Overview

This project uses Toronto Faces Dataset which contains over 4,000 grayscale images of faces. Each image is labelled as one of the following 7 categories based on its facial expression: 1-Anger, 2-Disgust, 3-Fear, 4-Happy, 5-Sad, 6-Surprise, 7-Neural. I am going to train a neural network to predict the facial expressions based on face images and test the model accuracy using Python Numpy & Pandas.

About the dataset

Each sample from Toronto Faces Dataset contains a grayscale image of size 48 x 48 representing a face photo and its corresponding expression from the following 7 categories: 1-Anger, 2-Disgust, 3-Fear, 4-Happy, 5-Sad, 6-Surprise, 7-Neural. The following graph shows one example image for each category:

Steps:

1. Partition the datasets into three parts: training, validation and testing.
2. Implement the neural network algorithm and train model using the training set.
3. Experiment with different values of hyper-parameters (numbers of hidden units, learning rate and mini-batch sizes). Use the validation set to evaluate the model performances and choose the best model.
4. Compute the prediction accuracy of the final model using testing set. Plot examples of predictions.

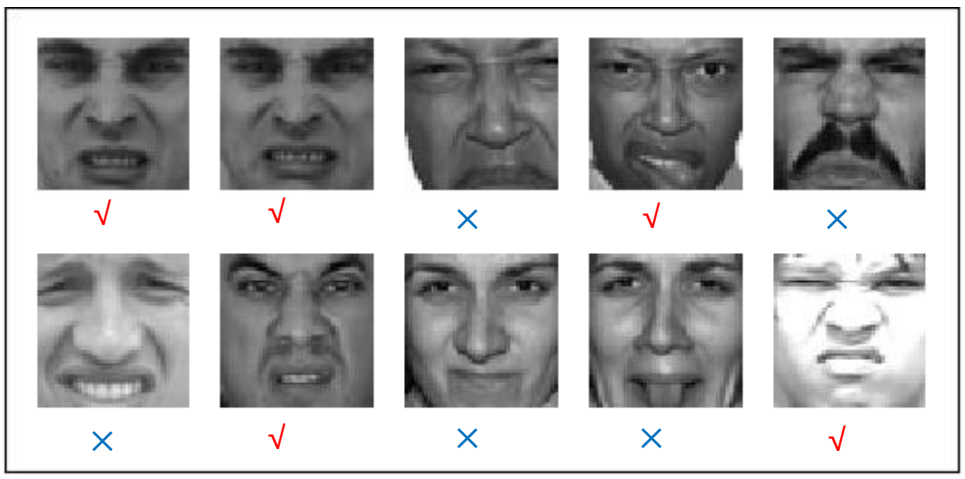
Results:

This part discusses results from the final model only. Please see appendix for model selection. The final model contains 2 hidden layers, the first layer contains 16 hidden units and 32 for the second layer. The learning rate is 0.01 and the mini-batch size is 100.

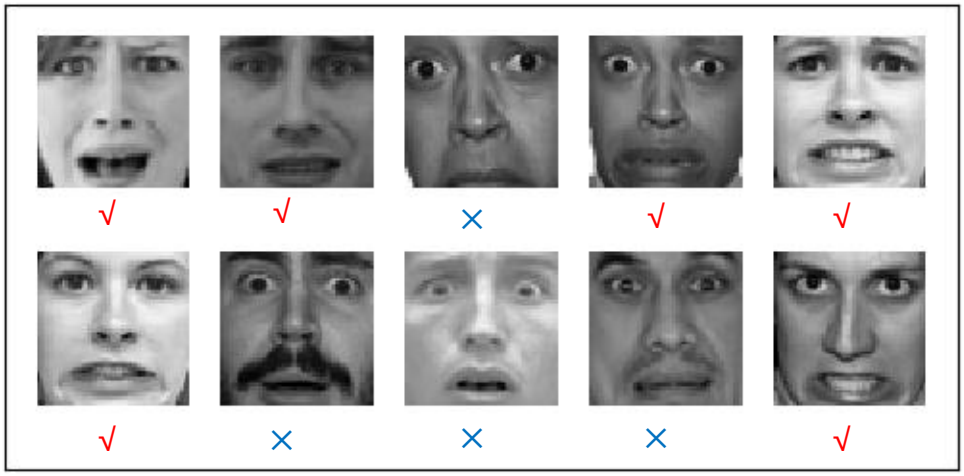
Here are some examples: (√ means an image is correctly classified and × means an image is misclassified)

