CFA Society India

NEURAL NETWORKS IN ACTION

Kriti Mahajan

11th April 2020

Summary of today's talk

1. Linear Regression
2. Structural Models – ARIMA, SARIMA
3. Neural networks
4. Convolutional neural networks, Long Short Term Memory neural networks
5. Tools used in the session:

   Python (Anaconda and Jupyter notebook)
   Tensorflow & Keras

Before we get started, some common notation

- $X_{p,i} = i^{th} observation\ of\ p^{th}\ independent\ variable$
- $Y_i = i^{th}\ observation\ of\ dependent\ variable$
- $\hat{Y}_i = predicted\ value\ of\ i^{th}\ observation\ of\ dependent\ variable$
- $P\ = total\ number\ of\ independent\ variables$
- $N = total\ number\ of\ observations$ (i=1..N)
- $w = weight$

Traditional Approaches to Forecasting

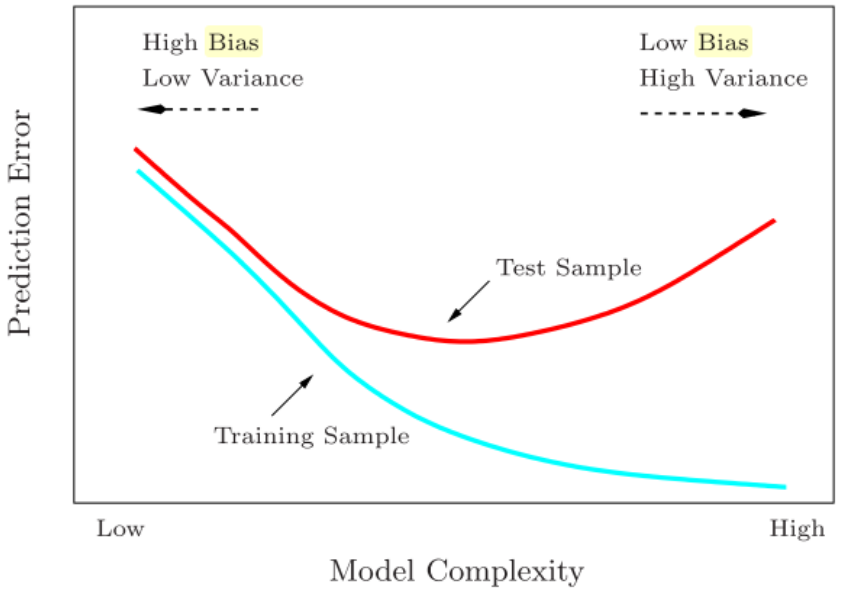
Structural	Non-structural/Time Series methods
• A theoretical model of the world is formulated and used as the basis for empirical forecasting • Often makes key assumptions about behaviour of agents	• Models that do not rely on theoretical models. • Focus on statistical models best forecast given the underlying data
Give conditional forecasts	Give unconditional forecasts
Example: DSGE	Example: Linear Regression , Autoregressive Integrated Moving Average (ARIMA) models , Vector Autoregressive (VAR) models - Require multiple decisions regarding model specification ↓ decisions force researcher to take strong positions on nature of underlying data generating process. ↓ **This is a problem because researchers choose non-structural forecasting approaches explicitly to avoid taking these types of positions.**

Potential Solution : Machine Learning Methods

Why Machine Learning?

Model	Key Features vis-à-vis Forecasting
Traditional	• Based on BLUE and thus due to the bias variance trade off "do not yield the most accurate predictions"(Chalfin et al. 2016).
Machine Learning	• "Designed for prediction… use the data to …trade off bias and variance to maximize out-of-sample prediction accuracy." (Chalfin et al. 2016) • No assumptions regarding (Smalter Hall and Cook, 2017): a) the underlying data generating process b) the underlying relationship with the independent variables • Models non-linearities automatically

Figure 4 : Bias Variance Trade Off



Source: Friedman, Hastie and Tibshirani (2001)

Expected prediction error = irreducible error + bias + variance

Fitting A Machine Learning Model

Objective of Machine Learning: Find a function that is 'generalizable' . How?

Step 1 Specify Hyper-parameters (Input arguments defining model structure/architecture) → Determined using **Hyper Parameter Search**

$$\begin{bmatrix} x_{1,0} & x_{1,1} & \dots & x_{1,n} \\ x_{2,0} & x_{2,1} & \dots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,0} & x_{m,1} & \cdots & x_{m,n} \end{bmatrix} \text{ and } \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix}$$

Step 2 Learn Parameters

$$\begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_n \end{bmatrix}$$

Step 3 Build Function

$$f(x;\theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

Step 4 Prediction

$$\hat{Y}_i$$

Training accuracy can be arbitrarily high through over fitting. Thus:

a) Each machine learning model needs a stopping/penalization/regularization criterion for in-sample training
b) Forecast accuracy must be measured out-of sample

Fitting A Machine Learning Model

Hyper Parameter Search

- The process of searching for the hyperprameters that result in ‘ideal’ model architecture (i.e. the model architecture that results in the highest predictive accuracy) is known as **hyper-parameter tuning**
- Hyper-parameter tuning can be done
  - Based on previous literature
  - Manually
  - Automatic search (Grid Search and Random Search)
  - Bayesian Optimization
- **Objective is to find hyper-parameters which work no matter what the underlying data.**
- Finding a generalizable set of hyper parameters requires careful specification of the **training and testing period**.

