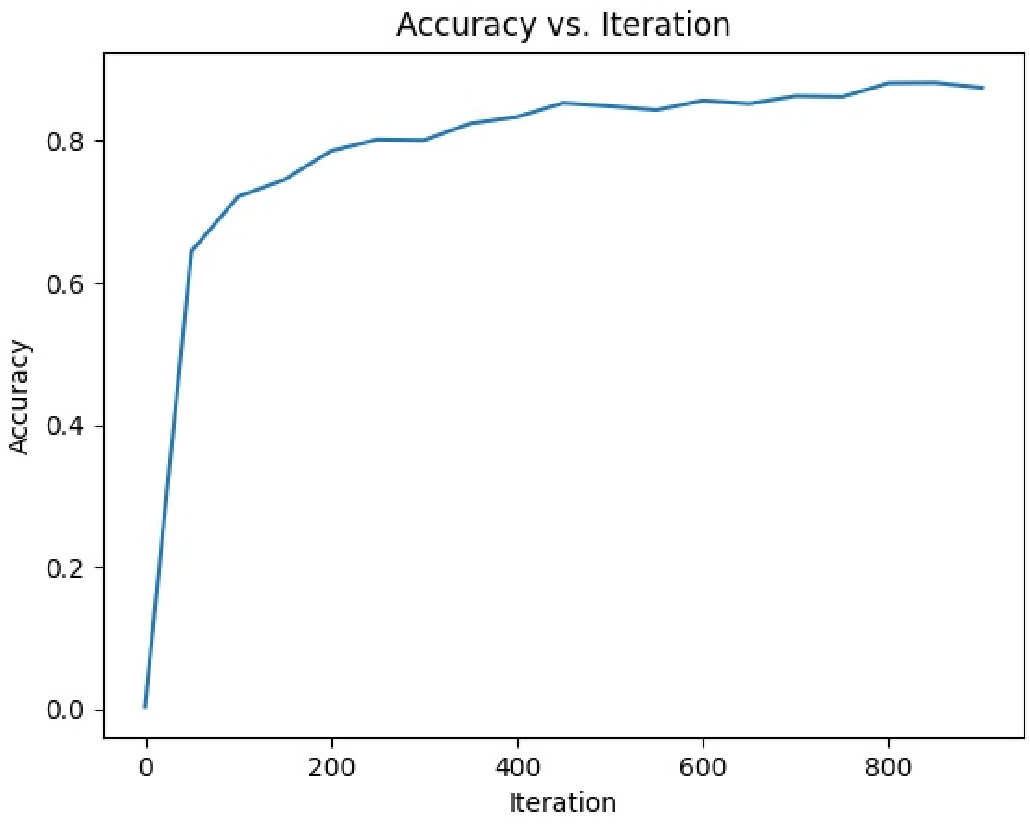
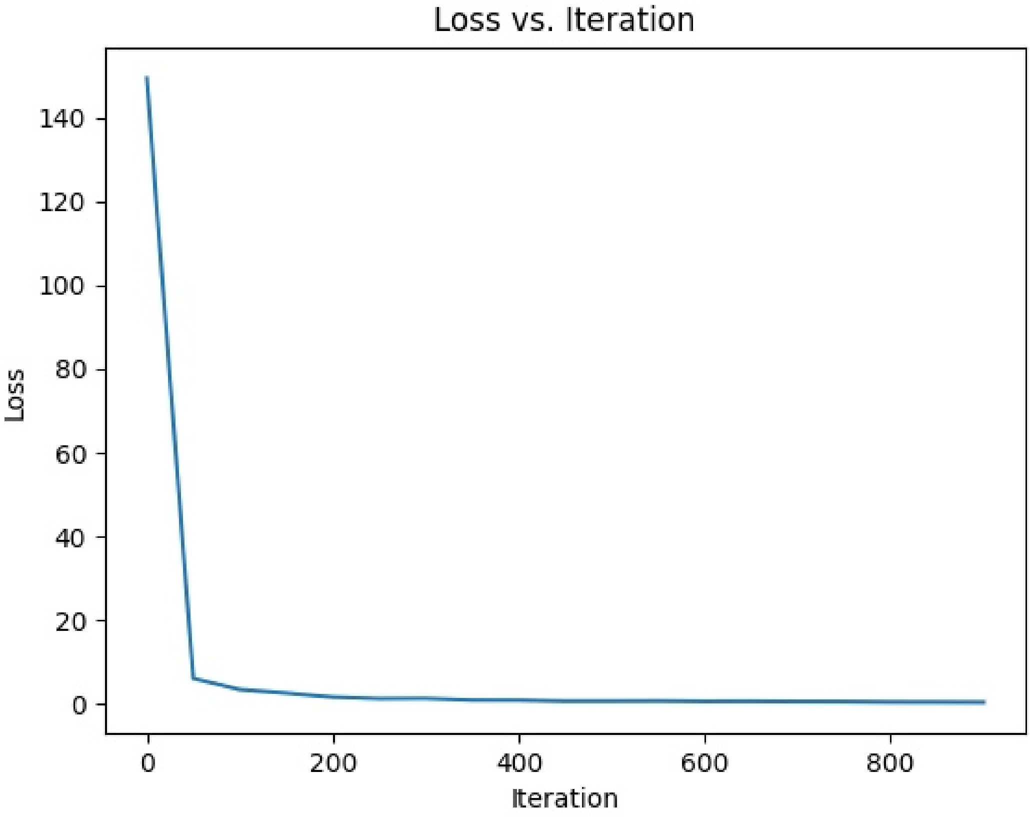
SI630 Homework 3 Report

