Homework 3 Question 2

Yanlin Li, Student #: 1003770305

March 19, 2021

2.1

The full code can be found at the end of this document

2.2

Performance:
1. Cross-Entropy: Train 0.38604, Validation 1.07142, Test 0.91233
2. Accuracy: Train 0.86396, Validation 0.71599, Test 0.71688
Here are the plots of training error and validation error:

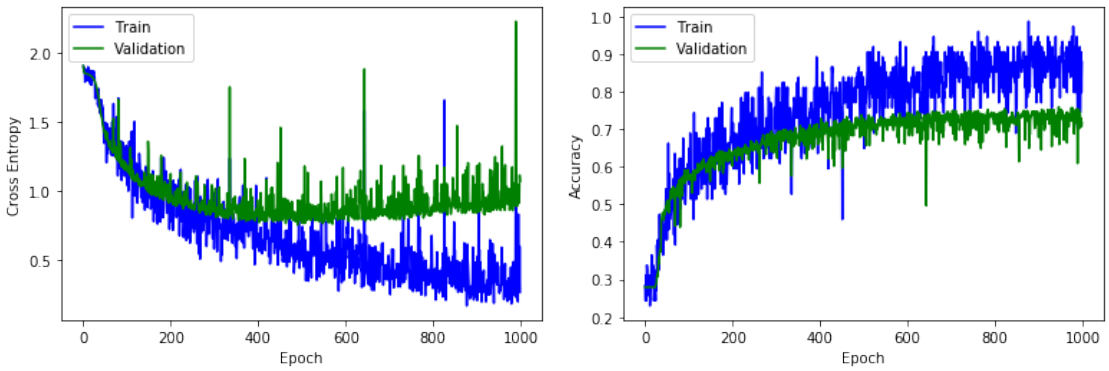

Figure 1: (1). Cross-entropy, (2). Accuracy

The network works better on the training set than on the validation set. This is a result of overfitting.

2.3

We keep other parameters to be the default when varying the one we are studying.

Changing η:

- Performance when $\eta = 0.001$:
1. Cross-Entropy: Train 1.10543, Validation 1.12684, Test 1.15172
2. Accuracy: Train 0.60670, Validation 0.57995, Test 0.56883

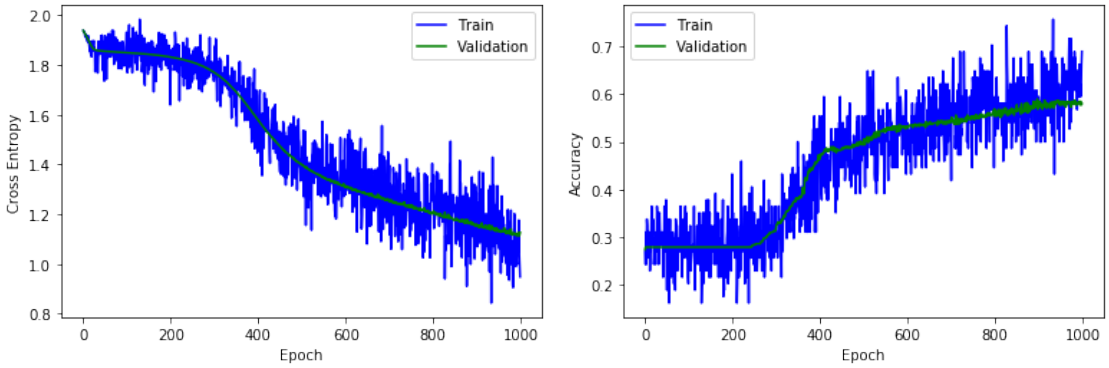

Figure 2: $\eta = 0.001$, (1). Cross-entropy, (2). Accuracy

- Performance when $\eta = 0.01$:
1. Cross-Entropy: Train 0.38604, Validation 1.07142, Test 0.91233
2. Accuracy: Train 0.86396, Validation 0.71599, Test 0.71688

