GEBZE TECHNICAL UNIVERSITY

CSE 655 DEEP LEARNING

HOMEWORK -1 REPORT

NAME: SUHA BERK KUKUK

STUDENT ID: 235008003024

Part 1

i) I built a 3 hidden layers model with each layer having 6 nodes. Train model with using Mean Square Error (MSE) loss and Scholastic Gradient Decent (SGD), learning rate 0.01 and epoch is 100. Set activation function swish, relu and sigmoid. Sigmoid function for output layer. This step I tried to many learning rate and epoch values, I choose best of them.

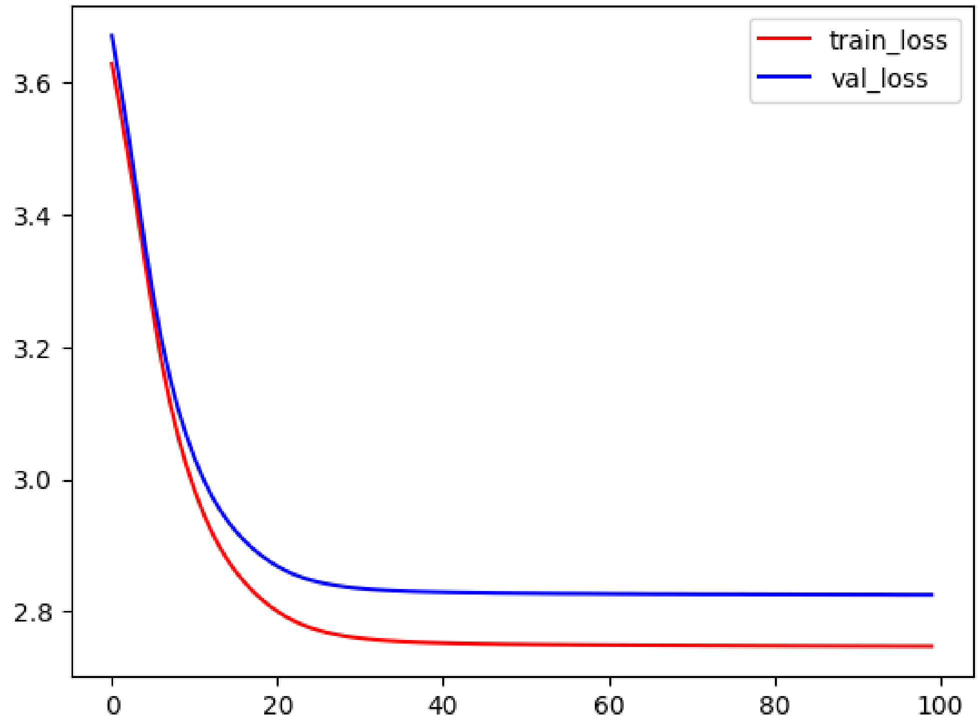

Figure 1: Loss Graphics

ii) Changing activation function with Elu without output layer activation.

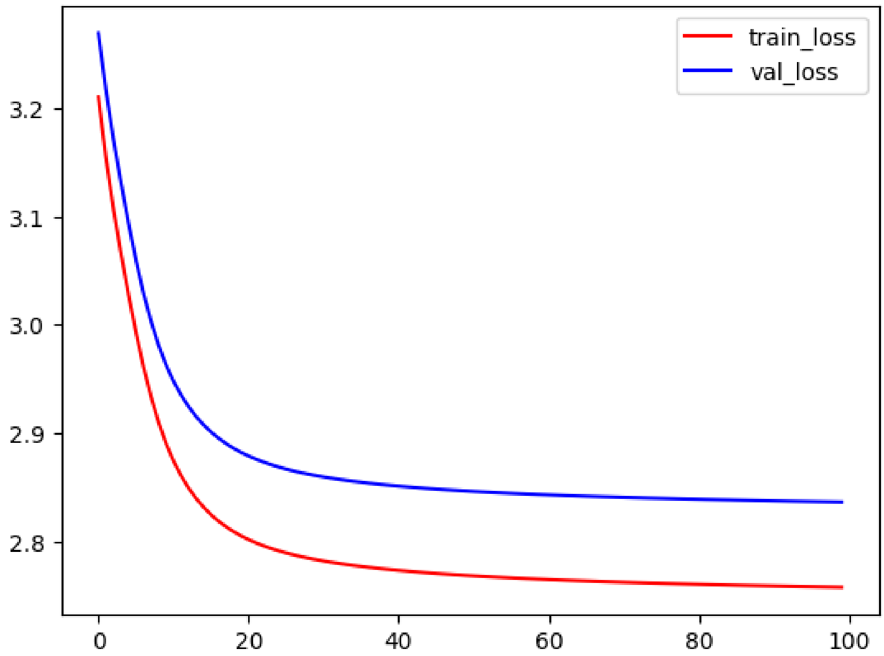

Figure 2: Loss Graphics

iii) Changing activation function with Swish without output layer activation.

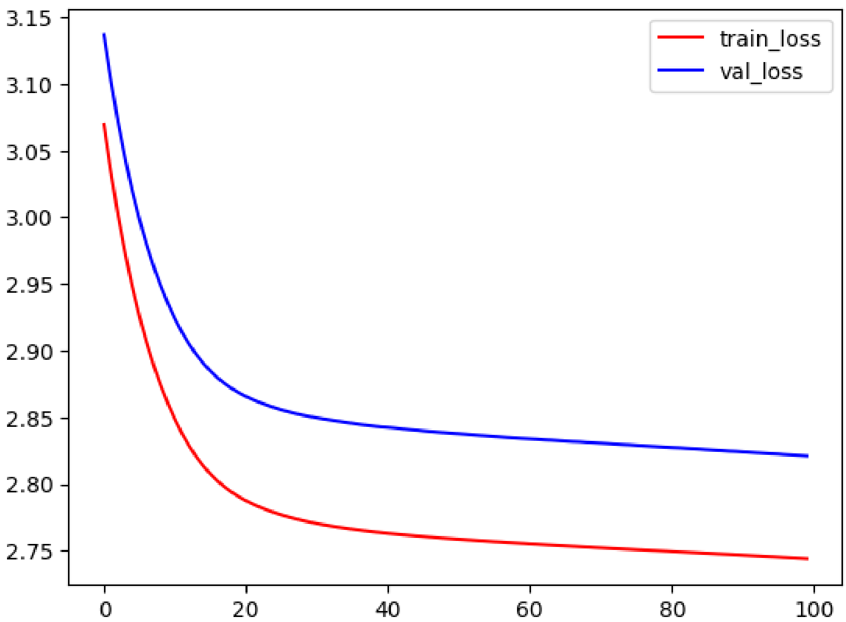

Figure 3: Loss Graphics

iv) Changing activation function with Relu without output layer activation.

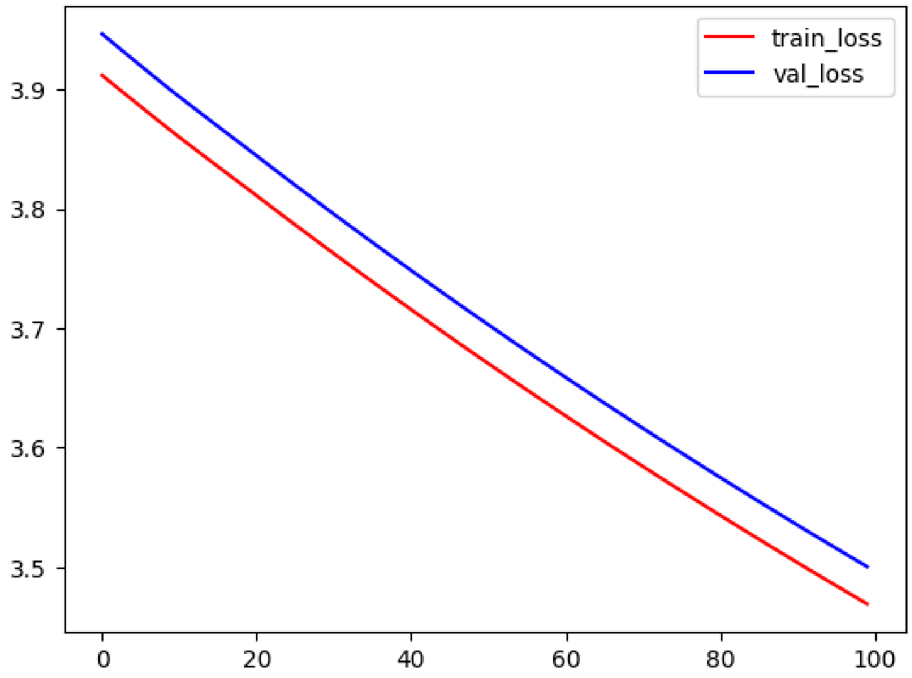

Figure 3: Loss Graphics

v) Combine activation function with relu and swish. First and second hidden layer use relu. Third hidden layer use swish.

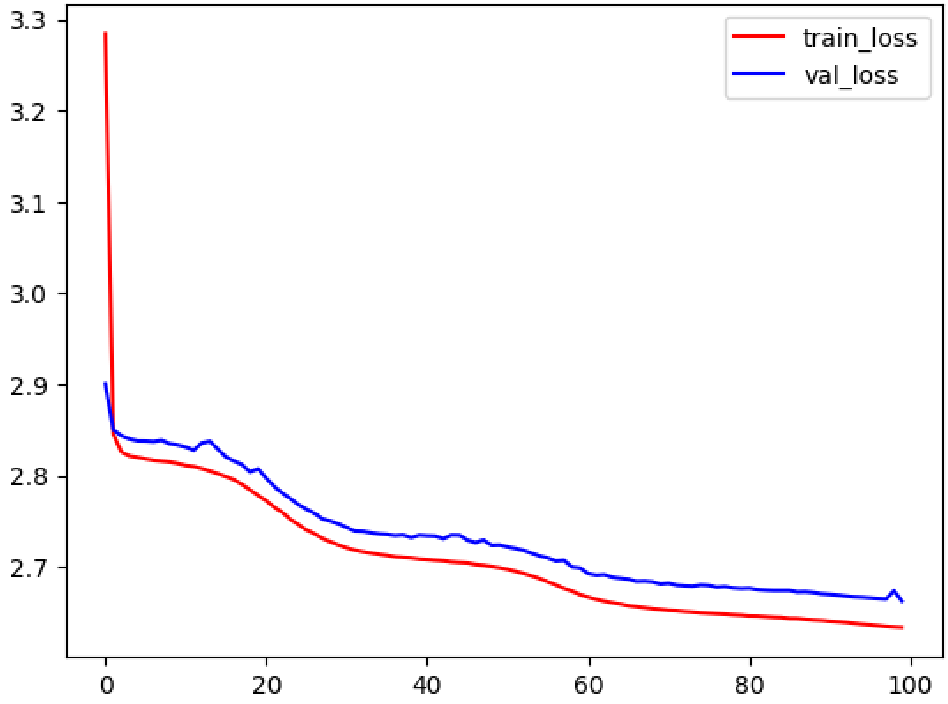

Figure 4: Loss Graphics

vi) Activation function set swish.

