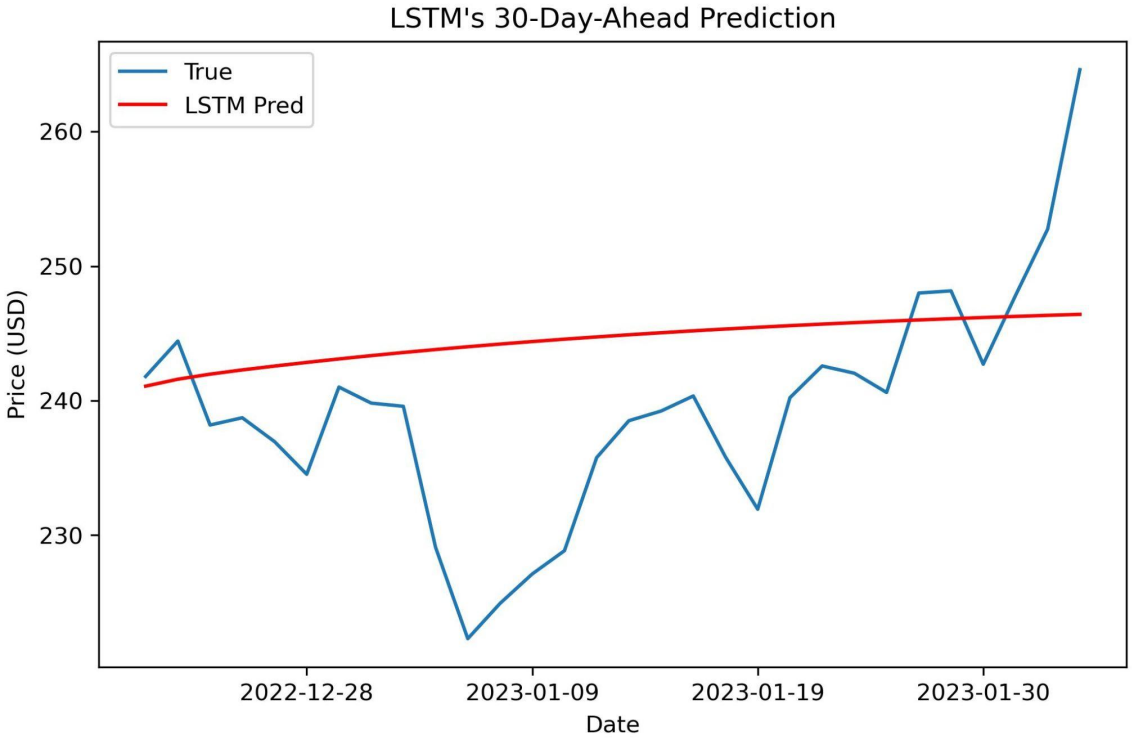
VAR Prediction



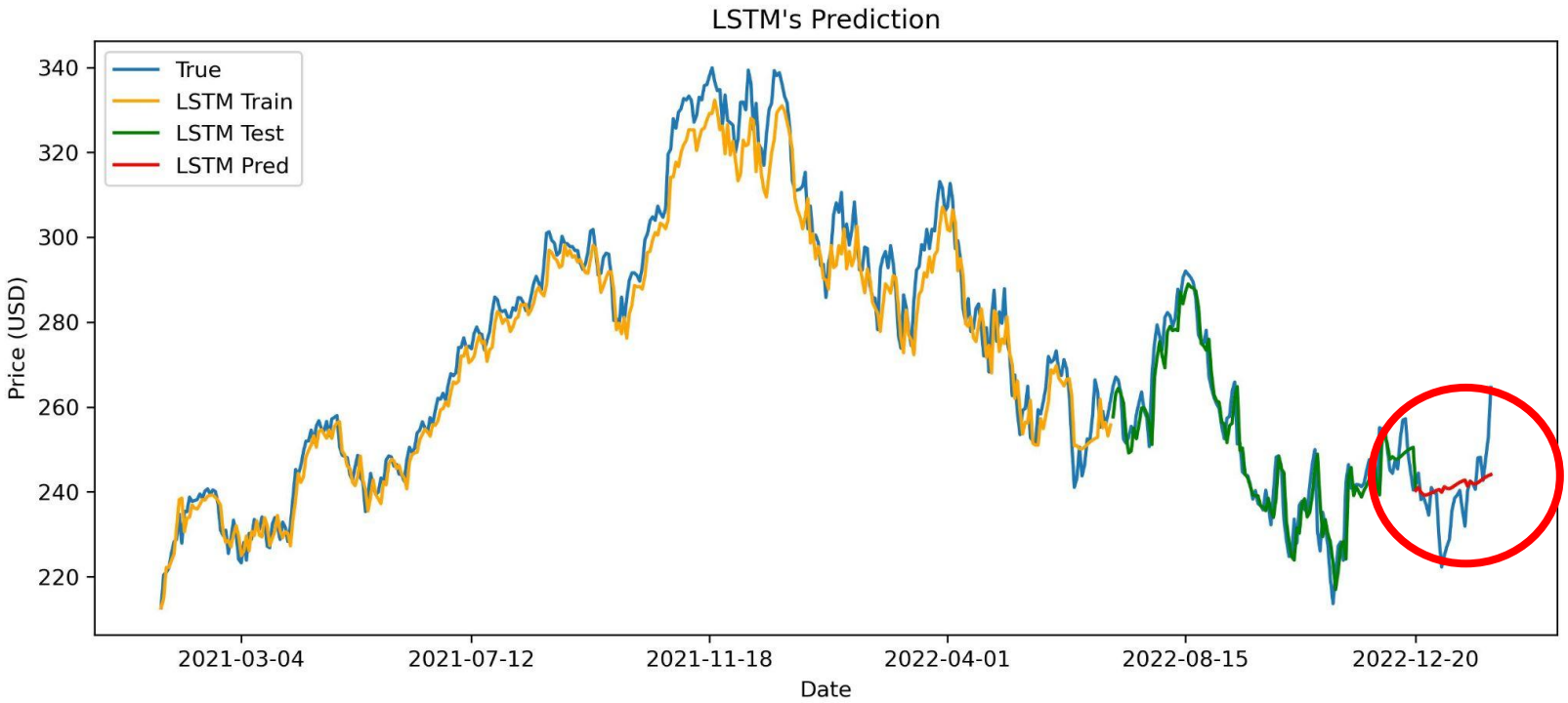
