Assignment 2 Report

Kareem Mokhtar	31-609
Nour ElDin Khaled	31-7974
Narihan Ellaithy	31-8027

1) Performance comparison using different number of neurons per hidden layer

a) Learning rate 0.001

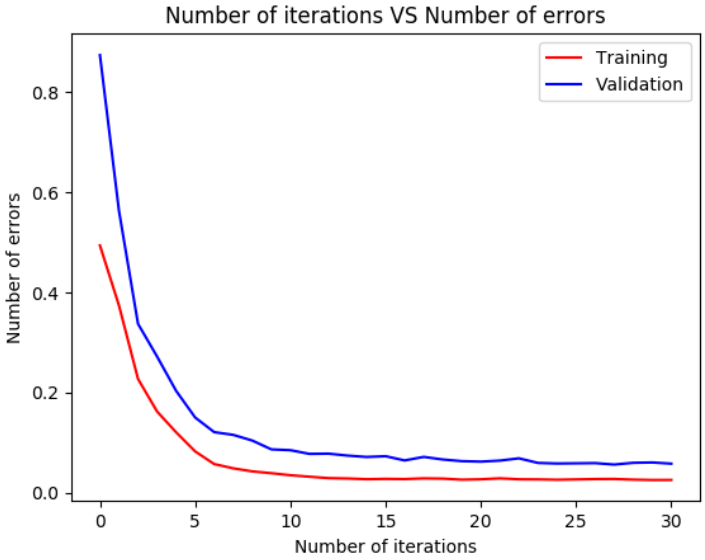
time_series_8_2	One Hidden Layer			Two Hidden Layers			Three Hidden Layers		
Number of neurons per layer	8	16	32	(16,8)	(32,16)	(32,8)	(32,16,8)	(64,16,8)	(64,8,8)
Mean Score	0.94	0.93	0.94	0.94	0.92	0.93	0.95	0.95	0.95
Mean Error	0.06	0.07	0.06	0.06	0.08	0.07	0.05	0.05	0.05
Maximum Error	0.08	0.10	0.12	0.08	0.2	0.12	0.09	0.07	0.08
Minimum Error	0.04	0.04	0.03	0.04	0.04	0.04	0.03	0.04	0.04
Mean Number of Epochs	171	117	116	156	89	147	136	102	136

b) Learning rate 0.01

time_series_8_2	One Hidden Layer			Two Hidden Layers			Three Hidden Layers		
Number of neurons per layer	8	16	32	(16,8)	(32,16)	(32,8)	(32,16,8)	(64,16,8)	(64,8,8)
Mean Score	0.94	0.94	0.95	0.94	0.95	0.95	0.95	0.95	0.95
Mean Error	0.06	0.06	0.05	0.06	0.05	0.05	0.05	0.05	0.05
Maximum Error	0.09	0.09	0.08	0.07	0.07	0.08	0.08	0.06	0.08
Minimum Error	0.04	0.04	0.04	0.04	0.03	0.04	0.04	0.03	0.03
Mean Number of Epochs	28	22	18	23	21	23	21	20	24

- Reasoning for hidden layers: After experimenting by training network with different hidden nodes and then choosing the one that gives the best results.
- Reasoning for learning rate: Learning rate 0.01 gave better results as it reached the minima faster than that of 0.001. This is because when the learning rate is too small, this means that it learns in a slow rate and reaches the minima in a longer time. However, if it is too big, it will learn fast and might pass the minima. Thus a balance must be struck, yet 0.01 is not big, it was just right and took less time to converge than 0.001.

2) Observe and report on the progress of both training and validation errors until the optimal performance is reached (in the graph). Take and report on adequate measures to guarantee that the regressor is not overfitted to the training data (notes below).



- As shown in the above figure, the training is stopped as soon as the validation error stops decreasing, thus avoiding overfitting (by using early stopping technique). This is because at some stage the error on the validation set will increase again after reaching its peak decrease, this is because the network has stopped learning about the function that generated the data and started learning on the noise instead of generalizing its learning(overfitting).

- The best accuracy that we have reached for timeseries_8_2 with the testing set is 89.297% with 3 hidden layers with number of neurons (64, 8, 8). This was discovered by learning on different numbers and layers till we reached the best accuracy level.

- Bonus: Justify the difference in number and size of hidden layers required to achieve high performance in timeseries_4_1 versus timeseries_8_2- if a difference was observed.

Timeseries_4_1	One Hidden Layer			Two Hidden Layers			Three Hidden Layers		
Number of neurons per layer	8	16	32	(16,8)	(32,16)	(32,8)	(32,16,8)	(64,16,8)	(64,8,8)
Mean Score	0.96	0.96	0.96	0.95	0.96	0.97	0.96	0.96	0.95
Mean Error	0.04	0.04	0.04	0.05	0.04	0.03	0.04	0.04	0.05
Maximum Error	0.11	0.1	0.08	0.09	0.12	0.05	0.07	0.09	0.1
Minimum Error	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02
Mean Number of Epochs	35	28	22	25	20	24	24	20	20

- In timeseries_4_1 only 2 hidden layers are required to reach highest performance with number of nodes (32, 8), unlike timeseries_8_2 which required 3 hidden layers. This is due to the fact that in timeseries_4_1 the number of inputs and outputs are smaller than that of timeseries_8_2, so there is no need for multiple layers and neurons.
- The best accuracy that we have reached for timeseries_4_1 with the testing set is 97.27%

  With 3 hidden layers with number of neurons (32, 8), which was also the result of pure trail and error.
- We got better performance in the timeseries_4_1 because we had to make less predict 1 week only as opposed to timeseries_8_2 where we predicted 2 weeks, thus the error is smaller.