1- Anger:

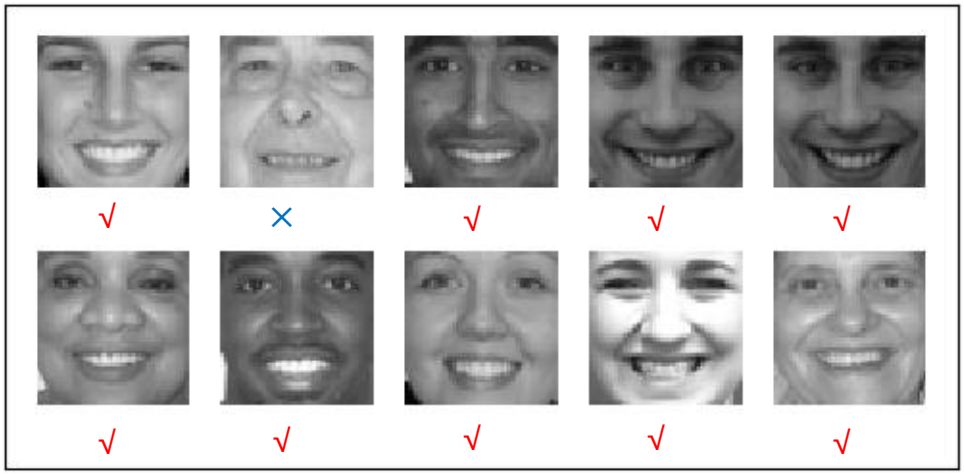
2- Disgust:



3- Fear:

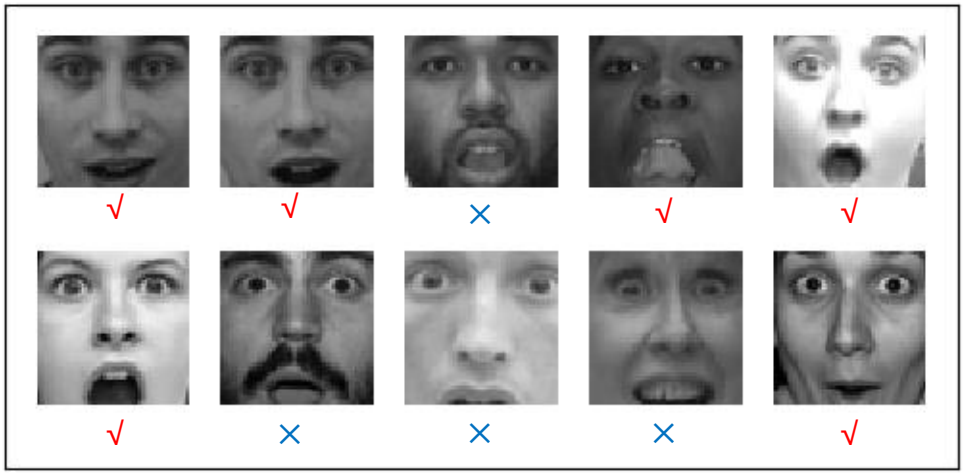


4- Happy:

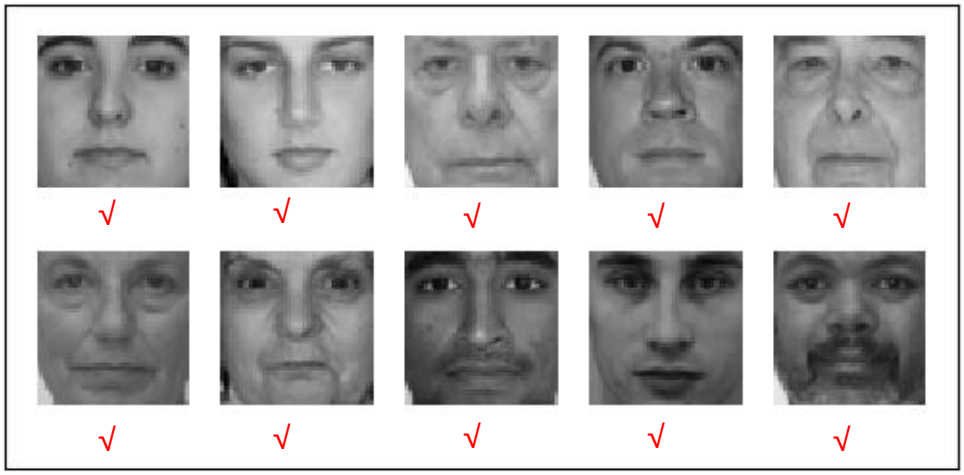


5- Sad:

6- Surprise:



7- Neural:



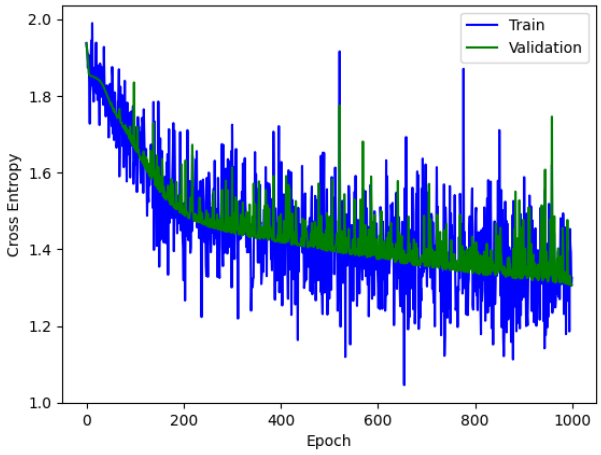
The final model accuracy is 72.7% on testing set.

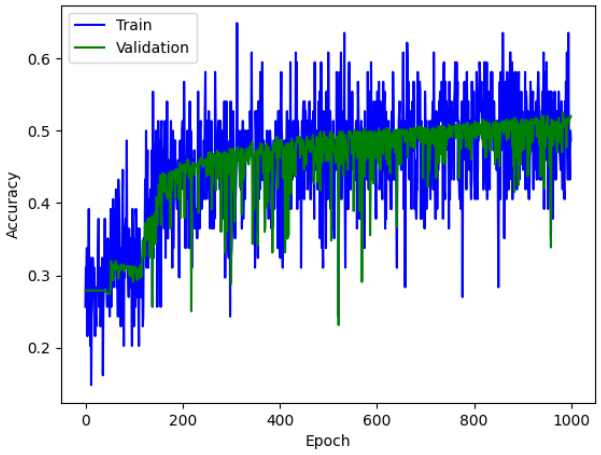
Appendix – model selection:

For each experimental values of hyper-parameters, I fit the corresponding model using the training set and measure the model performance using validation set. The maximum number of iterations is set to 1000. During the 1000 iterations, the cross entropy and accuracy on both sets are recorded and plotted as blow.

