Problem 1

1. **Flatten + FC (Fully Connected)**
   a. **Forward :**
   ```
   Difference:  4.0260162945880345e-09
   ```
   b. **Backward**
   ```
   dx Error:  8.416294705242632e-10
   dw Error:  3.4573909187264005e-09
   db Error:  1.8121943810977335e-11
   dinp Shape:  (15, 2, 2, 3) (15, 2, 2, 3)
   ```

2. **GELU**
   a. **Forward**
   ```
   Difference:  1.8037541876132445e-08
   ```
   b. **Backward**
   ```
   dx Error:  9.919403952243922e-10
   ```

3. **Dropout**
   a. **Forward**

   ```
   ----------------------------------------------------------------
   Dropout Keep Prob =  0
   Mean of input:  4.992425267027468
   Mean of output during training time:  4.992425267027468
   Mean of output during testing time:  4.992425267027468
   Fraction of output set to zero during training time:  0.0
   Fraction of output set to zero during testing time:  0.0
   ----------------------------------------------------------------
   Dropout Keep Prob =  0.25
   Mean of input:  4.992425267027468
   Mean of output during training time:  5.025032966043185
   Mean of output during testing time:  4.992425267027468
   Fraction of output set to zero during training time:  0.7486
   Fraction of output set to zero during testing time:  0.0
   ----------------------------------------------------------------
   Dropout Keep Prob =  0.5
   Mean of input:  4.992425267027468
   Mean of output during training time:  5.023324749546829
   Mean of output during testing time:  4.992425267027468
   Fraction of output set to zero during training time:  0.4957
   Fraction of output set to zero during testing time:  0.0
   ----------------------------------------------------------------
   Dropout Keep Prob =  0.75
   Mean of input:  4.992425267027468
   Mean of output during training time:  4.991425430625906
   Mean of output during testing time:  4.992425267027468
   Fraction of output set to zero during training time:  0.2496
   Fraction of output set to zero during testing time:  0.0
   ----------------------------------------------------------------
   Dropout Keep Prob =  1
   ```

```
Mean of input:  4.992425267027468
Mean of output during training time:  4.992425267027468
Mean of output during testing time:  4.992425267027468
Fraction of output set to zero during training time:  0.0
Fraction of output set to zero during testing time:  0.0
----------------------------------------------------------------
```

b. **Backward**

```
dx relative error:  3.003113496265614e-11
```

4. **FC+GELU**

```
dx error:  2.0201012078459058e-09
dw error:  7.302819581693971e-09
db error:  5.777871390451246e-10
```

Param names : fc_w, fc_b

5. **Softmax & Loss Layer**

```
Cross Entropy Loss:  1.7915748178170066
dx error:  7.3866310558791855e-09
```

6. **Test Small Fully connected Network**

```
Testing initialization ...
Passed!
Testing test-time forward pass ...
Passed!
Testing the loss ...
Passed!
Testing the gradients (error should be no larger than 1e-6) ...
fc1_b relative error: 5.94e-09
fc1_w relative error: 1.06e-08
fc2_b relative error: 4.01e-10
fc2_w relative error: 2.50e-08

Param names : fc1_w, fc1_b, fc2_w, fc2_b
```

7. **Test a fully connected network regularized + dropout**

```
Dropout p = 0
Error of gradients should be around or less than 1e-3
fc1_b relative error: 9.824168508277432e-08
fc1_w relative error: 4.706355825013066e-06
fc2_b relative error: 1.133402768221828e-08
fc2_w relative error: 3.1672231525171774e-05
fc3_b relative error: 2.0518174276833617e-10
fc3_w relative error: 2.720304740415546e-06

Dropout p = 0.25
Error of gradients should be around or less than 1e-3
fc1_b relative error: 1.894959185779182e-07
fc1_w relative error: 3.4287142983339667e-06
```