Task 2

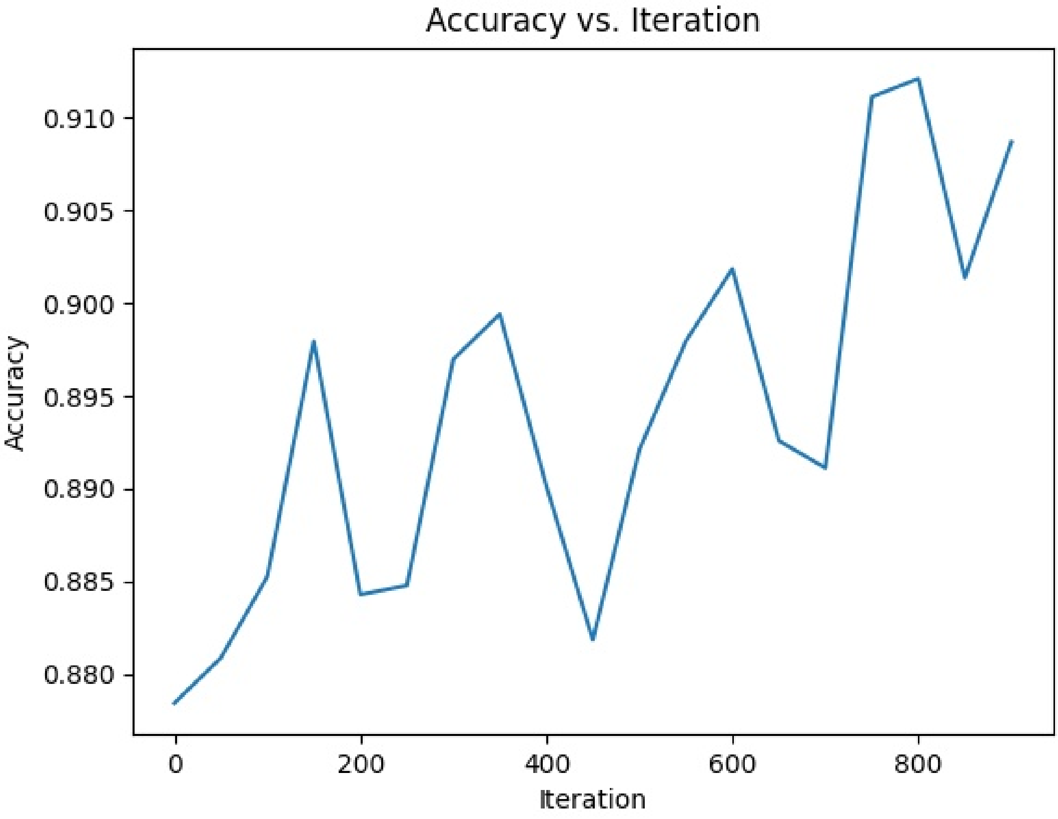
For each epoch, we draw the accuracy versus iteration plot and loss versus iteration plot. And then at last, we concatenate the loss curve and accuracy curve of those five epochs in a single plot for clearness.

