# *Assignment3*

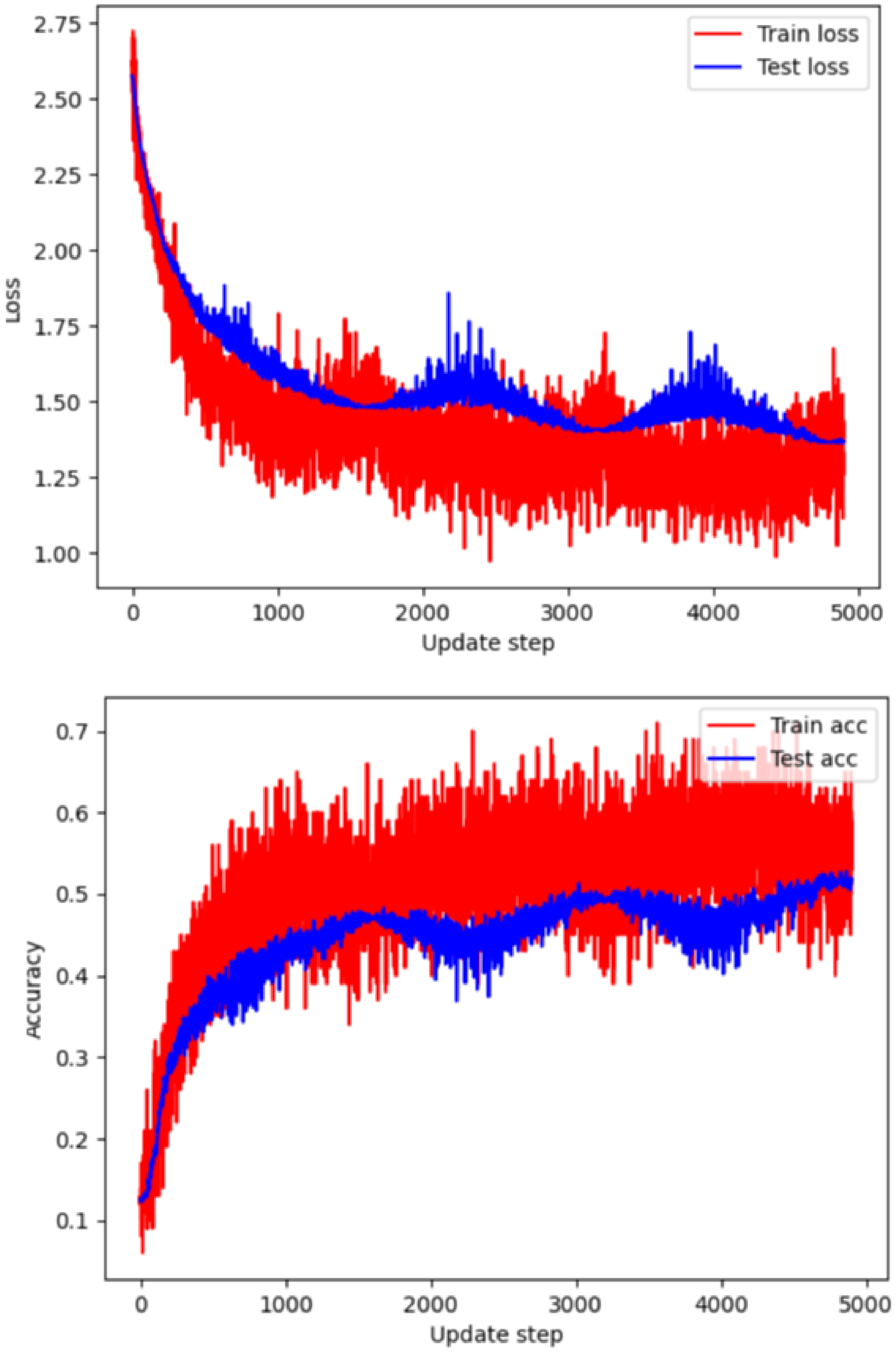
## 1. *The Gradient computation check:*

At this stage, I check several different values of my network, including P(forward pass result), W1, b1, W2, b2. The tablet shows the difference between my network result and pytorch result:

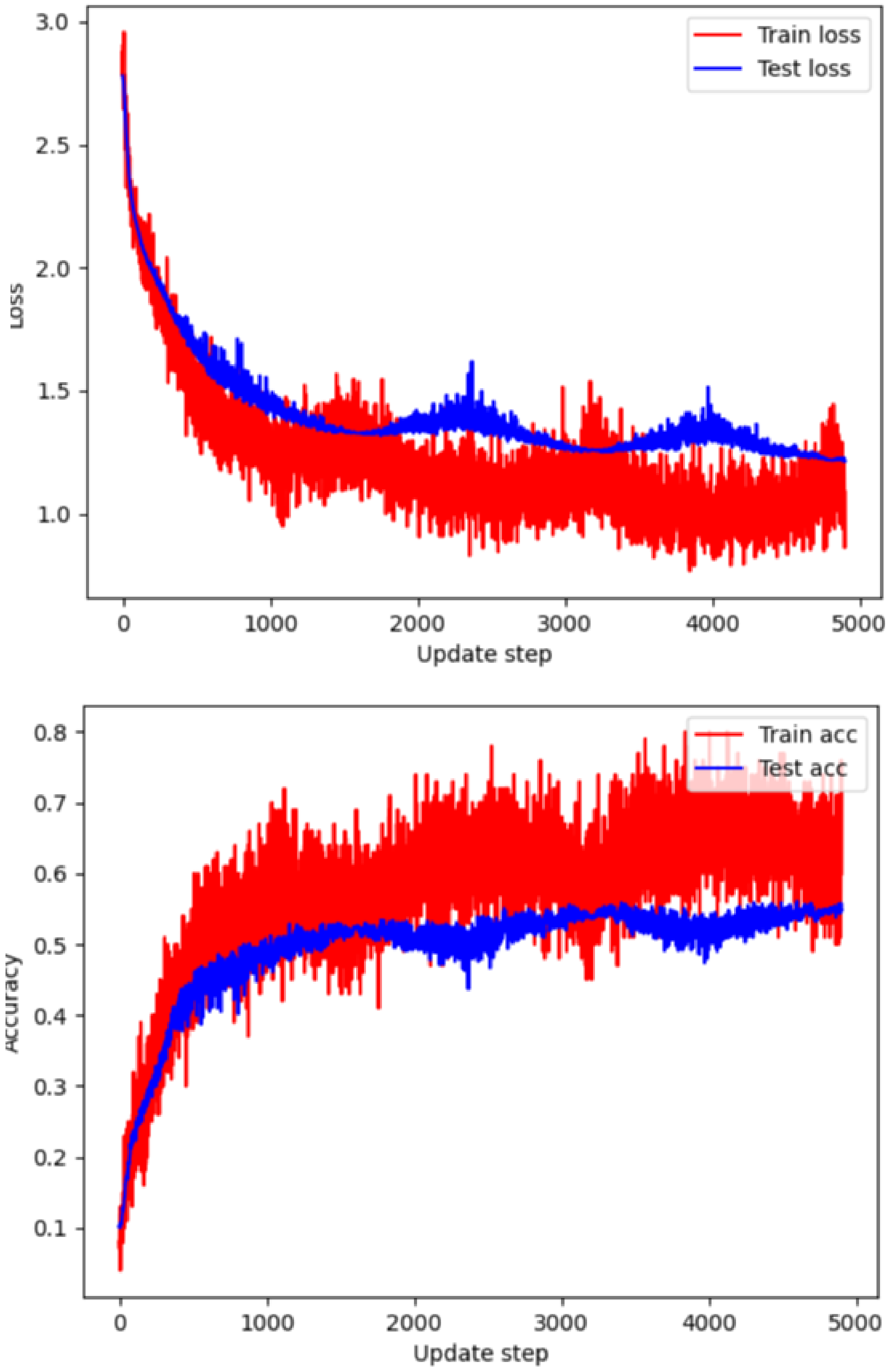
| | |
|---|---|
| W1 | 1.1102230246251565e-16 |
| W2 | 5.551115123125783e-17 |
| b1 | 3.469446951953614e-17 |
| b2 | 4.5102810375396984e-17 |
| Fs | 1.1102230246251565e-16 |
| P | 2.220446049250313e-16 |

