Northeastern University

Gesture Phase Segmentation

IE7615 Neural Networks and Deep Learning Project

Amina Tabassum

Zhibek Kassymkanova

Sai Malleswar Manikonda

April 20, 2023

Motivation

- Process of dividing continuous gestures into distinct phases
- Identify boundaries between different gestures

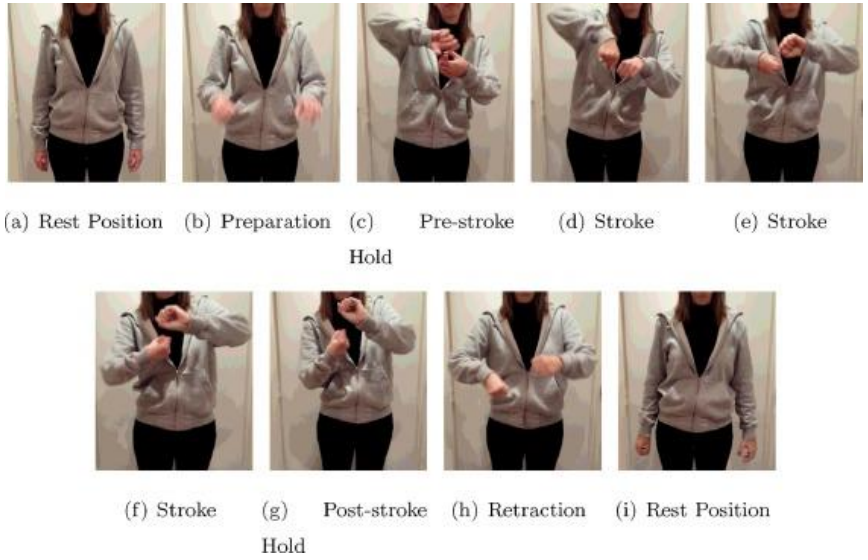

Figure 1: Gestures

Objectives

Gesture phase segmentation on temporal features extracted from video frames

Compare performance of different neural network models

Implementation

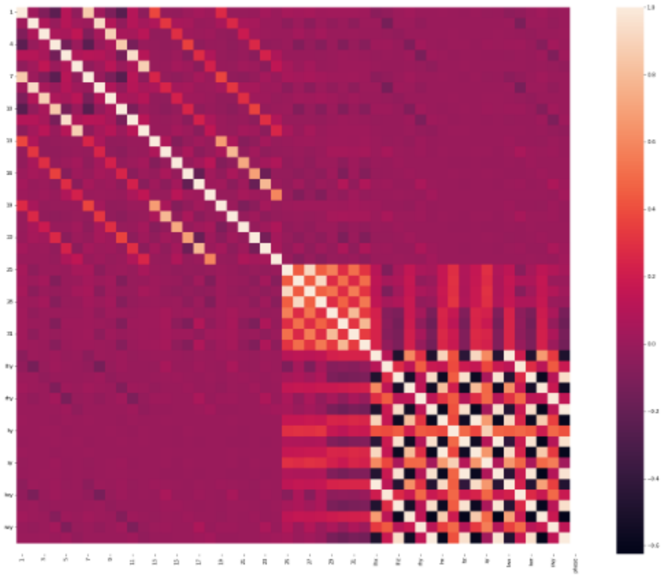

Figure2: Correlation Heatmap

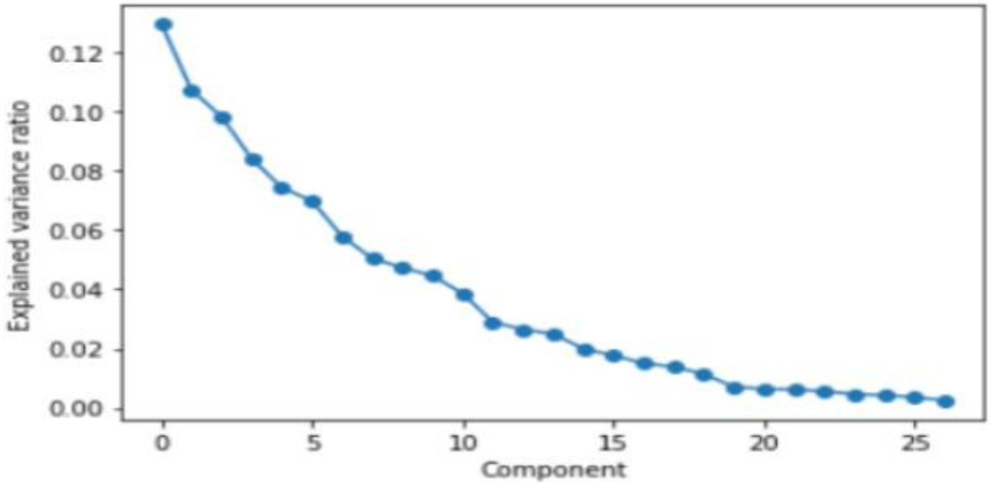

Figure3: Variance and PCA

Implementation

- Inspired by structure and functioning of human brain
- Neurons are fundamental units
- Layers of interconnected nodes
- Feed output of one layer to next layer