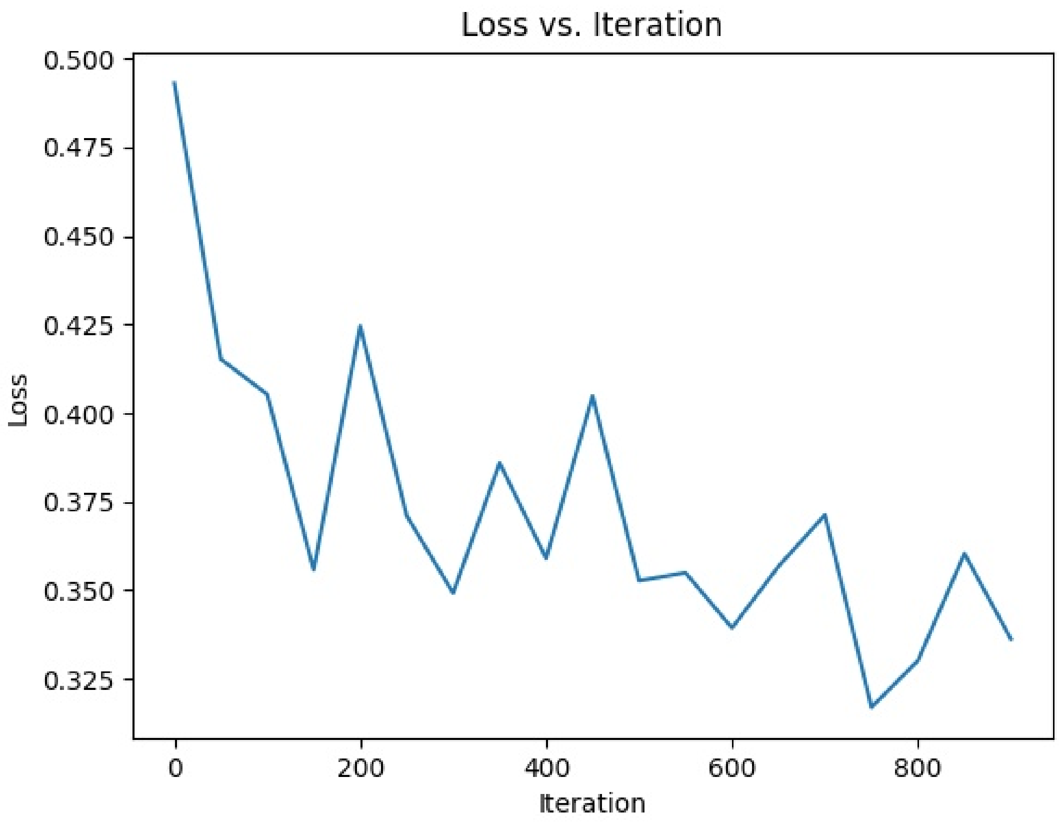
Epoch 1



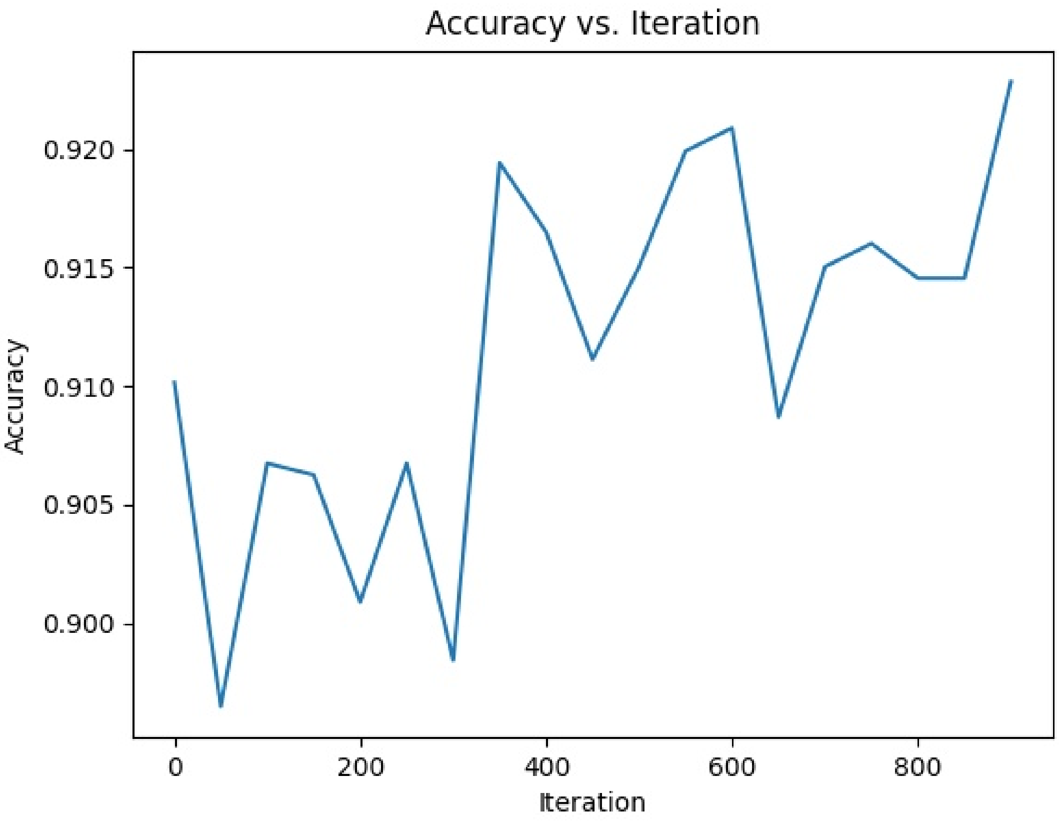

Loss vs. Iteration
Loss
140
120
100
80
60
40
20
0
0
200
400
600
800
Iteration

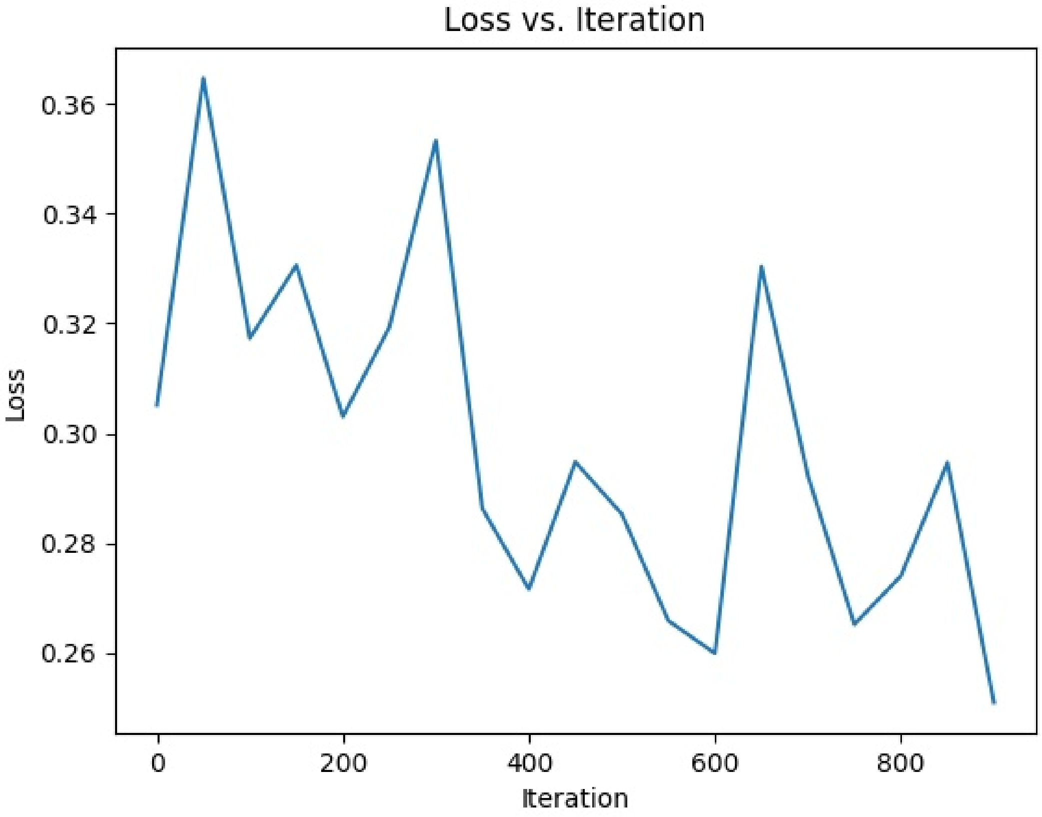
Epoch 2


Accuracy vs. Iteration
Accuracy
0.910
0.905
0.900
0.895
0.890
0.885
0.880
0
200
400
600
800
Iteration