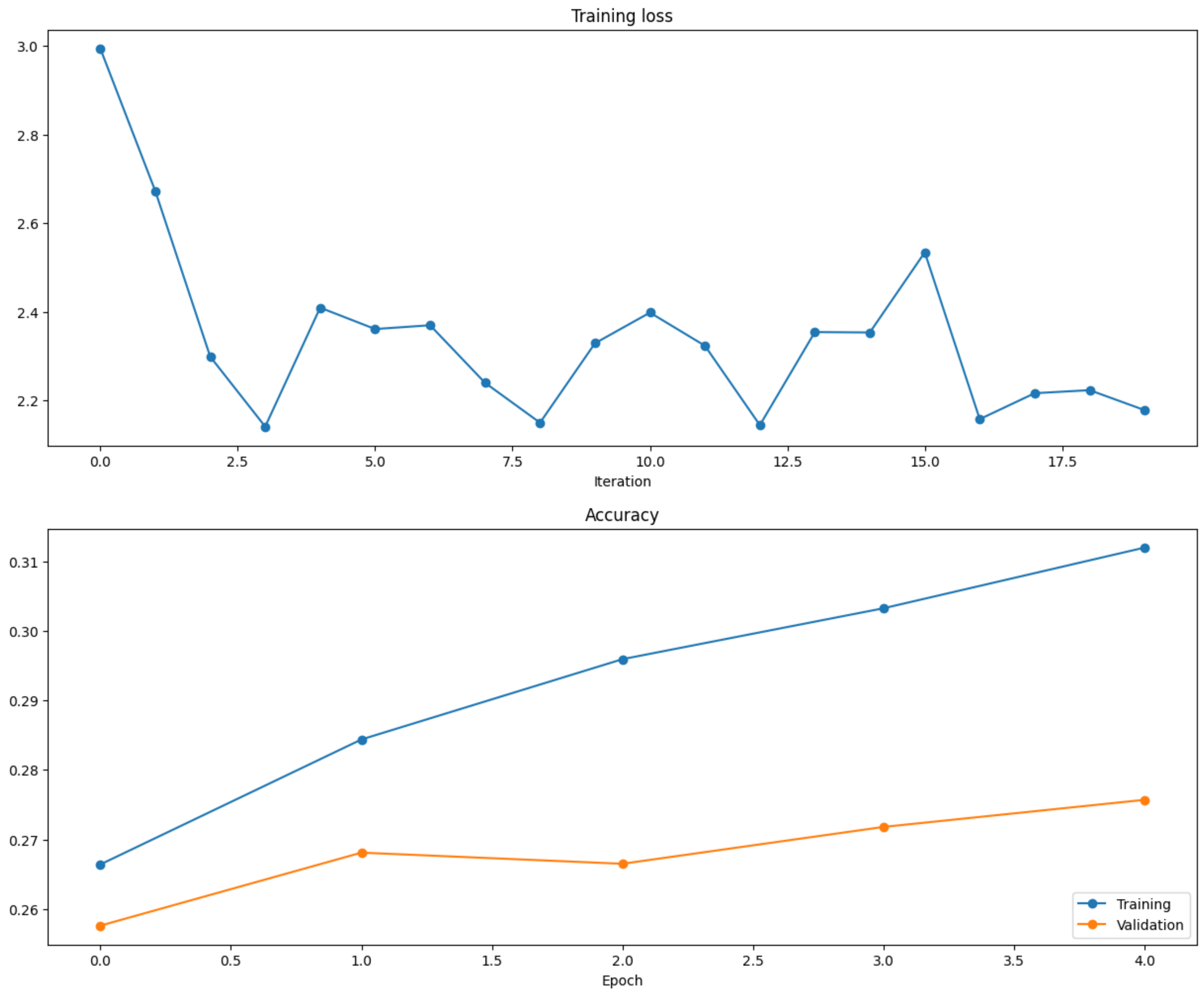
```
fc2_b relative error: 1.6435766065275814e-07
fc2_w relative error: 4.52072681731756e-05
fc3_b relative error: 2.1474160887299336e-10
fc3_w relative error: 7.9382903586546e-07

Dropout p = 0.5
Error of gradients should be around or less than 1e-3
fc1_b relative error: 3.613140820908816e-07
fc1_w relative error: 4.604428759190954e-07
fc2_b relative error: 1.7902141999906472e-08
fc2_w relative error: 7.923786506996994e-06
fc3_b relative error: 3.285178756580047e-10
fc3_w relative error: 1.103448593805289e-05
```

8. **Train a Network : Flatten->FC->GeLU->FC**

```
dict_keys(['fc1_w', 'fc1_b', 'fc2_w', 'fc2_b'])

Loading Params: fc1_w Shape: (3072, 48)
Loading Params: fc1_b Shape: (48,)
Loading Params: fc2_w Shape: (48, 20)
Loading Params: fc2_b Shape: (20,)
Validation Accuracy: 30.759999999999998%
Testing Accuracy: 30.55%
```