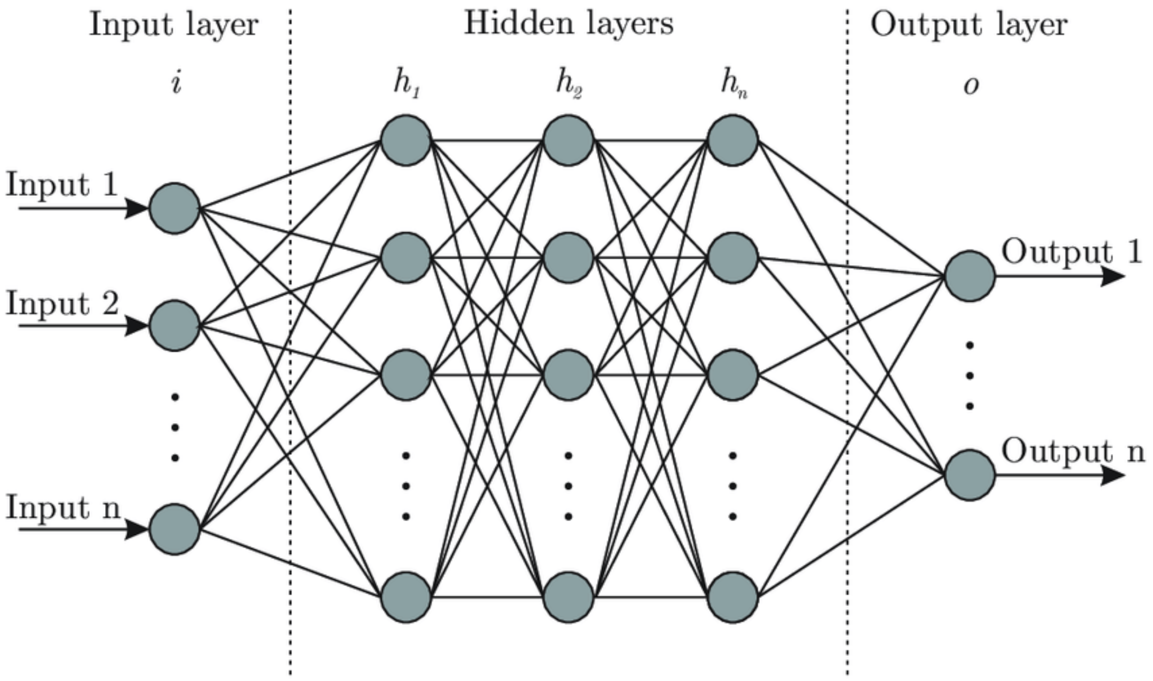

Figure4: Artificial Neural Networks

Implementation

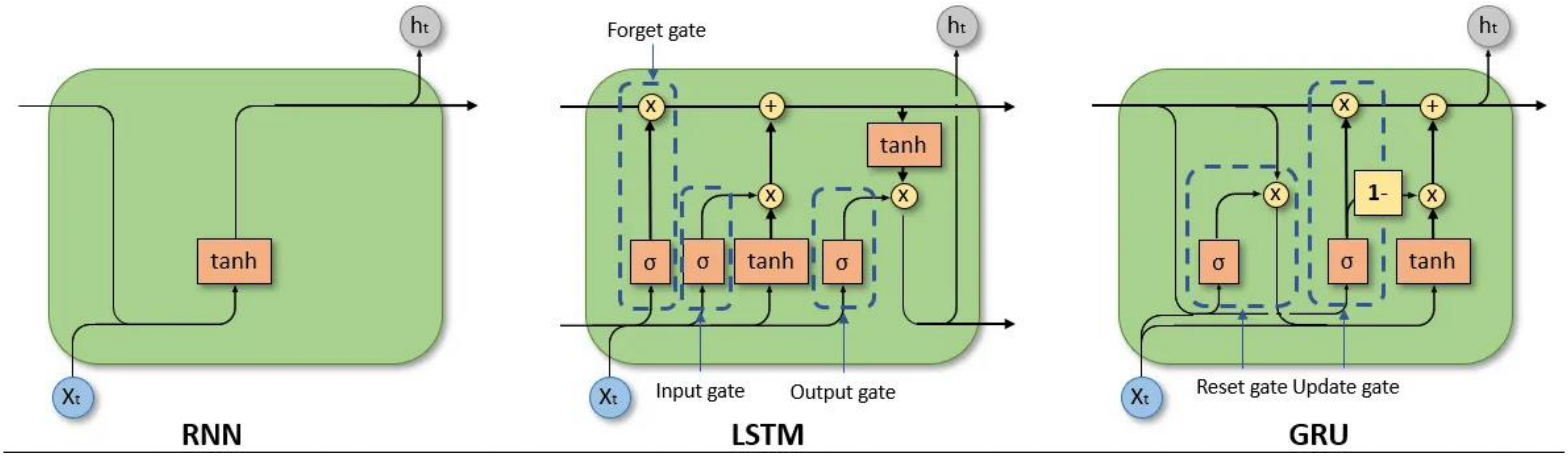

Figure5: Recurrent Neural Networks

Dataset

- Features extracted from 7 videos of people gesticulating for purpose of studying gesture phase segmentation problem
- Raw files are composed of user's position of hands, wrists , head and spine in each frame
- processed files are comprised of information regarding user's hands, wrists' velocity and acceleration