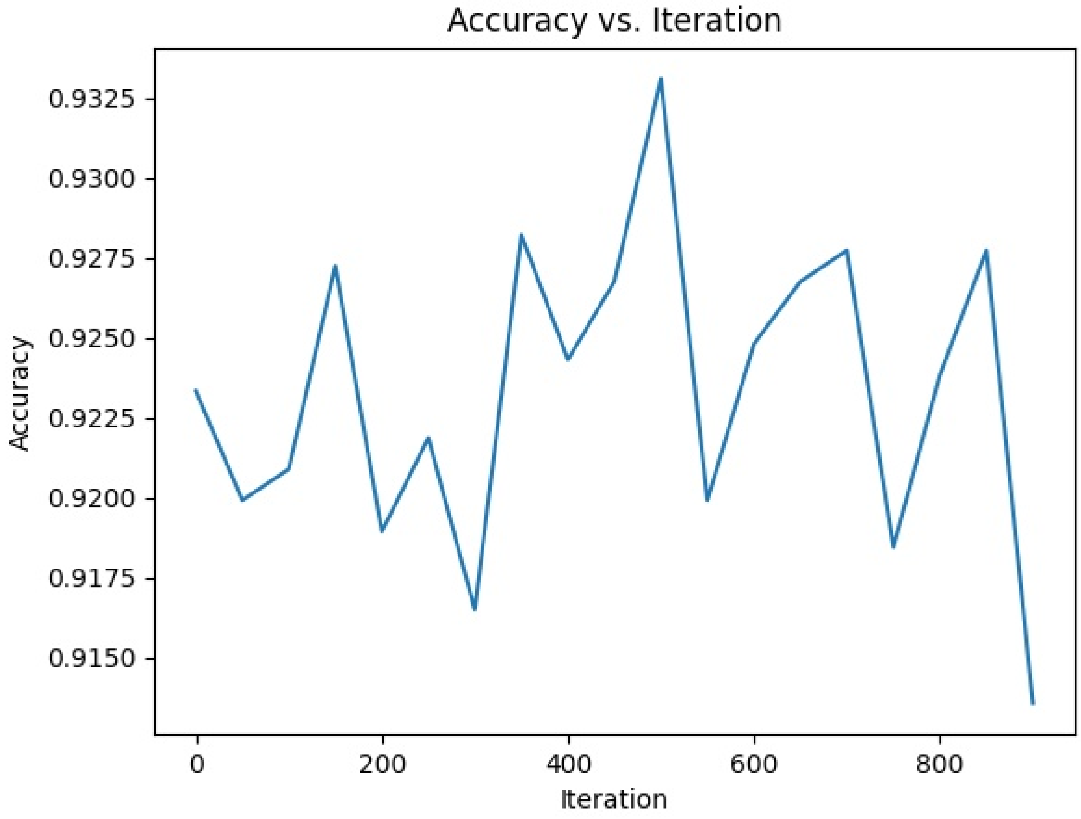

Loss vs. Iteration
Loss
0.500
0.475
0.450
0.425
0.400
0.375
0.350
0.325
0
200
400
600
800
Iteration