Fitting A Machine Learning Model

Training, Validation and Testing

- To measure the predictive accuracy of a model, the forecast accuracy must be measured out-of sample as the training accuracy can be made arbitrarily high through **over-fitting**.
- However, if we use the entire out-of-sample data for testing, we may over-fit to the out-of-sample data (a phenomenon known as ‘data leakage’), resulting poor true generalizability i.e. your model may not perform well on true unseen data
  Solution : Split out-of-sample data into two parts:
  - validation data
  - testing data

  Allows evaluation of the model on unseen data (validation data) to select the best model architecture, while still holding out a subset of data (testing data) for final evaluation after finding the best model.

Fitting A Machine Learning Model

Training, Validation and Testing (Contd.)

- The training, validation and testing data can be organized in many ways, namely:

  a) Cross validation (bootstrap sampling) : Not suitable for time series data

  b) Fixed window (training, validation and testing periods demarcated by dates)

  c) Rolling window/expanding window : Most suitable for time series data

Data Preparation

1) Deal with missing values
2) Make data stationary
3) Lag the independent variables
4) Split into training, validation and testing set
5) Normalize the data using the training sample
6) Standardize the data using the normalized training sample :For the deep neural networks (DNNs), after normalization, the data is rescaled to a suitable range (-1 to 1 in our case) as DNNs are not invariant to the magnitude of the data.

Note: The normalizing constants (mean and standard deviation) and the rescaling parameters from the training sample are used to normalize and rescale the data in the validation and testing samples to avoid look forward bias in the out-of-sample data.

So why is Linear Regression (OLS) not best suited for forecasting Time Series Data?

Properties of Time Series Data	Meaning?	Can OLS deal with this?	Why/Why Not?
Sequentially ordered	Outcome today depends on outcome yesterday	No	• Order of the data does not matter • OLS requires no correlation between observations of the error term (no autocorrelation)
Displays trends	Different data series may show high correlation even if their non-trend component is not correlated (i.e. spurious correlation)	No	• OLS requires no perfect correlation between independent variables (no multicolinearity)
May not be stationary	Mean of the data changes over time	No	• OLS requires that the variance of error does not change for each observation/ a range of observations (no heteroskedasticity)

Source: Author compiled ; *Basic Econometrics* ,Damodar N. Gujarati ; CMU Lecture notes

Potential Solution: SARIMA

What is SARIMA?

- ARIMA models describe how each successive observation is related to the previous observation
- The seasonal ARIMA (SARIMA) is capable of modelling the seasonal components in a univariate time series in addition to the autoregressive, moving average and trend components typically modelled by ARIMA.
- SARIMA is given by the following notation

$$ARIMA(p,d,q)(P,D,Q)m$$

p : trend autoregressive order

d : trend difference order

q : trend moving average order.

Determined using:
- Autocorrelation Function (ACF)
- Partial Autocorrelation Functions (PACF)
- Tests for stationary.

P : number of seasonal autoregressive terms

D : number of seasonal difference terms

Q : number of seasonal moving average terms

m : number of time steps for a seasonal period

SARIMA vs. OLS for Time Series Data

	Properties of Time Series Data	OLS	SARIMA
Differences	Sequentially ordered	No	Yes, but only for dependent variable
	Displays trends	No	Yes, but only for dependent variable
	The data may not be stationary	No	Yes, but only for dependent variable
Similarities	Linear	Yes	Yes
	Requires no. of obs. be less than no. of independent variables	Yes	Yes
	Utilizes sequential nature of independent variables	No	No

SARIMA : Disadvantages

General Disadvantages of ARIMA models

Forecasting Steps for ARMA Models

• Process:

- ARIMA model: $Y_t = \phi Y_{t-1} + a_t$

$\Downarrow$

- Estimation (Evaluation in-sample): $\hat{\phi}$ (Estimate of ϕ)

$\Downarrow$

$\hat{Y}_t = \hat{\phi} Y_{t-1}$ (Predictio n)

$\Downarrow$

- Forecast (Evaluation out-of-sample): $\hat{Y}_{t+1} = \hat{\phi}\hat{Y}_t$ (Forecast)

Source: Bauer College, RS –EC2 Lecture Notes

- Error propagation
- Backward looking: “they are generally poor at predicting turning points (unless the turning point represents a return to a long-run equilibrium)”.
- Mean-reverting: As the number of steps forecasted ahead in the future increases , the forecast converges to the mean

Potential Solution : Deep Neural Networks

Why Only Focus on Neural Networks?

Brief Overview of Machine Learning Methods

Category	Method	Type	Customized for Time Series?
Shrinkage Model	Elastic Net	Linear	No
Tree Based Models	Random Forests	Non-Linear	No
	XG Boost	Non-Linear	No
Neural Networks	CNN	Non-Linear	Yes
	LSTM	Non-Linear	Yes

Source: Author Complied

What is a Neural Network?

