Project Final Report: Human Activity Classification

Gurkan Kilicaslan
kilia064@umn

Understanding what people are doing based on how they move is important in many areas like healthcare, sports analysis, and automation. Being able to classify these activities can help create personalized plans and make systems work better. In this project, I build a model that can tell us what activity a person is doing by looking at their movements over time. The main goal of this project is to figure out and classify different activities people are doing. These activities are: going downstairs, sitting, going upstairs, jogging, walking, and standing. This project attempts to create a network that uses input from three different sensors across T = 150 time samples to classify human activities. There are 3000 samples in the training set and 600 samples in the test set. Wearable sensors that are attached to loose clothing are used to measure static postures. Despite the fact that the data I'm utilizing is a modified version of the data in [1]'s supplementary materials, it still provides a very clear picture of the project.

To do this, I considered applying three distinct types of methods—RNN, LSTM, and GRU—and contrasting them. At first, I had difficulties while implementing RNN. Hence, implementing the other two methods were not very feasible. Based on Professor Johnson's feedback, I first used my RNN algorithm with more epochs. Then, implemented LSTM to see if it is better than RNN. Then, I also had enough time to implement GRU. Overall, I did everything I anticipated to do when I decided to do this project.

Peer reviews have recommended using k-nearest neighbors, logistic regression, etc. to compare the outcomes to RNN in case I couldn't do LSTM and GRU. The worst-case scenario was to use one of the listed techniques rather than GRU and LSTM but luckily I had enough time to do both LSTM and GRU. I appreciate the reviews though.

Application of RNN

Following is a small description of the algorithm:

To classify human activity, I put into practice a single-layer RNN design with 128 hidden neuron numbers and hyperbolic tangent activation. With a mini-batch size of 32 samples, a learning rate of 0.1, a momentum rate of 0.85, and a maximum of 50 epochs, I utilized a stochastic gradient descent technique. I used the Xavier Uniform distribution for weight initialization [2]. The formula for it is as follows:

$$w_o = \sqrt{\frac{6}{L_{pre} + L_{post}}}, W = [-w_o, w_o]$$

The first layer is used as the recurrent layer during network training:

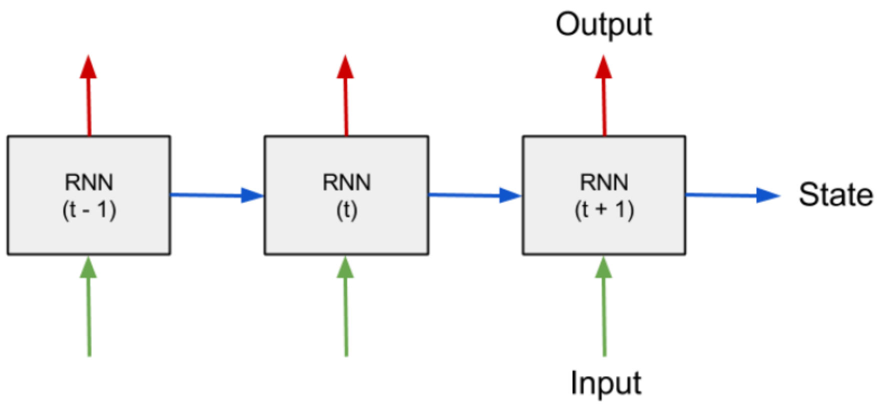


Figure 1: Recurrent Layer

For the issue, the RNN's mathematical meaning at each time step t is as follows:

$$h(t) = \tanh(x(t)W_{ih} + h(t-1)W_{hh} + bias)$$

where h is the hidden layer neuron number and i is the input neuron number.

We can see from the loss plot (Fig. 2) that there is fluctuation, which indicates that the network is not stable—something we don't want to see. Test accuracy is only 38.67%. The loss tends to increase extremely easily when learning rate rises. This is due to the cumulative nature of backpropagation through time (BPTT). Additionally, as shown in Fig. 3, the network only learns a part of the data and is unable to correctly predict the majority of classes (see **Appendices - RNN** for detailed training and validation loss and accuracy plots).