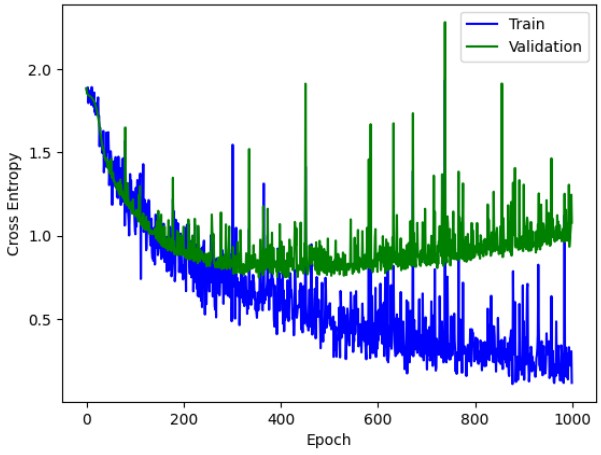
Hidden units:

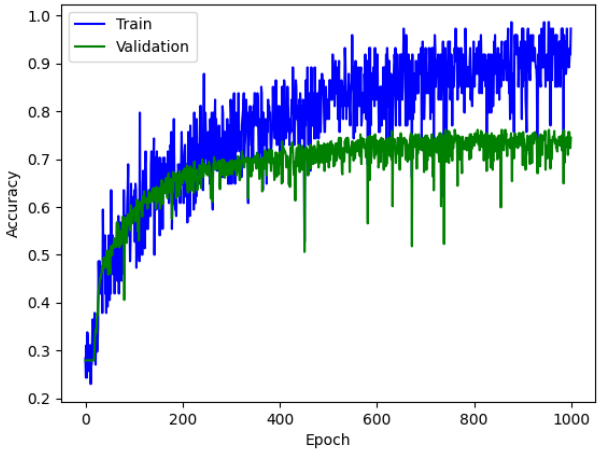
- Hidden units (4, 32), final validation accuracy 52.0%



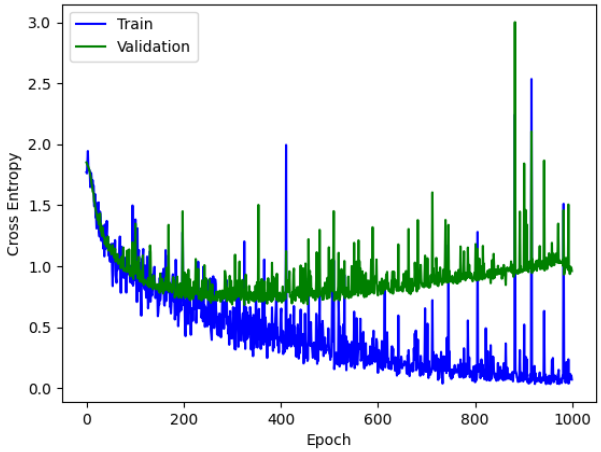


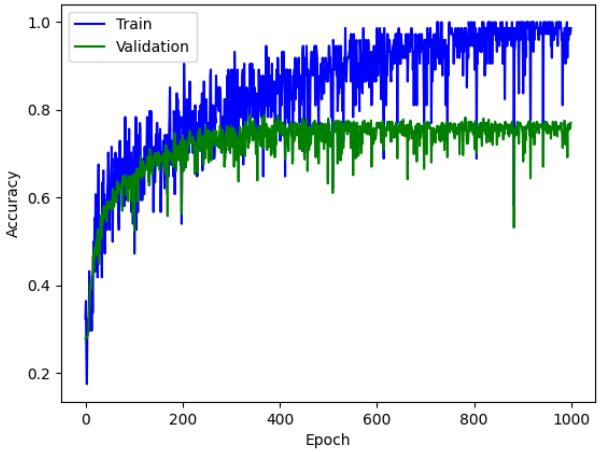
- Hidden units (16, 32), final validation accuracy 74.5%