Univariate LSTM Prediction



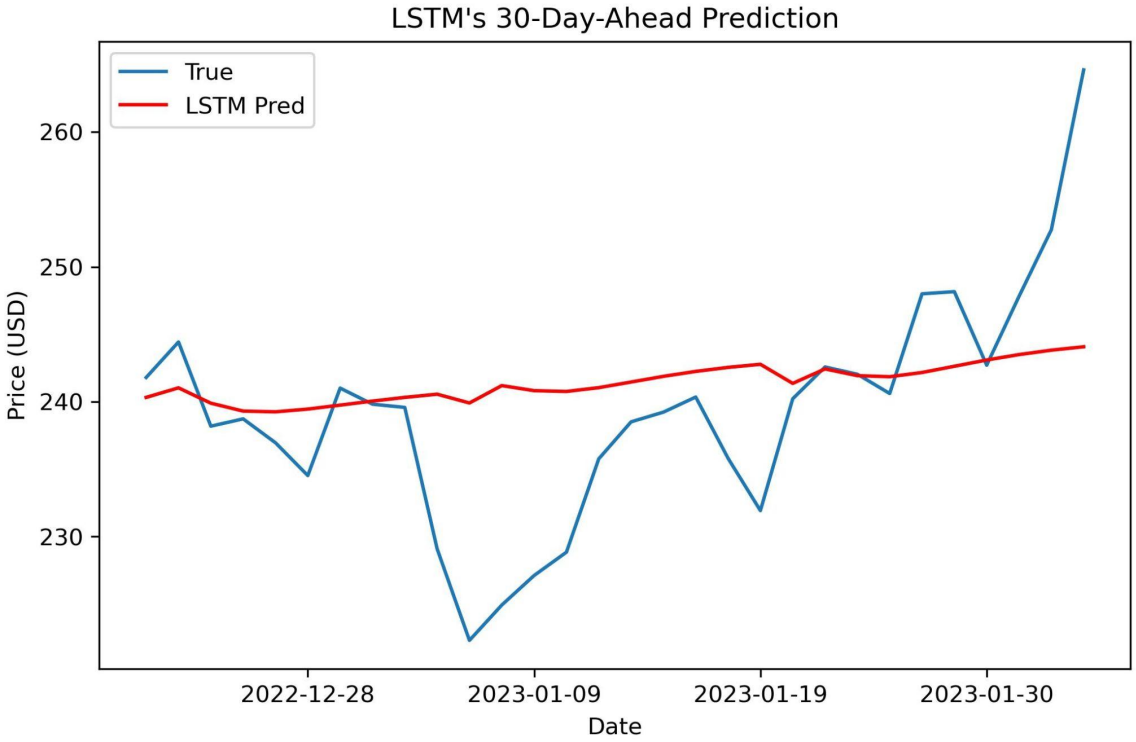
Univariate LSTM Prediction