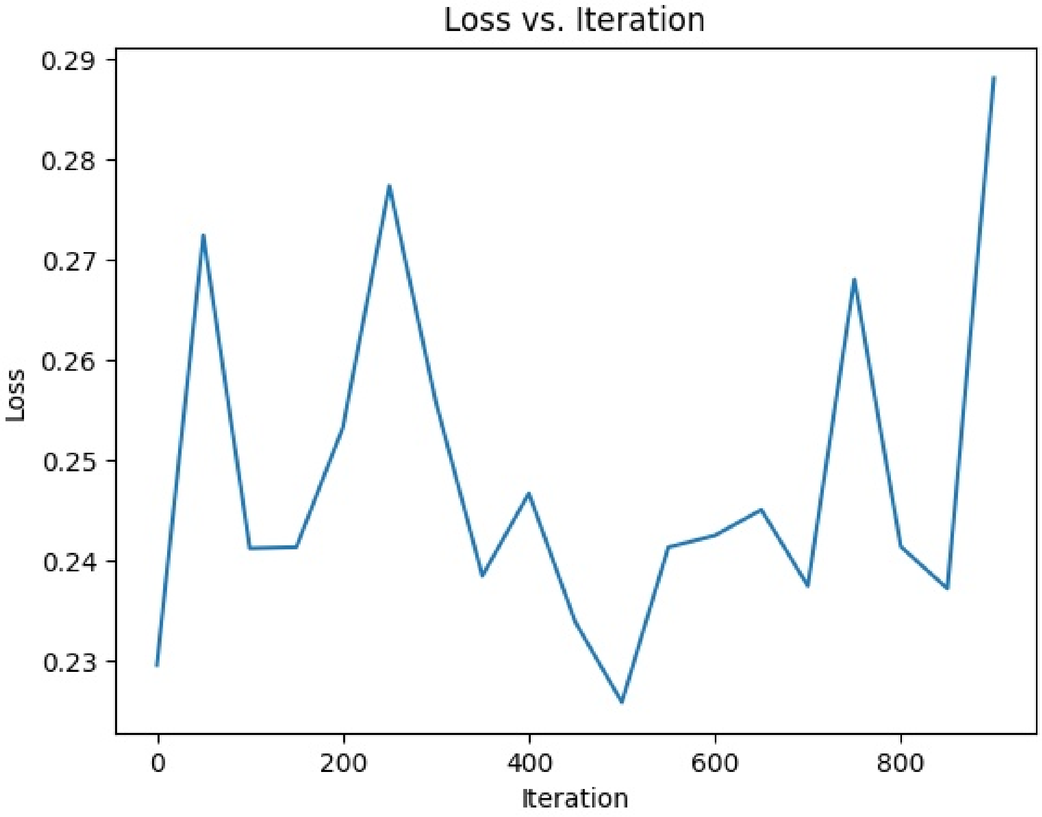
Epoch 3


Accuracy vs. Iteration
Accuracy
0.920
0.915
0.910
0.905
0.900
0
200
400
600
800
Iteration

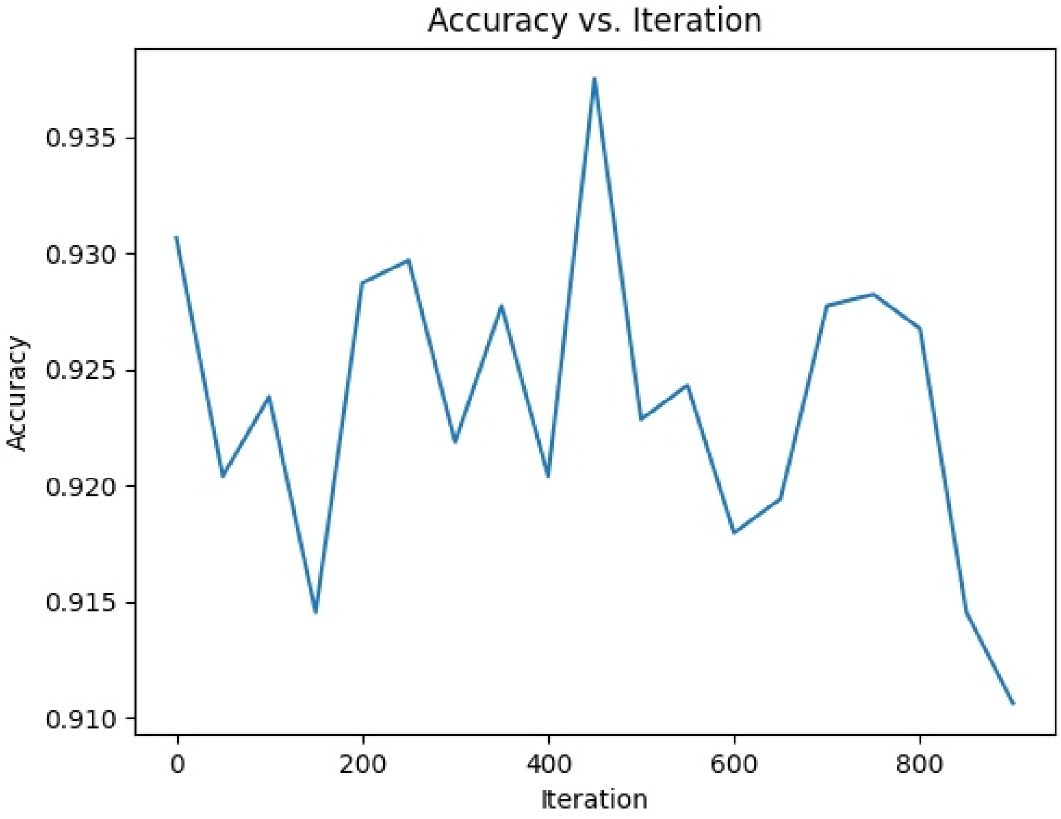

Loss vs. Iteration
Loss
0.36
0.34
0.32
0.30
0.28
0.26
0
200
400
600
800
Iteration