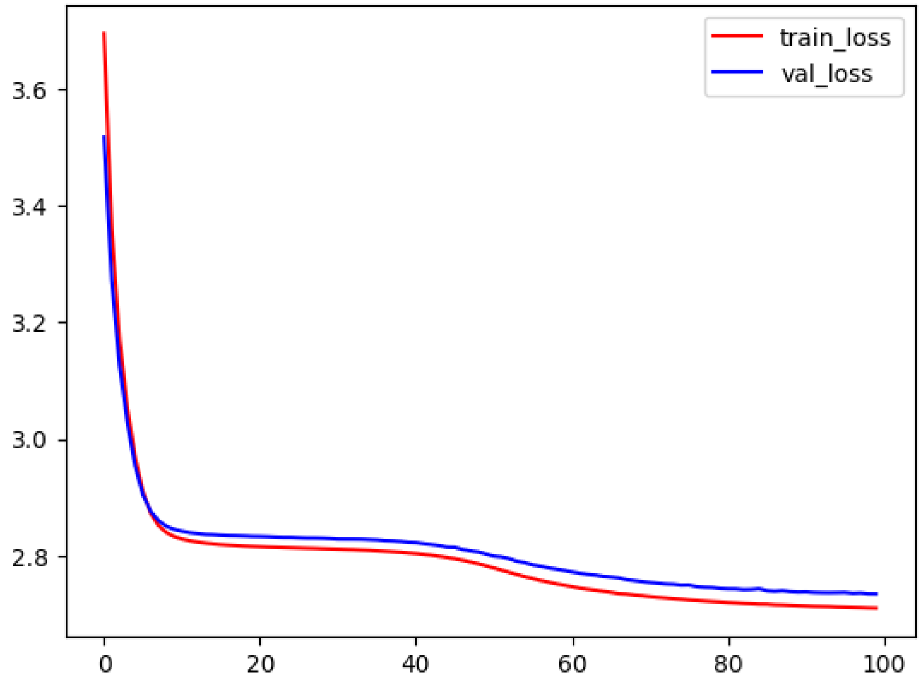

Figure 5: Loss Graphics

vii) Change learning rate parameters from 0.01 to 0.1. Train again model.

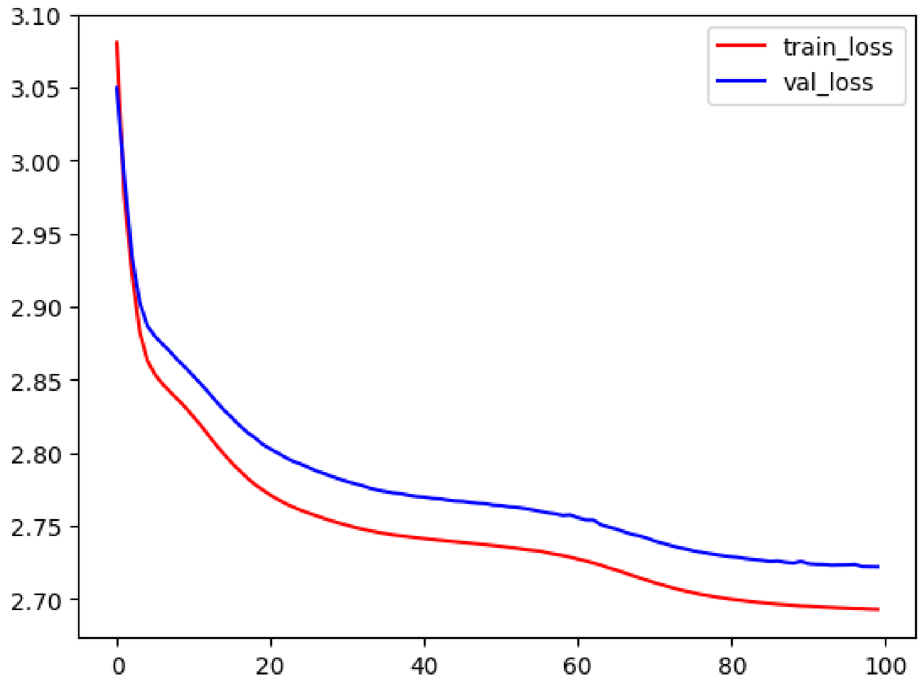

Figure 6: Loss Graphics

viii) Change activation function swish to tanh and train with learning rate 0.1.

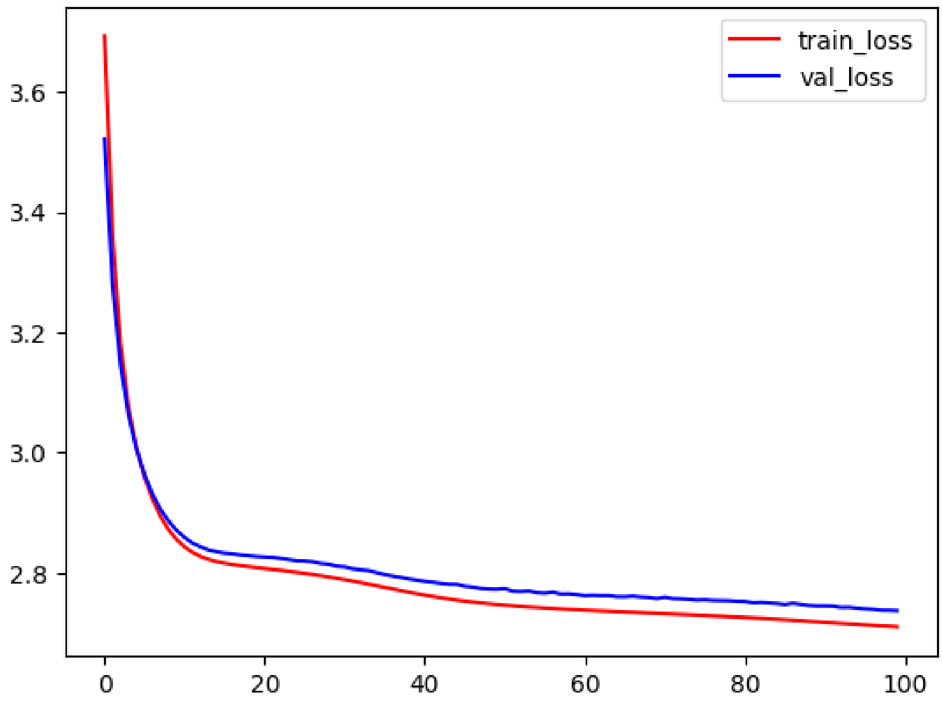

Figure 7: Loss Graphics

IX) Add new nodes at a time to each hidden layer and repeat steps above. Figure 8 – Figure 16