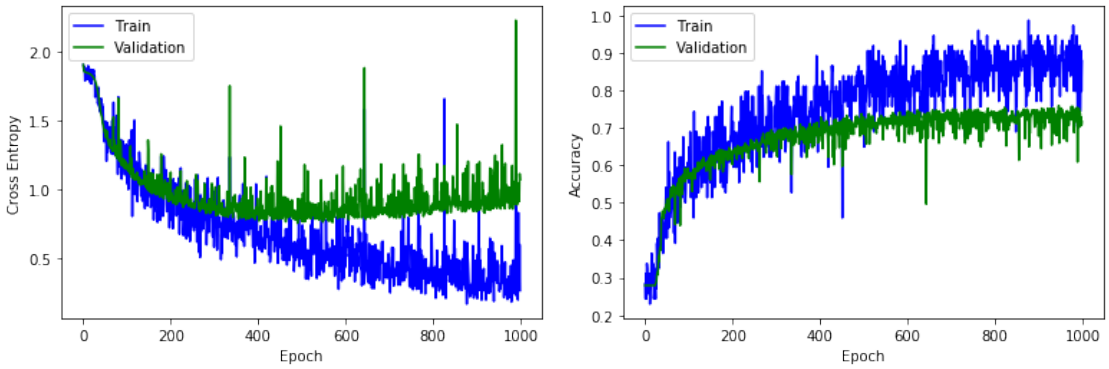

Figure 3: $\eta = 0.01$, (1). Cross-entropy, (2). Accuracy

- Performance when $\eta = 0.5$:
1. Cross-Entropy: Train 1.86108, Validation 1.85905, Test 1.83904
2. Accuracy: Train 0.28542, Validation 0.27924, Test 0.31688

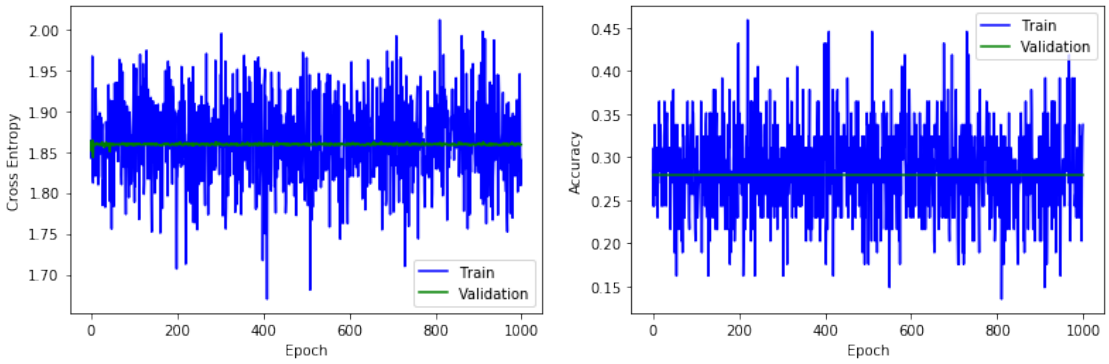

Figure 4: $\eta = 0.01$, (1). Cross-entropy, (2). Accuracy

Comment:
Cross-entropy is lowest and accuracy is the highest for medium large η, which is 0.01. If the learning rate becomes too small, the algorithm converges slowly and cannot converge within maximum iterations. When the learning rate is too large, the convergence process has a large noise and cannot converge to a minimum stably. In both cases, the change will decrease the accuracy.

Changing mini-batch sizes:

- Performance when mini-batch size is 10:
1. Cross-Entropy: Train 0.00293, Validation 2.53056, Test 2.17887
2. Accuracy: Train 1.00000, Validation 0.73508, Test 0.73766

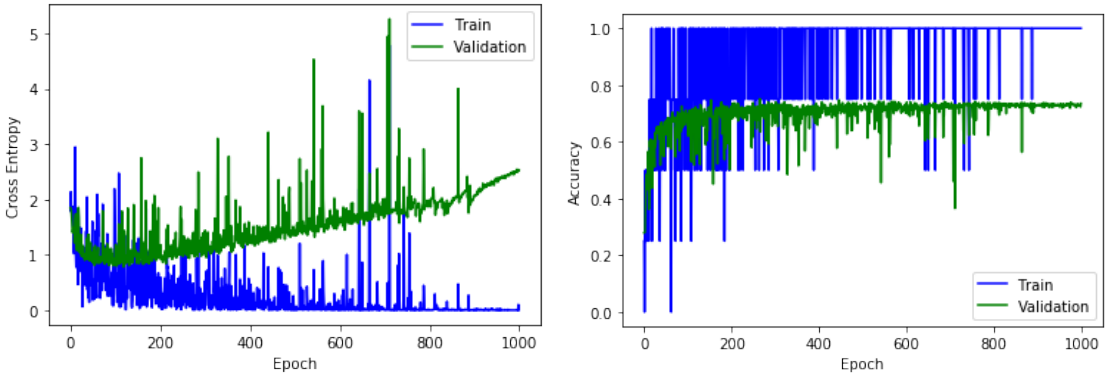

Figure 5: mini-batch size = 10, (1). Cross-entropy, (2). Accuracy

- Performance when mini-batch size is 100:
1. Cross-Entropy: Train 0.38604, Validation 1.07142, Test 0.91233
2. Accuracy: Train 0.86396, Validation 0.71599, Test 0.71688