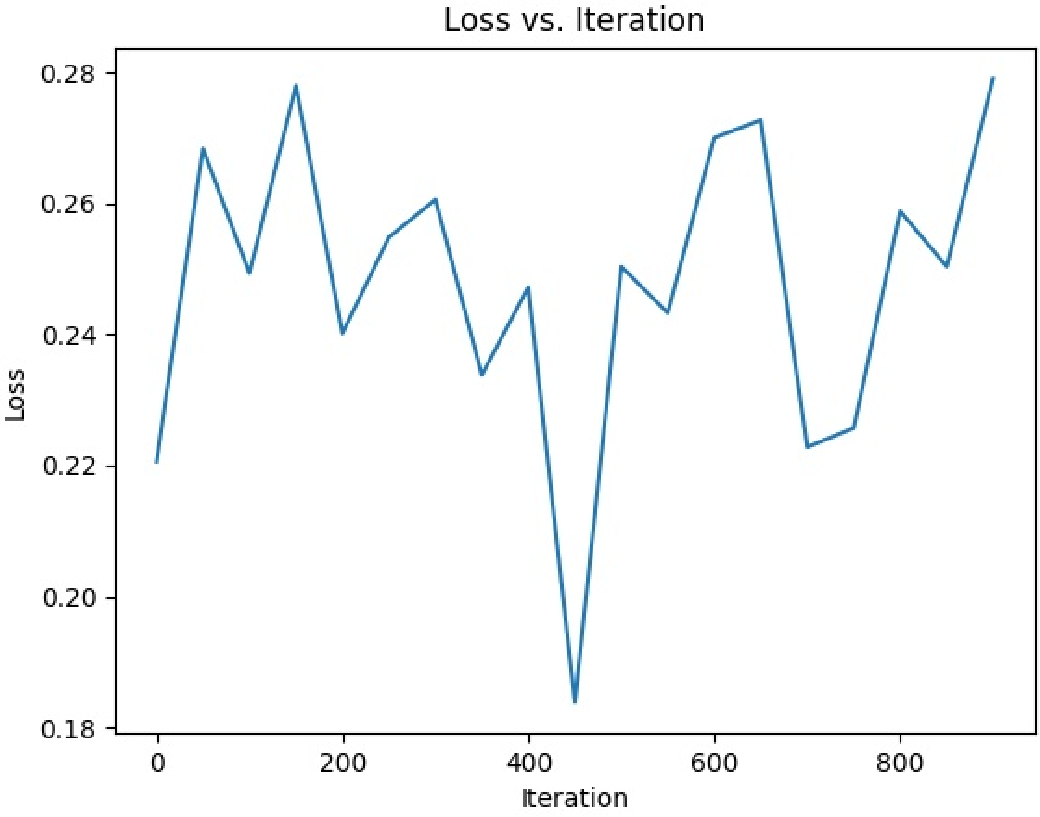
Epoch 4


Accuracy vs. Iteration
Accuracy
0.9325
0.9300
0.9275
0.9250
0.9225
0.9200
0.9175
0.9150
0
200
400
600
800
Iteration

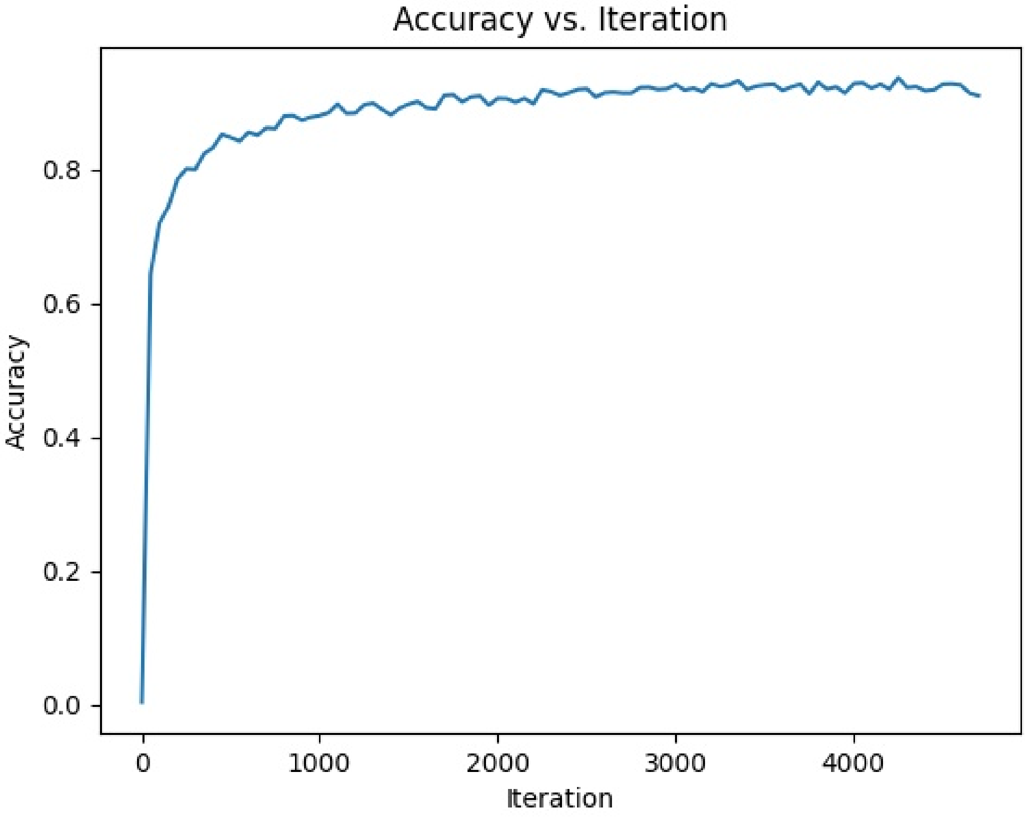

Loss vs. Iteration
Loss
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0
200
400
600
800
Iteration