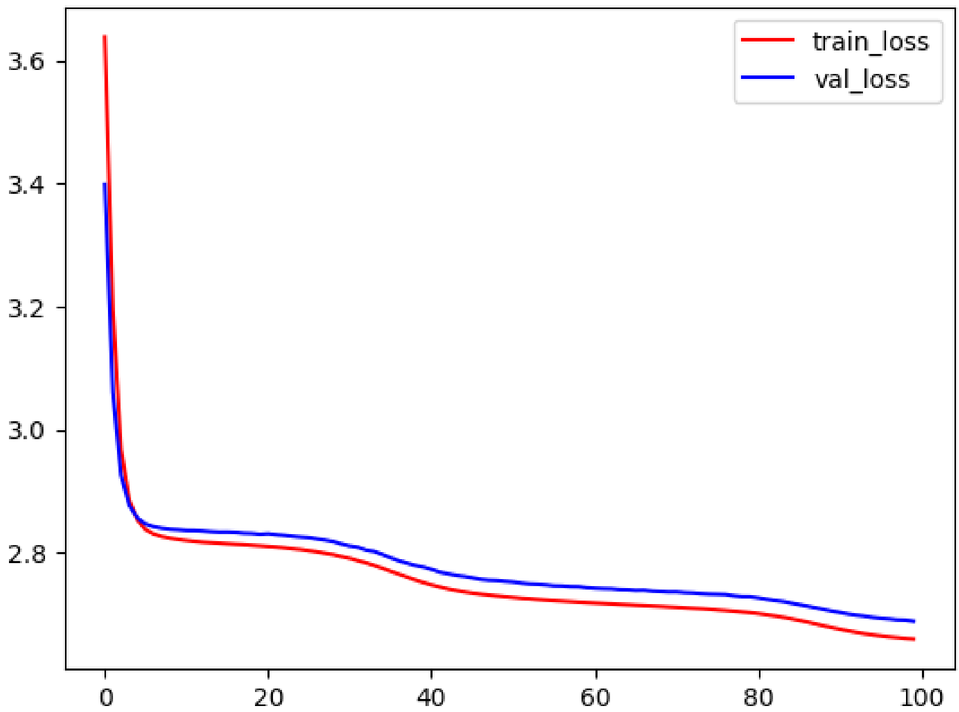

Figure 8: Loss Graphics

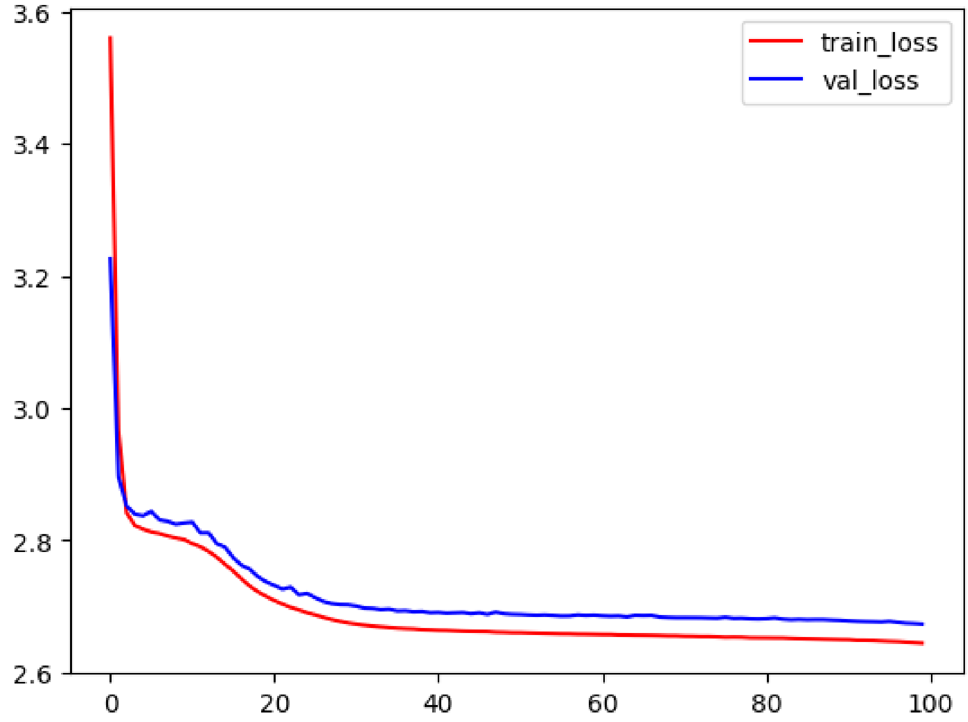

Figure 9: Loss Graphics

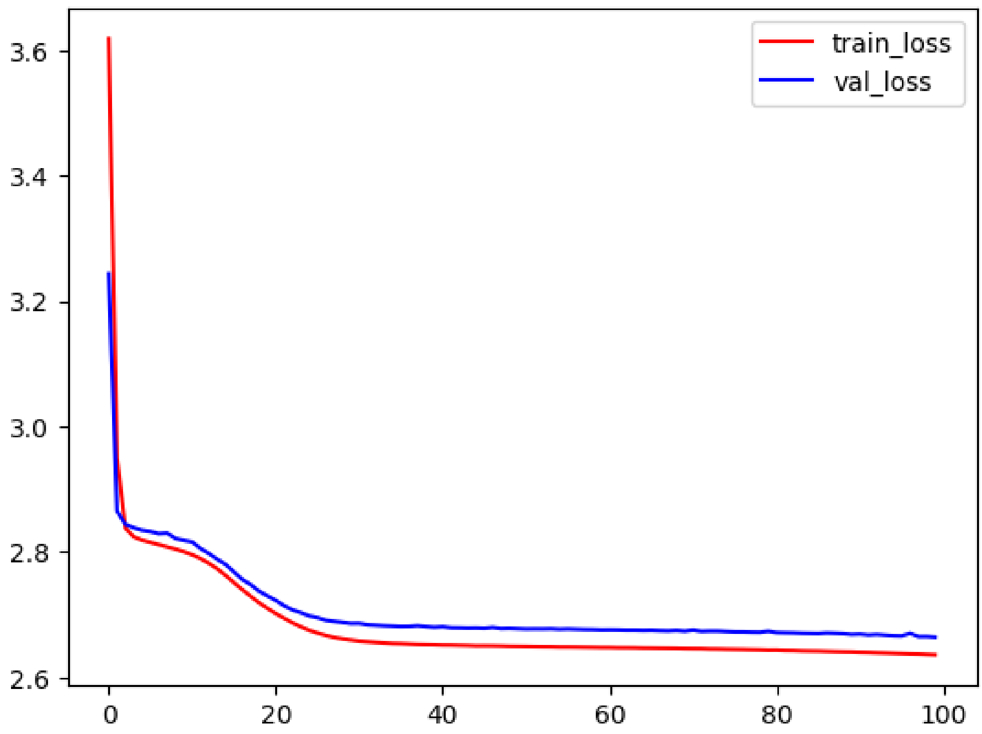

Figure 10: Loss Graphics

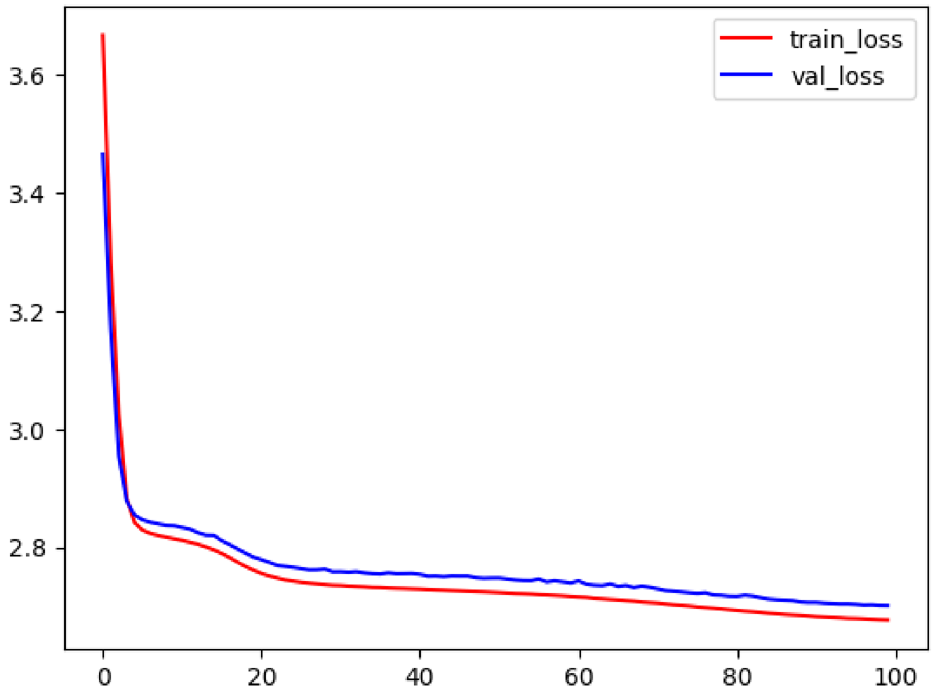

Figure 11: Loss Graphics

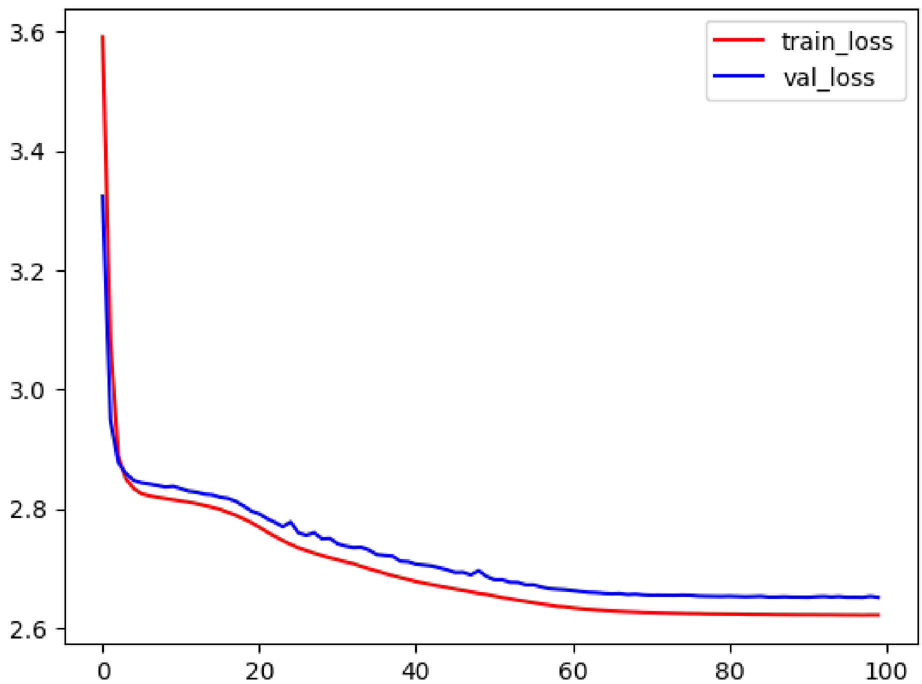

Figure 12: Loss Graphics