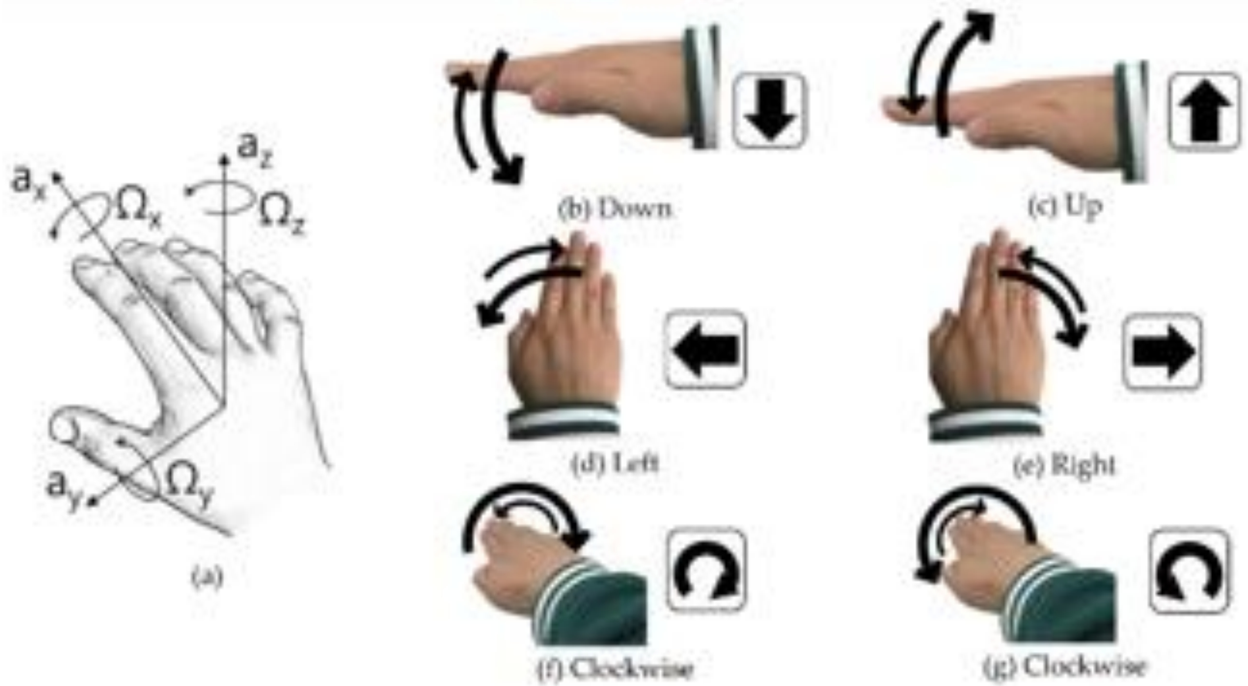

Figure 6

Results

Artificial Neural Networks

- Summary of ANN model

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_566 (Dense)           (None, 512)               3072