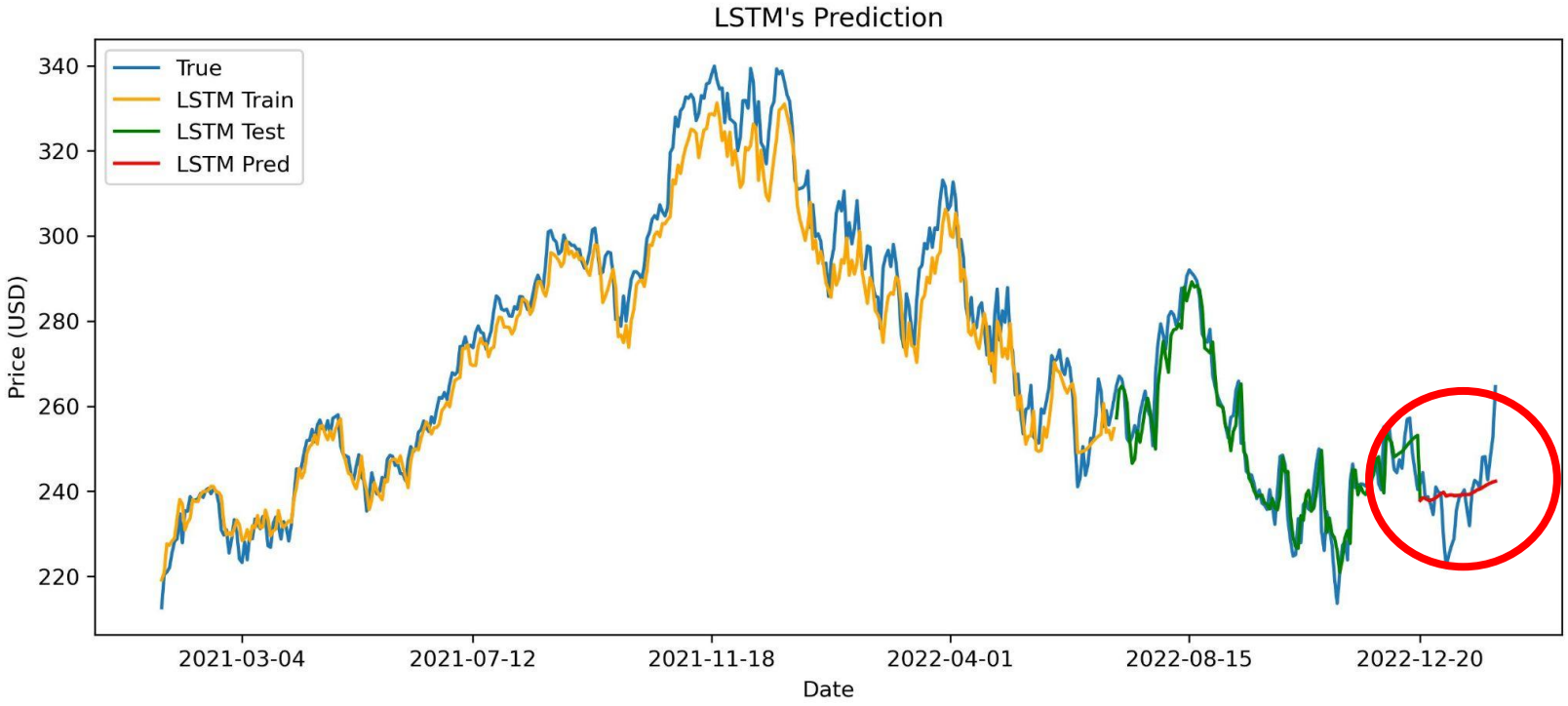
Multi-Step Multivariate LSTM without Sentiment Prediction