Training loss
3.0
2.8
2.6
2.4
2.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Iteration
Accuracy
0.31
0.30
0.29
0.28
0.27
0.26
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epoch
Training
Validation

9. SGD+Weight Decay

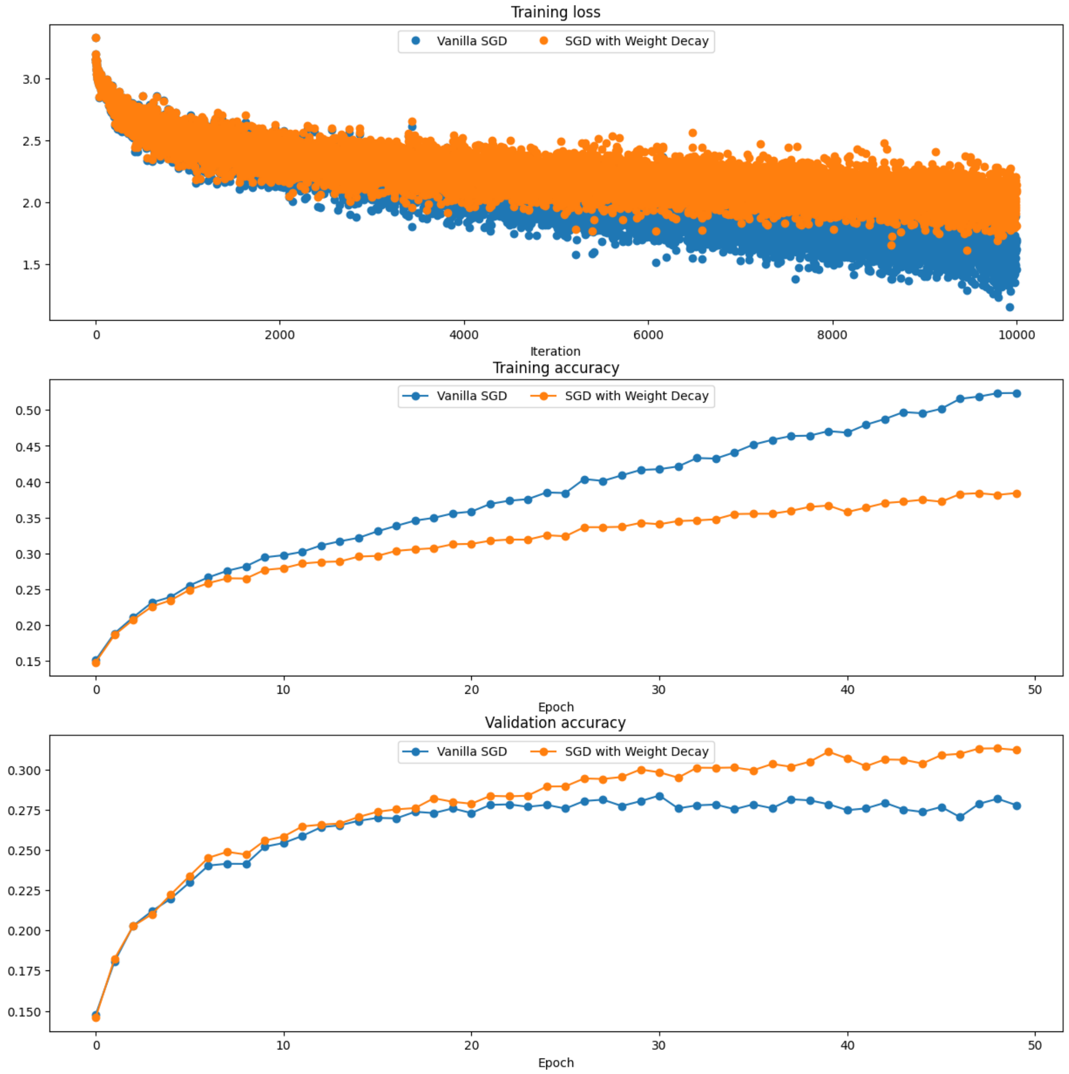
A.) Updated Error

```
The following errors should be around or less than 1e-6
updated_w error:  8.677112905190533e-08
```