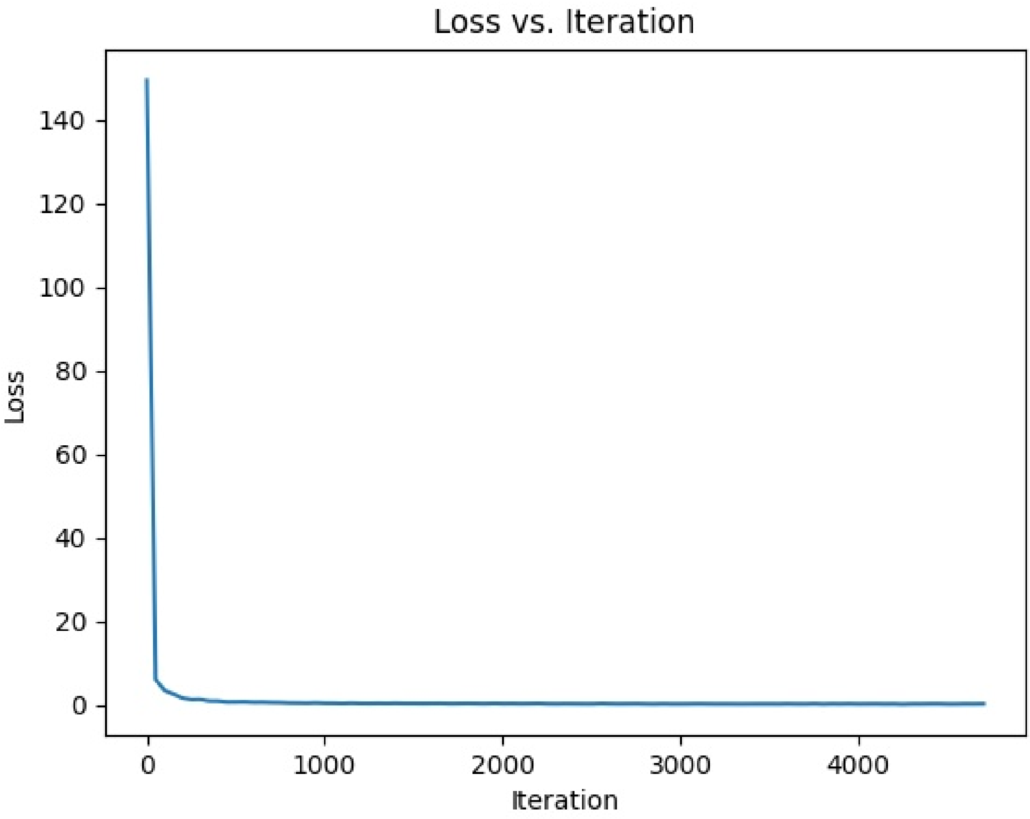
Epoch 5


Accuracy vs. Iteration
Accuracy
0.935
0.930
0.925
0.920
0.915
0.910
0
200
400
600
800
Iteration

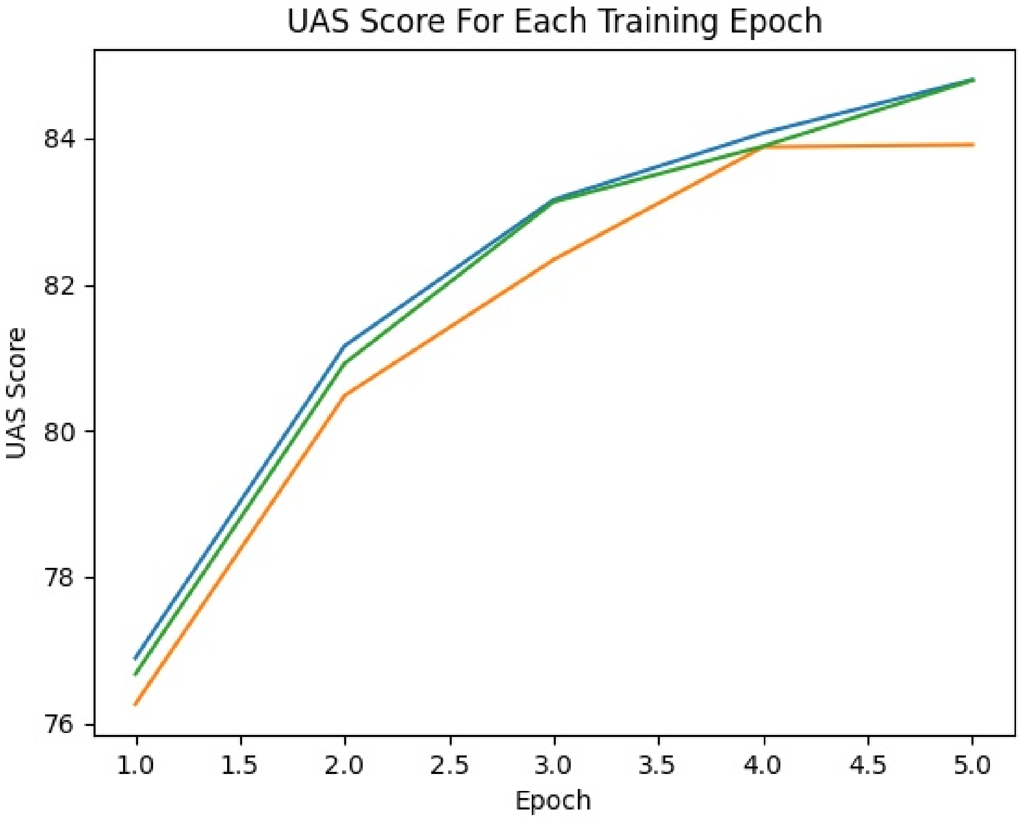


Total Accuracy and Loss (5 epochs concatenated)





UAS Score



In this plot, the model was trained for three times to show variance of the UAS Score on test data. The UAS Score result is shown in the table below.

	Experiment 1	Experiment 2	Experiment 3
Epoch 1	76.89	76.26	76.67
Epoch 2	81.16	80.48	80.92
Epoch 3	83.16	82.34	83.13
Epoch 4	84.07	83.88	83.89
Epoch 5	84.80	83.91	84.79