## 2. *Different result for the different parameters:*

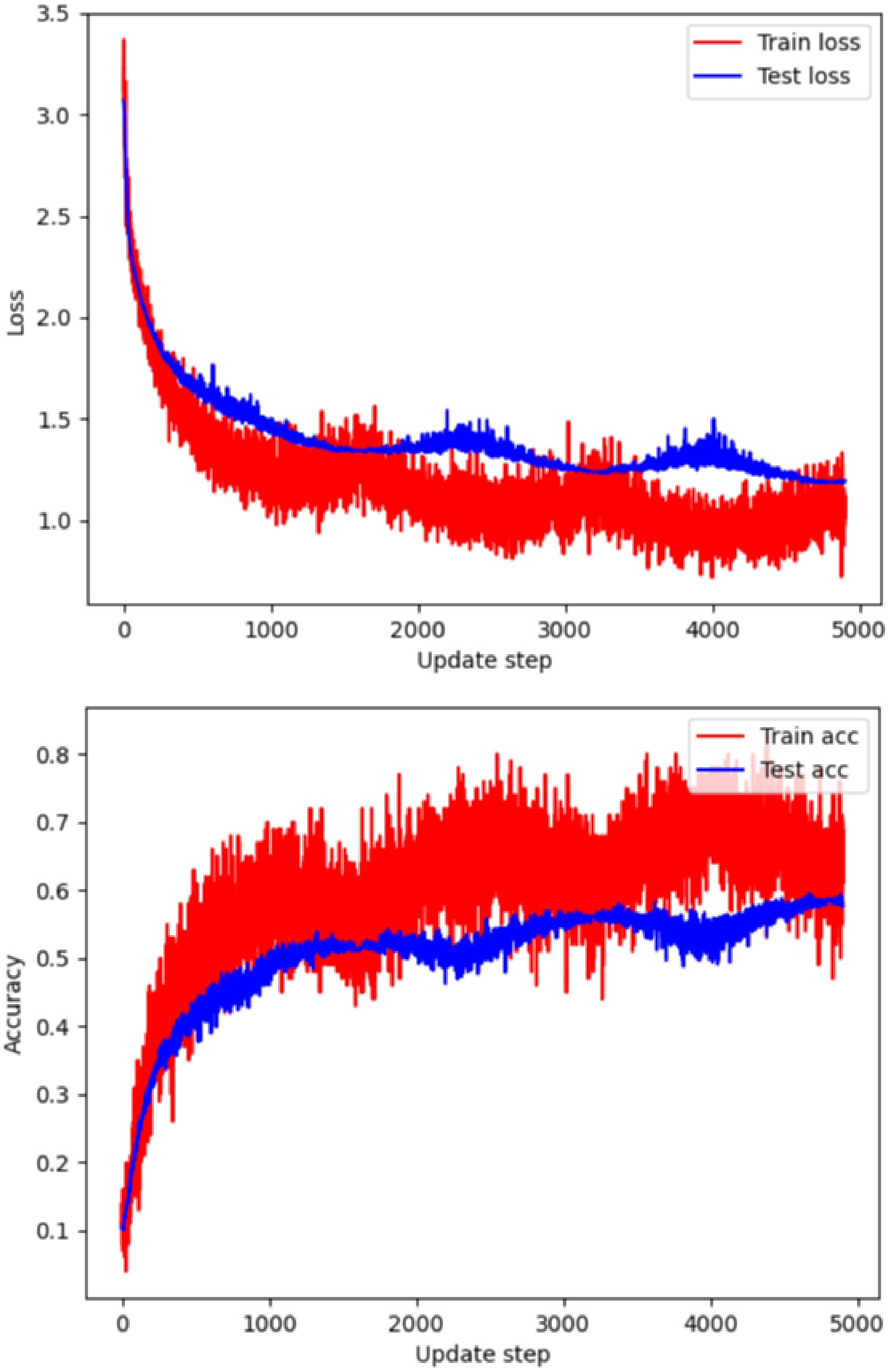
Architecture1:f=2, nf = 3, nh = 50 Accuracy = 50.04%



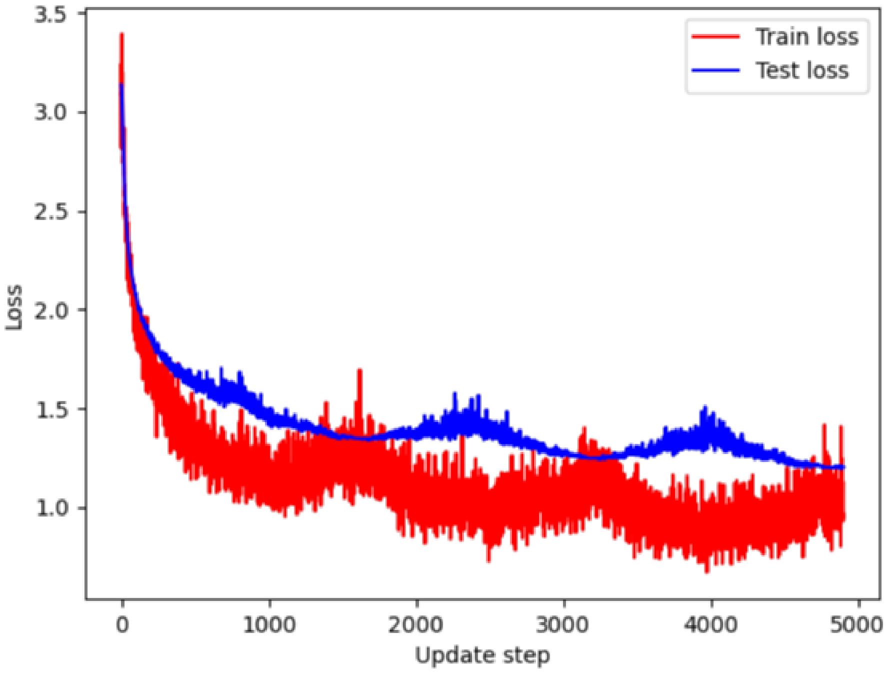
Architecture2: f=4, nf=10, nh=50, Accuracy = 55.87%