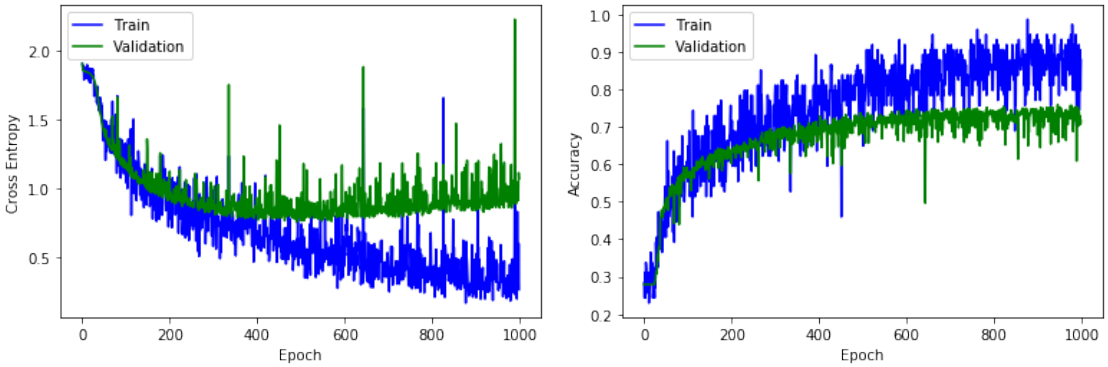

Figure 6: mini-batch size = 100, (1). Cross-entropy, (2). Accuracy

- Performance when mini-batch size is 1000:
1. Cross-Entropy: Train 1.06620, Validation 1.09098, Test 1.09856
2. Accuracy: Train 0.61440, Validation 0.58711, Test 0.57662

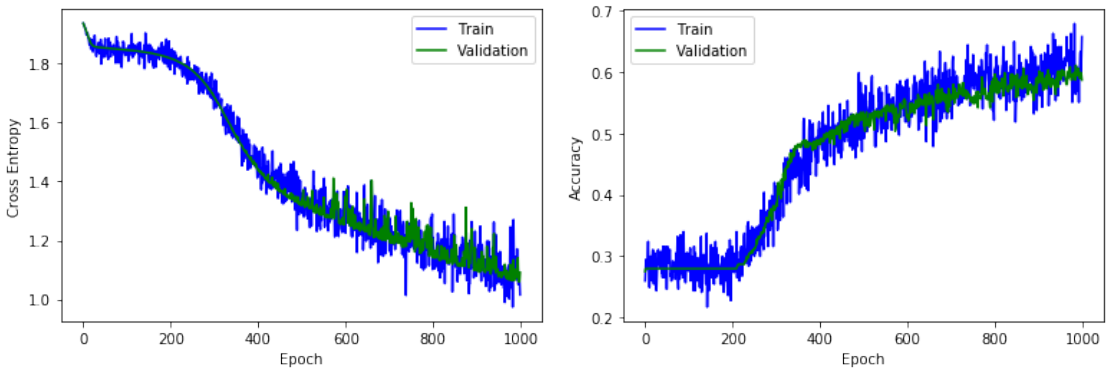

Figure 7: mini-batch size = 1000, (1). Cross-entropy, (2). Accuracy

Comment:
When mini-batch size increases, the accuracy decreases. Cross-entropy is lowest when we have a medium mini-batch size. With small mini-batch size, the algorithm converges quickly but resulting in a large noise in the training process. With large mini-batch size, the algorithm converges slowly with accurate estimates of the error gradient.

Choosing parameters:

We can choose the optimal parameter by cross validation. Choose the parameter that performs best on the cross validation set.

2.4

In this part, I left the first layer unchanged with default value 16 and changed the number of hidden units the second layer.

Hidden units in each layer:
- 2 hidden units:
1. Cross-Entropy: Train 1.01475, Validation 1.16106, Test 1.14734
2. Accuracy: Train 0.61915, Validation 0.60621, Test 0.59481

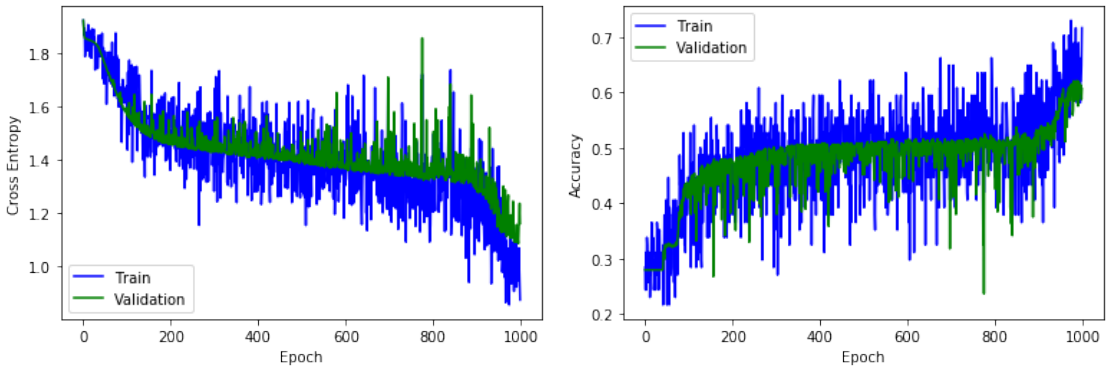

Figure 8: 2 hidden units, (1). Cross-entropy, (2). Accuracy

- 20 hidden units:
1. Cross-Entropy: Train 0.35273, Validation 1.02762, Test 0.94073
2. Accuracy: Train 0.86633, Validation 0.72315, Test 0.71948