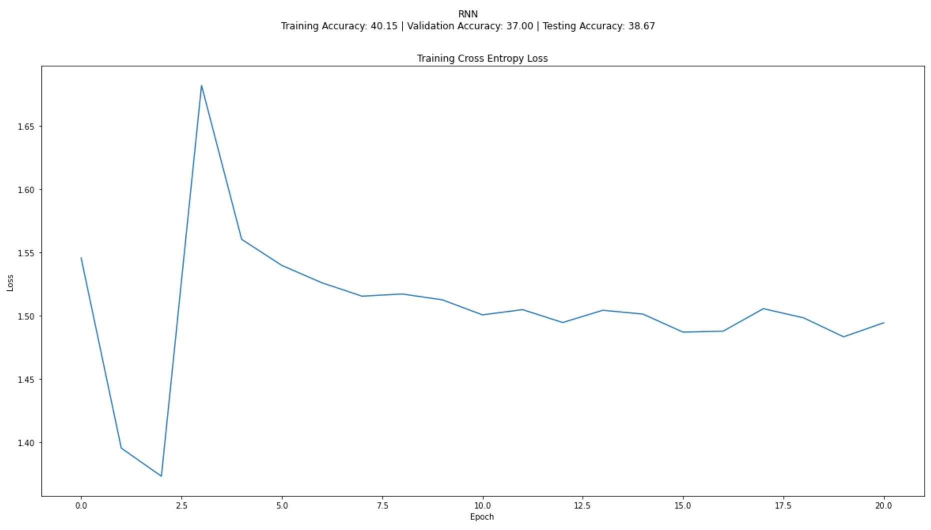


Figure 2: RNN Training Cross Entropy Loss by Epochs

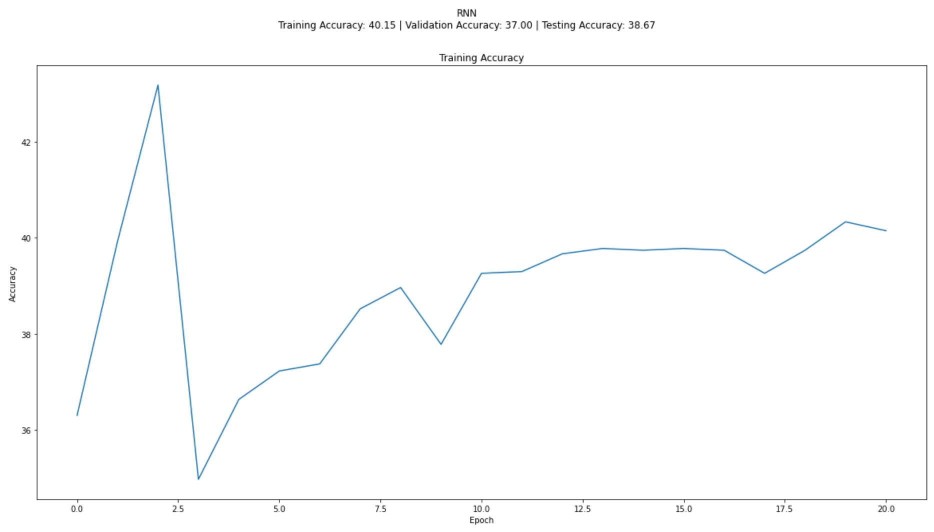


Figure 3: RNN Training Accuracy by Epochs

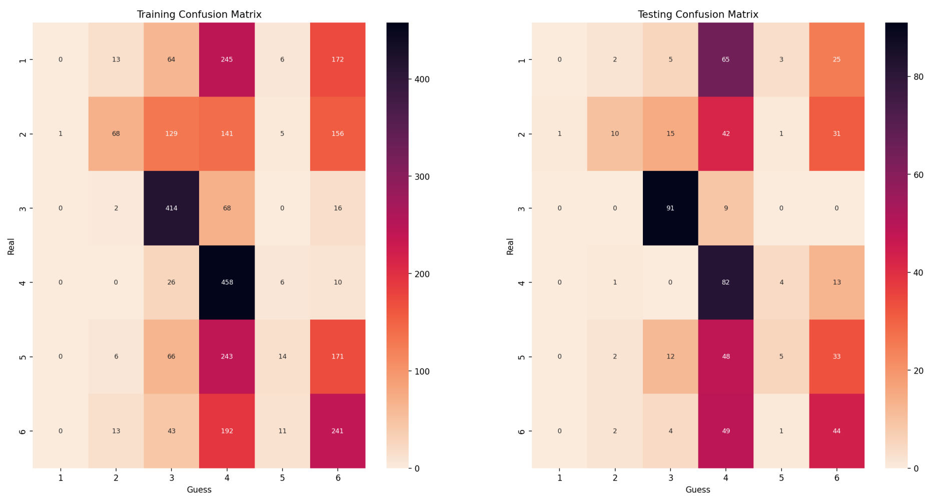


Figure 4: RNN Confusion Matrix

The accuracy and loss plots show some instability after RNN training. The network is not learning if, in the long term, it does not always approach a reduced loss as training progresses because of the vanishing and ballooning gradients problem of RNN. In this situation, gradients become extremely high or extremely low, which causes loss to shift quickly. Rapid gradient changes for RNN are also produced with 150 time samples and the BPTT cumulative features. The instability may be influenced by the parameters. For instance, lowering the learning rate can help reduce overshooting, but if it's done too much, it might also cause training to go slowly or even stop altogether. The goals of GRU and LSTM are to solve these issues.

Application of LSTM

This time, LSTM is used in the first layer to train the network.

The issue of exploding or disappearing gradients in RNN is resolved by LSTM. To improve network stability, it employs forget gates to erase a portion of the network's past memory.

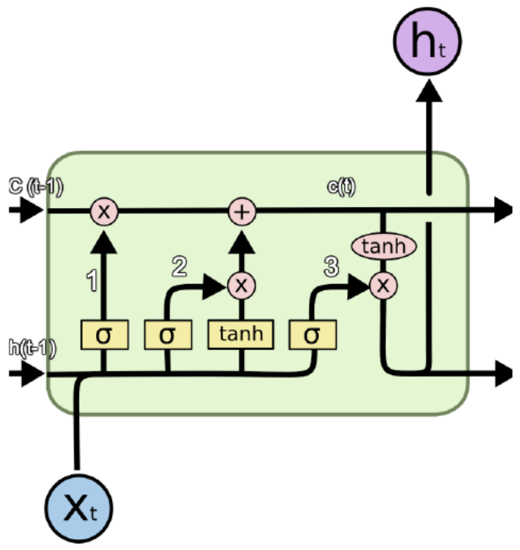


Figure 5: LSTM Layer

The structure of the algorithm is shown in Fig. 5 [3]. Equations of this algorithm are:

$$h_i(t) = \sigma([h(t-1), x(t)] \cdot weights_i + bias_i)$$

$$h_f(t) = \sigma([h(t-1), x(t)] \cdot weights_f + bias_f)$$

$$h_o(t) = \sigma([h(t-1), x(t)] \cdot weights_o + bias_o)$$

$$h_c(t) = tanh([h(t-1), x(t)] \cdot weights_c + bias_c)$$

$$c(t) = h_f(t) \cdot c(t-1) + h_i(t) \cdot h_c(t)$$

$$h(t) = h_o(t) \cdot tanh(c(t))$$

The results are shown below (see **Appendices - LSTM** for detailed training and validation loss and accuracy plots):

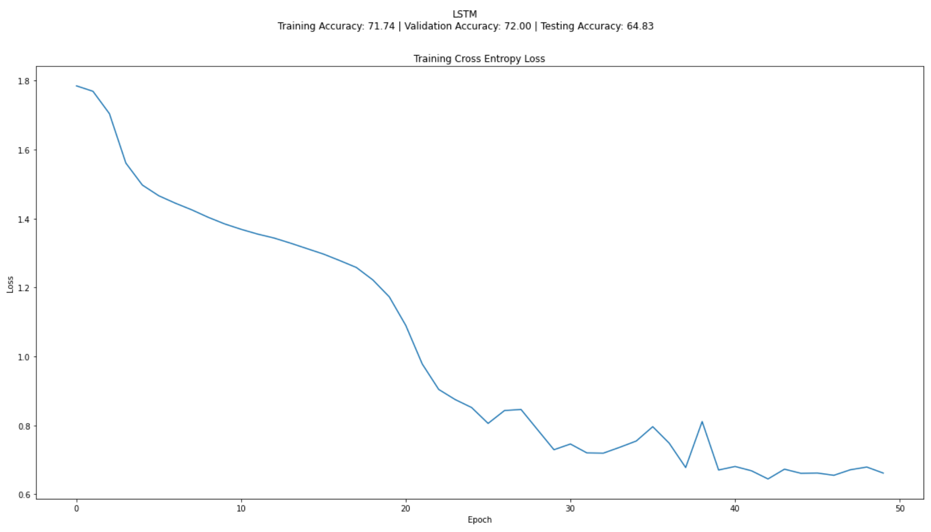


Figure 6: LSTM Training Cross Entropy Loss by Epochs

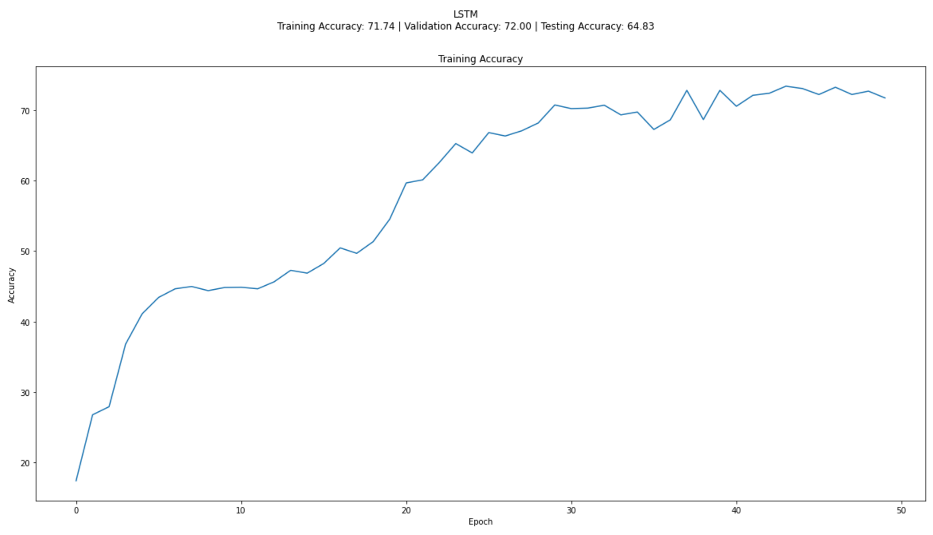


Figure 7: LSTM Training Accuracy by Epochs

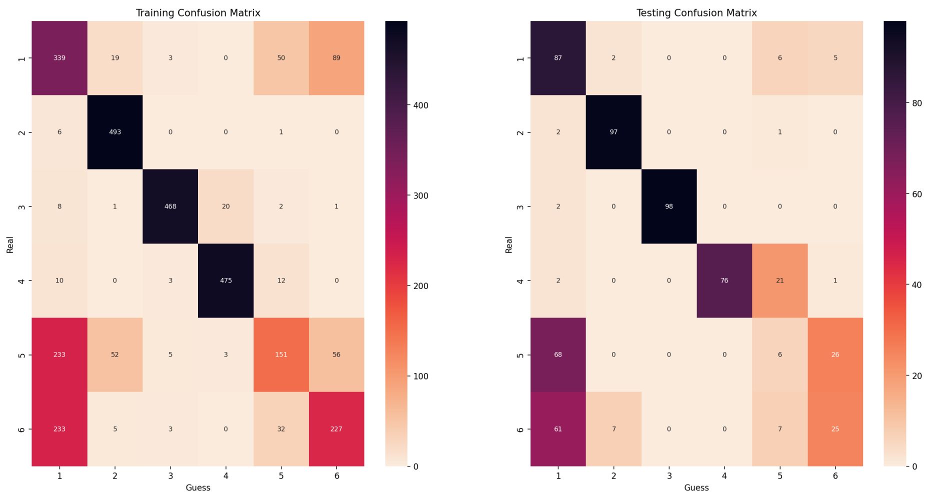


Figure 8: LSTM Confusion Matrix

The network accuracy has significantly improved over RNN. Confusion matrices and the loss plot demonstrate the network's ability to learn and make accurate predictions. Test accuracy increases to 64.83%, compared to 38.67% in RNN. While accuracy has improved, it can still be further boosted by raising the learning rate; nevertheless, in our situation, it should be maintained in order to determine which method produces the best outcomes. Furthermore, it is not desirable to increase learning gain, and doing so too much can cause gradients to explode.