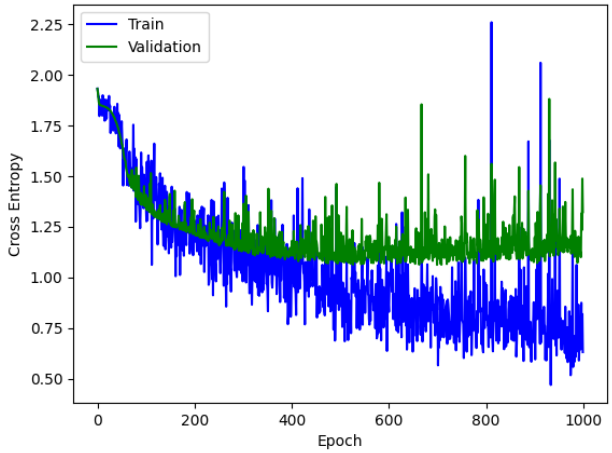
- Hidden units (64, 32), final validation accuracy 72.3%

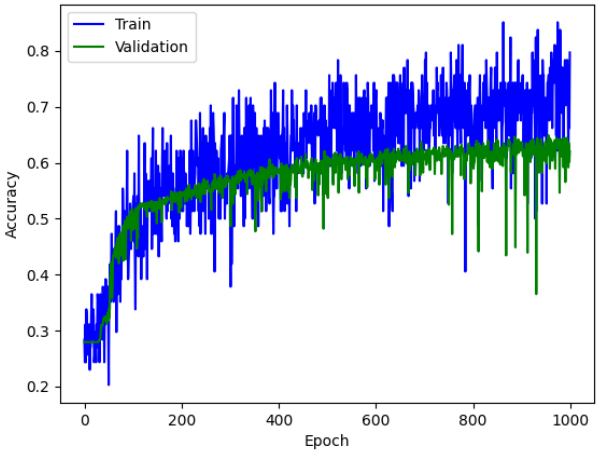




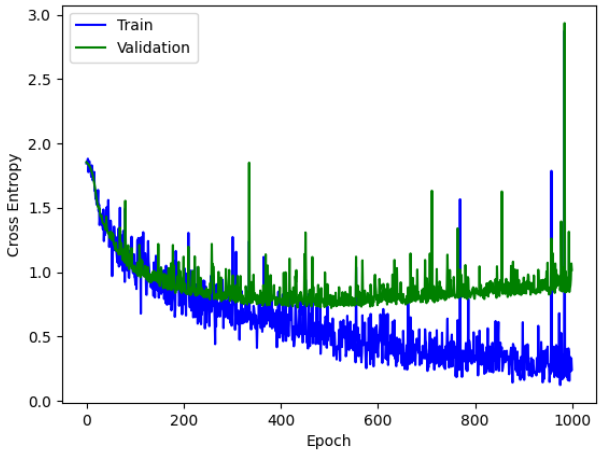
Here I chose 16 for the number of hidden units for the first layer.

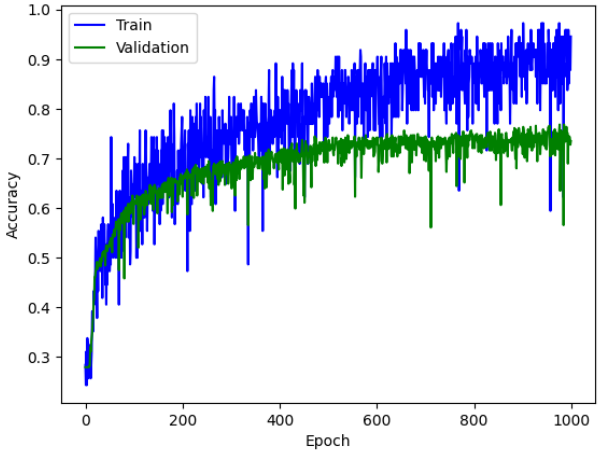
- Hidden units (16, 8), final validation accuracy 62.1%