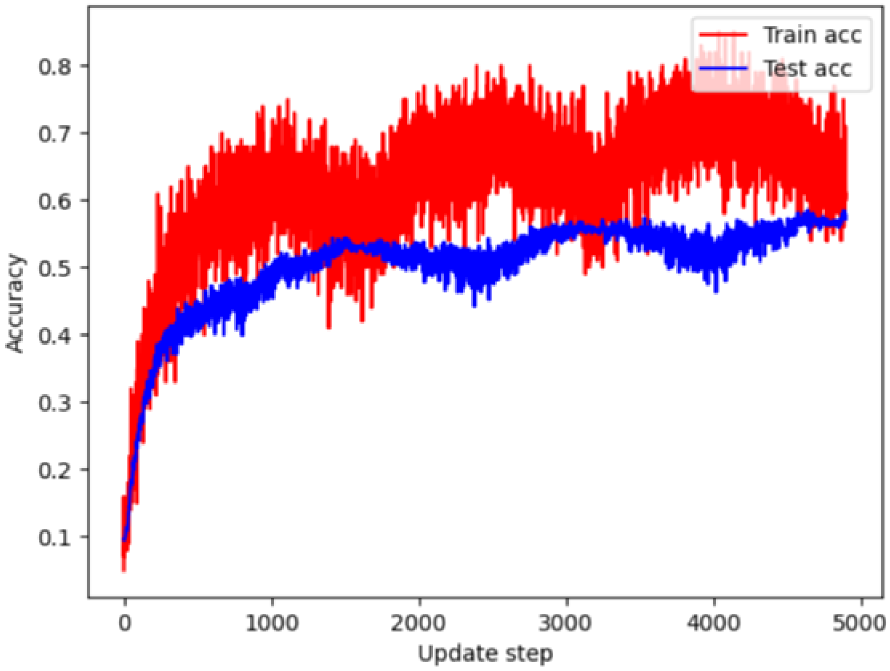


Architecture3: : f=8, nf=40, nh=50 Accuracy = 57.96%

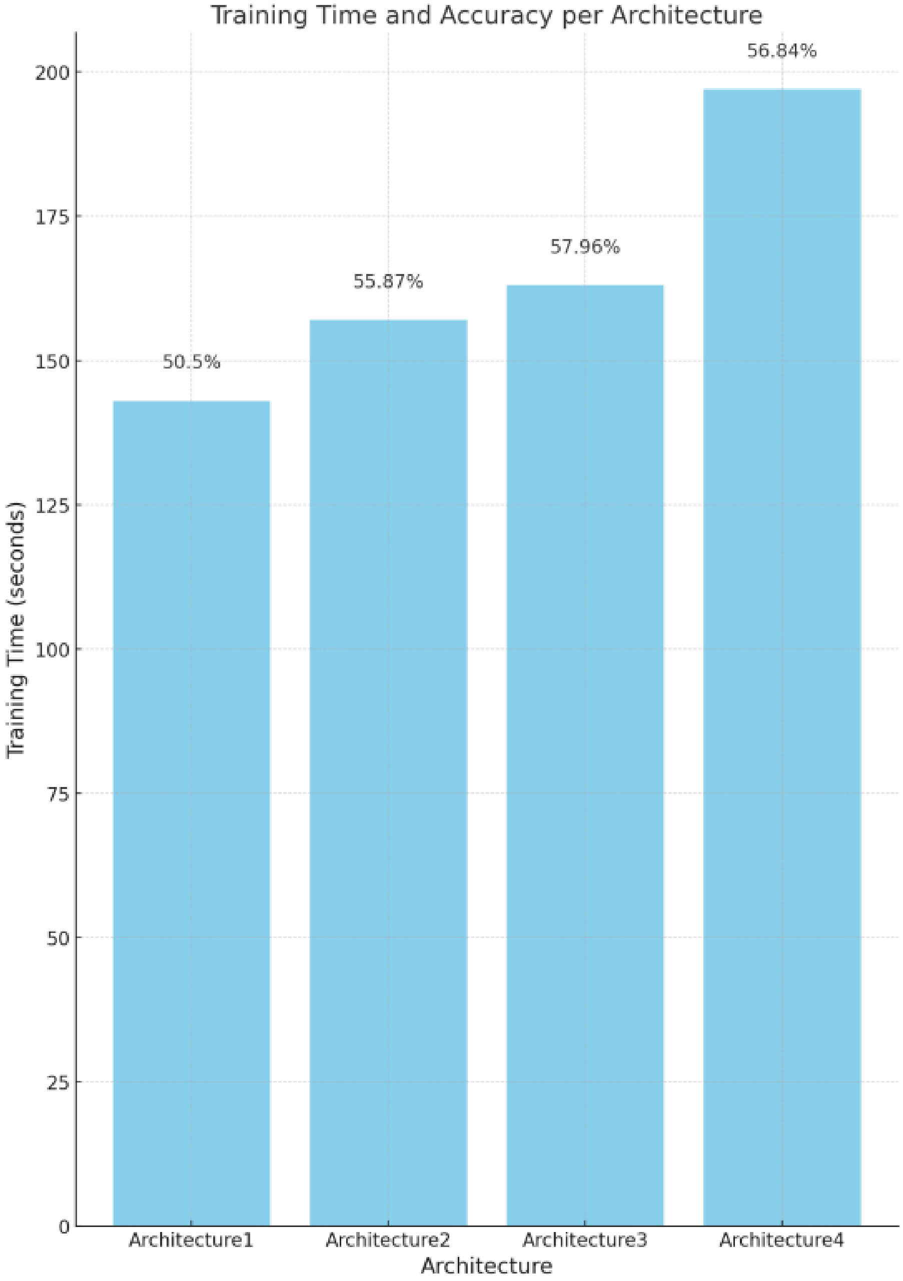