Application of GRU

Algorithm used in this section uses GRU as the first layer. With a few minor modifications, GRU is comparable to the LSTM network; it is less complex because it contains fewer gates and equations. Still, the performance is excellent and will even surpass the LSTM that will be demonstrated later.With a few minor modifications, GRU is comparable to the LSTM network; it is less complex because it contains fewer gates and equations. Still, the performance is excellent and will even surpass the LSTM that will be demonstrated later.

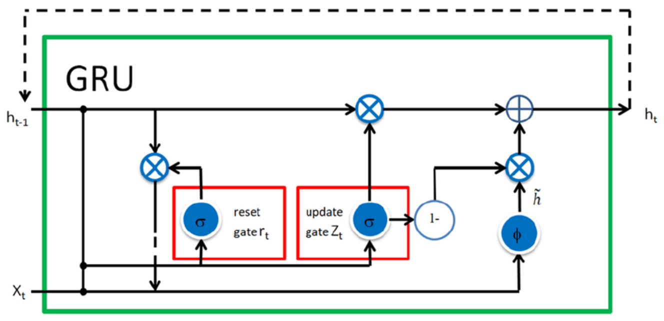


Figure 9: GRU Layer

The structure of the algorithm is shown in Fig. 9 [4]. The corresponding equations of this algorithm are:

$$z(t) = \sigma(x(t) \cdot weights_z + h(t-1)u_z + bias_z)$$
$$r(t) = \sigma(x(t) \cdot weights_r + h(t-1)u_r + bias_r)$$
$$\tilde{h}(t) = tanh(x(t) \cdot weights_h + (h(t-1) \cdot r(t))u_h + bias_h)$$
$$h(t) = (1 - z(t)) \cdot h(t-1) + z(t)\tilde{h}(t)$$

The results are shown below (see **Appendices - GRU** for detailed training and validation loss and accuracy plots):

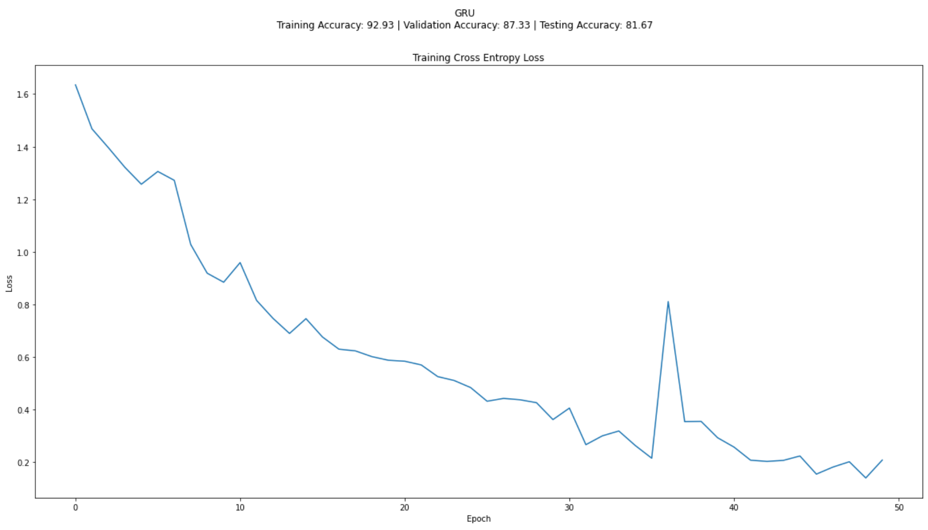


Figure 10: GRU Training Cross Entropy Loss by Epochs

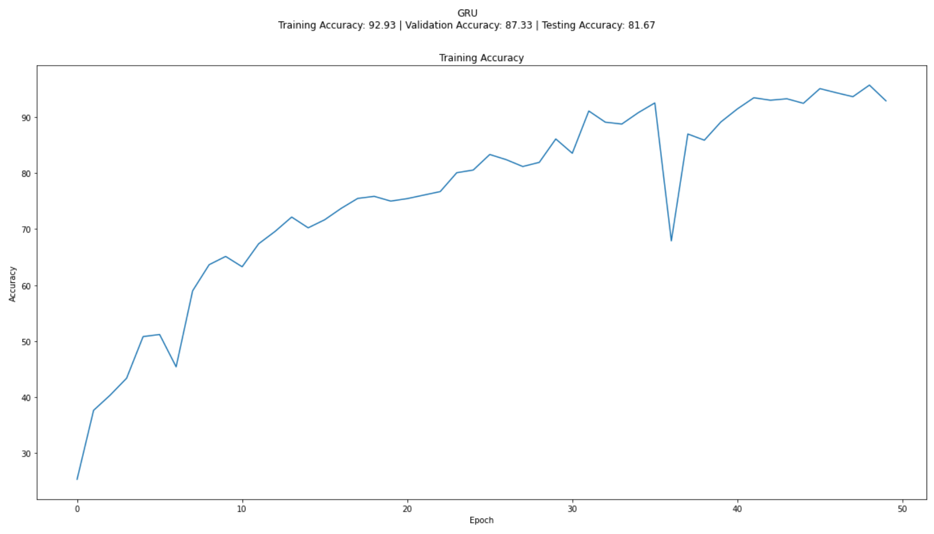


Figure 11: GRU Training Accuracy by Epochs

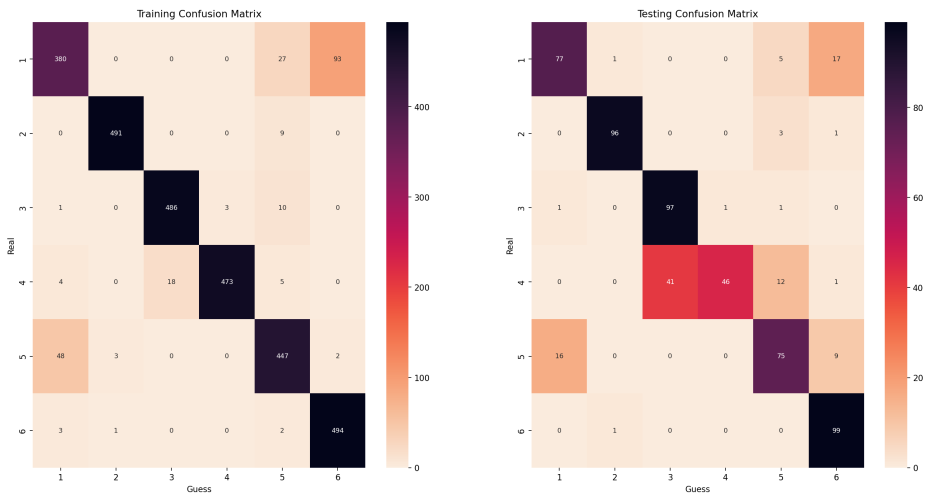


Figure 12: GRU Confusion Matrix

GRU performs better and has the highest accuracy when compared to the other networks. Its accuracy on tests is 81.67%. The fact that it ended with a steady loss indicates that it also resolves the issue of vanishing or explosion of gradients. GRU requires less training time than LSTM since it contains fewer gates. LSTM, on the other hand, is less susceptible to the vanishing or explosion of gradients yet is more stable. This is because GRU only reduces rather than completely eliminates vanishing gradients. Because LSTM handles this circumstance better, it becomes more stable. As a result, LSTM is designed to have a more precise and stable network, while GRU is meant to be efficient.

