Assignment 3

Alexander Falk

DD2424, May 2024

i) For evaluation of my analytical gradient computations I conducted experiments for a 2-layer, 3-layer and 4-layer neural network and calculated the relative error between my analytically computed gradients and numerically computed gradients using the centered difference method. Experiments were conducted on a small subset of the data and lambda = 0. In Table 1 the results are displayed for the 2-layer neural network with 50 hidden nodes. The differences are sufficiently small indicating that my gradient computations are correct.

Gradient	Difference
W1	7.145e-09
W2	5.068e-10
b1	3.846e-08
b2	3.865e-10

Table 1. Relative errors for weights and biases between analytic gradient computations and numerically computed gradient vectors using the centered difference method for a 2-layer neural network

ii) Below are the loss plots for my model without (Figure 2.1) and with (Figure 2.2) batch-normalization for a 3-layer network with 50 nodes in each hidden layer, initialized using He initialization. 49000 images were used for training and 1000 for validation. We can see that the loss was less when batch-normalization was used. The following hyper parameters were used:

- n_batch = 100
- eta_min = 1e-5
- eta_max = 1e-1
- lambda = 0.005
- n_cycles = 2
- n_s 5 * 45000 / n_batch

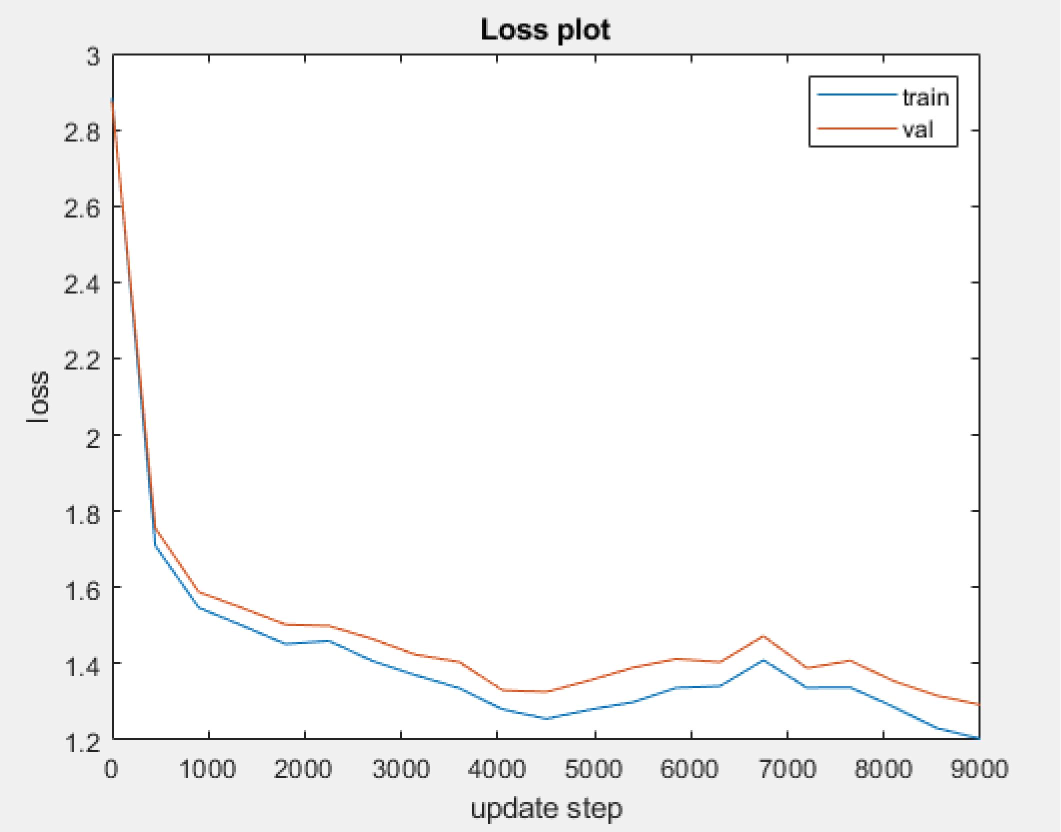

Figure 2.1. Loss plot for the 3-layer network without batch-normalization

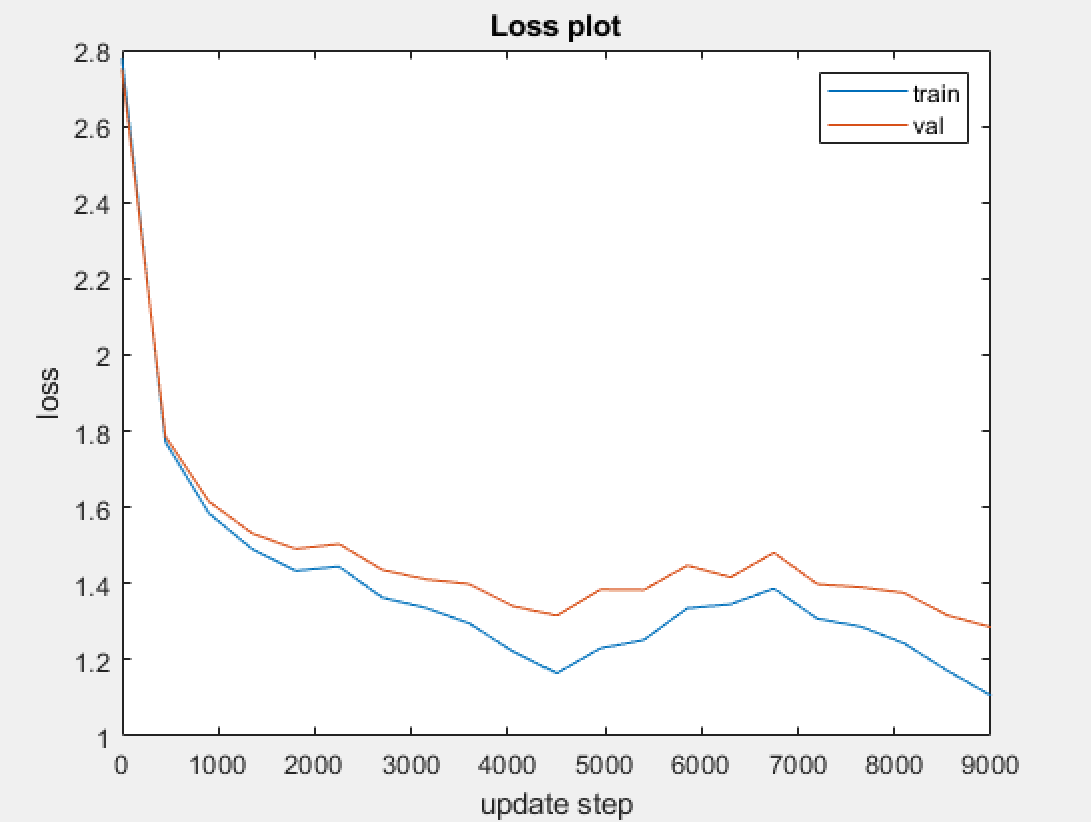

Figure 2.2. Loss plot for the 3-layer network with batch-normalization

iii) Below are the plots of the loss for a 9-layer neural network without (Figure 3.1) and with (Figure 3.2) batch-normalization. The number of nodes in each of the hidden layers were [50, 30, 20, 20, 10, 10, 10, 10]. 49000 images were used for training and 1000 for validation. Again, we can see that the loss decreased when using batch-normalization. The following hyper parameters were used:

- n_batch = 100
- eta_min = 1e-5
- eta_max = 1e-1
- lambda = 0.005
- n_cycles = 2
- n_s 5 * 45000 / n_batch