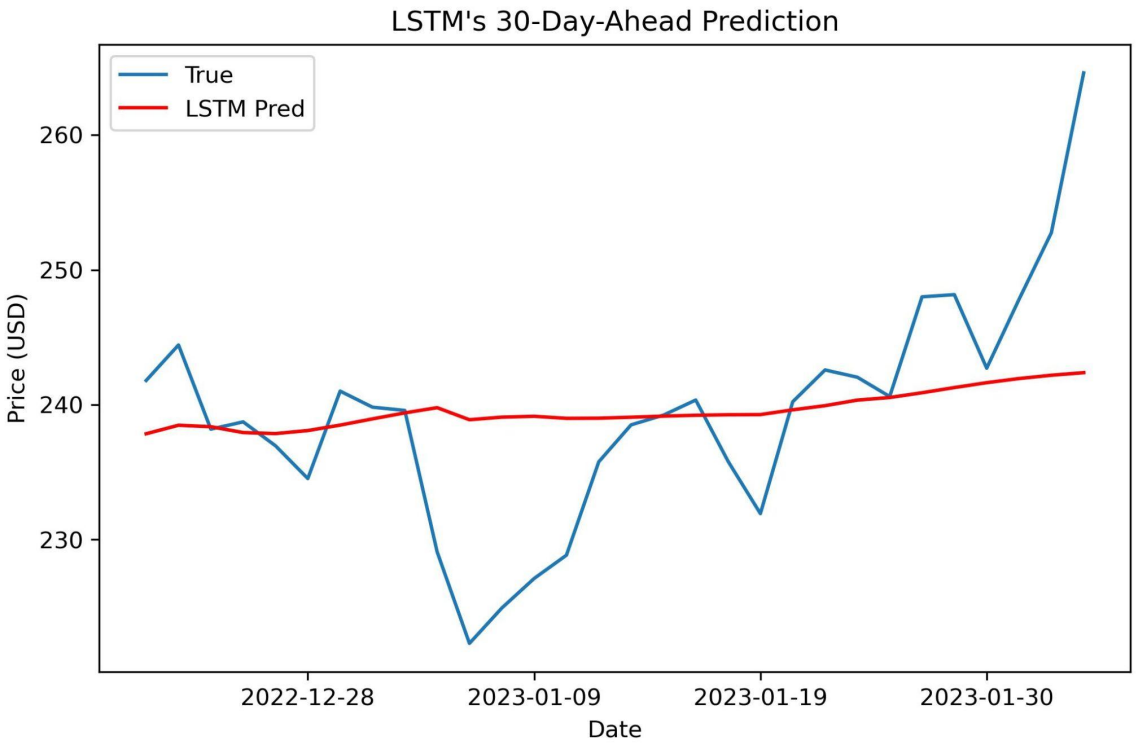
Multi-Step Multivariate LSTM without Sentiment Prediction



Multi-Step Multivariate LSTM with Sentiment Prediction



Multi-Step Multivariate LSTM with Sentiment Prediction



Results

30-day Out-of-sample Prediction Error

	MAE	MAPE
VAR	21.29	8.7%
Univariate LSTM	6.86	2.85%
Multi-Step Multivariate LSTM w/o Sentiment	6.63	2.80%
Multi-Step Multivariate LSTM w/ Mean Sentiment	**5.98**	**2.51%**
Multi-Step Multivariate LSTM w/ Median Sentiment	6.11	2.67%

Result Summary

- Multivariate LSTM models based on our accuracy metric clearly outperforms other models.
- Univariate LSTM model clearly overfits the training dataset but performs better than VAR.
- VAR model cannot capture non-linearity as well as LSTM models.