Conclusion

In conclusion, the application of recurrent neural networks (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU) in classifying human activities using sensor data yielded insightful results.

The RNN implementation faced challenges with stability, exhibiting fluctuations in loss and limited accuracy (38.67%). The inherent issues of vanishing and exploding gradients were apparent, hindering long-term learning and prediction capabilities.

The LSTM model, designed to address gradient instability, significantly improved performance. The accuracy increased to 64.83%, demonstrating the effectiveness of LSTM in learning from sequential data and mitigating the challenges encountered with RNN.

The GRU outperformed both RNN and LSTM, achieving an impressive test accuracy of 81.67%. GRU's ability to balance performance and efficiency, resolving gradient issues while requiring less training time, showcased its suitability for the task.

In summary, the project successfully compared and implemented RNN, LSTM, and GRU for human activity classification, with GRU emerging as the most effective model in terms of accuracy and stability. The findings emphasize the importance of choosing appropriate recurrent neural network architectures for specific applications, considering both performance and computational efficiency. **Possible Next Steps:** Refine hyperparameters for better convergence, explore ensemble methods for model combination, and enhance interpretability using attention mechanisms. Further, investigate real-time implementation on wearable devices for practical deployment.

References

[1] Jayasinghe, U., Janko, B., Hwang, F., & Harwin, W. S. (2023). Classification of static postures with wearable sensors mounted on loose clothing. *Scientific Reports, 13 (1).* https://doi.org/10.1038/s41598-022-27306-4

[2] Dou, R., Principe, J. (2023) Dynamic Analysis and an Eigen Initializer for Recurrent Neural Networks. *2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 2023, pp. 1-6.* https://doi.org/10.1109/IJCNN54540.2023.10191986

[3] Kalita D. An overview on Long short term memory (lstm) [Online]. *Analytics Vidhya: 2022.* https://www.analyticsvidhya.com/blog/2022/03/an-overview-on-long-short-term-memory-lstm

[4] Gated Recurrent Units (GRU) - Dive into Deep Learning 1.0.3 documentation [Online]. https://d2l.ai/chapter_recurrent-modern/gru.htm

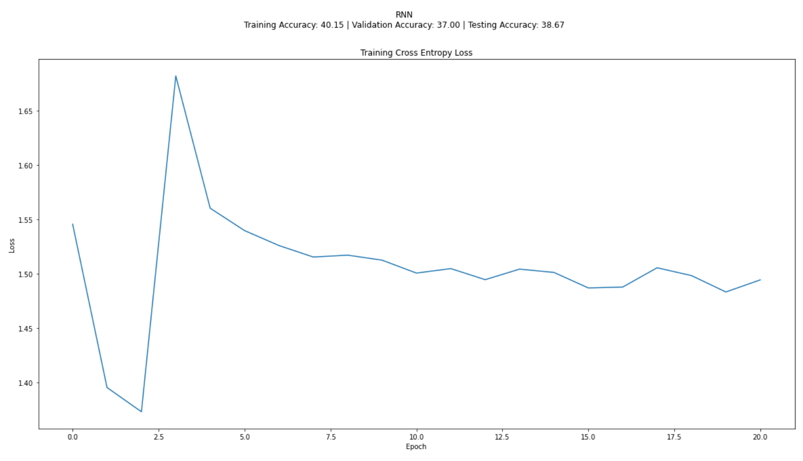
Appendices - RNN

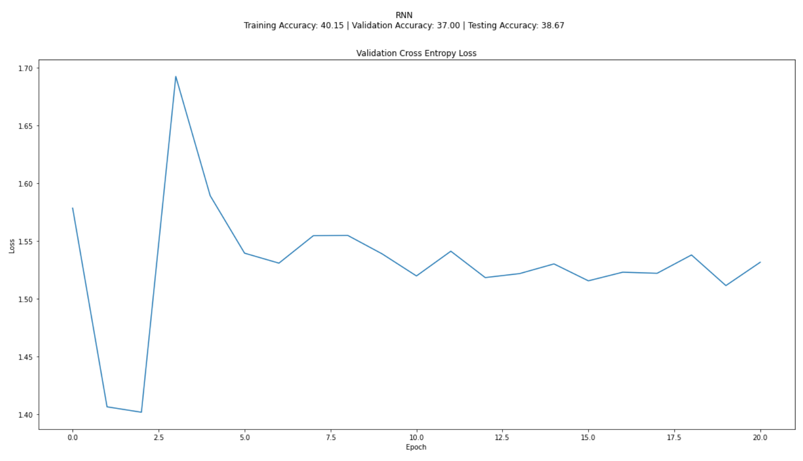
RNN Layer

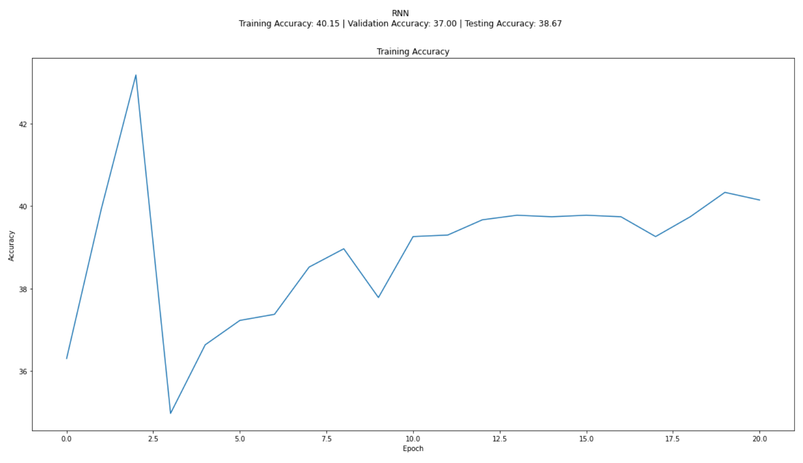
```
Epoch: 1=== Training Loss: 1.545,Validation Loss: 1.579,Training Accuracy: 36.296,Validation Accuracy: 31.000
Epoch: 2=== Training Loss: 1.395,Validation Loss: 1.406,Training Accuracy: 39.926,Validation Accuracy: 38.333
Epoch: 3=== Training Loss: 1.373,Validation Loss: 1.402,Training Accuracy: 43.185,Validation Accuracy: 41.000
Epoch: 4=== Training Loss: 1.682,Validation Loss: 1.693,Training Accuracy: 34.963,Validation Accuracy: 31.667
Epoch: 5=== Training Loss: 1.560,Validation Loss: 1.589,Training Accuracy: 36.630,Validation Accuracy: 32.000
Epoch: 6=== Training Loss: 1.540,Validation Loss: 1.540,Training Accuracy: 37.222,Validation Accuracy: 35.667
Epoch: 7=== Training Loss: 1.526,Validation Loss: 1.531,Training Accuracy: 37.370,Validation Accuracy: 37.000
Epoch: 8=== Training Loss: 1.515,Validation Loss: 1.555,Training Accuracy: 38.519,Validation Accuracy: 33.667
Epoch: 9=== Training Loss: 1.517,Validation Loss: 1.555,Training Accuracy: 38.963,Validation Accuracy: 37.667
Epoch: 10=== Training Loss: 1.512,Validation Loss: 1.539,Training Accuracy: 37.778,Validation Accuracy: 37.000
Epoch: 11=== Training Loss: 1.501,Validation Loss: 1.520,Training Accuracy: 39.259,Validation Accuracy: 37.667
Epoch: 12=== Training Loss: 1.505,Validation Loss: 1.541,Training Accuracy: 39.296,Validation Accuracy: 35.333
Epoch: 13=== Training Loss: 1.494,Validation Loss: 1.518,Training Accuracy: 39.667,Validation Accuracy: 36.333
Epoch: 14=== Training Loss: 1.504,Validation Loss: 1.522,Training Accuracy: 39.778,Validation Accuracy: 39.333
Epoch: 15=== Training Loss: 1.501,Validation Loss: 1.530,Training Accuracy: 39.741,Validation Accuracy: 36.000
Epoch: 16=== Training Loss: 1.487,Validation Loss: 1.516,Training Accuracy: 39.778,Validation Accuracy: 38.333
Epoch: 17=== Training Loss: 1.488,Validation Loss: 1.523,Training Accuracy: 39.741,Validation Accuracy: 36.000
Epoch: 18=== Training Loss: 1.505,Validation Loss: 1.522,Training Accuracy: 39.259,Validation Accuracy: 36.333
Epoch: 19=== Training Loss: 1.498,Validation Loss: 1.538,Training Accuracy: 39.741,Validation Accuracy: 33.667
Epoch: 20=== Training Loss: 1.483,Validation Loss: 1.511,Training Accuracy: 40.333,Validation Accuracy: 38.000
Epoch: 21=== Training Loss: 1.494,Validation Loss: 1.532,Training Accuracy: 40.148,Validation Accuracy: 37.000
```

