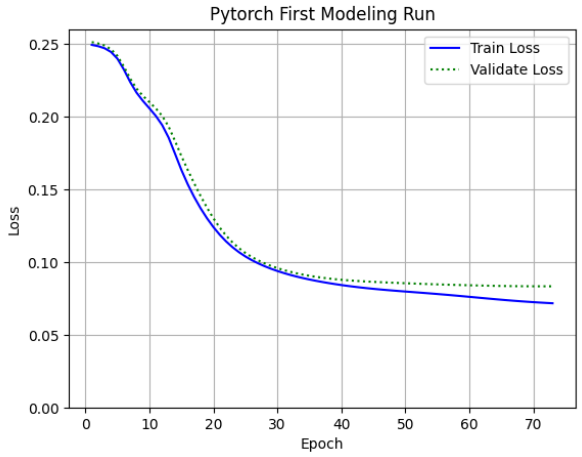
1 Point -- Your updated BlinkNeuralNetwork.py file.

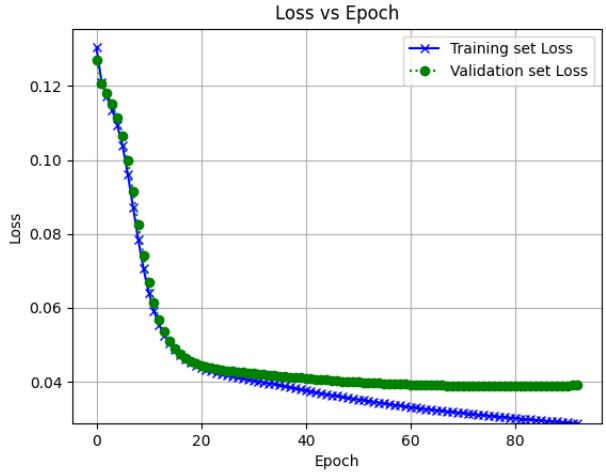
See attached file

3 Point -- Two charts and less than two hundred words describing your comparison. Did the models
converge similarly? Did they generalize similarly? What differences remain between the implementations?
How much does randomization affect the PyTorch run (vs your implementation where the framework set a seed)?

Pytorch run:



Handcrafted neural network:



In both the models pytorch and handcrafted, I achieved the same accuracy of 89.66% (50% confidence interval 88.69% - 90.62%) using the same hyperparameters step size (learning rate) = 0.01, convergence = 0.0001 and model structure = [7, 2]. The loss in the 2 models however is different with minimum loss of 0.04 in the handcrafted model

and approximately 0.08 in the pytorch model. The models converge similarly and do not show any signs of overfitting until convergence. I found two differences between pytorch and the handcrafted model :

1. We set the randomization seed manually in the handcrafted model, and therefore we got consistent result run-after run. Whereas in the pytorch model we do not set a seed and therefore there is a variance between the runs. I used a pytorch method torch.manual_seed() to test my framework first.
2. In the handcrafted model, the normalization was performed across the entire dataset (validation, training and testing separately), whereas in the pytorch model, the individual images are normalized.