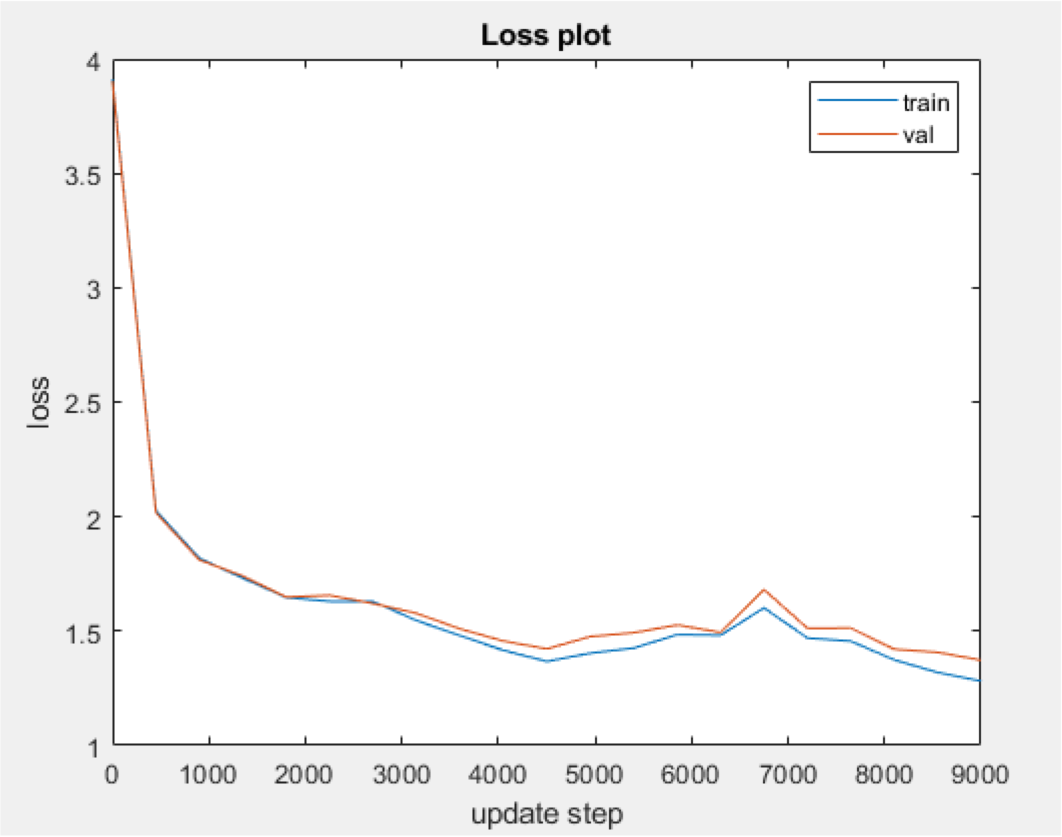

Figure 3.1. Loss plot for the 9-layer neural network without batch normalization

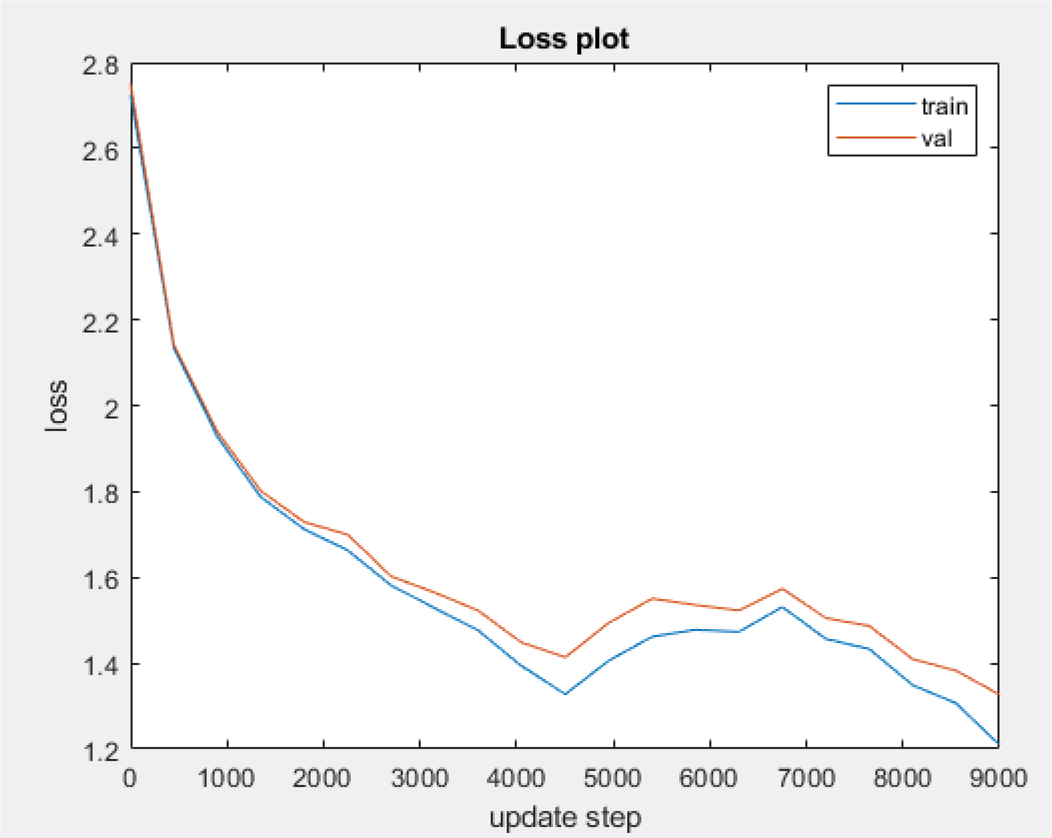

Figure 3.2. Loss plot for the 9-layer neural network with batch normalization

iv) When searching for an optimal lambda, I used l_min = -5 and l_max -1. 10 values of lambda was sampled and the 3 most accurate are displayed in Table 1. I used a 3-layer network with 50 hidden nodes in each hidden layer, batch-normalization was also used. Training was run for 2 cycles and used 49000 images.

Lambda	Test accuracy
0.0075	53.80 %
0.0145	53.58 %
0.0120	53.11 %

Table 1. Top 3 values of lambda found during the fine search. In descending order.

v) Below are the loss plots for training a 3-layer neural network with 50 hidden nodes in each hidden layer, with and without batch normalization. Instead of using He initialization as previously I initialized the weight parameters to be normally distributed with sigma = 1e-1 (Figure 5.1 & 5.2), sigma = 1e-3 (Figure 5.3 & 5.4) and sigma = 1e-4 (Figure 5.5 & 5.6). I used the aforementioned (basic) hyperparameters.