Architecture4: f=16, nf=160, nh=50 Accuracy = 56.84%



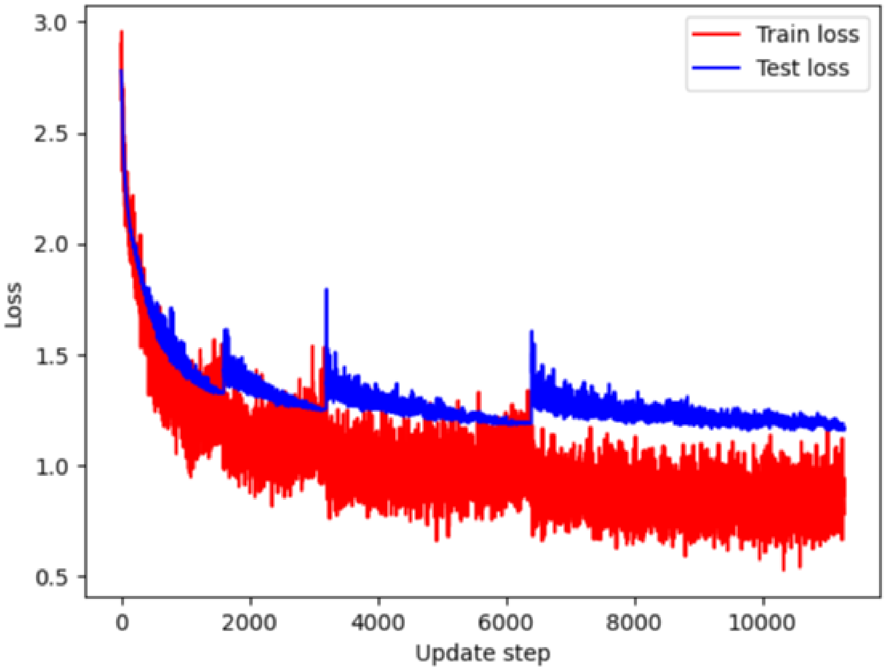


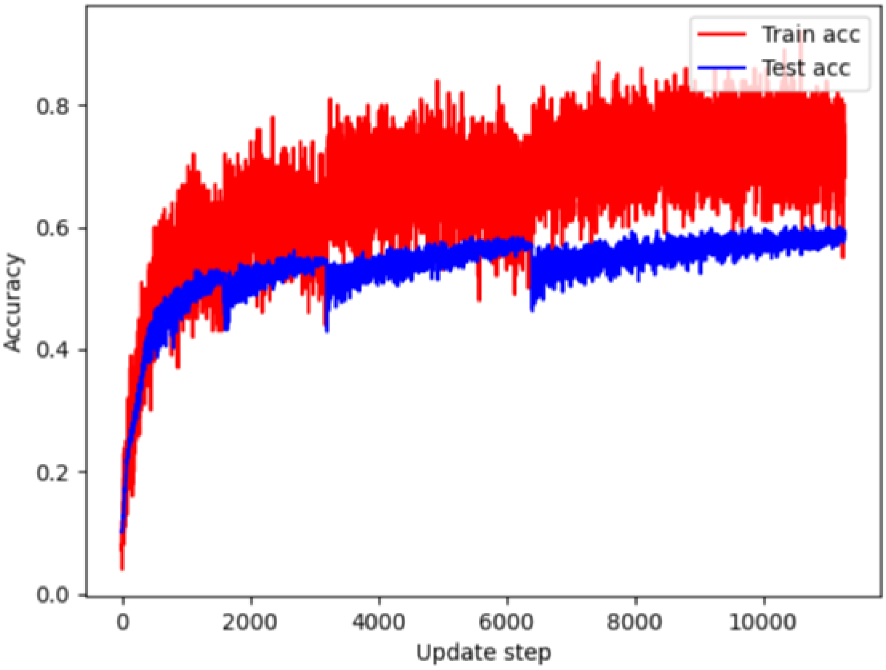
Bar chart from architecture 1 – 4:



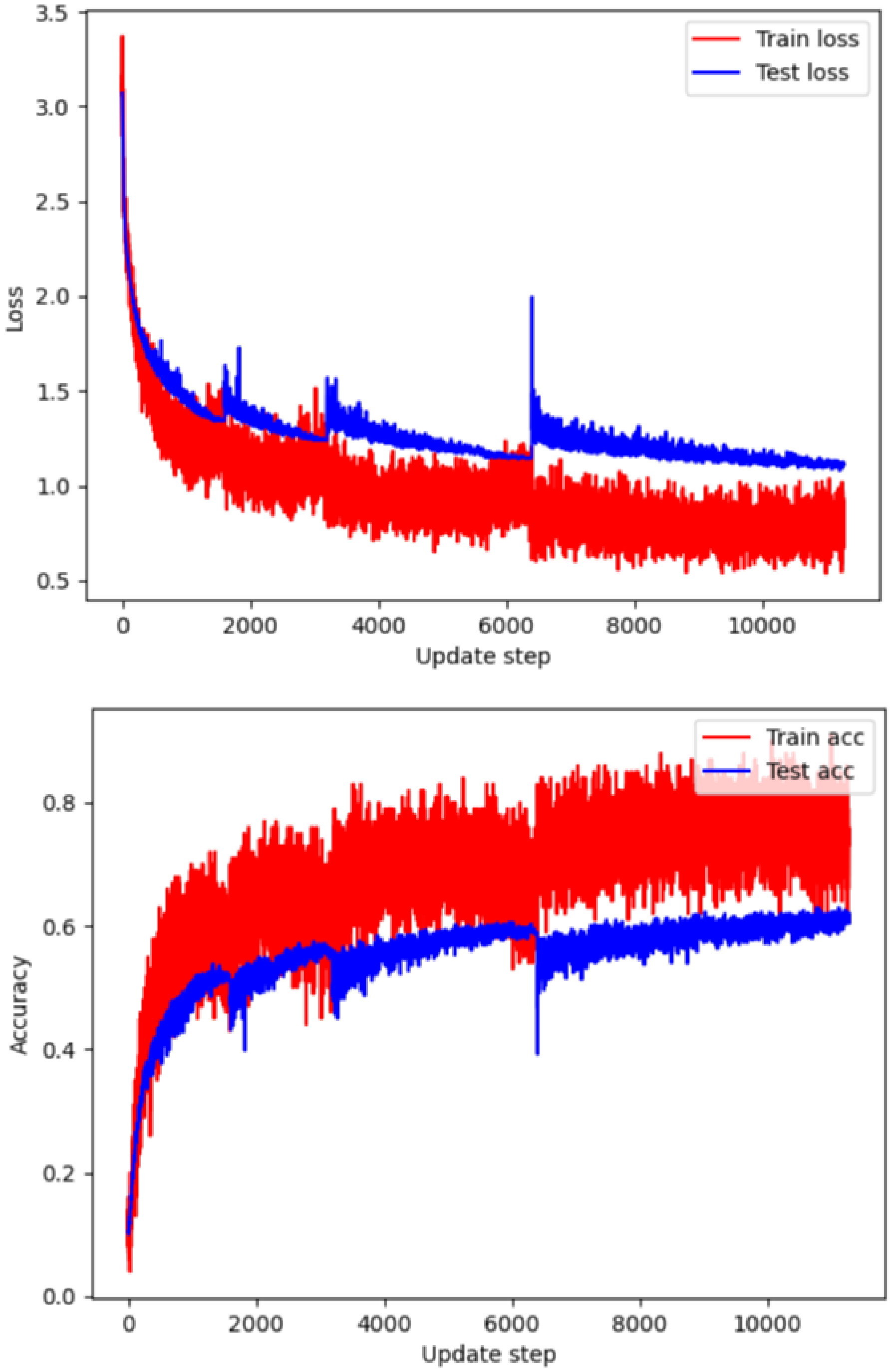
3. ***Train for longer:***

Architecture2, Accuracy = 57.18% (5 minutes 15 seconds)