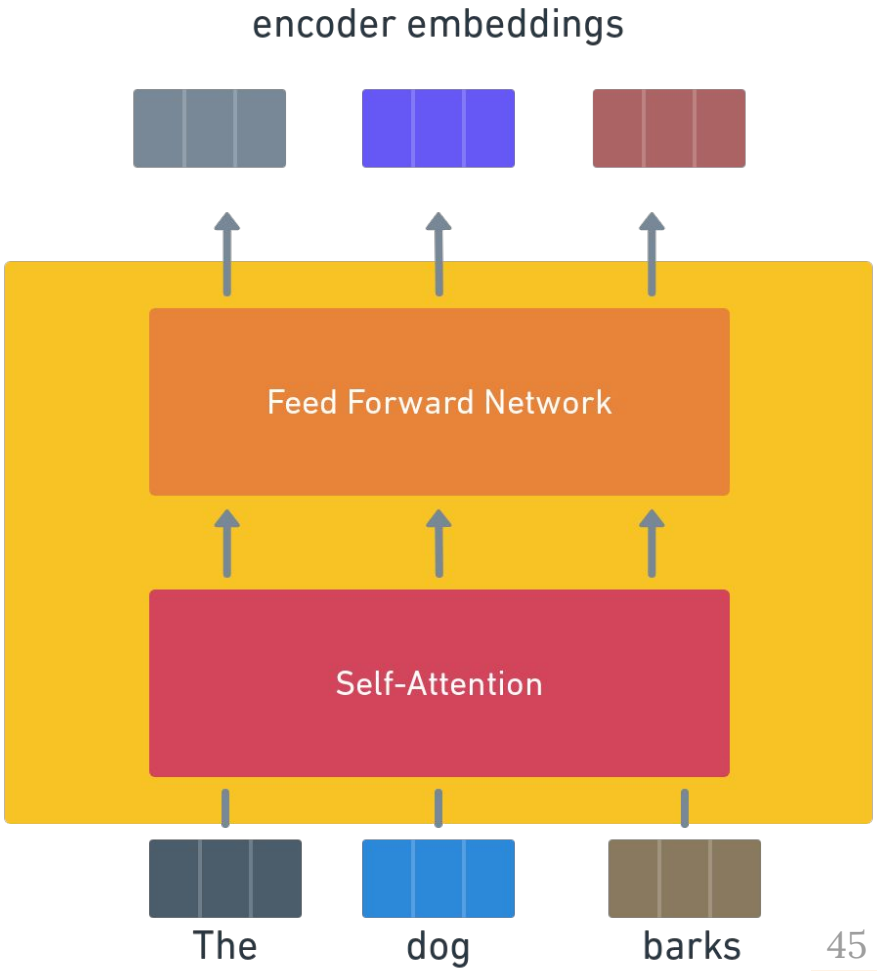
Limitations

- The stocks data are only limited to one index (MSFT).
- Tweets can contain irrelevant information and noisy data.
- Our current implementation of the model is not space and time optimized.
- More granular data (hourly) can improve the performance of our model.

Future work

Transformer-based Language Models

BERT (Bidirectional Encoder Representations from Transformers)



Acknowledgement

Deepak

Elon Musk

The Statistics Department

Mike Tie

Our family and friends

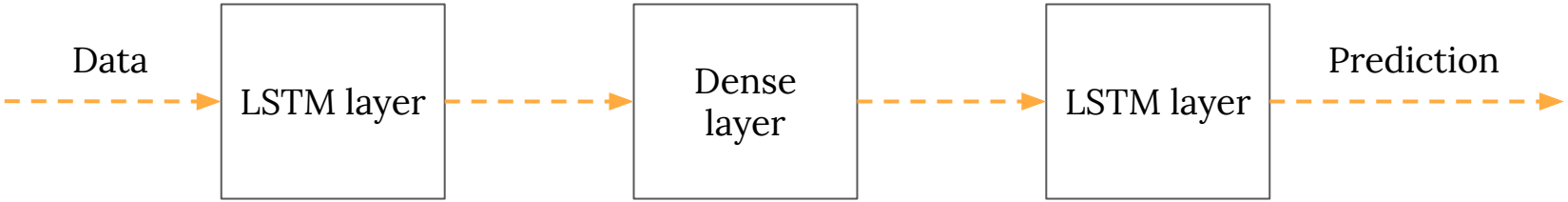
ChatGPT

Thank you :)

Questions for us?

