

CM4

1. Runtime performance for training and testing on both networks for all three labels

- Default Network (per epochs)

Confirmed		Deaths		Recovered	
Training	Testing	Training	Testing	Training	Testing
2 – 3 sec	2 sec	2 - 3 sec	2 sec	2 – 3 sec	2 sec

- LSTM (per epochs)

Confirmed		Deaths		Recovered	
Training	Testing	Training	Testing	Training	Testing
3 – 4 sec	2 sec	3 - 4 sec	2 sec	3 – 4 sec	2 sec

This is mainly due to the sequential computation in the LSTM layer. Remember that LSTM requires sequential input to calculate the hidden layer weights iteratively, in other words, you must wait for the hidden state at time $t-1$ to calculate the hidden state at time t .

2. Comparison of different algorithms and parameters tried

By changing the number of epochs

Labels	Accuracy					
	Epochs = 50		Epochs = 100		Epochs = 200	
	Training	Testing	Training	Testing	Training	Testing
Confirmed	0.9655	0.9504	0.9626	0.9504	0.9724	0.9545
Deaths	0.9310	0.8967	0.9287	0.8967	0.9452	0.8884
Recovered	0.9345	0.9380	0.9402	0.9380	0.9460	0.9356

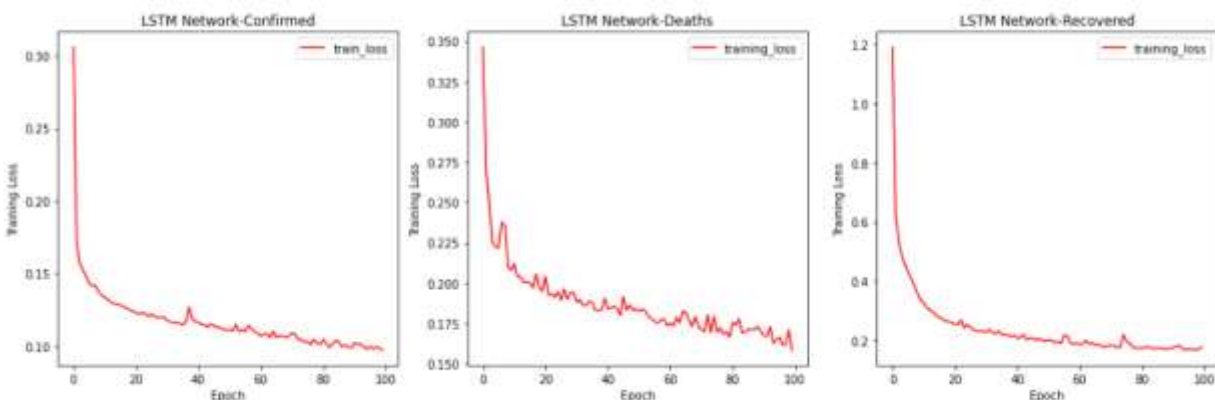
Here we had used different number of epochs for training the model for confirmed, deaths and recovered so by analyzing the differences in the accuracy that we get had slight difference in that so here in this code we had used the value of epochs as 100 because using more number of epochs makes the model more precise and accurate but leads to much more over fitting.