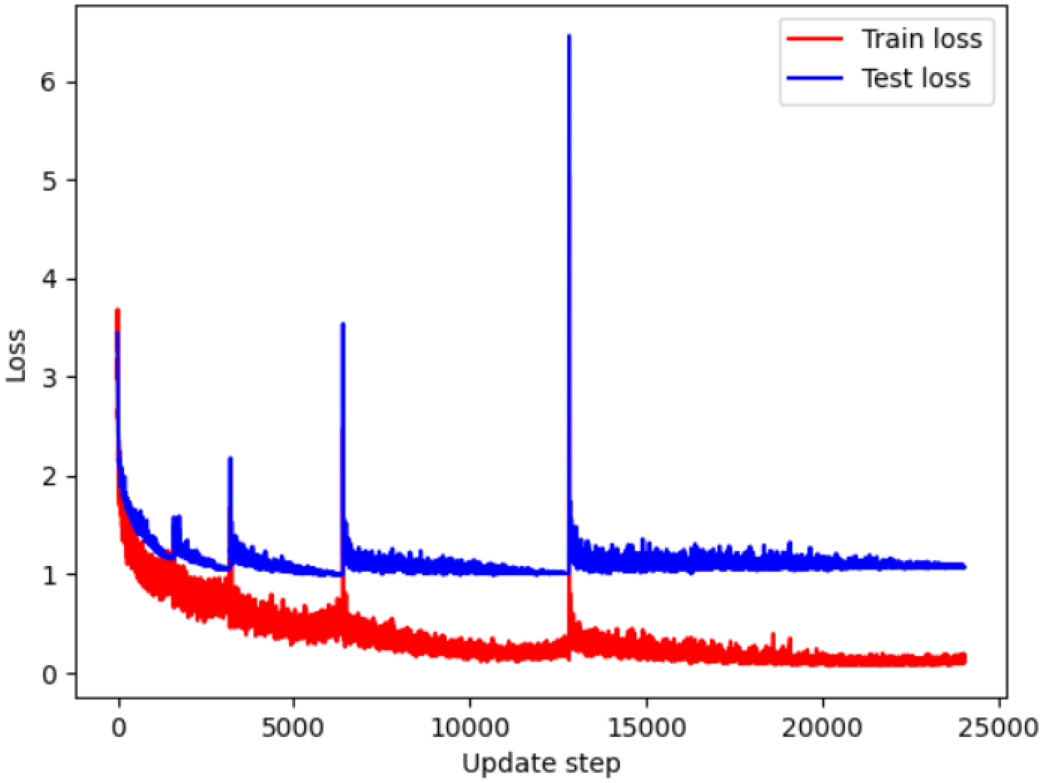


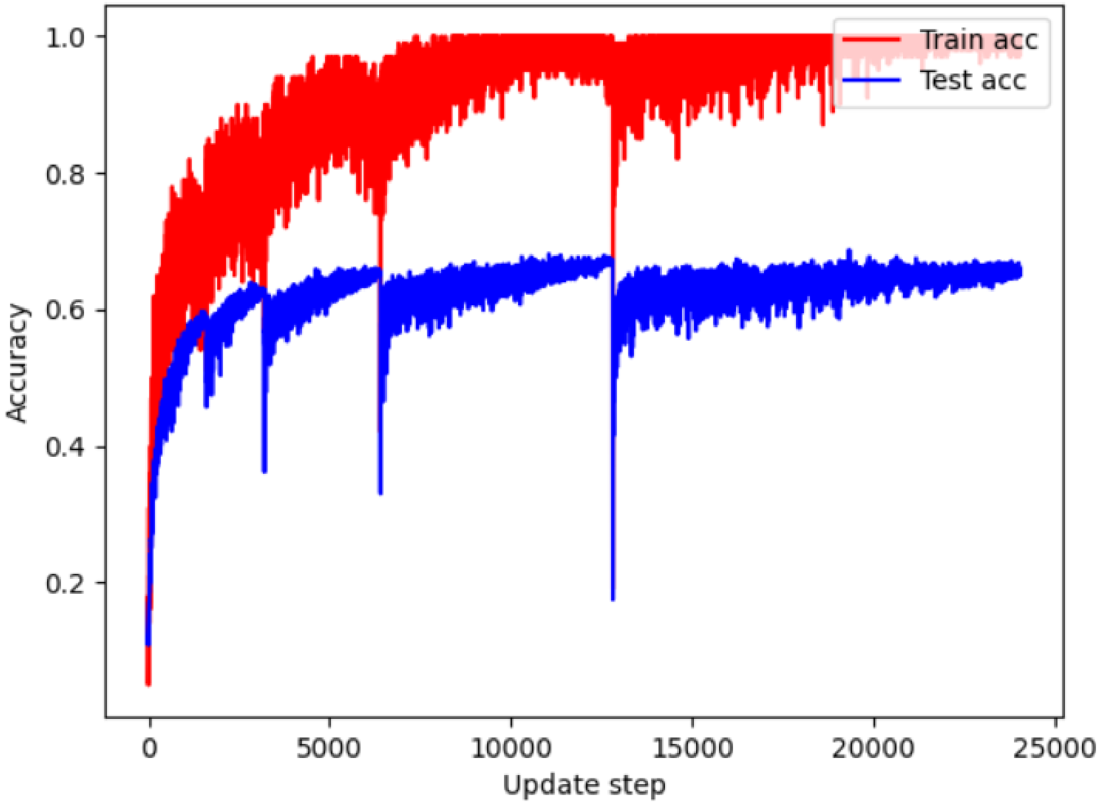
Architecture3: Accuracy = 61.26% (About 5 minutes 58 seconds)