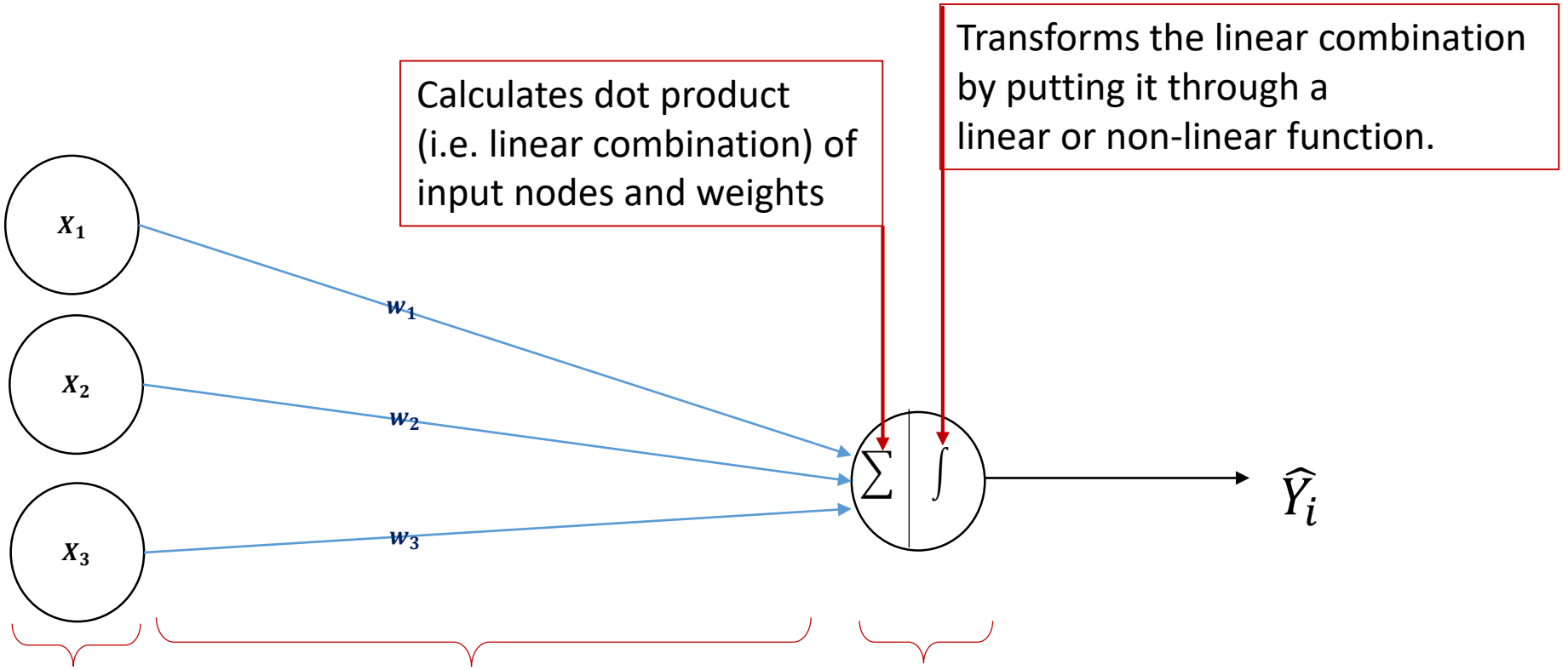
- **In general, neural networks are composite functions which are universal function approximators.**

- Every neural network is broadly composed of three types of layers:
  a. the input layer
  b. the hidden layer / the computational layer
  c. an output layer

- Each of the 3 layers is comprised of multiple nodes and is connected to the subsequent layer through weights

Neural Networks (DNNs): Perceptron

Linear Regression as a Neural Network



Input layer

- Each node is an independent variable.