By changing hidden units

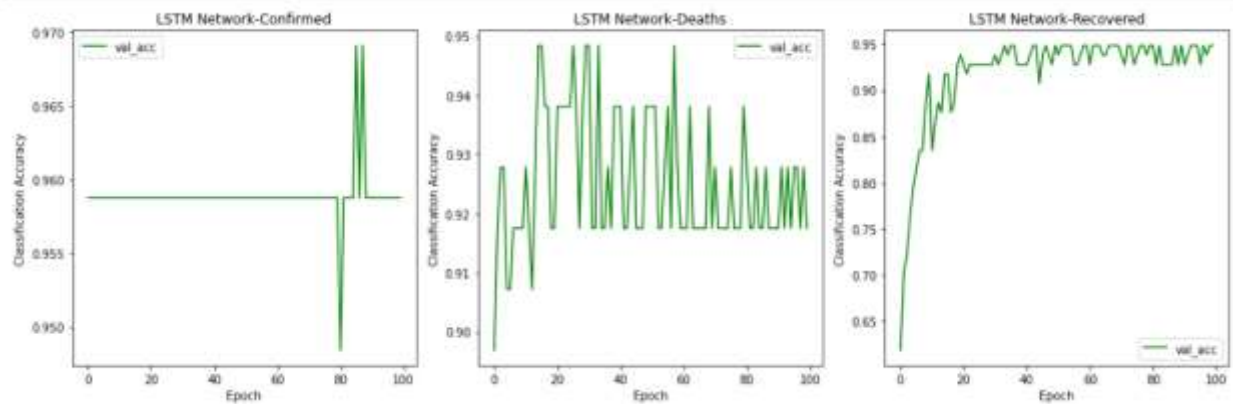
Label	Accuracy			
	Hidden units = 32		Hidden units = 64	
	Training	Testing	Training	Testing
Confirmed	0.9626	0.9504	0.9643	0.9504
Deaths	0.9287	0.8967	0.9264	0.8926
Recovered	0.9402	0.9380	0.9379	0.9463

Increasing the number of hidden units might improve the accuracy or might not, depends upon the complexity and increasing the hidden units leads to increase the accuracy.

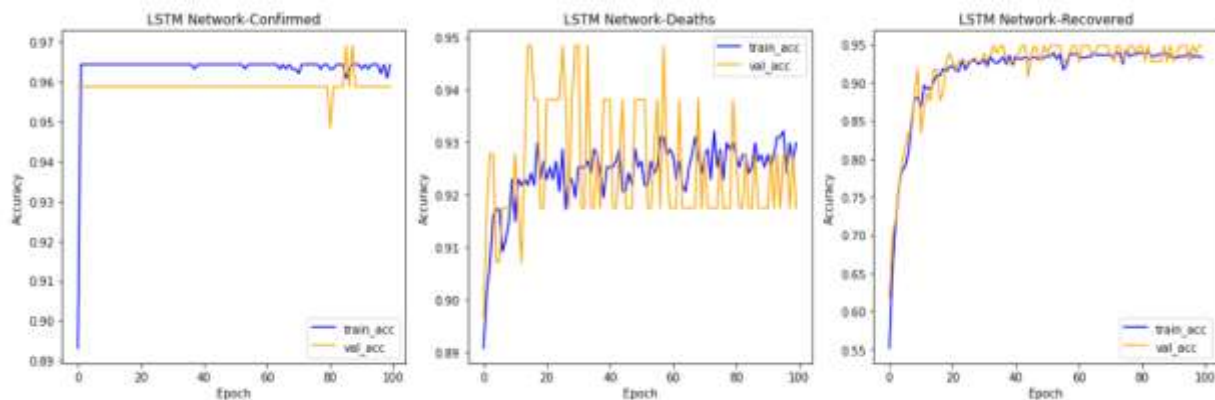
3. You can use any plots to explain the performance of your approach. But at the very least produce two plots, one of training loss vs. training epoch and one of classification accuracy vs. training epoch on both your training and test set.



- As epochs increase, the network learns better and the weights adjust their value to predict the target variable better. There is a decrease in loss with each gradient update.
- The output of the training loop is shown in the figure above; as the number of training epochs increases, the loss reduces progressively. Initially, it declines extremely quickly, but after a few epochs, it converges to around 0.12, 0.200 and 0.3 respectively in confirmed, deaths and recovered and then continues to fall again, eventually converging to training loss of around 30 in confirmed, deaths and recovered.



From the above graph, classification accuracy Vs epoch of confirmed, deaths and recovered in recovered the classification accuracy is increasing as epochs increases after the value of epochs crosses 20 the classification accuracy fluctuates between 90 to 95%.



- As epochs increases accuracy also increases. An epoch is mixed up with an iteration the more number of epoch will leads to over fitting.
- Analyzing the above graphs we can conclude that, in confirmed the accuracy drastically picks up from 0.90 to 0.96 and after that its being stable there. In deaths and recovered both train and val accuracy increases as increase in epochs having minor ups and downs in the accuracy.