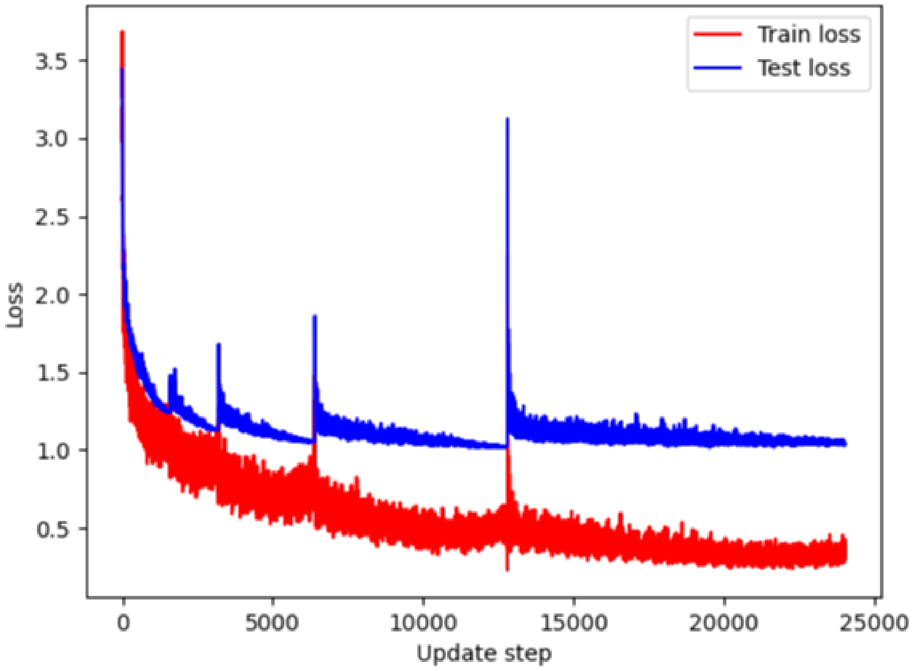
## 4. *Compare the L2 and smooth label with Architecture5*

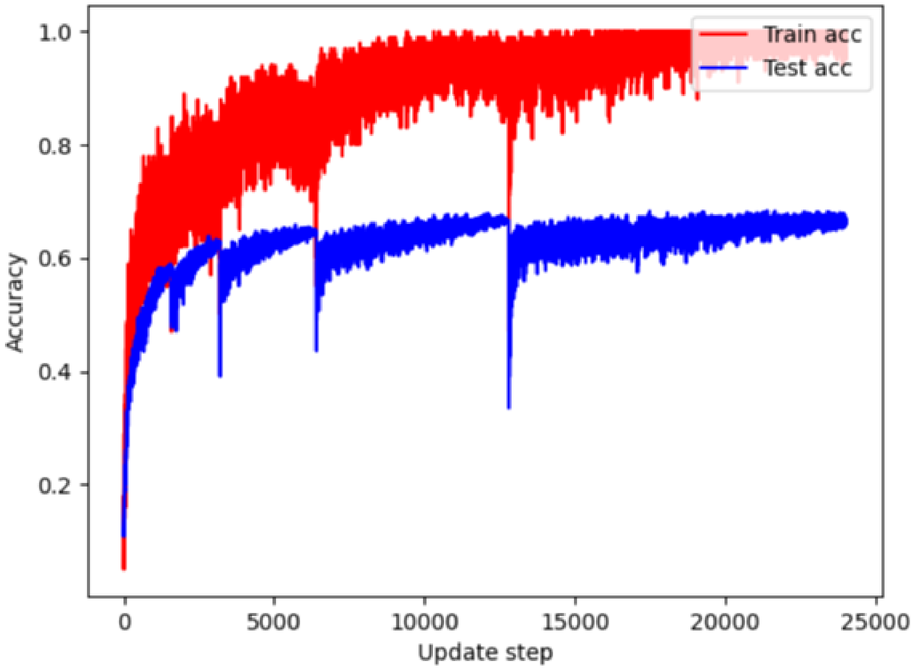
L2: f=4, nf=40, nh=300 Accuracy = 65.66%(10 minutes 40 seconds)





Smoothing label: Accuracy = 65.56% （11 minutes 32 seconds）





With Label Smoothing, the training process is smoother, and although the training error is slightly higher, it significantly slows down the overfitting phenomenon, resulting in a model that generalizes better on the test set.

## 5. *Implement ways to increase the performance*

1) Make the network wider: Widening the network (i.e., increasing the number of hidden units in each layer) allows the model to learn more complex and diverse feature representations. Compared to deepening the network, the increase in width is more stable and less prone to the problem of vanishing or exploding gradients.
2) Data-augmentation: Data augmentation (e.g., random flipping, cropping, panning, etc.) can create artificial data diversity while avoiding actually having to collect new data. It can significantly enhance the generalization ability of the model and reduce overfitting while keeping the training set closer to the test set distribution
3) Decay eta max: When using the Cyclical Learning Rate (CLR) strategy, the gradual decay of the maximum learning rate (eta_max) helps the model to better fine-tune the parameters in the later stages of training to avoid skipping the optimal point and thus achieve a better local minimum.