For sigma = 1e-1 the training and validation loss plots look okay even without batch normalization, however we can see that the loss was decreased when batch normalization was implemented. For the other values of sigma (1e-3 & 1e-4) we can see that the training became very unstable when no batch normalization was used. This shows that my experiments are consistent with the stated pros for batch normalization i.e. it makes training more stable. The instability shown in Figure 5.4 and 5.6 also shows that using a careful initialization of weights, such as He initialization, is important to make training more stable considering that the un-normalized batches from exercise 2 & 3 did not show this degree of instability.

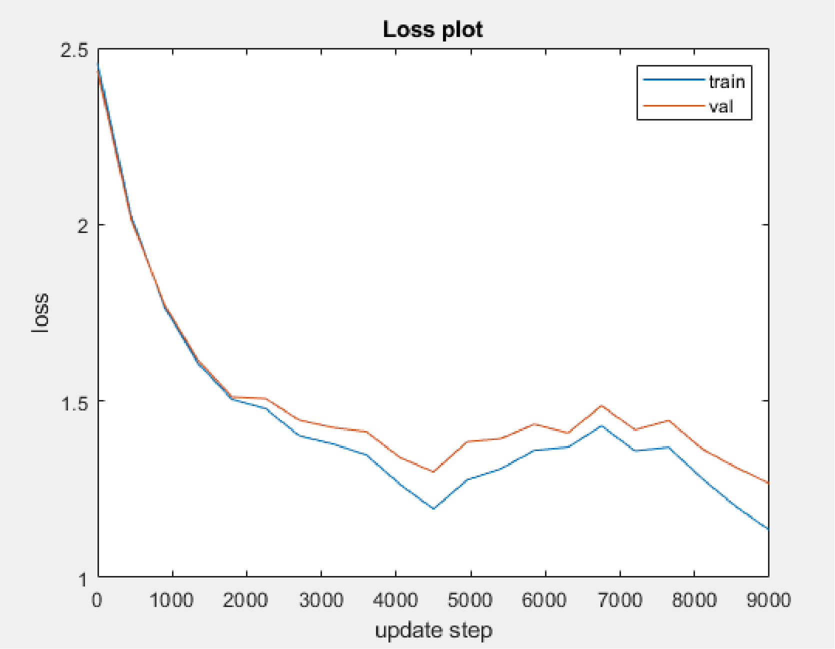

Figure 5.1. Training and validation loss plot for sigma = 1e-1 with batch normalization.