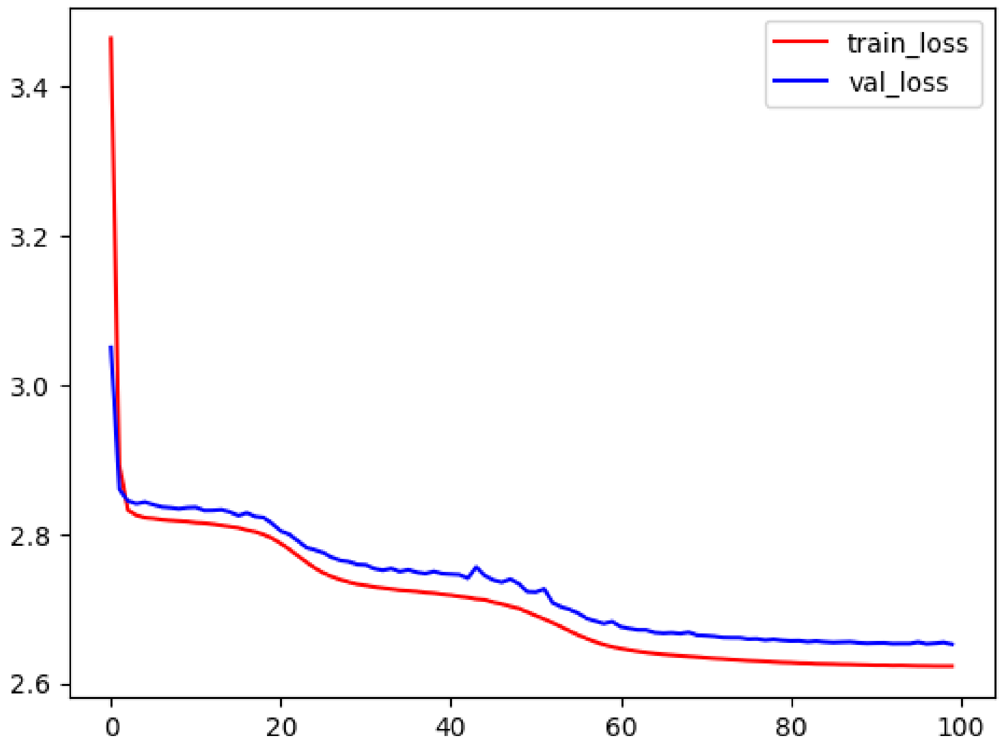

Figure 13: Loss Graphics

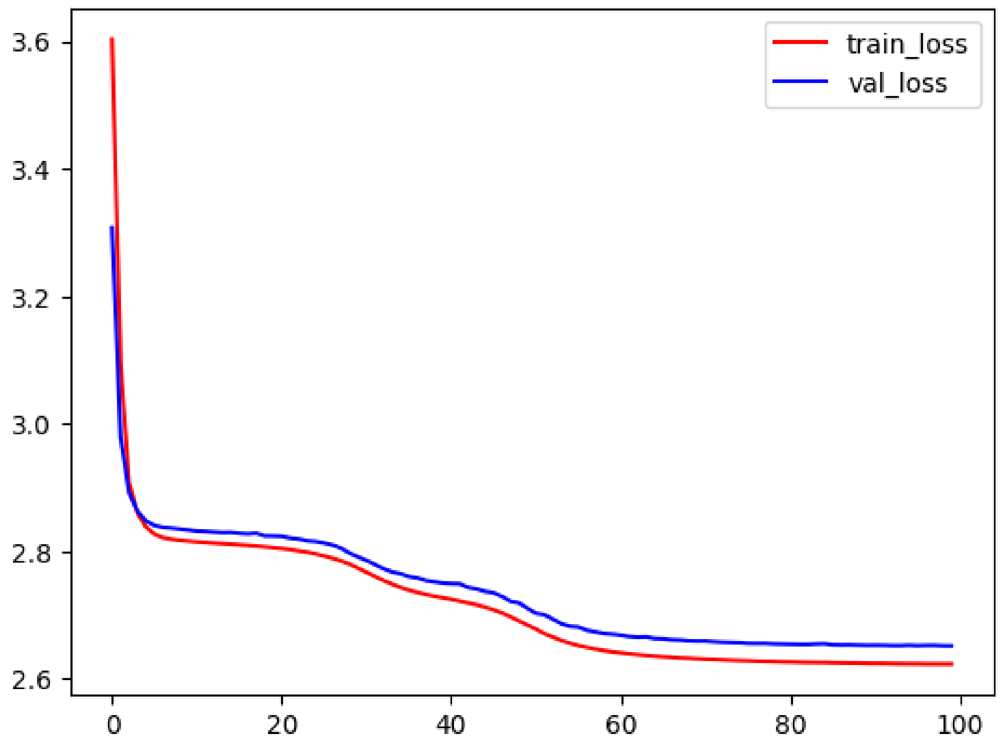

Figure 14: Loss Graphics

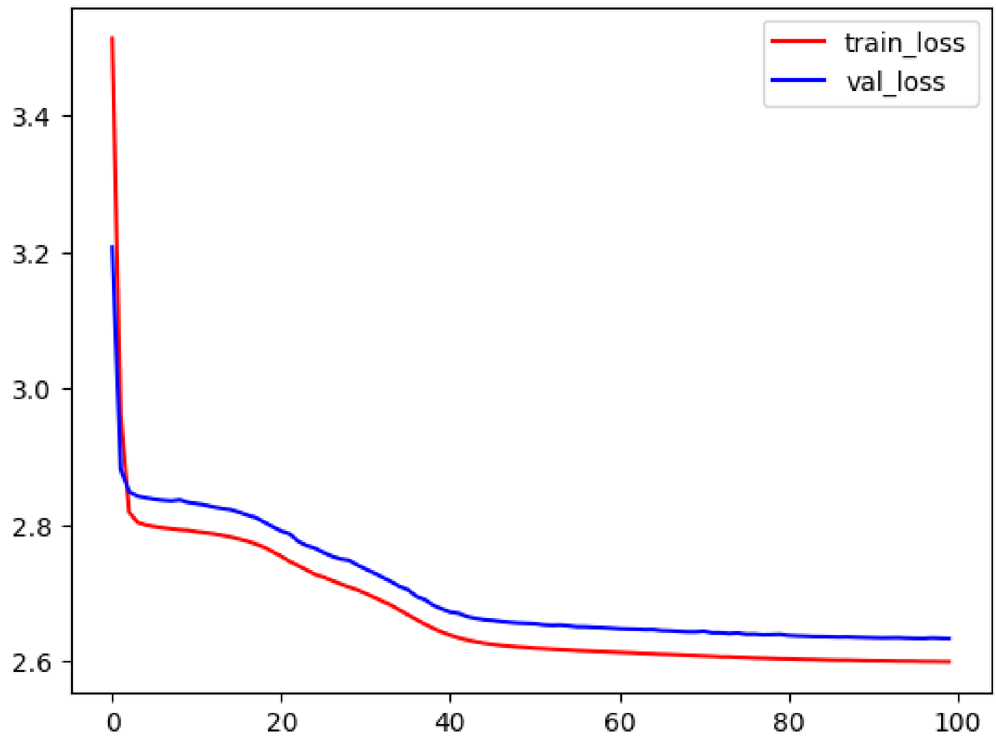

Figure 15: Loss Graphics

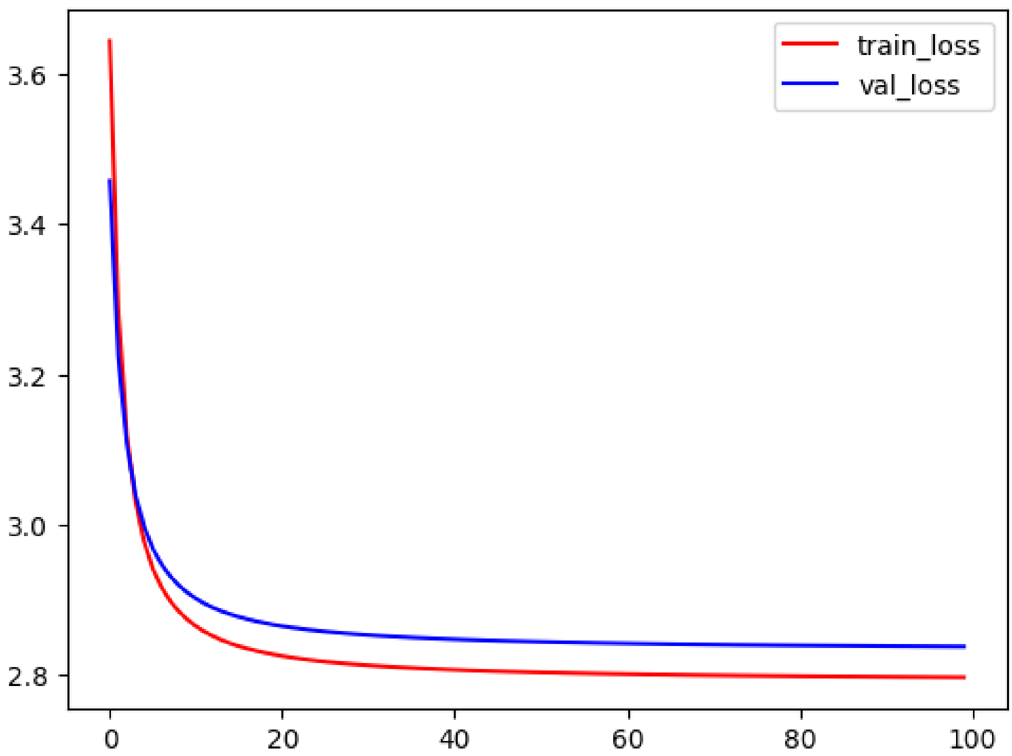

Figure 16: Loss Graphics

V) Increasing date %10 and train again with above parameters. Figure 17 – Figure 22