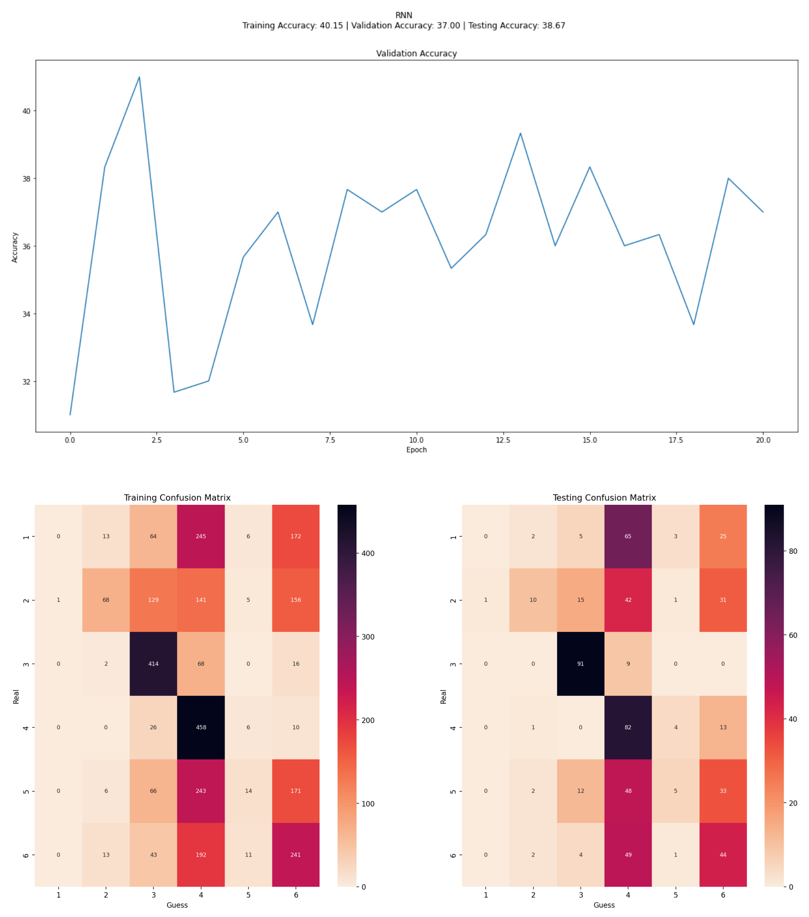
Training stopped since validation C-E reached convergence.