APPENDIX

Multi-Step Multivariate LSTM Model Architecture



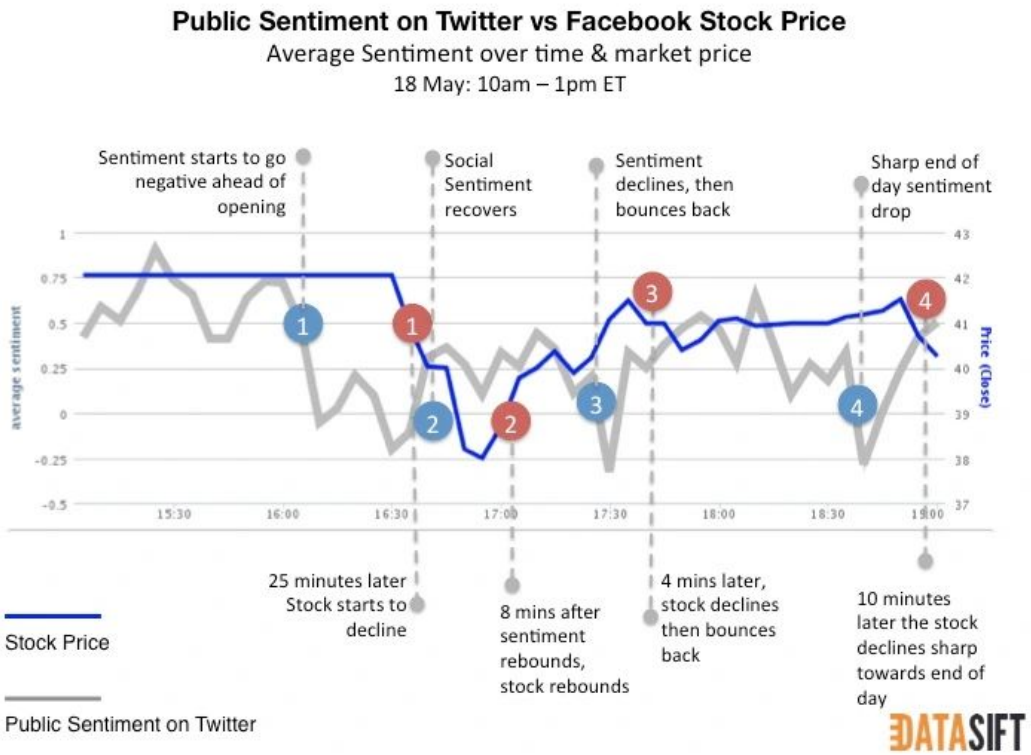
LSTM Hyperparameters and Model Summary

- Keras Tuner
- Random Search

LSTM Hyperparameters Optimization	
Hyperparameters	Value
Number of LSTM Layer	2
1st LSTM Units	336
2nd LSTM Units	80
Number of Dense Layer	1
1st Dense Units	256
Dense Layer Activation Function	Tanh
Number of Dropout Layer	2
1st Dropout Rate	0.1
2nd Dropout Rate	0.05

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 336)               460992

 repeat_vector (RepeatVector  (None, 10, 336)          0
 )

 dropout (Dropout)           (None, 10, 336)           0

 dense (Dense)               (None, 10, 256)           86272

 dropout_1 (Dropout)         (None, 10, 256)           0

 lstm_1 (LSTM)               (None, 10, 80)            107840

 time_distributed (TimeDistr  (None, 10, 6)            486
 ibuted)