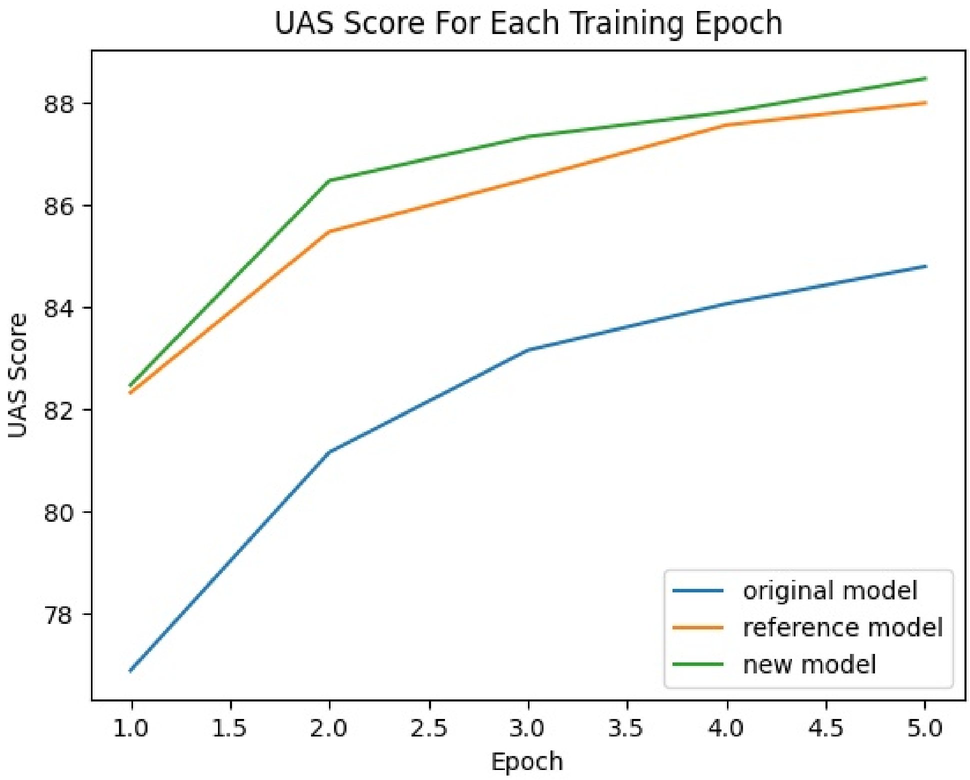
Findings

From the loss and accuracy plot for each epoch, we can see the accuracy curve climbed fast and loss curve also declined fast at epoch 1. For epoch 2 and 3, the speed of accuracy climbing and loss declining slowed down, but the tendency still remained. From epoch 4 the accuracy stopped climbing and the loss stopped decreasing, both the plots fluctuate in a random way. So 3 epochs training should be appropriate for this model.

Task 3

In my new parser model TwoHiddenParserModel, I tried different network designs and hyperparameters. The model is different from the original model in:

- There are 2 hidden layers with l1_hidden_size = 500 and l2_hidden_size = 200.
- The activation function is changed from cube to ReLU.
- The l1_hidden_size was changed from 200 to 500.



Then we measure the test UAS score of these two models for each training epoch. The result of a model with single hidden layer but using ReLU activation function is also shown for reference.

	Single Hidden Layer + Cube (Old)	Single Hidden Layers + ReLU (Reference)	Two Hidden Layers + ReLU (New)
1	76.89	82.33	82.48
2	81.86	85.48	86.48
3	83.16	86.51	87.34
4	84.07	87.57	87.82
5	84.80	88.00	88.48

From the result above we can see, the change of activation function from cube to ReLU improved the UAS Score a lot. The change from one hidden layer to two hidden layers further improved the performance a bit further.

Task 4

The output of parsing sentence "The big dog ate my homework" was shown below. The parser made some mistakes when parsing the first half of the sentence.

```
----
buffer: ['the', 'big', 'dog', 'ate', 'my', 'homework']
stack:  ['<root>']
action: shift
----
buffer: ['big', 'dog', 'ate', 'my', 'homework']
stack:  ['<root>', 'the']
action: shift
----
buffer: ['dog', 'ate', 'my', 'homework']
stack:  ['<root>', 'the', 'big']
action: shift
----
buffer: ['ate', 'my', 'homework']
stack:  ['<root>', 'the', 'big', 'dog']
action: shift
----
buffer: ['my', 'homework']
stack:  ['<root>', 'the', 'big', 'dog', 'ate']
action: left arc, <d>:compound
----
buffer: ['my', 'homework']
stack:  ['<root>', 'the', 'big', 'ate']
action: left arc, <d>:amod
----
buffer: ['my', 'homework']
stack:  ['<root>', 'the', 'ate']
action: left arc, <d>:det
----
buffer: ['my', 'homework']
stack:  ['<root>', 'ate']
action: shift
----
buffer: ['homework']
stack:  ['<root>', 'ate', 'my']
```

```
action: shift
----
buffer: []
stack:  ['<root>', 'ate', 'my', 'homework']
action: left arc, <d>:nmod:poss
----
buffer: []
stack:  ['<root>', 'ate', 'homework']
action: right arc, <d>:dep
----
buffer: []
stack:  ['<root>', 'ate']
action: right arc, <d>:root
    <root>
      |
     ate
  ____|___________
 |    |     |  homework
 |    |     |     |
the  big   dog    my
```

The correct operation for each step should be:

```
----
buffer: ['the', 'big', 'dog', 'ate', 'my', 'homework']
stack:  ['<root>']
action: shift
----
buffer: ['big', 'dog', 'ate', 'my', 'homework']
stack:  ['<root>', 'the']
action: shift
----
buffer: ['dog', 'ate', 'my', 'homework']
stack:  ['<root>', 'the', 'big']
action: shift
----
buffer: ['ate', 'my', 'homework']
stack:  ['<root>', 'the', 'big', 'dog']
action: left arc
----
buffer: ['ate', 'my', 'homework']
stack:  ['<root>', 'the', 'dog']
action: left arc,
----
buffer: ['ate', 'my', 'homework']
stack:  ['<root>', 'dog']
action: shift,
----
buffer: ['my', 'homework']
stack:  ['<root>', 'dog', 'ate']
action: left arc,
----
buffer: ['my', 'homework']
stack:  ['<root>', 'ate']
action: shift
----
buffer: ['homework']
```

```
stack:  ['<root>', 'ate', 'my']
action: shift
----
buffer: []
stack:  ['<root>', 'ate', 'my', 'homework']
action: left arc,
----
buffer: []
stack:  ['<root>', 'ate', 'homework']
action: right arc,
----
buffer: []
stack:  ['<root>', 'ate']
action: right arc,
```