 dropout_201 (Dropout)       (None, 512)               0

 dense_567 (Dense)           (None, 256)               131328

 dropout_202 (Dropout)       (None, 256)               0

 dense_568 (Dense)           (None, 128)               32896

 dropout_203 (Dropout)       (None, 128)               0

 dense_569 (Dense)           (None, 256)               33024

 dense_570 (Dense)           (None, 128)               32896

 dense_571 (Dense)           (None, 64)                8256

 dropout_204 (Dropout)       (None, 64)                0

 dense_572 (Dense)           (None, 64)                4160
...
Total params: 248,325
Trainable params: 248,325
Non-trainable params: 0
_________________________________________________________________
```

Artificial Neural Networks

Hyperparameter	Value
Epoch	100
Batch Size	16
L2 Regularizer	0.01
No of principal components	5
Activation function	ReLU
Dropout	0.01
Adam Optimizer	0.01

Table 1: Hyperparameters for ANN for 79.99 % Accuracy

Hyperparameter	Value
Epoch	300
Batch Size	32
L2 Regularizer	0.01
No of principal components	5
Activation function	ReLU
Dropout	0.02
Adam Optimizer	0.01

Table 2: Hyperparameters for ANN for 74.43 % Accuracy

Artificial Neural Networks

Hyperparameter	Value
Epoch	100
Batch Size	8
L2 Regularizer	0.01
No of principal components	5
Activation function	ReLU
Dropout	0.02
Adam Optimizer	0.01