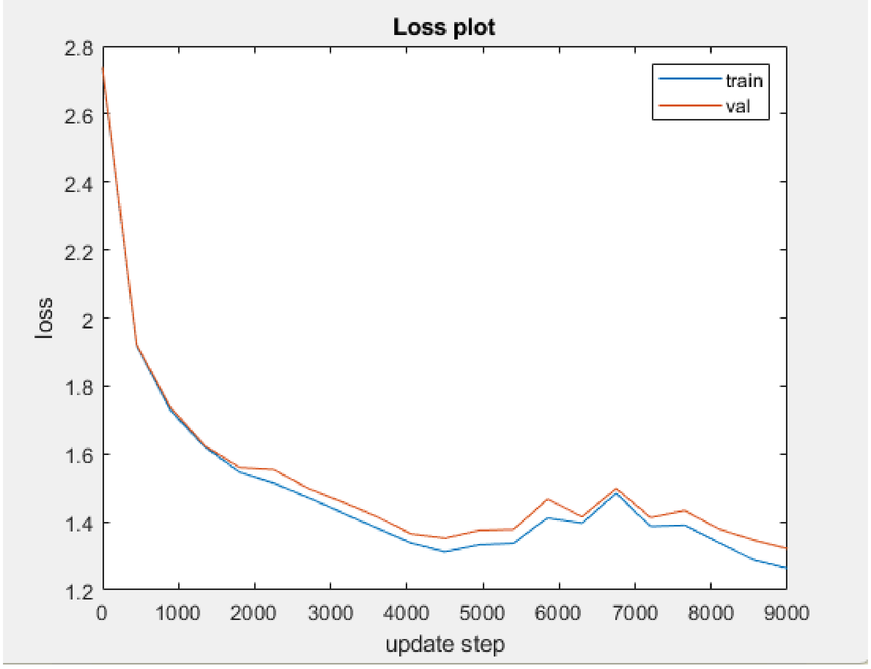

Figure 5.2. Training and validation loss plot for sigma = 1e-1 without batch normalization.