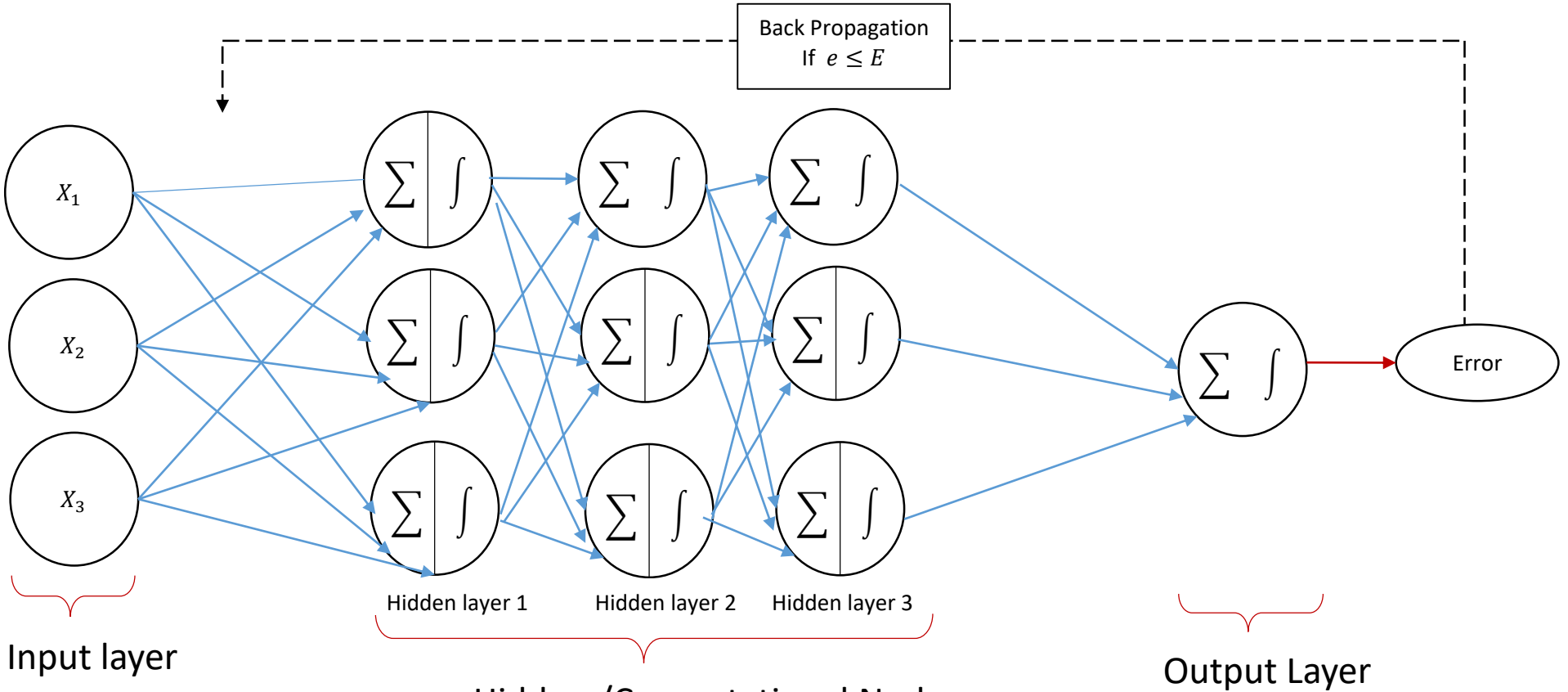
Weights

Hidden / Computational Node

- Data is revealed to each hidden node simultaneously.

Deep Neural Networks (DNNs)

Where does the 'learning' in a neural network come from?



- Data is revealed to each hidden node simultaneously
- As a result the data is treated as being cross sectional

Feed Forward Mechanism

But Not All Neural Networks Are Suited for Time Series…

- The input layer in a feed forward neural network is a column vector which means that the data is revealed to each hidden node simultaneously.
- As a result the data is treated as being cross sectional because the spatial and sequential nature of the data is not exploited.
- Thus, it is not recommended that a fully connected feed forward neural network because it is dominated by more complex deep neural networks that can infer information from time series data
- Today we consider :

- Convolutional Neural Networks (CNNs)
- Long Short Term Memory (LSTM) Networks

What makes CNNs and LSTMs different?

- Like OLS and SARIMA, conventional neural networks depend on hand-selected independent variables supplied manually
  - They do *not* use these independent variables to create *new independent variables*
- However, both CNNs and LSTMs can “learn an internal representation of the time series data and ideally achieve comparable performance to models fit on a version of the dataset with engineered features”
  - Briefly: CNNs and LSTMs extract (learn) new features (independent) variables from the independent variables supplied manually

How do CNNs and LSTMs extract new information from the independent variables supplied?

1) The input layer preserves the temporal structure of data

2) The hidden layer/s use complex computational nodes:
- The different computational nodes used in a CNN vis-à-vis a LSTM network result in the two type of networks processing the data differently.

- Both CNNs and LSTMs are followed by a regression layer which gives the final output.
  - the regression layer is a fully connected feed forward neural network followed by the output layer
- The feed forward mechanism and modified versions of back propagation are used to train the model

Convolutional Neural Networks (CNNs)

- Convolutional neural networks (CNNs) can extract information from the temporal structure of the data by
  a) preserving the spatial/ temporal structure of the data in the input layer
  b) using filters which look for patterns in spatially adjacent data
  - For instance, one filter could finding pseaks, another could find troughs while another could find a linear trend.
  - The information extracted by the filters is known as a feature map.
  - Each additional feature results in a new feature map
- The CNNs layer is may be followed by a sub-sampling layer
  - reduces the noise in the learned features (i.e. the feature maps)
- The sub-sampling layer is followed by a regression layer.

Convolutional Neural Network (CNN)

What does preserving the temporal structure of data mean?

$$\begin{matrix} x_{1,1} & x_{2,1} & x_{3,1} \\ x_{1,1+1} & x_{2,1+1} & x_{3,1+1} \\ x_{1,1+2} & x_{3,1+2} & x_{3,1+2} \end{matrix}$$

Input layer

The input layer of a CNN preserves the temporal structure of the data by accepting data in a 3 dimensional format:

$width$: number of the independent variables

$height$: number of observations we assume to be related across time

$depth$: equals 1 for time series data

Long Short Term Memory (LSTM) Network

What are they?

- Shortcoming of CNNs for time series data: they do not draw information from the sequential nature of the data independent
- Solution : Long Short Term Memory (LSTM) Network
- LSTM networks draw information from the sequential nature of the data because they have memory.
- LSTMs belong to a larger class of neural networks known as Recurrent Neural Network, all of which have memory.
- Of many RNNs , LSTM chosen here over a vanilla RNN as the former can learn from long sequences while the latter may not.