Table 3: Hyperparameters for ANN for 65.55 % Accuracy

Hyperparameter	Value
Epoch	300
Batch Size	16
L2 Regularizer	0.01
No of principal components	5
Activation function	ReLU
Dropout	0.02
SGD Optimizer	0.01 , 0.9

Table 4: Hyperparameters for ANN for 64.33 % Accuracy

Recurrent Neural Networks

Hyperparameter	Value
Epoch	300
Batch Size	32
L2 Regularizer	0.01
No of features	50
Activation function	ReLU, Softmax
Dropout	0.02
Adam Optimizer	0.01

Table 6: Hyperparameters for 'Vanilla' RNN for 66.6 % Accuracy

Hyperparameter	Value
Epoch	100
Batch Size	32
L2 Regularizer	0.01
No of features	50
Activation function	ReLU, Softmax
Dropout	0.02
SGD Optimizer	0.01, 0.09

Table 7: Hyperparameters for 'Vanilla' RNN for 39.9 % Accuracy

Recurrent Neural Networks

Hyperparameter	Value
Epoch	50
Batch Size	32
L2 Regularizer	0.01
No of features	50
Activation function	ReLU, Softmax
Dropout	0.02
Adam Optimizer	0.01

Table 8: Hyperparameters for LSTM RNN for 70.4 % Accuracy

Hyperparameter	Value
Epoch	50
Batch Size	132
L2 Regularizer	0.01
No of features	50
Activation function	ReLU, Softmax
Dropout	0.02
SGD Optimizer	0.01 , 0.9

Table 9: Hyperparameters for LSTM RNN for 71.1 % Accuracy

Recurrent Neural Networks

Hyperparameter	Value
Epoch	100
Batch Size	32
L2 Regularizer	0.01
No of features	50
Activation function	ReLU, Softmax
Dropout	0.2
Adam Optimizer	0.01