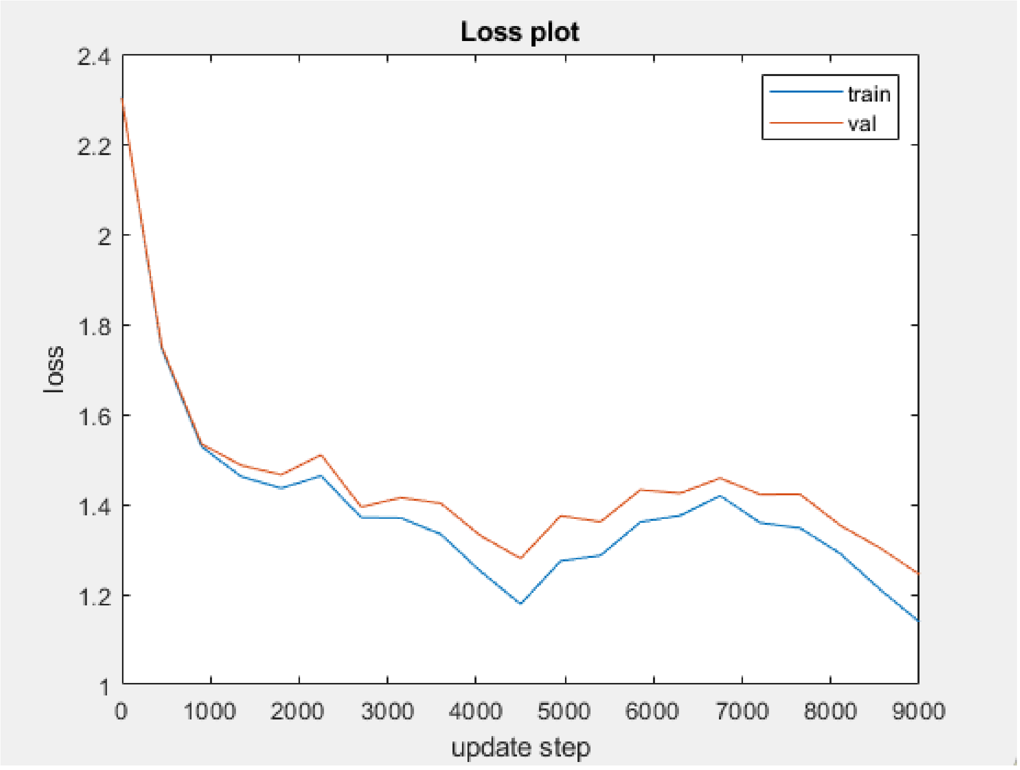

Figure 5.3. Training and validation loss plot for sigma = 1e-3 with batch normalization.

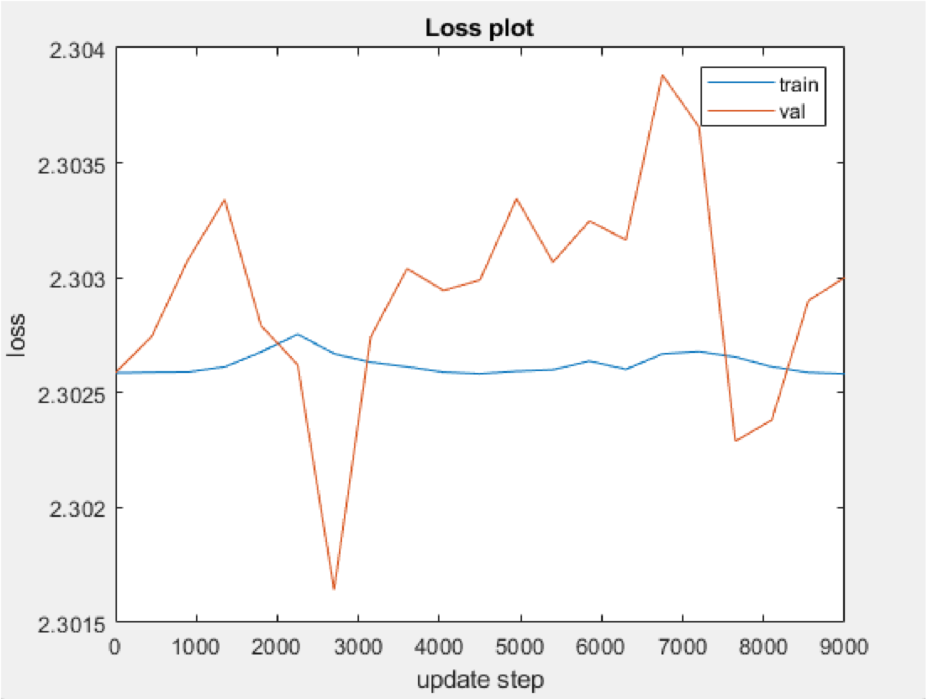

Figure 5.4. Training and validation loss plot for sigma = 1e-3 without batch normalization.