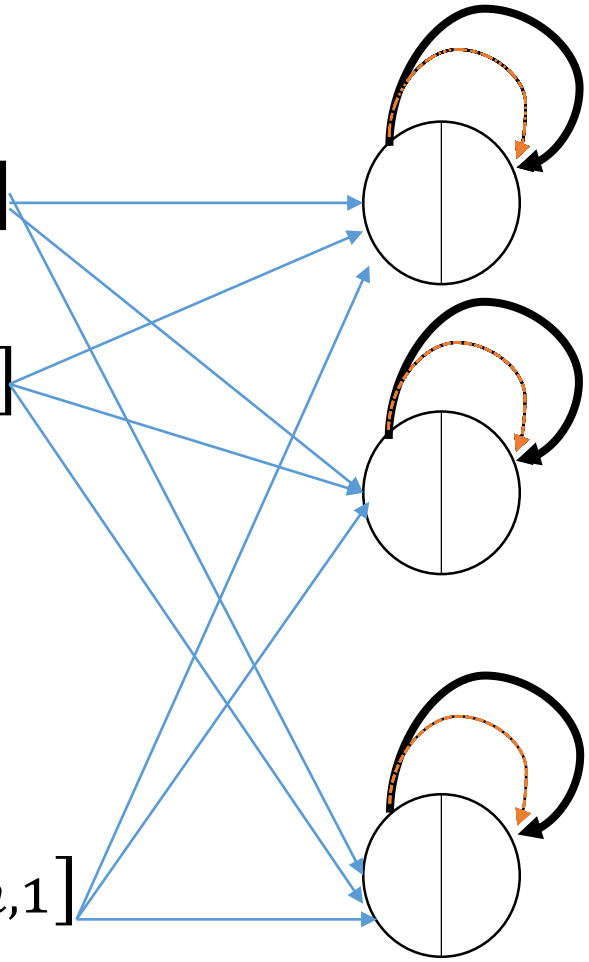
LSTM (Contd.)

How does it build short and long term memory? Sequential Revealing of Data

$[x_{1,1+2} \quad x_{1,1+1} \quad x_{1,1}]$

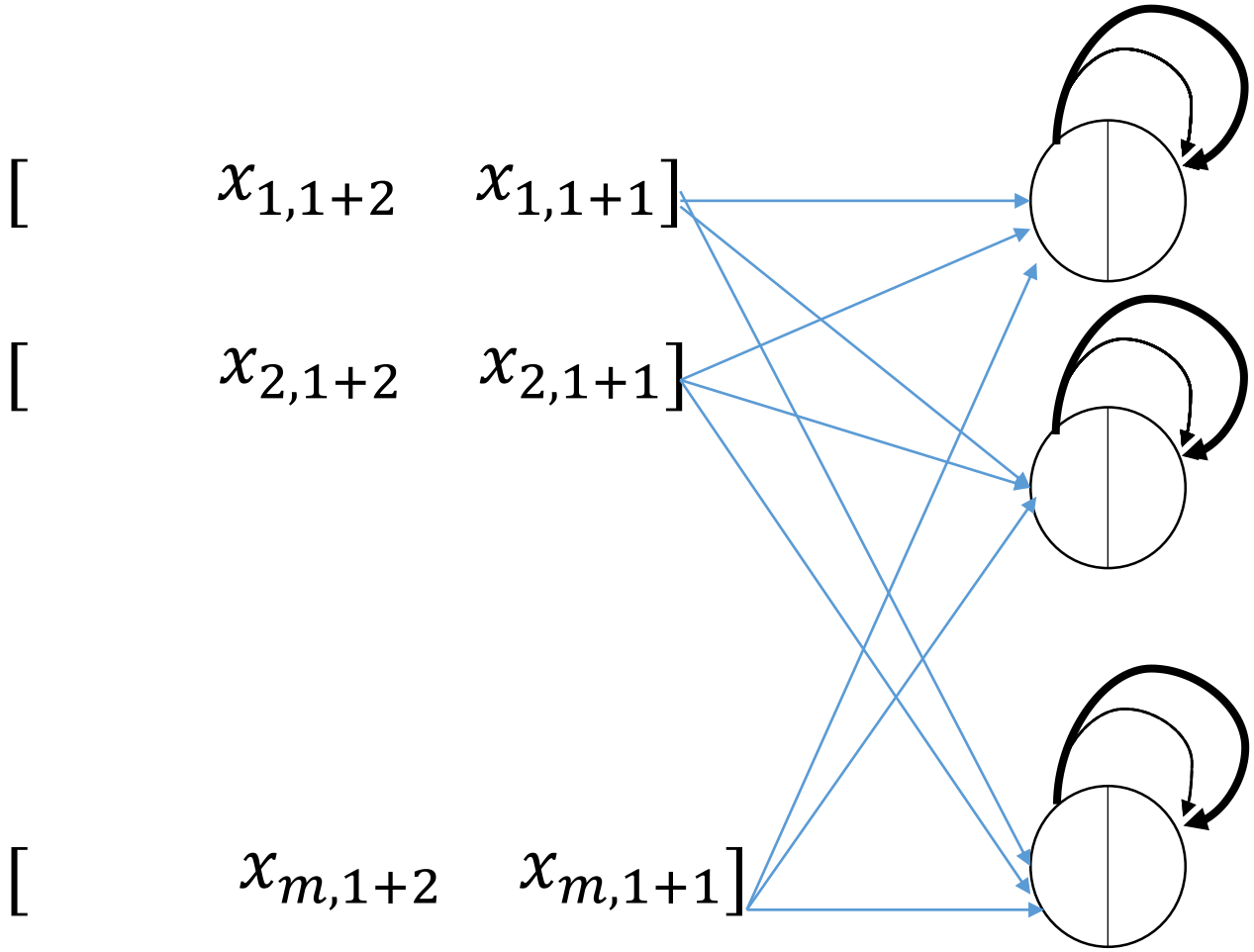
$[x_{2,1+2} \quad x_{2,1+1} \quad x_{2,1}]$