Test Accuracy: 38.666666666666664









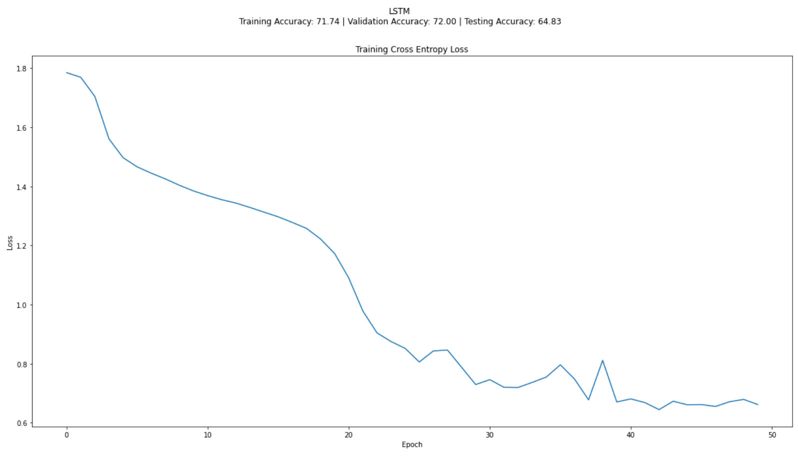
Appendices - LSTM

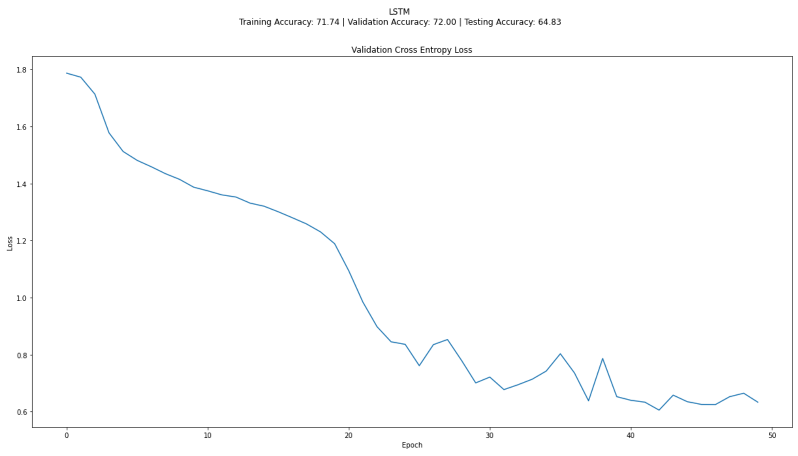
LSTM Layer

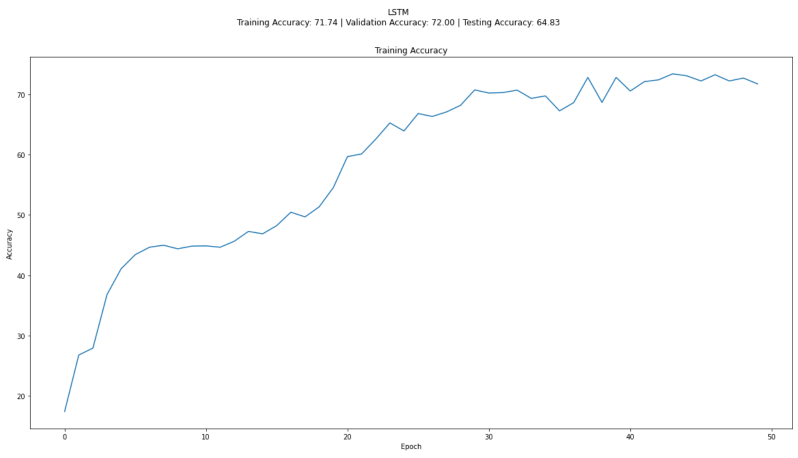
```
Epoch: 1=== Training Loss: 1.785,Validation Loss: 1.787,Training Accuracy: 17.370,Validation Accuracy: 14.333
Epoch: 2=== Training Loss: 1.769,Validation Loss: 1.773,Training Accuracy: 26.741,Validation Accuracy: 27.000
Epoch: 3=== Training Loss: 1.704,Validation Loss: 1.713,Training Accuracy: 27.889,Validation Accuracy: 28.000
Epoch: 4=== Training Loss: 1.561,Validation Loss: 1.578,Training Accuracy: 36.778,Validation Accuracy: 37.000
Epoch: 5=== Training Loss: 1.497,Validation Loss: 1.512,Training Accuracy: 41.074,Validation Accuracy: 40.333
Epoch: 6=== Training Loss: 1.466,Validation Loss: 1.481,Training Accuracy: 43.407,Validation Accuracy: 41.333
Epoch: 7=== Training Loss: 1.445,Validation Loss: 1.459,Training Accuracy: 44.630,Validation Accuracy: 43.667
Epoch: 8=== Training Loss: 1.425,Validation Loss: 1.434,Training Accuracy: 44.963,Validation Accuracy: 43.667
Epoch: 9=== Training Loss: 1.404,Validation Loss: 1.415,Training Accuracy: 44.370,Validation Accuracy: 43.000
Epoch: 10=== Training Loss: 1.384,Validation Loss: 1.387,Training Accuracy: 44.815,Validation Accuracy: 44.333
Epoch: 11=== Training Loss: 1.369,Validation Loss: 1.374,Training Accuracy: 44.852,Validation Accuracy: 45.333
Epoch: 12=== Training Loss: 1.355,Validation Loss: 1.360,Training Accuracy: 44.630,Validation Accuracy: 44.333
Epoch: 13=== Training Loss: 1.343,Validation Loss: 1.352,Training Accuracy: 45.630,Validation Accuracy: 44.667
Epoch: 14=== Training Loss: 1.329,Validation Loss: 1.331,Training Accuracy: 47.259,Validation Accuracy: 46.333
Epoch: 15=== Training Loss: 1.313,Validation Loss: 1.320,Training Accuracy: 46.852,Validation Accuracy: 46.333
Epoch: 16=== Training Loss: 1.297,Validation Loss: 1.301,Training Accuracy: 48.222,Validation Accuracy: 48.667
Epoch: 17=== Training Loss: 1.278,Validation Loss: 1.280,Training Accuracy: 50.444,Validation Accuracy: 49.667
Epoch: 18=== Training Loss: 1.258,Validation Loss: 1.258,Training Accuracy: 49.667,Validation Accuracy: 50.667
Epoch: 19=== Training Loss: 1.222,Validation Loss: 1.230,Training Accuracy: 51.333,Validation Accuracy: 52.667
Epoch: 20=== Training Loss: 1.173,Validation Loss: 1.189,Training Accuracy: 54.519,Validation Accuracy: 55.000
Epoch: 21=== Training Loss: 1.090,Validation Loss: 1.094,Training Accuracy: 59.667,Validation Accuracy: 60.667
Epoch: 22=== Training Loss: 0.978,Validation Loss: 0.984,Training Accuracy: 60.111,Validation Accuracy: 60.000
Epoch: 23=== Training Loss: 0.904,Validation Loss: 0.898,Training Accuracy: 62.556,Validation Accuracy: 63.667
Epoch: 24=== Training Loss: 0.875,Validation Loss: 0.845,Training Accuracy: 65.259,Validation Accuracy: 68.000
Epoch: 25=== Training Loss: 0.852,Validation Loss: 0.836,Training Accuracy: 63.926,Validation Accuracy: 62.667
Epoch: 26=== Training Loss: 0.806,Validation Loss: 0.761,Training Accuracy: 66.815,Validation Accuracy: 66.333
Epoch: 27=== Training Loss: 0.843,Validation Loss: 0.835,Training Accuracy: 66.333,Validation Accuracy: 68.333
Epoch: 28=== Training Loss: 0.846,Validation Loss: 0.853,Training Accuracy: 67.074,Validation Accuracy: 68.667
Epoch: 29=== Training Loss: 0.787,Validation Loss: 0.779,Training Accuracy: 68.185,Validation Accuracy: 67.333
Epoch: 30=== Training Loss: 0.729,Validation Loss: 0.700,Training Accuracy: 70.741,Validation Accuracy: 71.667
Epoch: 31=== Training Loss: 0.746,Validation Loss: 0.721,Training Accuracy: 70.222,Validation Accuracy: 70.333
Epoch: 32=== Training Loss: 0.720,Validation Loss: 0.677,Training Accuracy: 70.296,Validation Accuracy: 72.333
Epoch: 33=== Training Loss: 0.719,Validation Loss: 0.694,Training Accuracy: 70.704,Validation Accuracy: 73.000
Epoch: 34=== Training Loss: 0.736,Validation Loss: 0.713,Training Accuracy: 69.333,Validation Accuracy: 66.667
Epoch: 35=== Training Loss: 0.754,Validation Loss: 0.742,Training Accuracy: 69.741,Validation Accuracy: 69.333
Epoch: 36=== Training Loss: 0.796,Validation Loss: 0.803,Training Accuracy: 67.259,Validation Accuracy: 66.000
Epoch: 37=== Training Loss: 0.748,Validation Loss: 0.736,Training Accuracy: 68.630,Validation Accuracy: 71.000
Epoch: 38=== Training Loss: 0.678,Validation Loss: 0.638,Training Accuracy: 72.815,Validation Accuracy: 72.000
Epoch: 39=== Training Loss: 0.811,Validation Loss: 0.786,Training Accuracy: 68.667,Validation Accuracy: 69.000
Epoch: 40=== Training Loss: 0.670,Validation Loss: 0.652,Training Accuracy: 72.815,Validation Accuracy: 74.000
Epoch: 41=== Training Loss: 0.681,Validation Loss: 0.640,Training Accuracy: 70.556,Validation Accuracy: 71.667
Epoch: 42=== Training Loss: 0.668,Validation Loss: 0.633,Training Accuracy: 72.111,Validation Accuracy: 72.000
Epoch: 43=== Training Loss: 0.644,Validation Loss: 0.605,Training Accuracy: 72.407,Validation Accuracy: 73.333
Epoch: 44=== Training Loss: 0.673,Validation Loss: 0.658,Training Accuracy: 73.407,Validation Accuracy: 73.333
Epoch: 45=== Training Loss: 0.661,Validation Loss: 0.635,Training Accuracy: 73.074,Validation Accuracy: 74.000
```

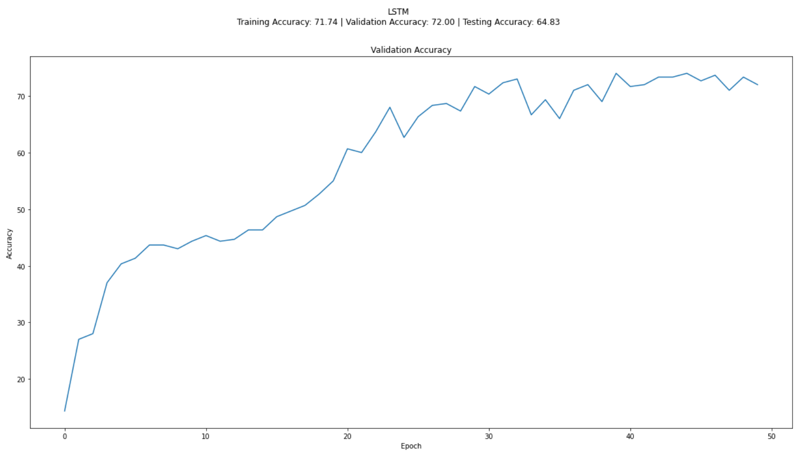
```
Epoch: 46=== Training Loss: 0.662,Validation Loss: 0.625,Training Accuracy: 72.222,Validation Accuracy: 72.667
Epoch: 47=== Training Loss: 0.655,Validation Loss: 0.625,Training Accuracy: 73.259,Validation Accuracy: 73.667
Epoch: 48=== Training Loss: 0.671,Validation Loss: 0.652,Training Accuracy: 72.222,Validation Accuracy: 71.000
Epoch: 49=== Training Loss: 0.679,Validation Loss: 0.664,Training Accuracy: 72.704,Validation Accuracy: 73.333
Epoch: 50=== Training Loss: 0.661,Validation Loss: 0.633,Training Accuracy: 71.741,Validation Accuracy: 72.000
```