Table 10: Hyperparameters for GRU RNN for 75.2 % Accuracy

Hyperparameter	Value
Epoch	100
Batch Size	32
L2 Regularizer	0.01
No of features	50
Activation function	ReLU, Softmax
Dropout	0.2
Adam Optimizer	0.01

Table 10: Hyperparameters for GRU RNN for 75.2 % Accuracy

Results of performance evaluation

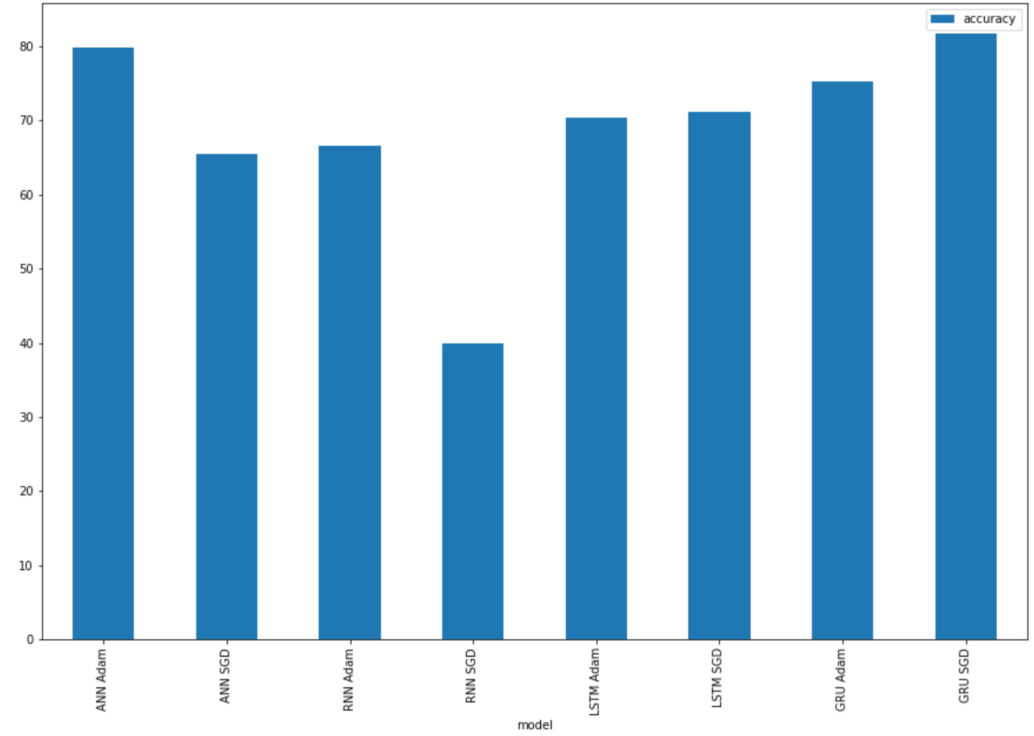

Figure 7

Model	Optimizer	Accuracy
ANN	Adam	79.99%
ANN	SGD	65.55%
RNN	Adam	66.60%
RNN	SGD	39.90%
LSTM	Adam	70.40%
LSTM	SGD	71.10%
GRU	Adam	75.20%
GRU	SGD	81.70%

Drawbacks

- **Artificial Neural Networks**

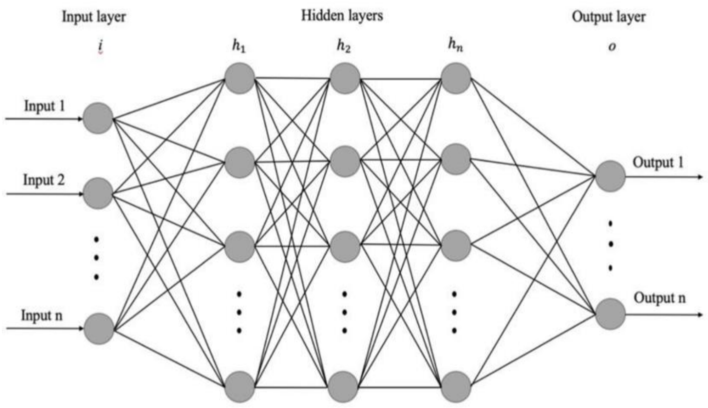

Figure 8

Adam Optimization:

- ReLU activation
- 0.01 optimizer
- 0.02 dropout
- 0.01 regularizer

SGD Optimization:

- ReLU activation
- 0.01, 0.09 optimizer
- 0.02 dropout
- 0.01 regularizer

Drawbacks:

- **Not suitable for time-series data**
- **Risk of overfitting**
- **Long training**

Drawbacks

- **'Vanilla' Recurrent Neural Networks**

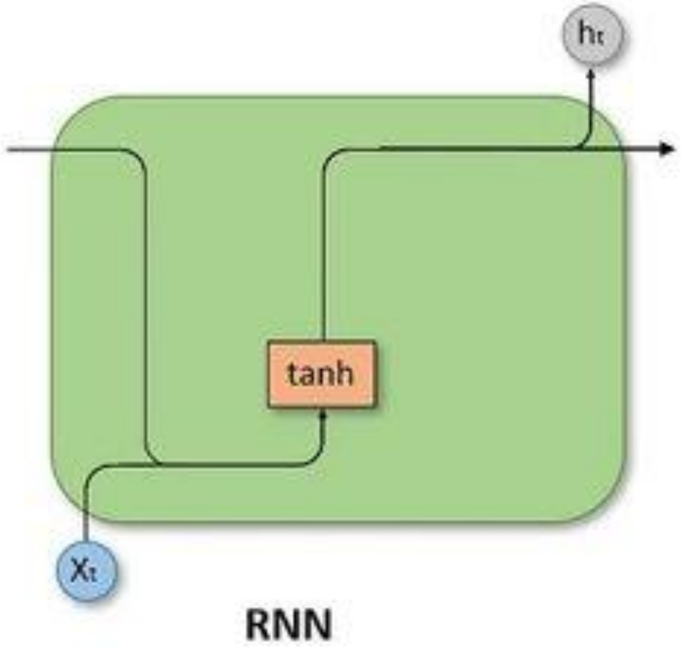

Figure 9

Adam Optimization:

- ReLU activation
- 0.01 optimizer
- 0.02 dropout
- 0.01 regularizer

SGD Optimization:

- ReLU activation
- 0.01, 0.09 optimizer
- 0.02 dropout
- 0.01 regularizer

Drawbacks:

- **Too simple for time series**
- **No control of amount of information**
- **Long training**
- **Complex training**

Drawbacks

- **Long Short-Term Memory RNN**

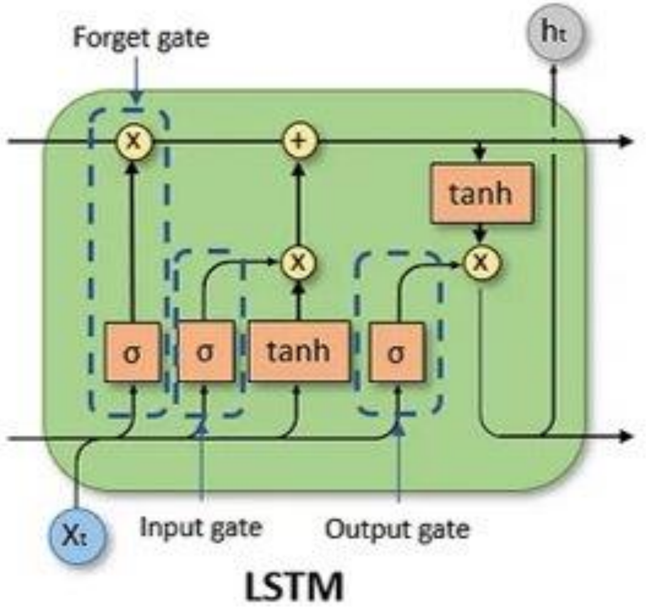