$[x_{m,1+2} \quad x_{m,1+1} \quad x_{m,1}]$



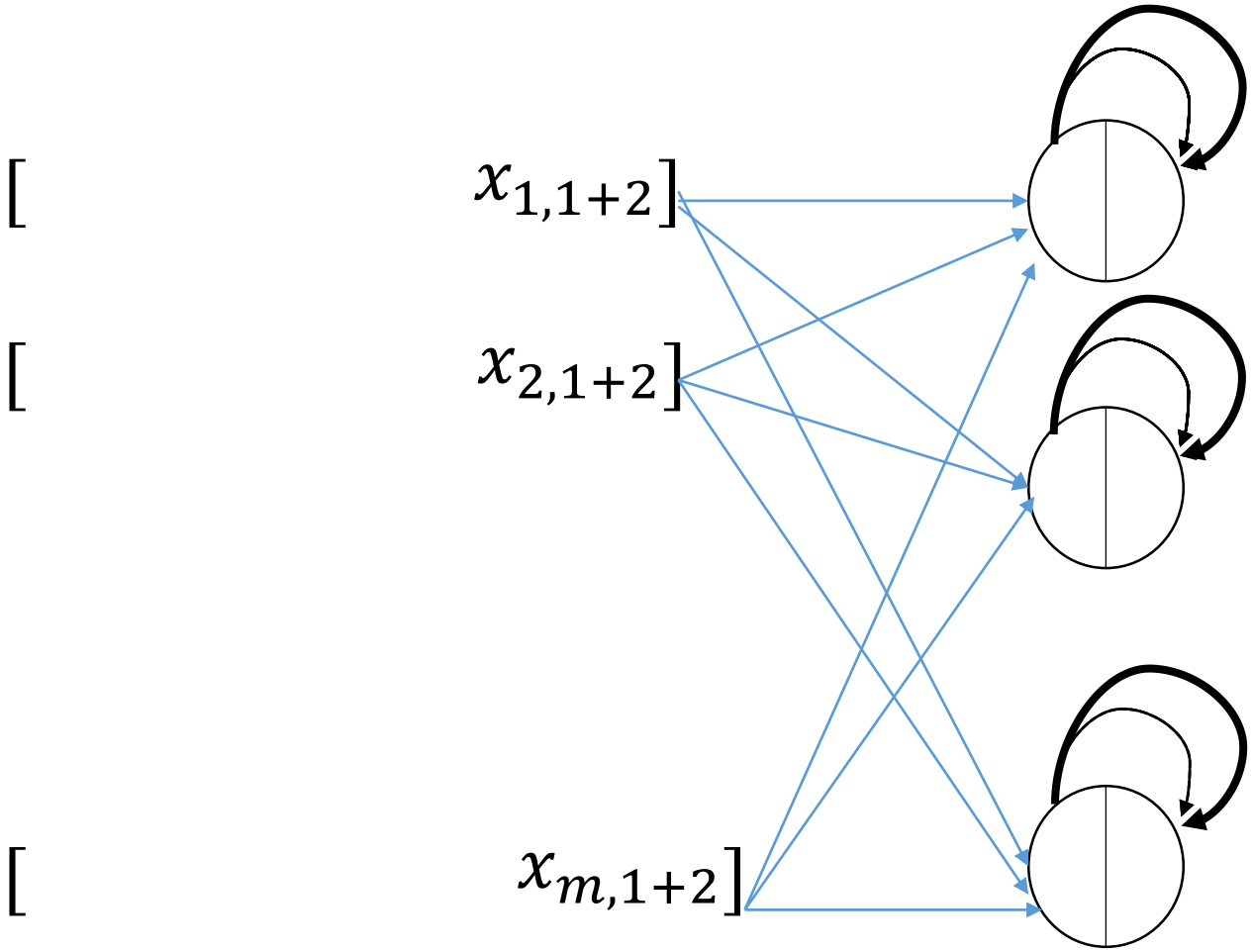
LSTM (Contd.)

How does it build short and long term memory? Sequential Revealing of Data



LSTM (Contd.)

How does it build short and long term memory? Sequential Revealing of Data



How does an LSTM Node Build Memory?

Components of an LSTM cell

1) Component 1 : the long term memory

- represents the understanding of the LSTM network at the current member of the sequence given the information that has been revealed to it.
- is updated upon being exposed to each subsequent element of the sequence , conditional on the output of each 'gate'.