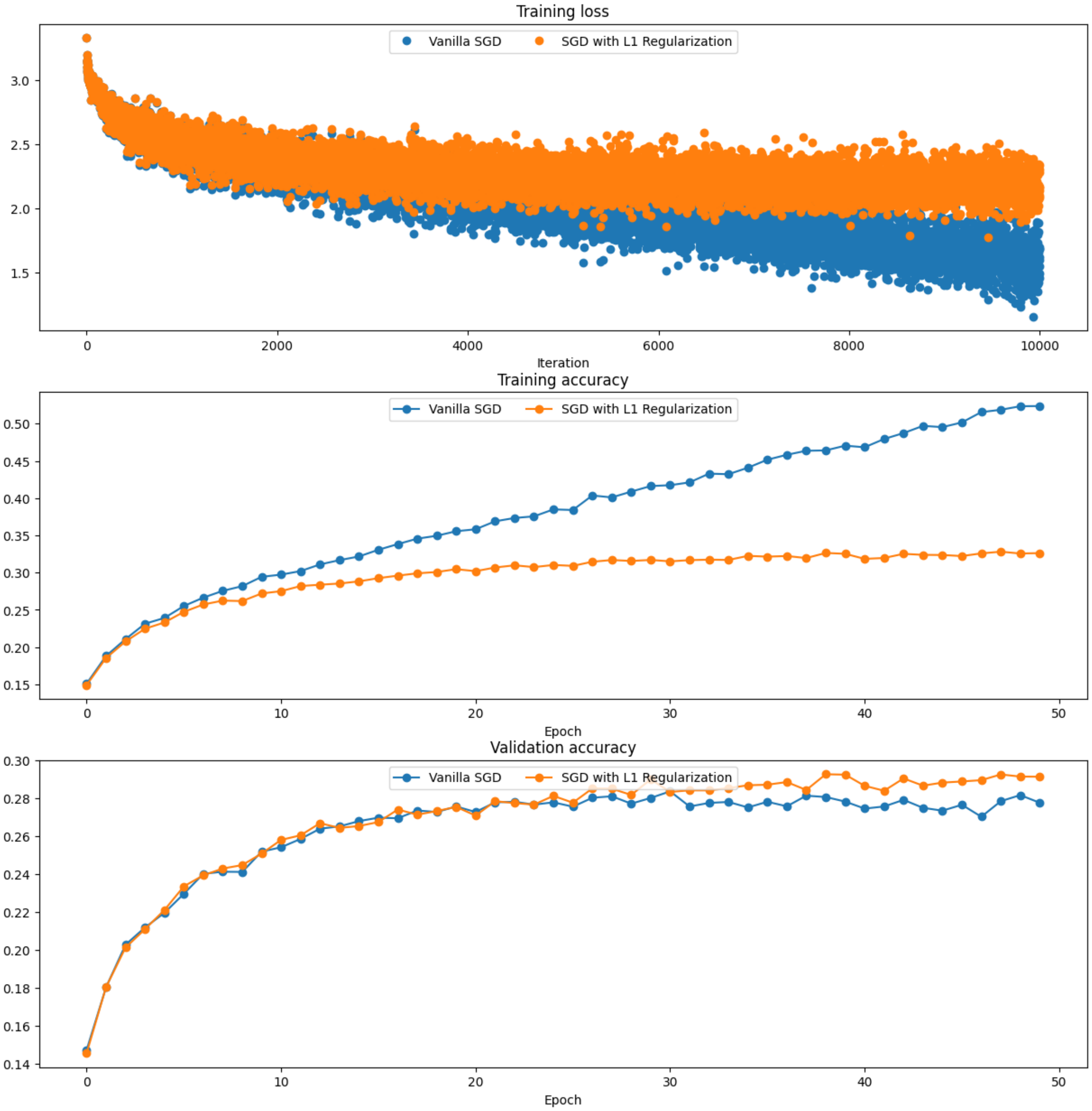
B.) Comparison

Weight Decay has better validation accuracy than vanilla SGD, as the curve is higher in each epoch as represented in graph & calculations.

```
Average Accuracy SGD:
 Training: 38.18000000000001% , Validation : 26.179999999999996%
Average Accuracy SGD + weight Decay:
 Training: 31.96% , Validation: 27.980000000000004%
```