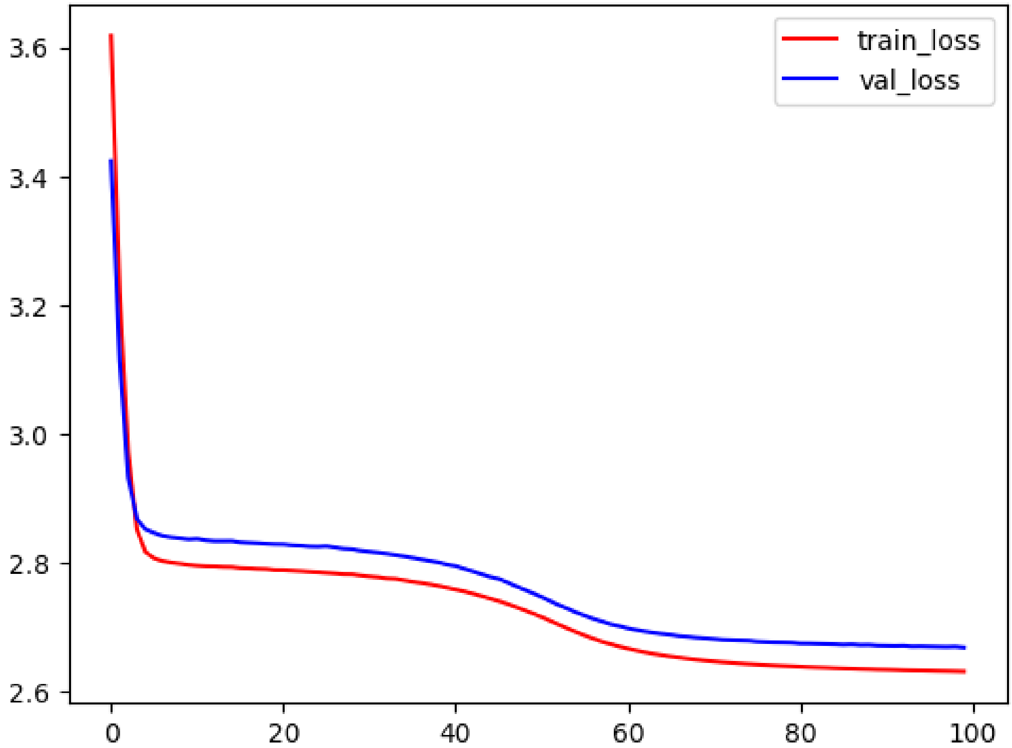

Figure 17: Loss Graphics

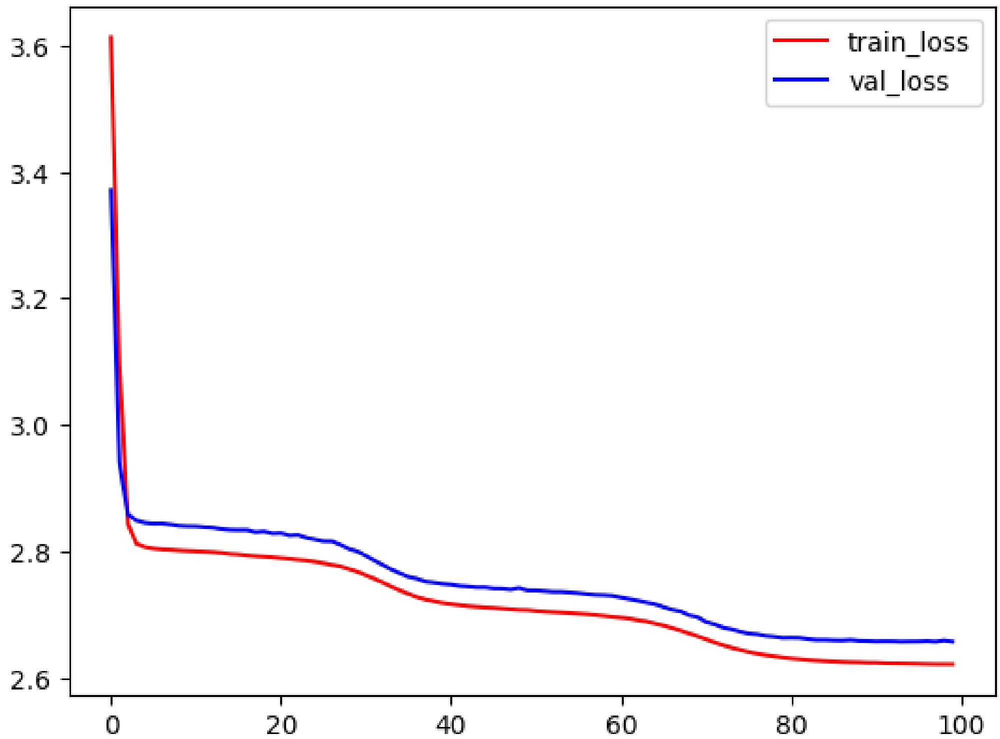

Figure 18: Loss Graphics

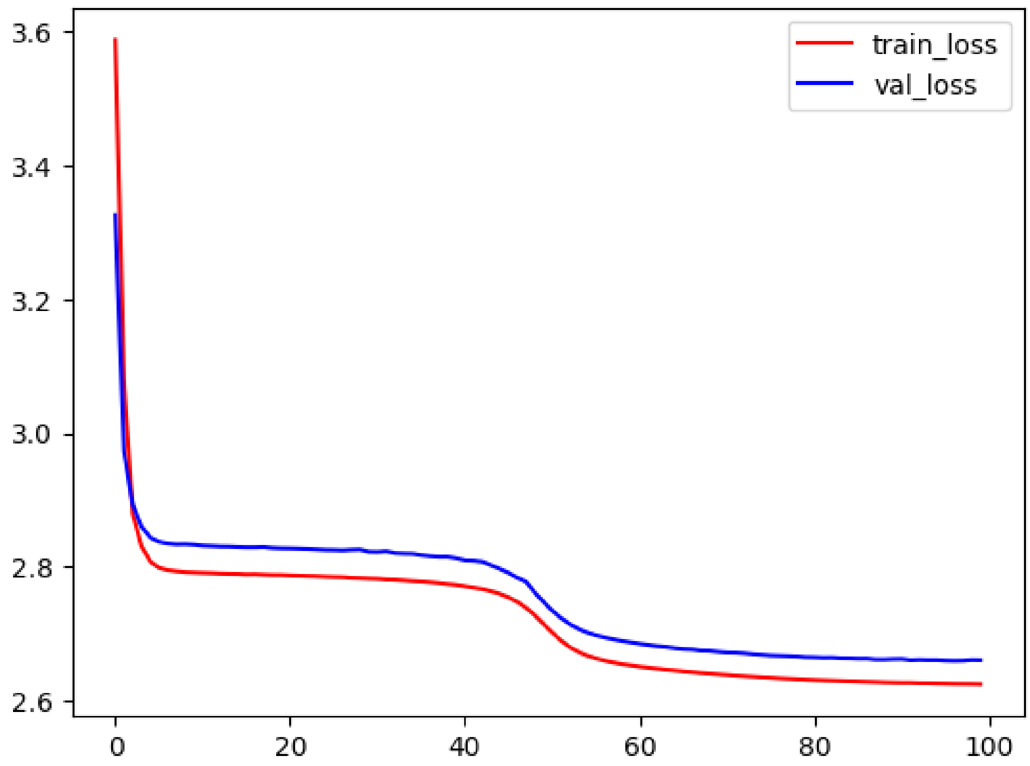

Figure 19: Loss Graphics

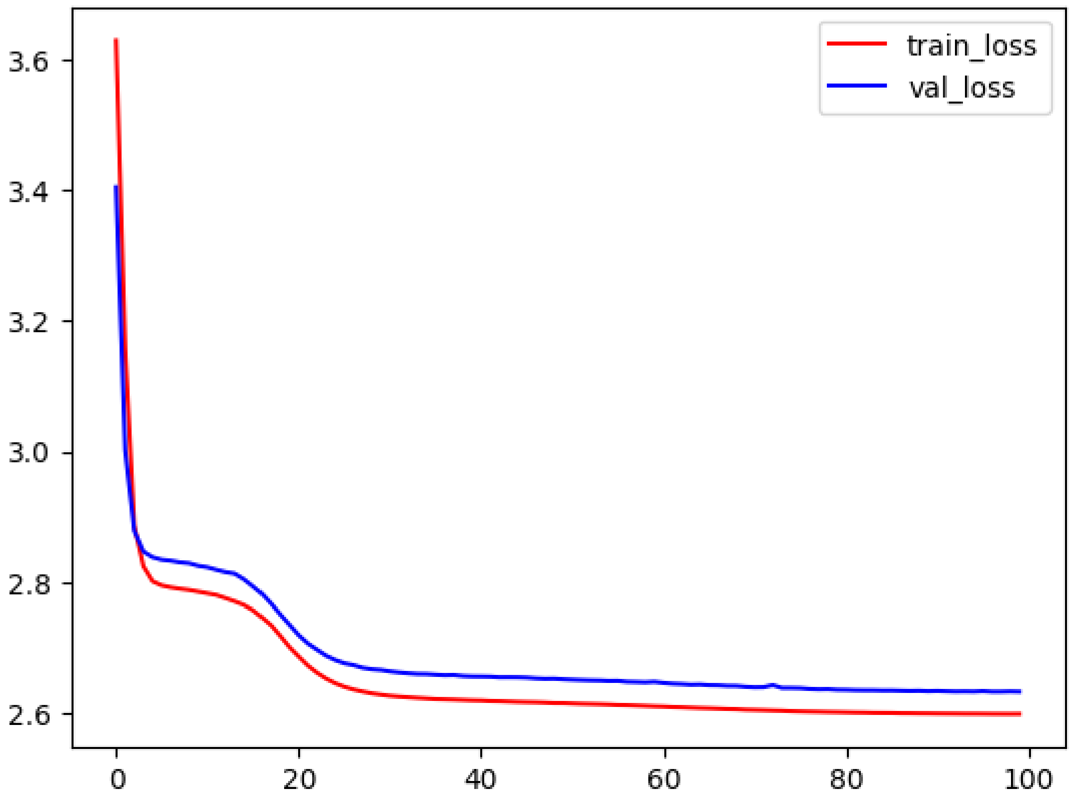

Figure 20: Loss Graphics

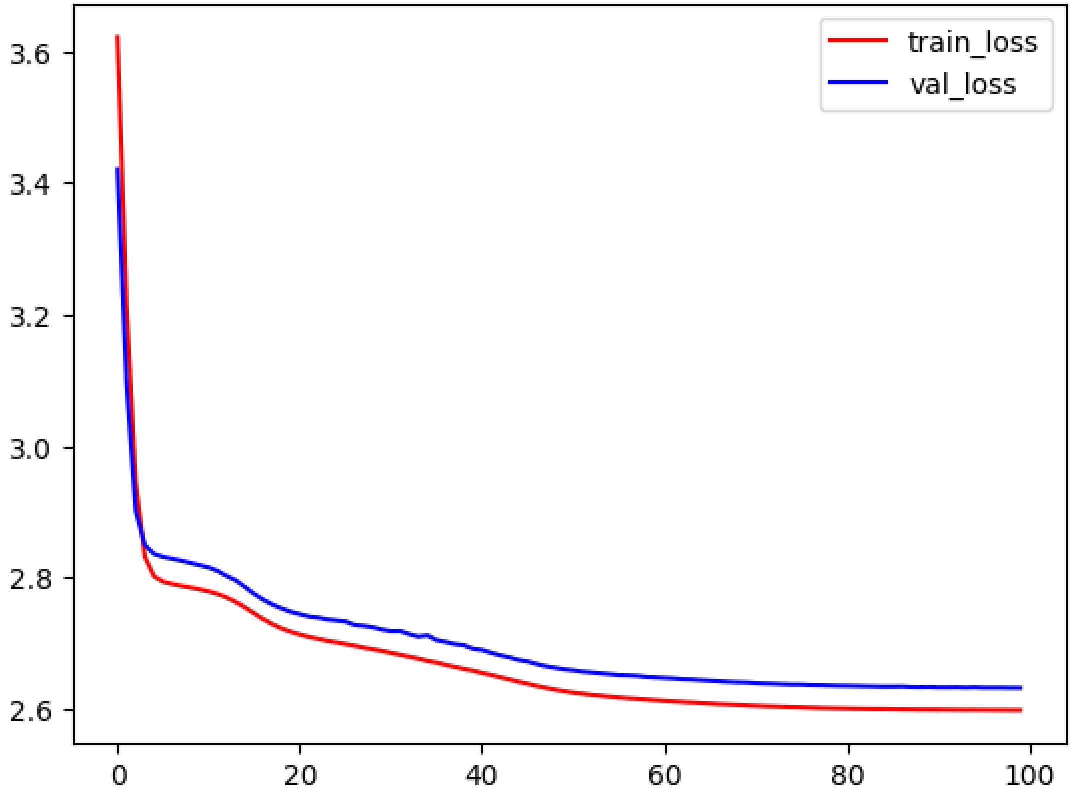

Figure 21: Loss Graphics

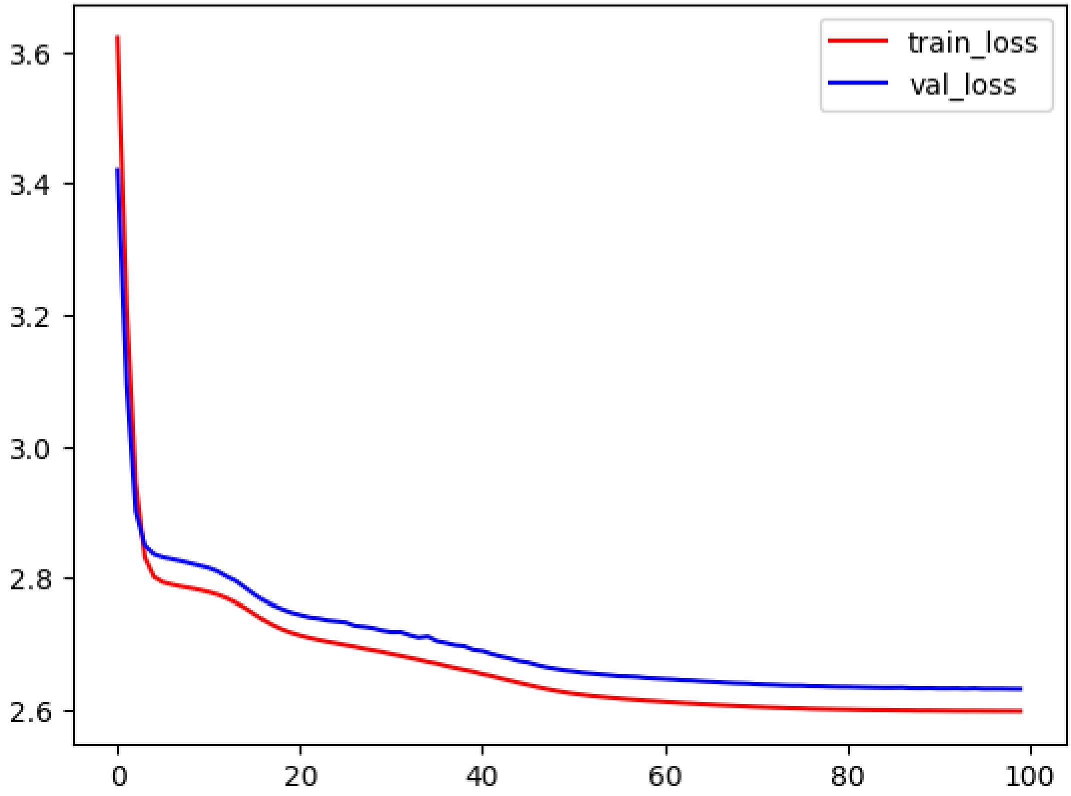

Figure 22: Loss Graphics

My comments:

Until this stage, I have implemented the creation of a dataset and the training of a sequential model. I created a function called 'create data' that takes an input 'Nt' for the number of instances in the data parameter and a Boolean value to determine if the data is used for training. If so, noise is added, along with the parameters 'mu' and 'standard validation' value. I generated random numbers for the input using the random function from the NumPy library.

Another function, 'train,' was implemented to train a model with specific parameters. At the end of the training process, I plotted the loss graphs with respect to epoch. The resulting figure represented the loss graph (Figure 1). According to this graph, the model fitted the data well up to a certain point, after which the training loss decreased significantly more than the validation loss. This phenomenon indicates that the model was overfitting, and the goal was to bring the validation loss closer to the training loss.

To address the overfitting issue, I experimented with changing the activation functions. In Figure 2, I replaced the activation function with tanh, but the model failed to learn, displaying a clear case of underfitting, which was worse than the initial model (Figure 1). Additionally, changing the activation function to ELU (Figure 3) resulted in a similar underfitting issue, indicating that continuous functions better represented the problem compared to piecewise linear functions like ELU.

Intriguingly, when I switched to the Swish activation function (Figure 4), the model's performance improved significantly. The loss graphs showed a better fit, indicating that continuous functions better addressed the problem compared to linear functions. However, using ReLU (Figure 5) demonstrated that a purely linear activation function may not adequately represent the output, leading to higher losses and parallel decrease in both training and validation losses.

By combining ReLU and Swish, I achieved a more relative model, with the validation loss decreasing in tandem with the training loss, and the gap between the two minimized. This suggested a well-fitted model. Continuing with the Swish activation function, a learning rate of 0.1, and the last layer's output set as sigmoid, I experimented with increasing the number of nodes in each layer. Adding two nodes to each layer improved the model's performance significantly, as shown in Figure 8, where the loss values decreased equally. However, adding two nodes to the second and third hidden layers demonstrated a diminishing return, indicating that increasing model complexity is not always beneficial, especially for less complex problems.

This trend was observed consistently when I increased the data by 10%. The findings from systematic experimentation emphasized that the Swish activation function yielded the most effective results, leading to a well-fitted and optimally performing model. Additionally, it underscored the importance of balancing model complexity with data size and ensuring that the chosen model adequately generalizes the problem at hand.