2) Component 2: "gates"

- Each "gate" is a neural network with a sigmoid/logistic activation function which determines how the long term memory is updated by accepting as inputs:
    a) the current element of the sequence
    b) the outputs of the previous LSTM cell ($st-1$ and $ht-1$) to.
    c) Each gate outputs a number between 0 (meaning no information is transferred to **long term memory**) and 1 (meaning all information is transferred to **long term memory**)
- The gates put a constraint on how the long term memory is updated : In their absence, the long term memory would change very chaotically.
- The Forget Gate / Remember Vector
- The Input Gate / Save Vector
- The Output gate / Focus Vector .

3) Component 3: the short term memory

How does an LSTM Node Build Memory?

Components of an LSTM cell (Contd.)

- **3) Component 3: the short term memory**

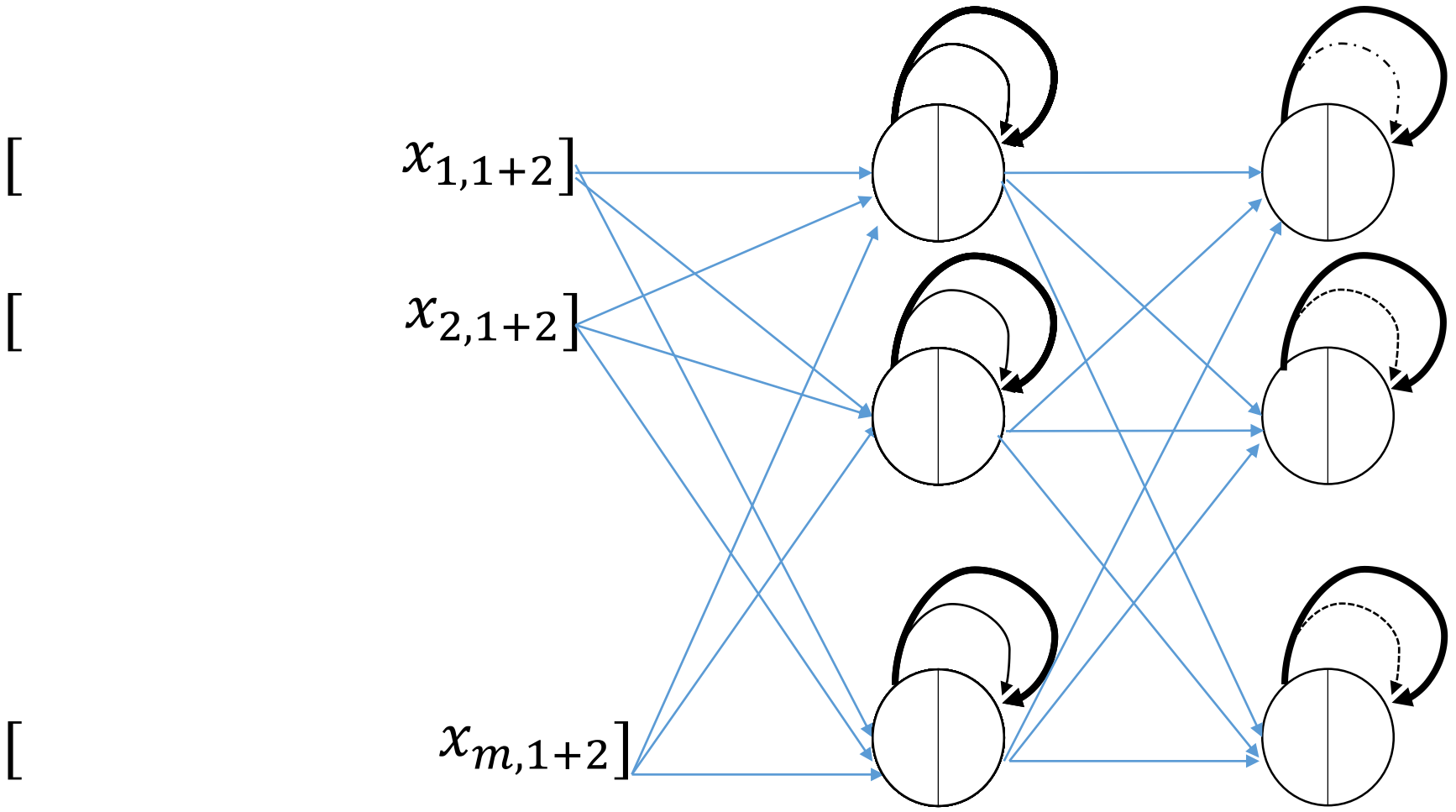
  - Based on the updated long term memory, the LSTM cell updates the working memory.

  - The updated working memory at the end of the sequence is the output of an LSTM node.

  - This working memory or hidden memory is the new internal representation of the data that is learnt by an LSTM network i.e. the working memory is the new feature used as an input in the regression layer.