- Hidden units (16, 128), final validation accuracy 70.4%

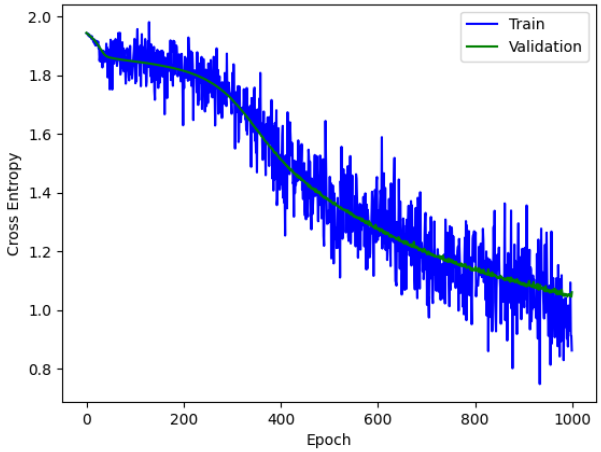


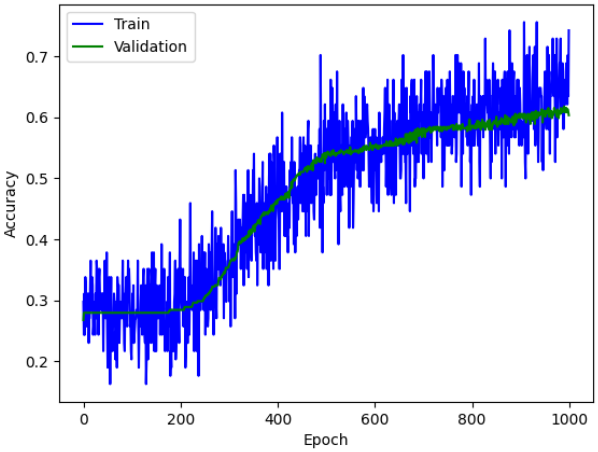


Hence the number of hidden units for the second layer was chosen to 32.

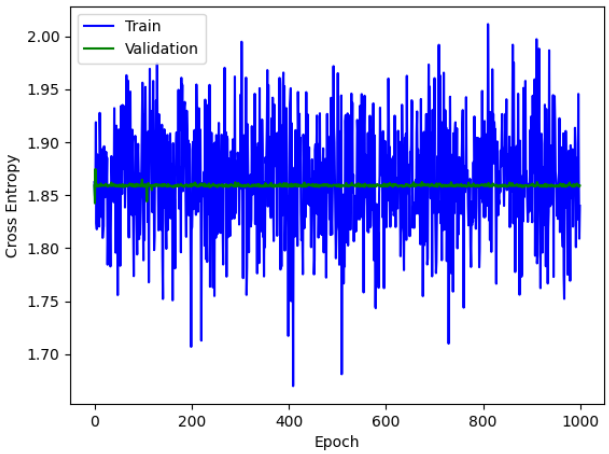
Learning rate:

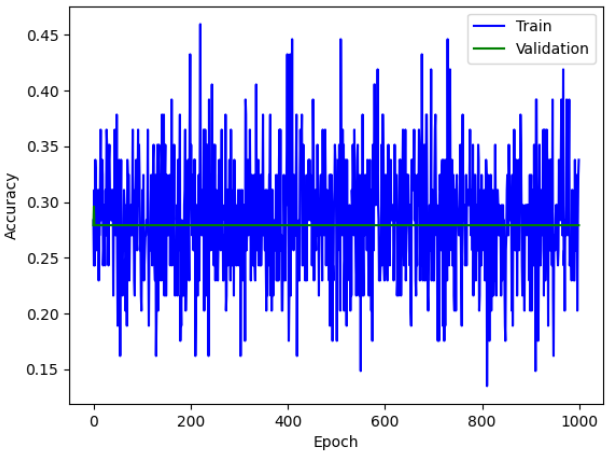
- Learning rate 0.001, final validation accuracy 60.4%