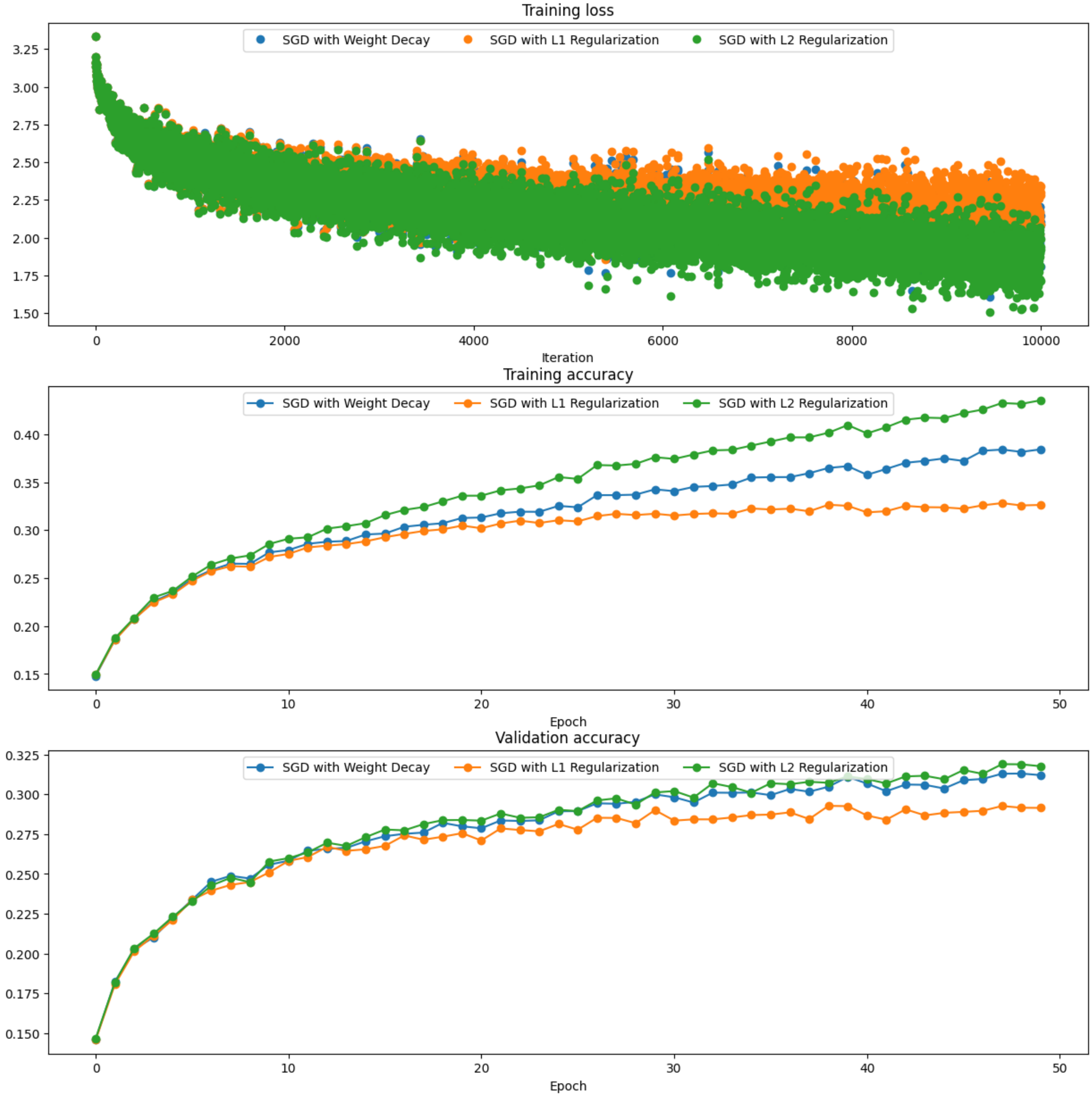
10. SGD + L1 regularization



11. SGD + L2 Regularization

```
lambda = 5e-3
```

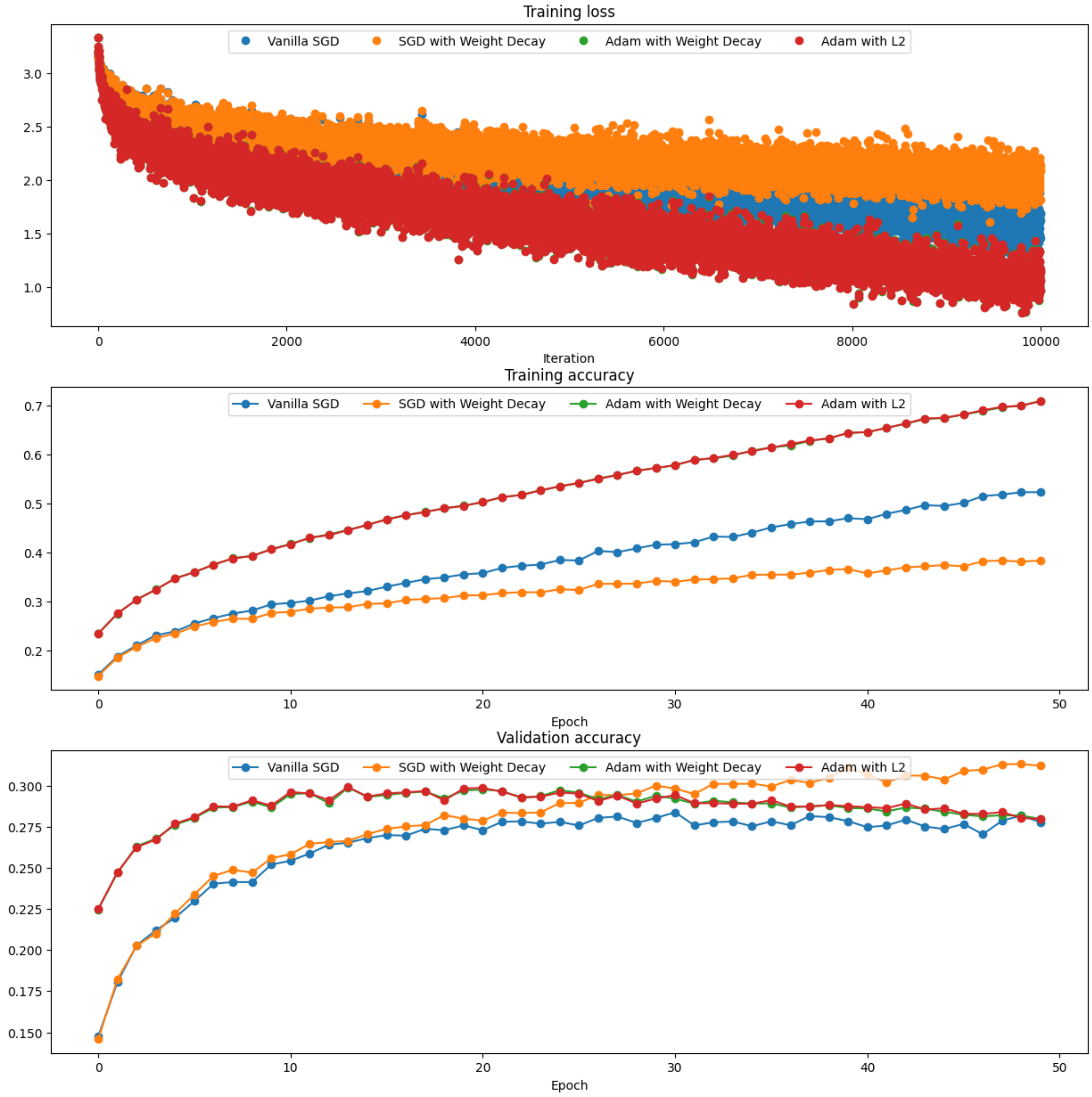
Such that it functions same as weight decay parameter 1e-4 Regularization



12. Adams

```
## Inline Answer
- Yes, they are still the same, as the below graphs are overlapping.
- We can do the same in adams also, by setting the hyperparameter (lambda).
```

```
The following errors should be around or less than 1e-7
updated_w error:  1.1395691798535431e-07
mt error:  4.214963193114416e-09
vt error:  4.208314038113071e-09
```


Training loss
● Vanilla SGD
● SGD with Weight Decay
● Adam with Weight Decay
● Adam with L2
Iteration
Training accuracy
Vanilla SGD
SGD with Weight Decay
Adam with Weight Decay
Adam with L2
Epoch
Validation accuracy
Vanilla SGD
SGD with Weight Decay
Adam with Weight Decay
Adam with L2
Epoch