Figure 10

Adam Optimization:

- ReLU activation
- 0.01 optimizer
- 0.02 dropout
- 0.01 regularizer

SGD Optimization:

- ReLU activation
- 0.01, 0.09 optimizer
- 0.02 dropout
- 0.01 regularizer

Drawbacks:

- **Time inefficiency due to complexity**
- **Low learning efficiency**
- **Complex training**

Drawbacks

- **Gated Recurrent Unit RNN**

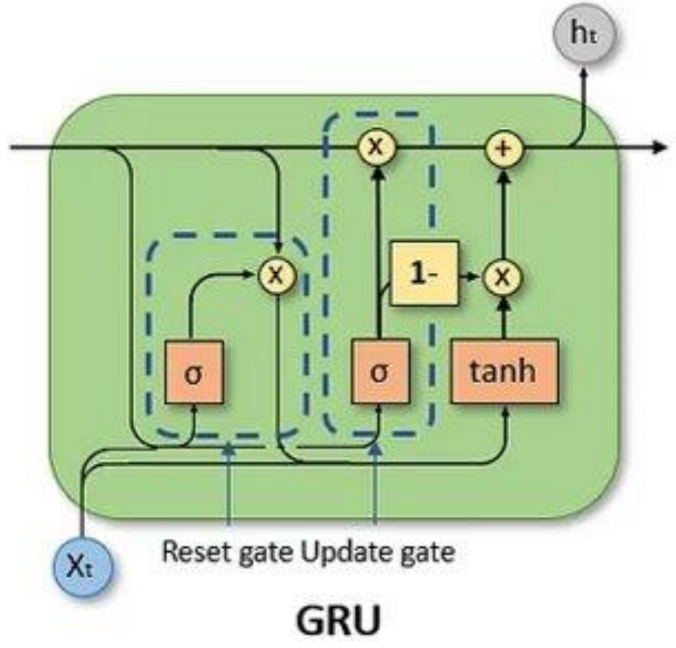

Figure 11

Adam Optimization:

- ReLU activation
- 0.01 optimizer
- 0.02 dropout
- 0.01 regularizer

SGD Optimization:

- ReLU activation
- 0.01, 0.09 optimizer
- 0.02 dropout
- 0.01 regularizer

Drawbacks:

- **Slow convergence**
- **Low learning efficiency**
- **Long training**
- **Risk of underfitting**

Conclusion

RNN performs slightly better than ANN. Out of variety of RNN models, GRU works the best.

We consider this as expected discovery, since our dataset consists of timesteps and RNN is focused on fitting time-series data.

Additionally, we have observed how different optimizers and hyperparameters may influence the performance of models and the difference in the time efficiency between models.

GRU = process twice more epochs than LSTM/same time, which makes GRU more efficient model to implement for larger datasets.

LSTM might have performed better for larger number of epochs, but it will not be time efficient.

Questions?

Thank You!