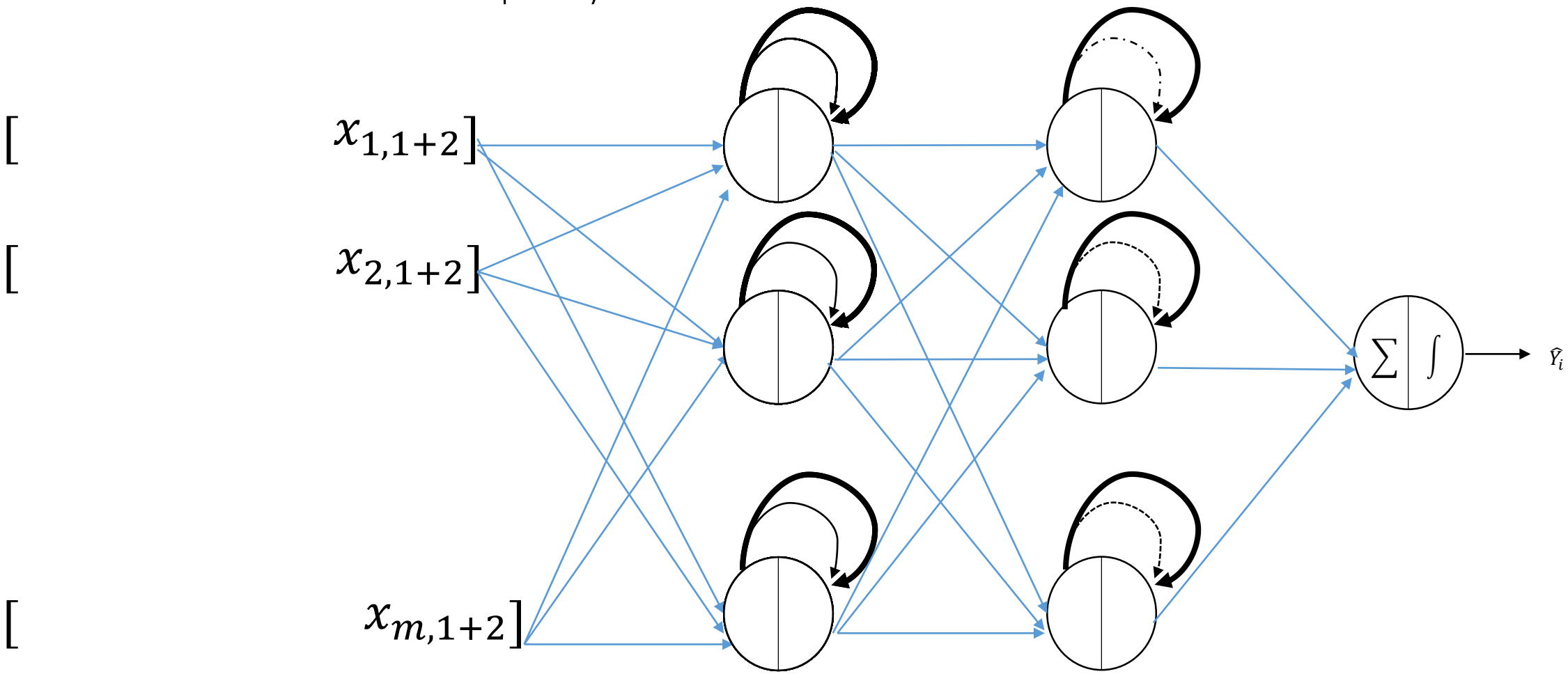
LSTM (Contd.)

An LSTM Network Can Have Multiple Layers



LSTM (Contd.)

An LSTM Network Can Have Multiple Layers



This completes one forward pass of the data and is followed by back proapagation to update the weights

Overview of All the Methods Considered

Nature of Time Series Data	OLS	SARIMA	FFN	CNN	LSTM
Sequentially ordered	No	Yes, but only for Y	No	No	Yes
Displays trends	No	Yes, but only for Y	Ambivalent	Yes	Yes
The data may not be stationary	No	Yes	No	Yes	Yes
Requires no. of obs. to be less than no. of independent variables	No	No	Yes	Yes	Yes
Linear	Yes	Yes	No	No	No
Temporal structure of the data	No	No	No	Yes	Yes
New Features Created	No	No	No	Yes (Feature Maps)	Yes (Working Memory)

Practical Considerations

Which hyper-parameters need to be fine tuned?

Hyper-parameter	Hyper-parameter Domain	CNN	LSTM
Patience	0, ∞	x	x
Learning Rate	0,∞	x	x
No. of Epochs	0,∞	x	x
Optimizer Type	SGD/ADAM	x	x
Batch Size	1, No. of Obs.	x	x
No. of Steps In	1, No. of Obs. in Test Set	x	x
No. of Conv. Layers	1, ∞	x	
Conv. Activation Layer Type	Linear/ Tanh /Logistic/ReLU	x	
No. of Filters per Conv. Layer	1, ∞	x	
Filter Size	1, No. of Steps In*No. Independent Variables	x	
Stride Size	1, ∞	x	
Sub-sampling Layer type	Max Pooling / Average Pooling	x	x
Sub-sampling Layer Size	1, No. of Steps In – Filter Size	x	x
No. of Dropout layers	0,No. of Hidden Layers	x	x
Dropout Percentage	0,1	x	x
No. of Full Connected Hidden Layers	0, ∞	x	x
No. of Hidden Nodes	0, ∞	x	x
Batch Normalization	Yes/No	x	x
Output Layer Activation Type	Linear, Tanh, Logistic, ReLU	x	x
No. of LSTM layers	1, ∞		x
No. of LSTM nodes	1, ∞		x

Source: Author Complied

Sources

- Tibshirani and Friedman (2017)
- Hastie, James , Tibshirani and Witten (n.d.)
- Goodfellow , Bengio and Courville (2016),