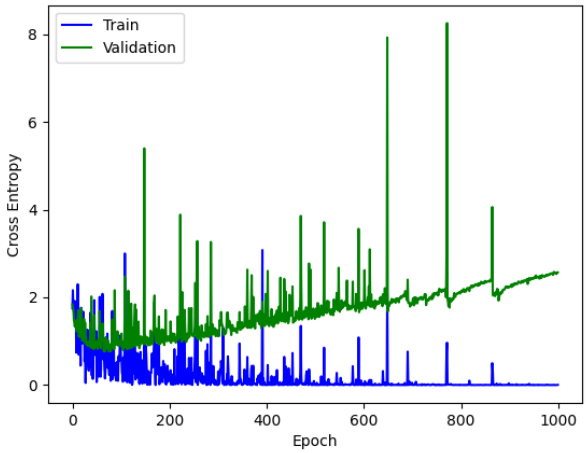
- Learning rate 0.5, final validation accuracy 27.9%.

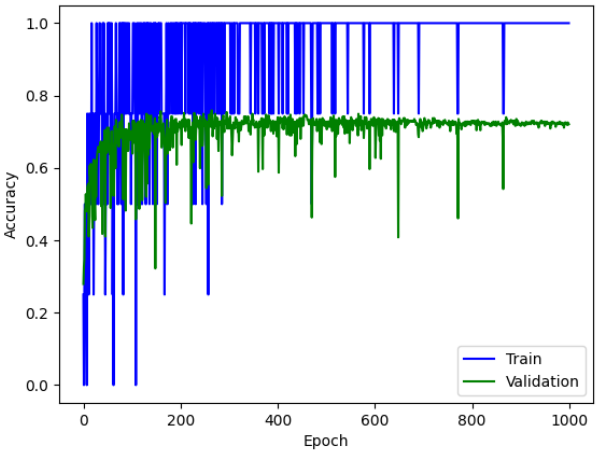




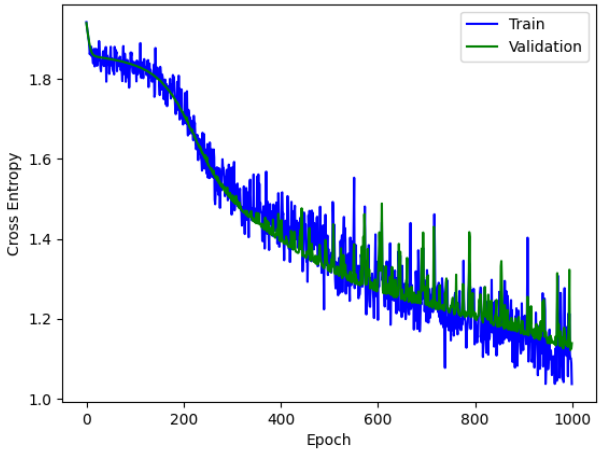
Mini-batch size:

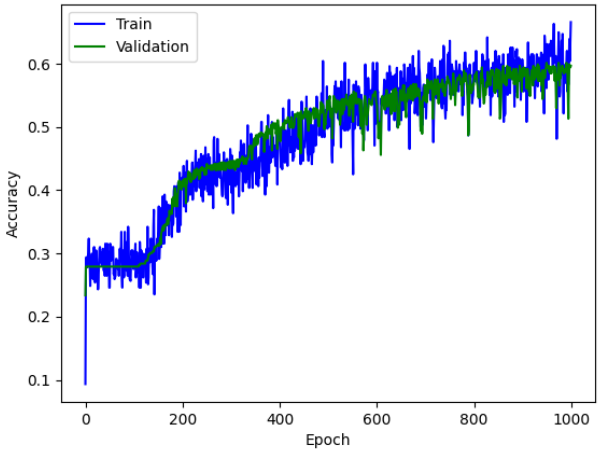
- Mini-batch size 10, final validation accuracy 72.1%.





- Mini-batch size 1000, final validation accuracy 59.67%


Train
Validation
Cross Entropy
Epoch
1.0
1.2
1.4
1.6
1.8
0
200
400
600
800
1000


Train
Validation
Accuracy
Epoch
0.1
0.2
0.3
0.4
0.5
0.6
0
200
400
600
800
1000