=================================================================
Total params: 655,590
Trainable params: 655,590
Non-trainable params: 0
_________________________________________________________________
```

Sentiment Analysis

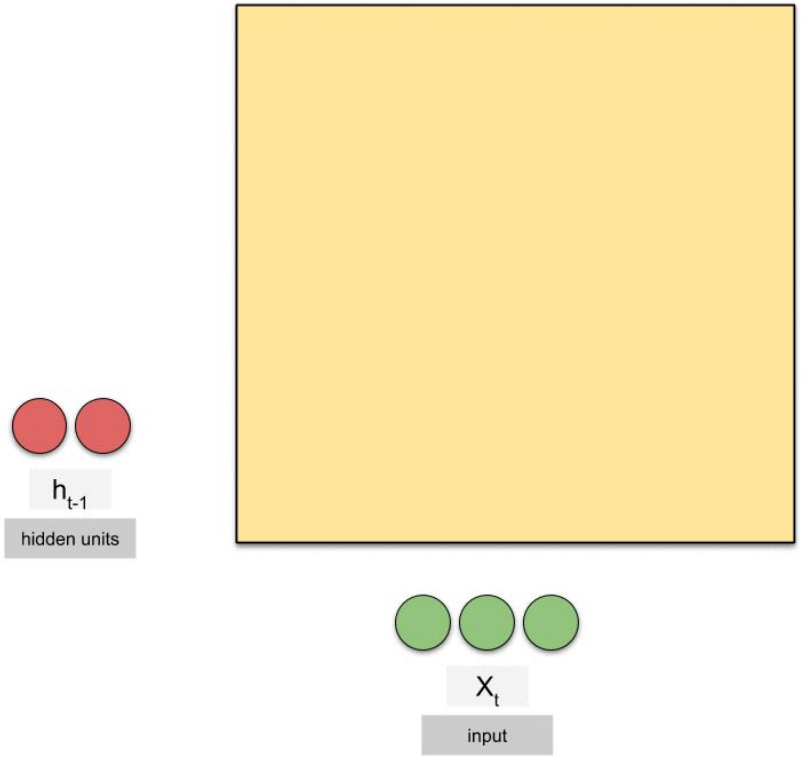


Facebook stock price mirrors Twitter sentiment throughout the day

Of course, sentiment is not the only factor that affects stock price, but this influence is pretty significant

Lardinois, F. (2012, May 18). *Public Sentiment on Twitter vs Facebook Stock Price*. TechCrunch. https://techcrunch.com/2012/05/18/study-twitter-sentiment-mirrored-facebooks-stock-price-today/

Recurrent Neural Networks



Word Embedding Layer

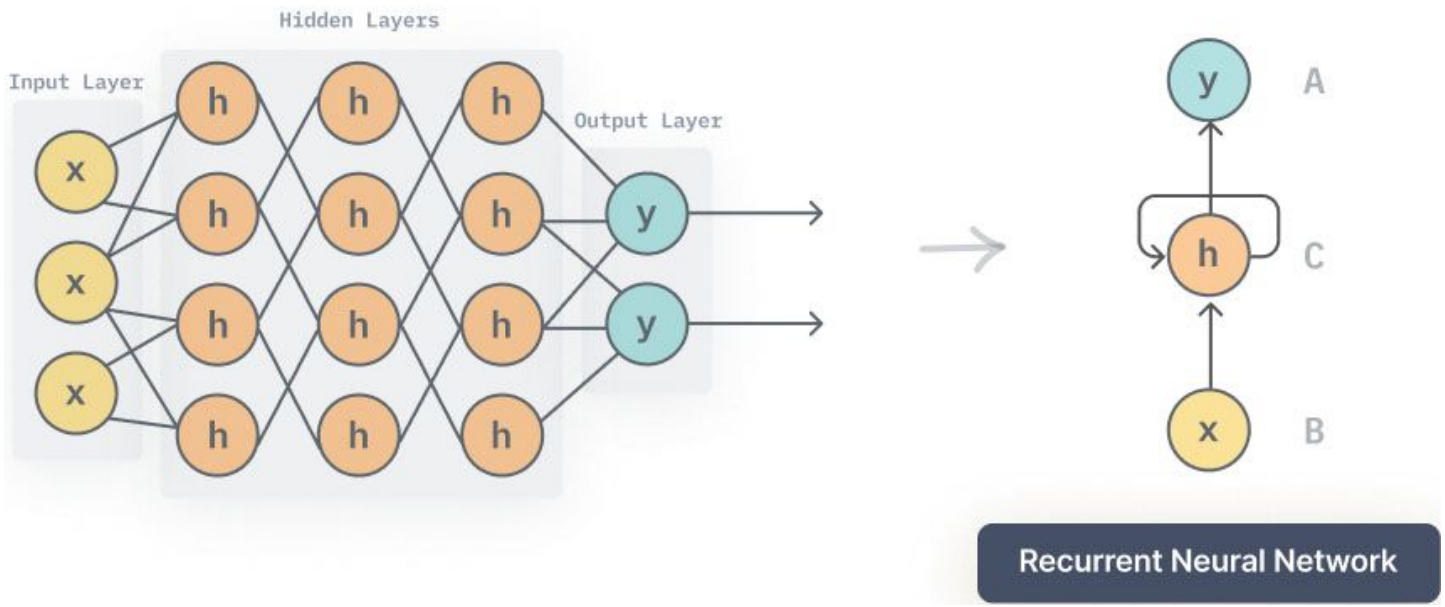
[1] I love my dog.
[2] I like ice cream.

[1] i
[2] love
[3] my
[4] dog
[5] like
[6] ice
[7] cream

Word vector size = 4

	[, 1]	[, 2]	[, 3]	[, 4]
[1,]	1	2	3	4
[2,]	1	5	6	7

Recurrent Neural Networks



RMSE

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$$