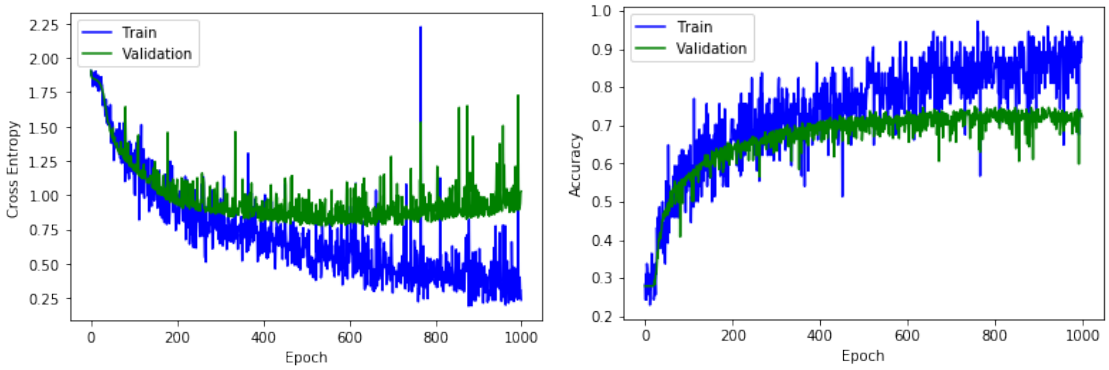

Figure 9: 20 hidden units, (1). Cross-entropy, (2). Accuracy

- 80 hidden units:
1. Cross-Entropy: Train 0.31556, Validation 1.08321, Test 0.98279
2. Accuracy: Train 0.87908, Validation 0.71360, Test 0.73247

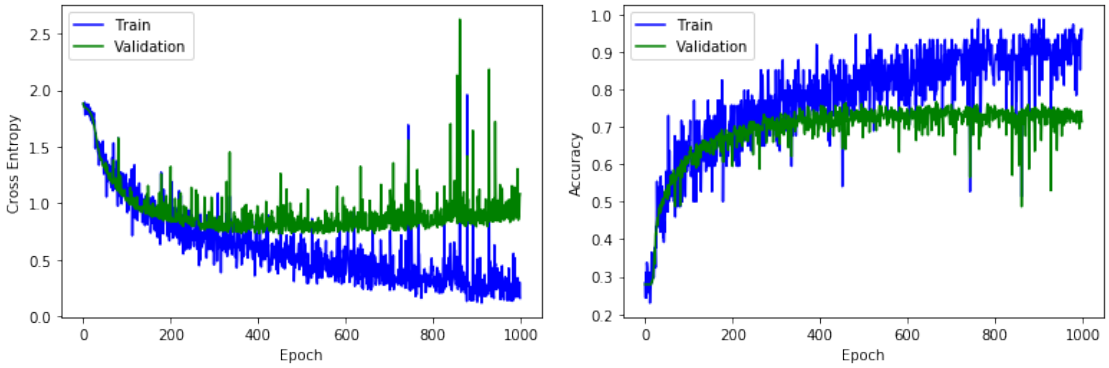

Figure 10: 80 hidden units, (1). Cross-entropy, (2). Accuracy

Comment:
When the hidden unit increases, the accuracy increases, but test accuracy increase much slower than training accuracy. The cross-entropy is smallest when we take a medium value of hidden layers. This is because when number of hidden layers is too small, the number of parameters is also small and the model tend to underfit. This will decrease both train and test accuracy. When there are too many hidden units, there will be too many parameters and the model tend to overfit. There will be a large gap between the accuracy of train and test sets. The accuracy on test data will not improve much with the expense of complexity. So choosing a moderate value of hidden units is important.

2.5

We use the default parameters when doing this part. I chose the 5 least certain expressions. Here are the figures:

Figure 11: Five examples

Comment:
These expressions are all hard to classify even for human, so the model can only give the category with the highest possibility and are not confident.
The classifier can be correct if it outputs the top scoring class anyways. For the five images above, the true class is almost the same as the top scoring class. So this classifier can be correct, although not 100% sure.

The codes begin on the next page.