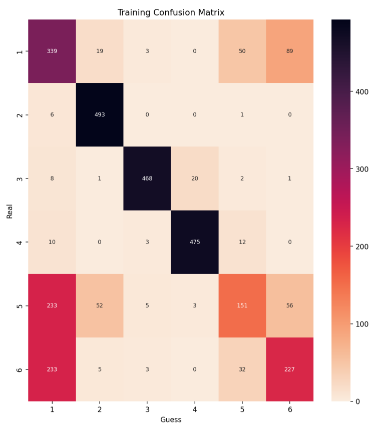
```
    Test Accuracy: 64.83333333333333
```

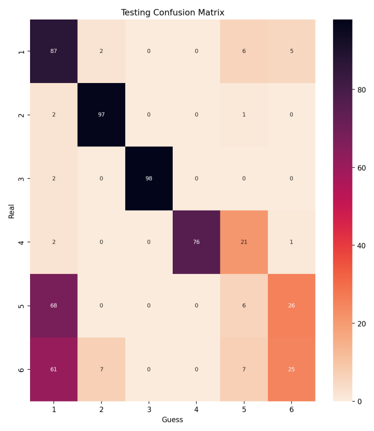










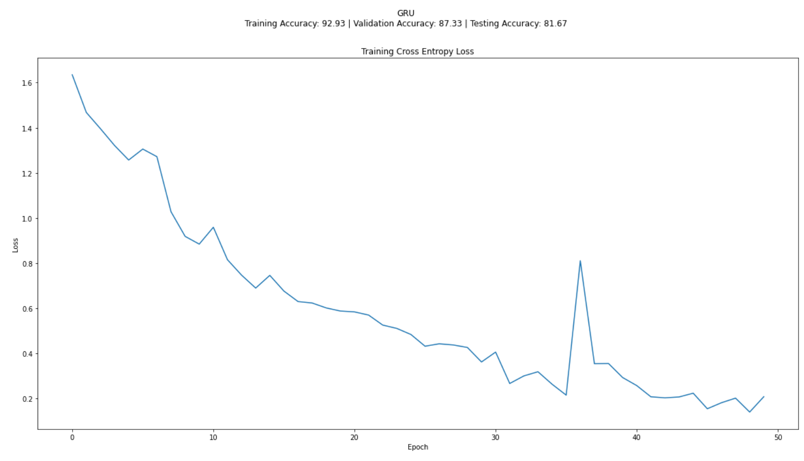


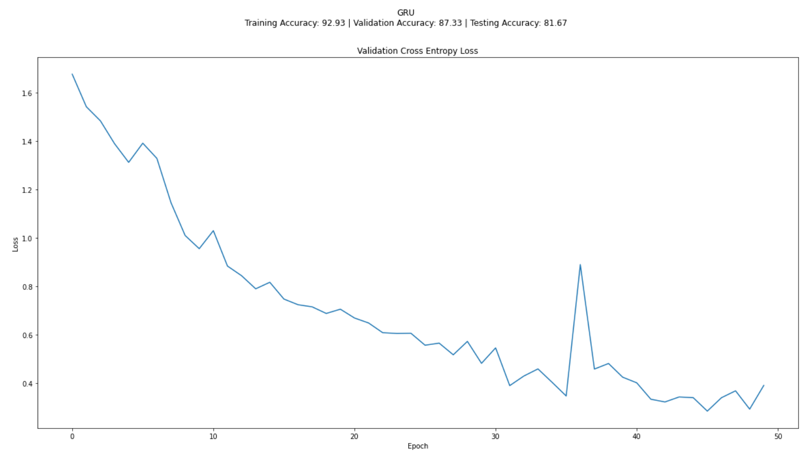
Appendices - GRU

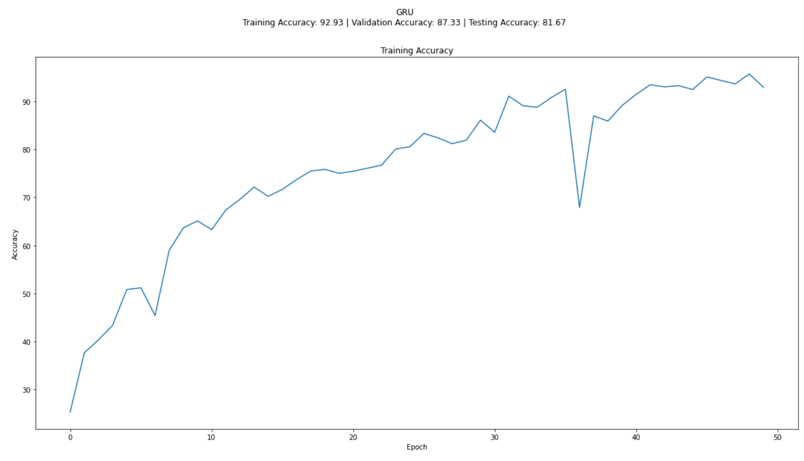
GRU Layer

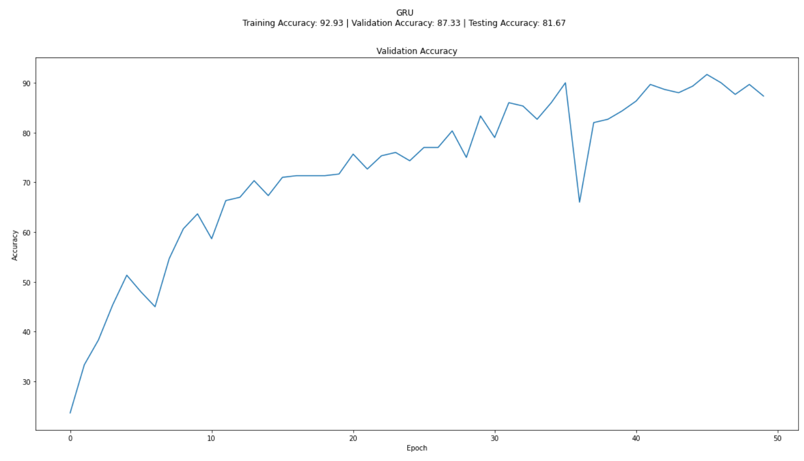
```
Epoch: 1=== Training Loss: 1.635,Validation Loss: 1.677,Training Accuracy: 25.259,Validation Accuracy: 23.667
Epoch: 2=== Training Loss: 1.468,Validation Loss: 1.542,Training Accuracy: 37.593,Validation Accuracy: 33.333
Epoch: 3=== Training Loss: 1.396,Validation Loss: 1.484,Training Accuracy: 40.296,Validation Accuracy: 38.333
Epoch: 4=== Training Loss: 1.321,Validation Loss: 1.389,Training Accuracy: 43.333,Validation Accuracy: 45.333
Epoch: 5=== Training Loss: 1.257,Validation Loss: 1.312,Training Accuracy: 50.778,Validation Accuracy: 51.333
Epoch: 6=== Training Loss: 1.306,Validation Loss: 1.392,Training Accuracy: 51.148,Validation Accuracy: 48.000
Epoch: 7=== Training Loss: 1.272,Validation Loss: 1.329,Training Accuracy: 45.370,Validation Accuracy: 45.000
Epoch: 8=== Training Loss: 1.028,Validation Loss: 1.146,Training Accuracy: 58.963,Validation Accuracy: 54.667
Epoch: 9=== Training Loss: 0.919,Validation Loss: 1.010,Training Accuracy: 63.630,Validation Accuracy: 60.667
Epoch: 10=== Training Loss: 0.884,Validation Loss: 0.956,Training Accuracy: 65.111,Validation Accuracy: 63.667
Epoch: 11=== Training Loss: 0.959,Validation Loss: 1.030,Training Accuracy: 63.259,Validation Accuracy: 58.667
Epoch: 12=== Training Loss: 0.815,Validation Loss: 0.884,Training Accuracy: 67.370,Validation Accuracy: 66.333
Epoch: 13=== Training Loss: 0.747,Validation Loss: 0.844,Training Accuracy: 69.593,Validation Accuracy: 67.000
Epoch: 14=== Training Loss: 0.690,Validation Loss: 0.790,Training Accuracy: 72.148,Validation Accuracy: 70.333
Epoch: 15=== Training Loss: 0.746,Validation Loss: 0.817,Training Accuracy: 70.222,Validation Accuracy: 67.333
Epoch: 16=== Training Loss: 0.676,Validation Loss: 0.748,Training Accuracy: 71.667,Validation Accuracy: 71.000
Epoch: 17=== Training Loss: 0.630,Validation Loss: 0.724,Training Accuracy: 73.704,Validation Accuracy: 71.333
Epoch: 18=== Training Loss: 0.623,Validation Loss: 0.716,Training Accuracy: 75.481,Validation Accuracy: 71.333
Epoch: 19=== Training Loss: 0.601,Validation Loss: 0.688,Training Accuracy: 75.852,Validation Accuracy: 71.333
Epoch: 20=== Training Loss: 0.588,Validation Loss: 0.706,Training Accuracy: 75.000,Validation Accuracy: 71.667
Epoch: 21=== Training Loss: 0.584,Validation Loss: 0.670,Training Accuracy: 75.444,Validation Accuracy: 75.667
Epoch: 22=== Training Loss: 0.570,Validation Loss: 0.649,Training Accuracy: 76.074,Validation Accuracy: 72.667
Epoch: 23=== Training Loss: 0.526,Validation Loss: 0.609,Training Accuracy: 76.704,Validation Accuracy: 75.333
Epoch: 24=== Training Loss: 0.511,Validation Loss: 0.606,Training Accuracy: 80.074,Validation Accuracy: 76.000
Epoch: 25=== Training Loss: 0.484,Validation Loss: 0.607,Training Accuracy: 80.556,Validation Accuracy: 74.333
Epoch: 26=== Training Loss: 0.432,Validation Loss: 0.557,Training Accuracy: 83.333,Validation Accuracy: 77.000
Epoch: 27=== Training Loss: 0.442,Validation Loss: 0.566,Training Accuracy: 82.407,Validation Accuracy: 77.000