Part 2

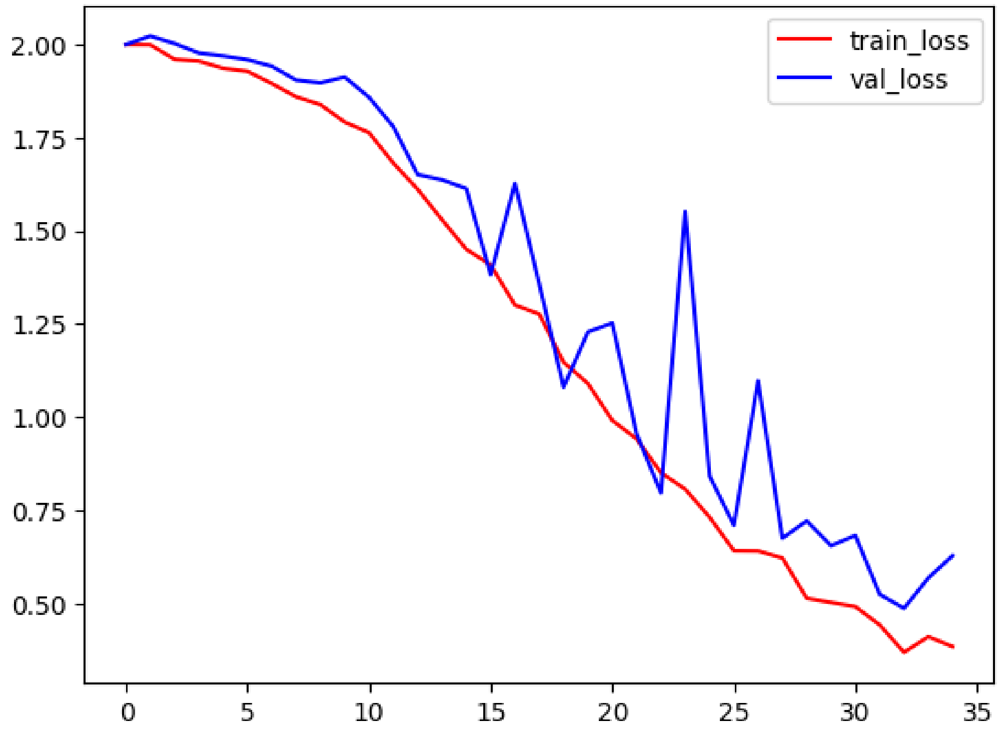

Figure 23: Loss Graphics

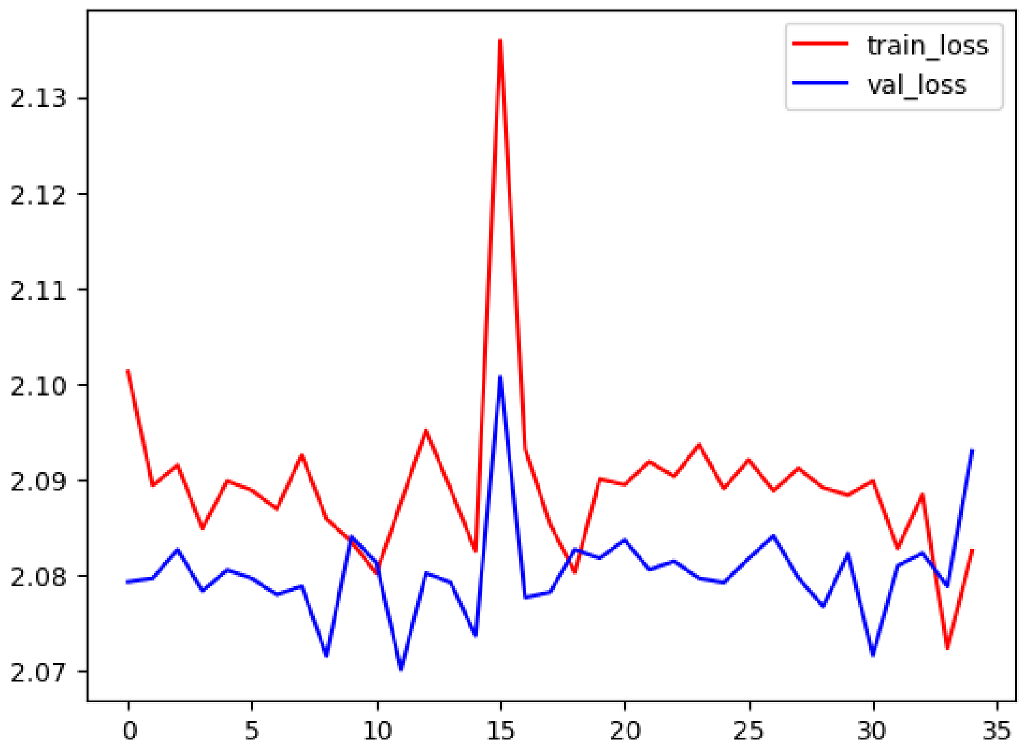

Figure 24: Loss Graphics

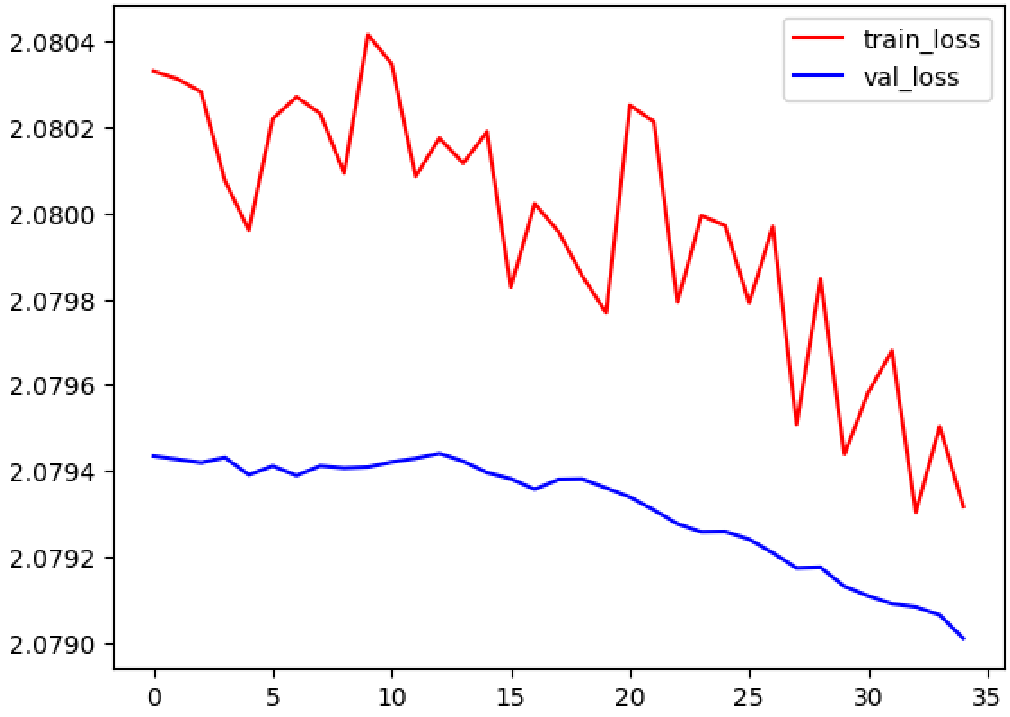

Figure 25: Loss Graphics

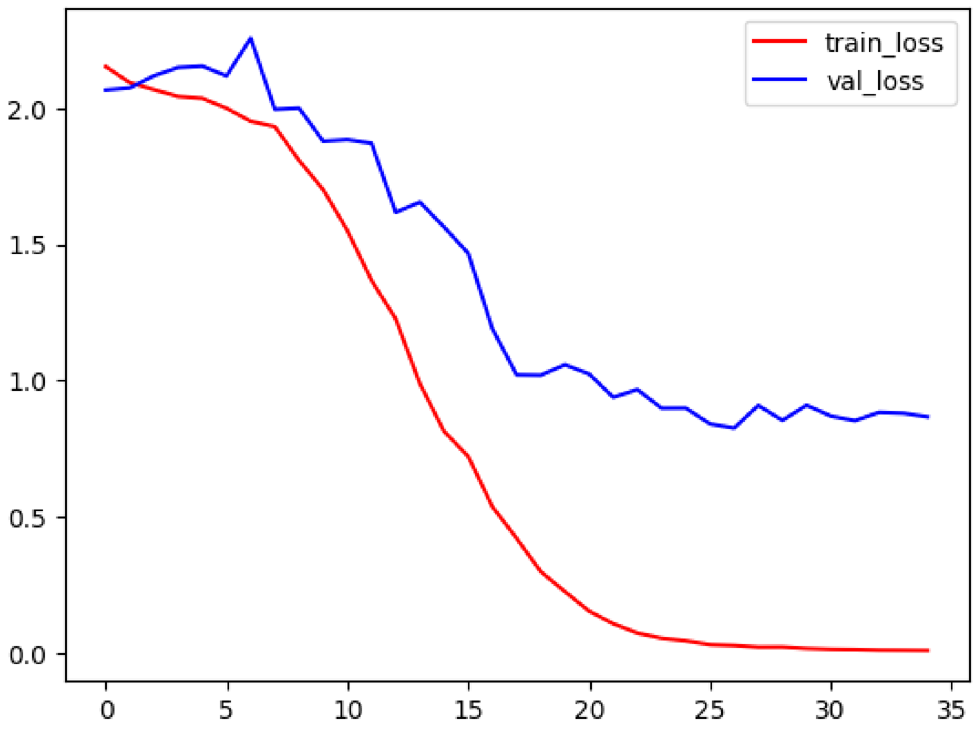

Figure 26: Loss Graphics

Figure 27: Loss Graphics

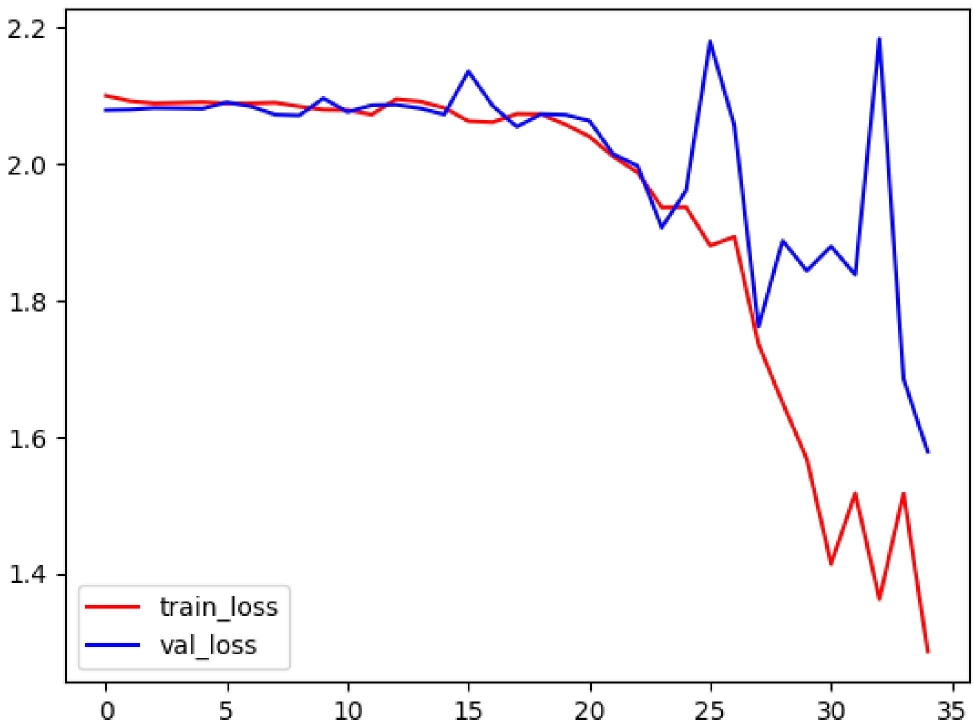

Figure 28: Loss Graphics

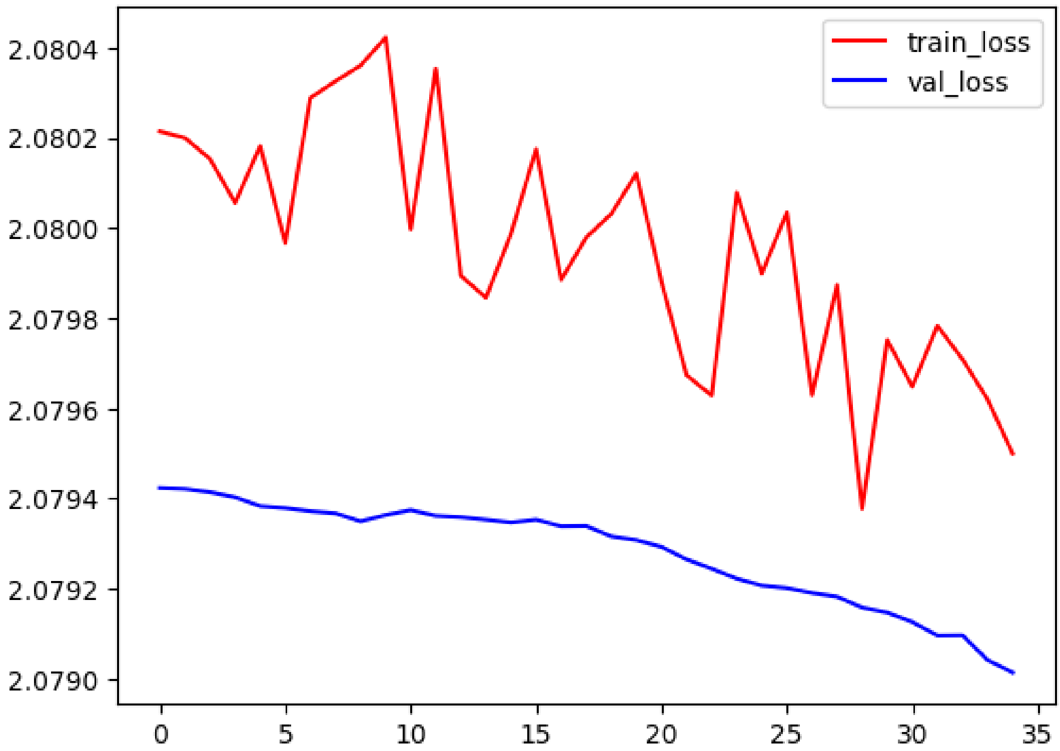

Figure 29: Loss Graphics

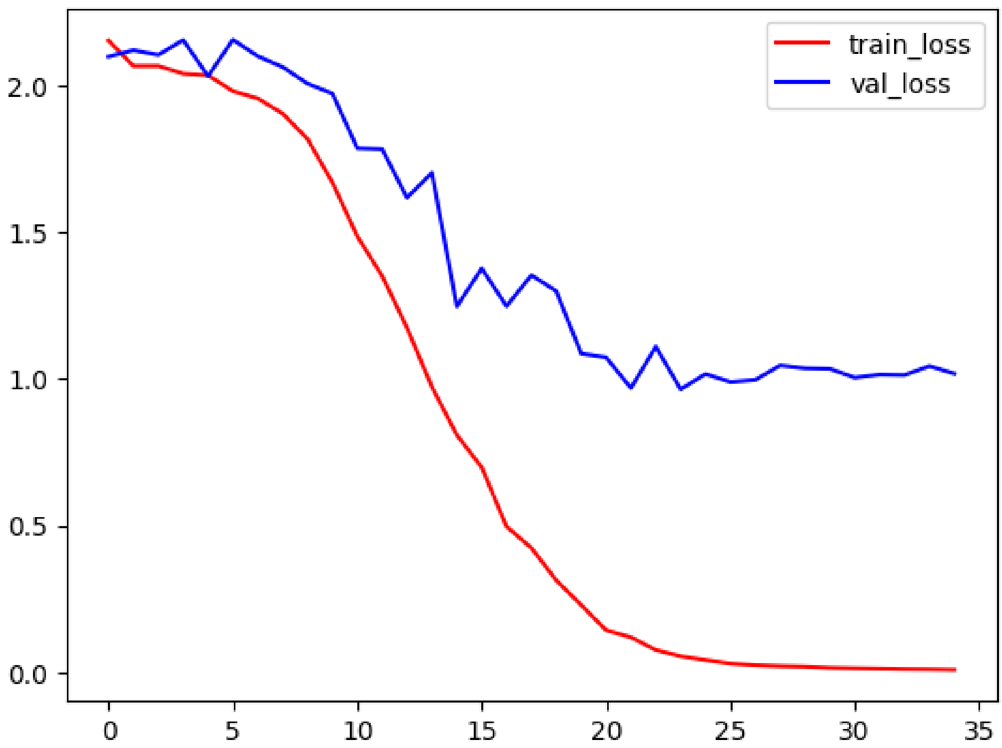

Figure 30: Loss Graphics

Figure 31: Loss Graphics

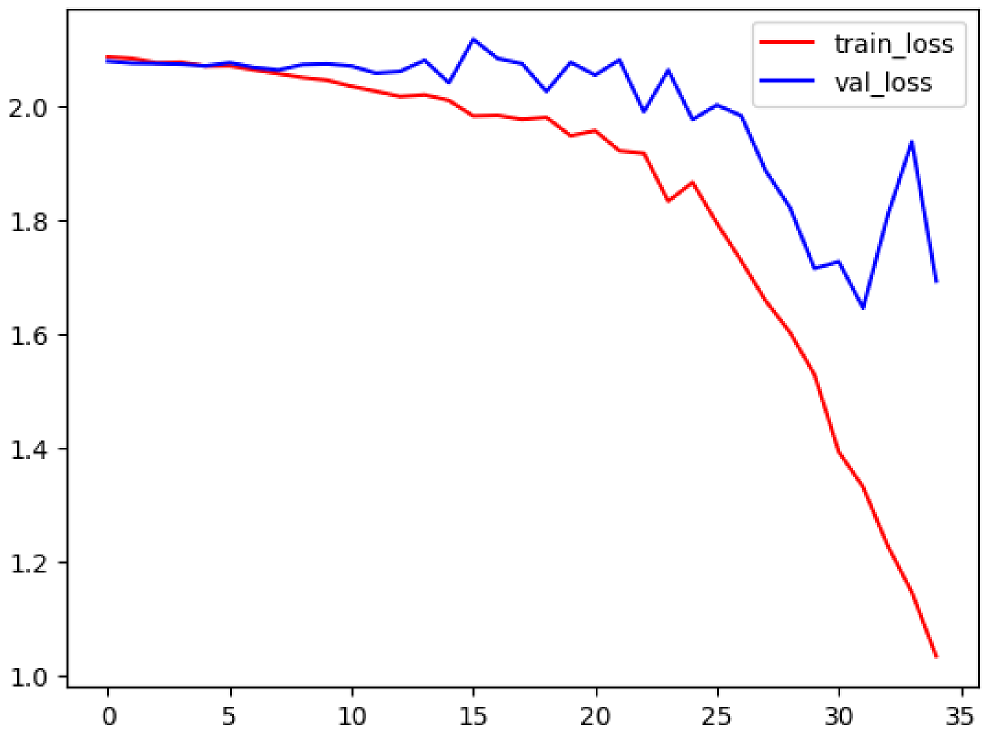

Figure 32: Loss Graphics

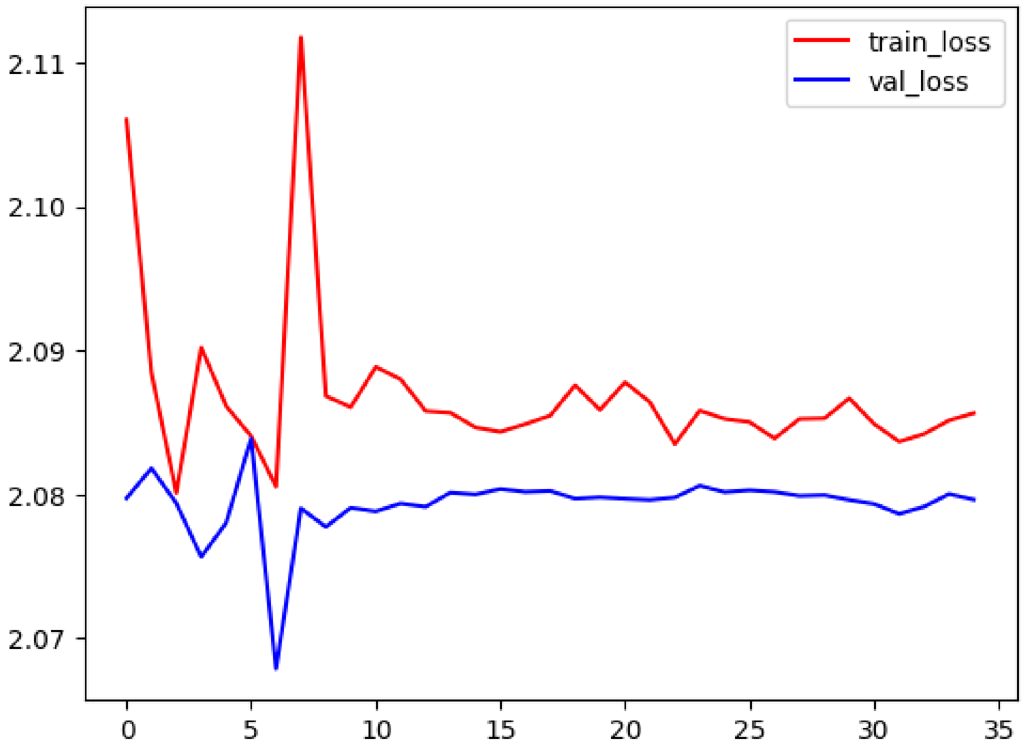

Figure 33: Loss Graphics

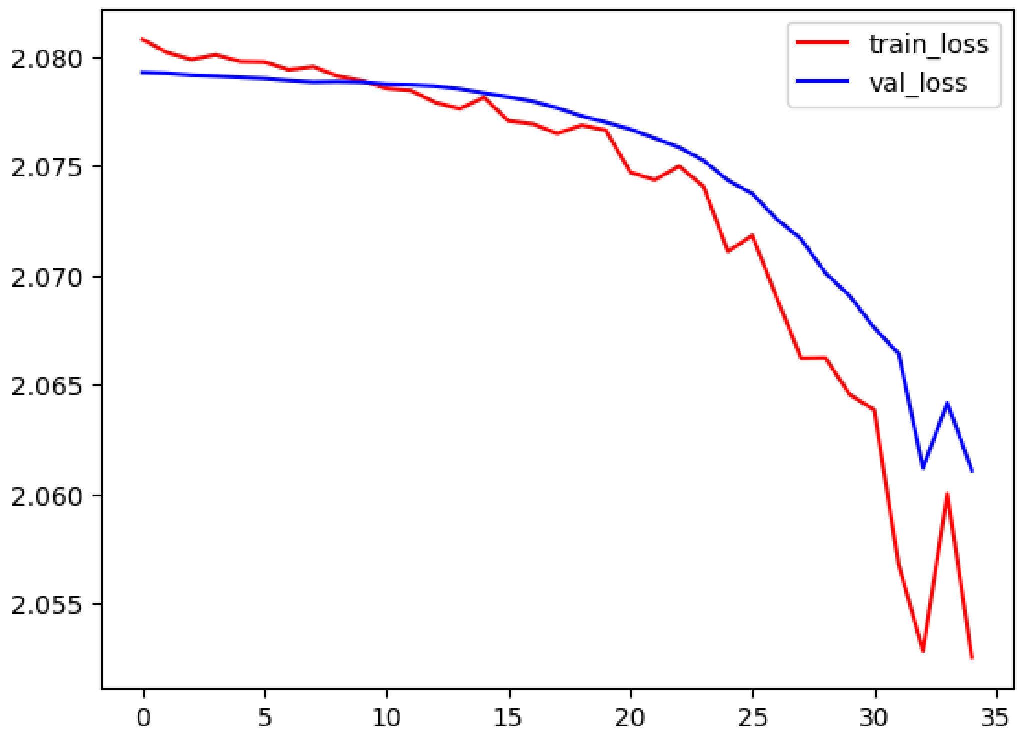

Figure 34: Loss Graphics

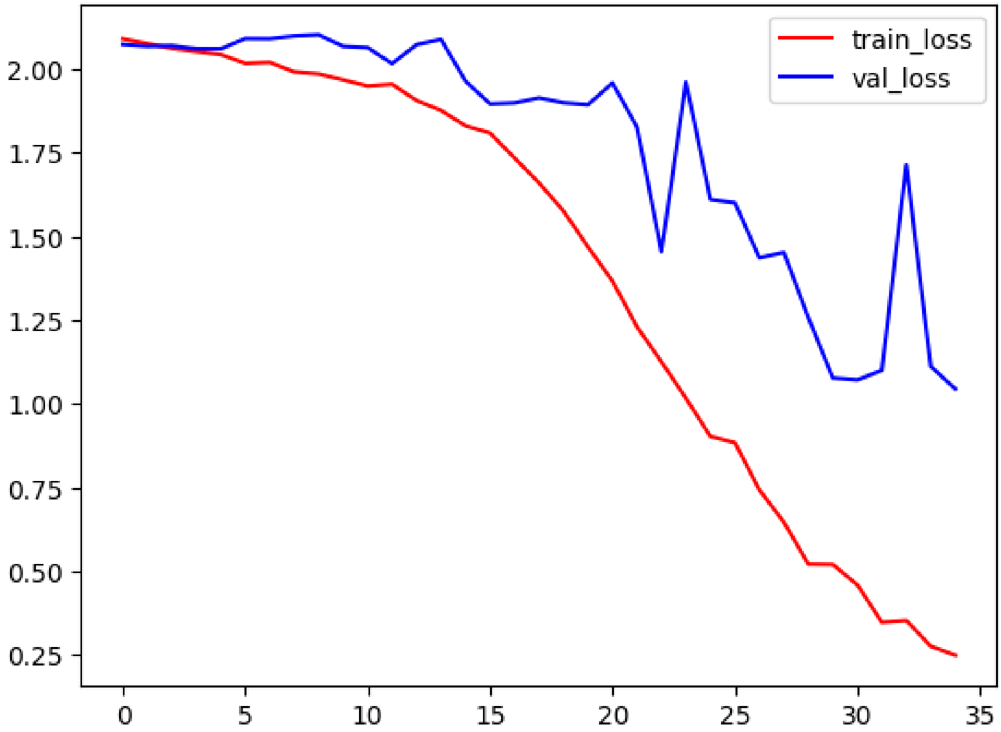

Figure 35: Loss Graphics

My comment:

Generate 2D gray images, I create a bash file so draw shapes automatically. I saved in dataset folder and the folder has subfolder corresponding to labels. There is a problem with size. When i tried to get images width and high respectively 128*128 px. The shapes is not center of the image. I solved problem with center cropped images. So i created image bigger size much more black areas. Applied center crop so i get much more contains shape. You can see implementation in createDataset function. The other function is adding noise. The function add randomly noise to pixel. Our problem is multiclass classification so i implement one hot decoder. Finished preprocessing steps i implemented Alexnet model.

I trained the first model with default parameters. In the training and validation loss graph, although there were points where they intersected, it was observed that they decreased at the same rate. However, at one point, while the training loss continued to decrease, the validation loss started to increase, indicating that the model was tending to overfit.

When setting the learning rate to 0.1 and training the model, it is clear from the loss graph that the model did not learn effectively. The AlexNet uses the ReLU function as the default parameter for activation function. In the next step, I replaced the ReLU function with the Swish function. Although it had a better effect in the previous problem, it failed to yield the same impact in our current problem. As evident from the model's graph, it is not able to learn effectively.

When I trained the model with the tanh activation function, the training loss was good, but it did not yield effective results in the validation. Overfitting was also observed in this case. When I tried reducing the density by 10% based on the steps we applied so far, it is evident from the losses that the model could not fit the problem effectively. Removing the last layers also resulted in the model underfitting.