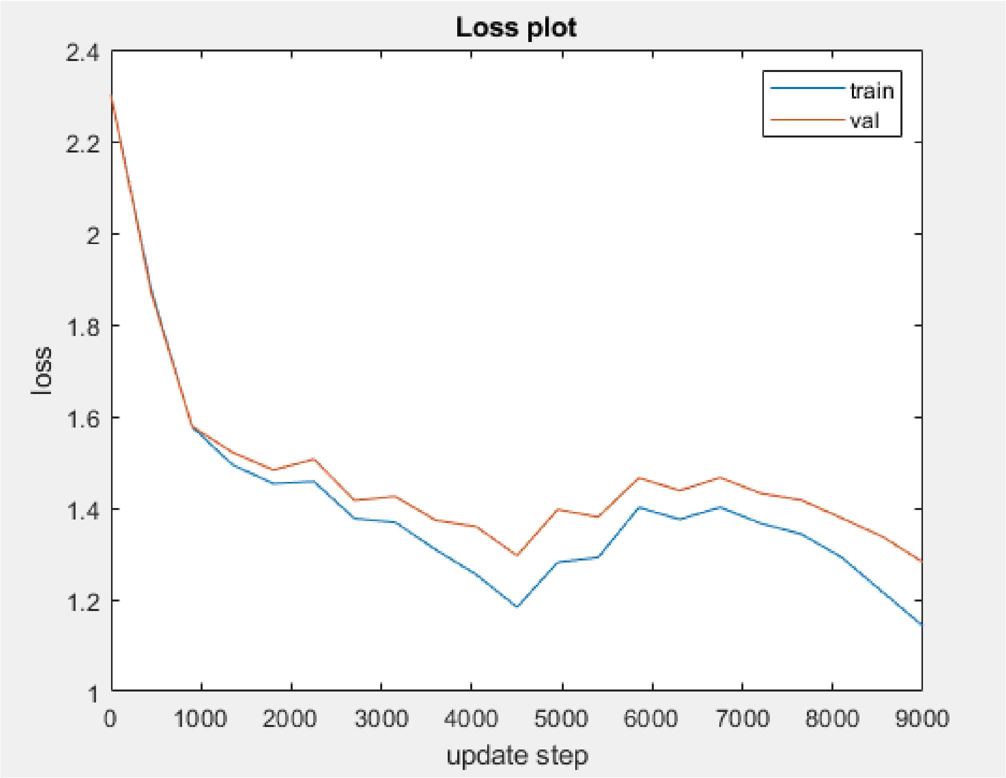

Figure 5.5. Training and validation loss plot for sigma = 1e-4 with batch normalization.

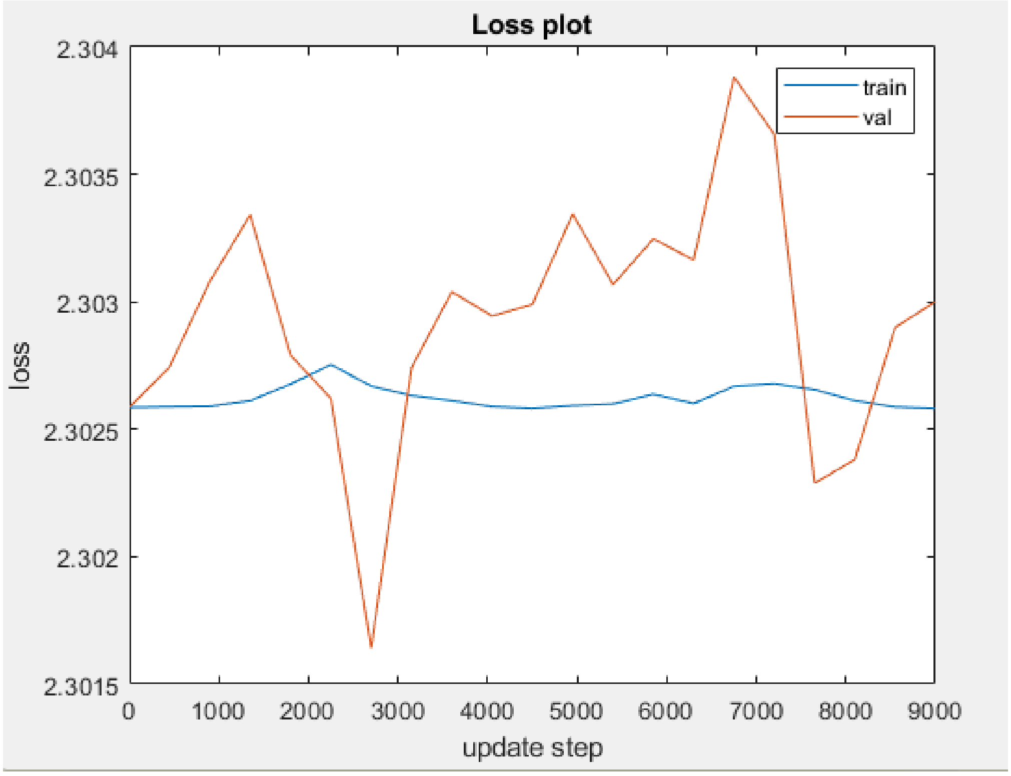

Figure 5.6. Training and validation loss plot for sigma = 1e-4 without batch normalization.