Epoch: 28=== Training Loss: 0.437,Validation Loss: 0.518,Training Accuracy: 81.185,Validation Accuracy: 80.333
Epoch: 29=== Training Loss: 0.426,Validation Loss: 0.573,Training Accuracy: 81.926,Validation Accuracy: 75.000
Epoch: 30=== Training Loss: 0.362,Validation Loss: 0.482,Training Accuracy: 86.111,Validation Accuracy: 83.333
Epoch: 31=== Training Loss: 0.406,Validation Loss: 0.546,Training Accuracy: 83.556,Validation Accuracy: 79.000
Epoch: 32=== Training Loss: 0.267,Validation Loss: 0.390,Training Accuracy: 91.111,Validation Accuracy: 86.000
Epoch: 33=== Training Loss: 0.300,Validation Loss: 0.430,Training Accuracy: 89.111,Validation Accuracy: 85.333
Epoch: 34=== Training Loss: 0.319,Validation Loss: 0.459,Training Accuracy: 88.778,Validation Accuracy: 82.667
Epoch: 35=== Training Loss: 0.263,Validation Loss: 0.404,Training Accuracy: 90.815,Validation Accuracy: 86.000
Epoch: 36=== Training Loss: 0.215,Validation Loss: 0.348,Training Accuracy: 92.556,Validation Accuracy: 90.000
Epoch: 37=== Training Loss: 0.811,Validation Loss: 0.890,Training Accuracy: 67.889,Validation Accuracy: 66.000
Epoch: 38=== Training Loss: 0.354,Validation Loss: 0.459,Training Accuracy: 87.000,Validation Accuracy: 82.000
Epoch: 39=== Training Loss: 0.355,Validation Loss: 0.482,Training Accuracy: 85.889,Validation Accuracy: 82.667
Epoch: 40=== Training Loss: 0.293,Validation Loss: 0.425,Training Accuracy: 89.148,Validation Accuracy: 84.333
Epoch: 41=== Training Loss: 0.257,Validation Loss: 0.402,Training Accuracy: 91.481,Validation Accuracy: 86.333
Epoch: 42=== Training Loss: 0.208,Validation Loss: 0.334,Training Accuracy: 93.481,Validation Accuracy: 89.667
Epoch: 43=== Training Loss: 0.203,Validation Loss: 0.323,Training Accuracy: 93.037,Validation Accuracy: 88.667
Epoch: 44=== Training Loss: 0.207,Validation Loss: 0.343,Training Accuracy: 93.296,Validation Accuracy: 88.000
Epoch: 45=== Training Loss: 0.224,Validation Loss: 0.341,Training Accuracy: 92.481,Validation Accuracy: 89.333
Epoch: 46=== Training Loss: 0.155,Validation Loss: 0.285,Training Accuracy: 95.111,Validation Accuracy: 91.667
Epoch: 47=== Training Loss: 0.181,Validation Loss: 0.340,Training Accuracy: 94.370,Validation Accuracy: 90.000
Epoch: 48=== Training Loss: 0.202,Validation Loss: 0.369,Training Accuracy: 93.667,Validation Accuracy: 87.667
Epoch: 49=== Training Loss: 0.140,Validation Loss: 0.293,Training Accuracy: 95.741,Validation Accuracy: 89.667
Epoch: 50=== Training Loss: 0.208,Validation Loss: 0.391,Training Accuracy: 92.926,Validation Accuracy: 87.333
```

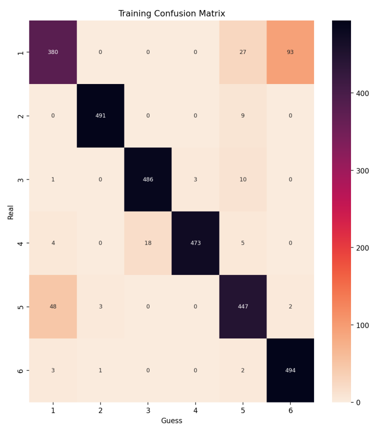
Test Accuracy: 81.66666666666667

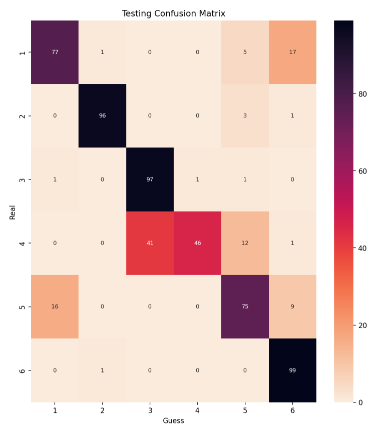

GRU
Training Accuracy: 92.93 | Validation Accuracy: 87.33 | Testing Accuracy: 81.67
Training Cross Entropy Loss
Loss
Epoch


GRU
Training Accuracy: 92.93 | Validation Accuracy: 87.33 | Testing Accuracy: 81.67
Validation Cross Entropy Loss
Loss
Epoch


GRU
Training Accuracy: 92.93 | Validation Accuracy: 87.33 | Testing Accuracy: 81.67
Training Accuracy
Accuracy
Epoch


GRU
Training Accuracy: 92.93 | Validation Accuracy: 87.33 | Testing Accuracy: 81.67
Validation Accuracy
Accuracy
Epoch


Training Confusion Matrix
Real
Guess
380 0 0 0 27 93
0 491 0 0 9 0
1 0 486 3 10 0
4 0 18 473 5 0
48 3 0 0 447 2
3 1 0 0 2 494


Testing Confusion Matrix
Real
Guess
77 1 0 0 5 17
0 96 0 0 3 1
1 0 97 1 1 0
0 0 41 46 12 1
16 0 0 0 75 9
0 1 0 0 0 99