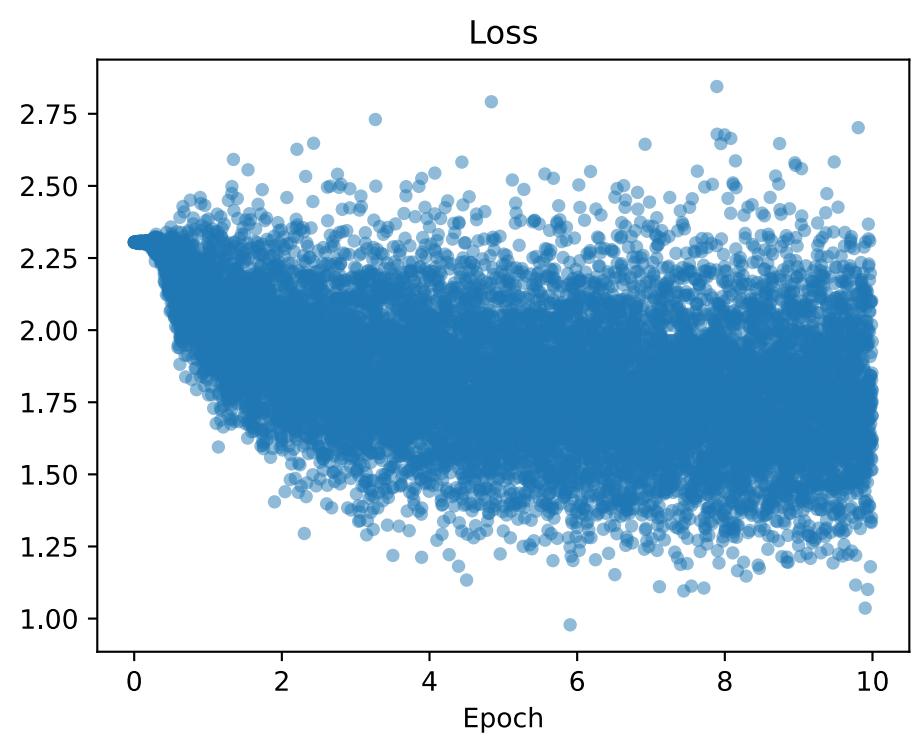
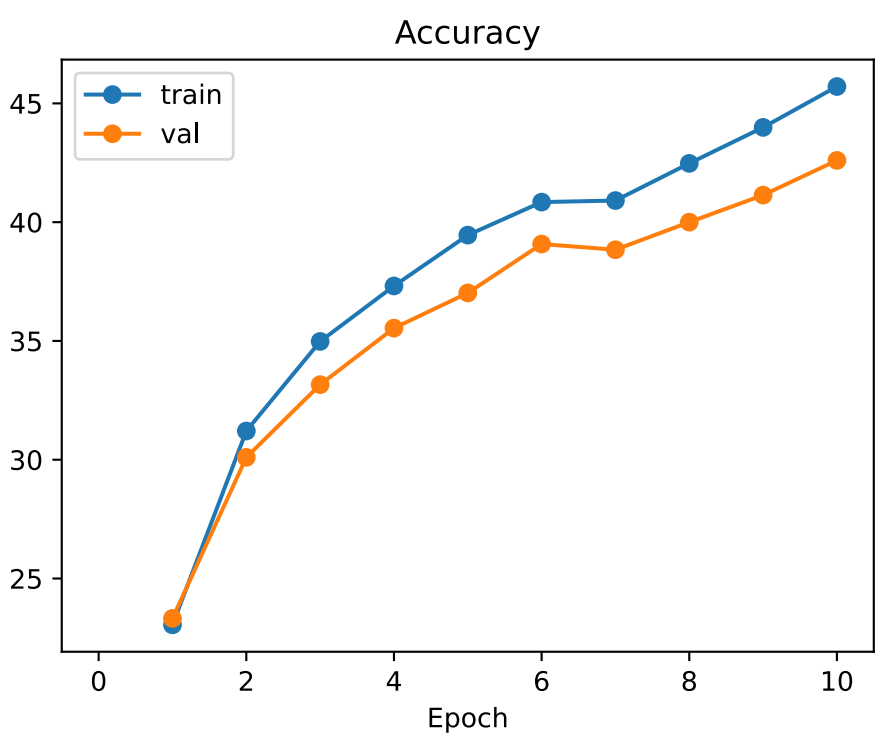
```
Running numeric gradient check for fc
  grad_x difference:  1.2427170403839227e-10
  grad_w difference:  1.8818568925382806e-10
  grad_b difference:  5.866240826435387e-11
Running numeric gradient check for relu
  grad_x difference:  1.573126073850517e-10
Running numeric gradient check for softmax loss
  grad_x difference:  0.08402553245025368
Running numeric gradient check for L2 regularization
  grad_w difference:  9.897935249192358e-11
```

```
Running numeric gradient check for LinearClassifier
  Max diff for grad_W:  1.6875389974302 38e-13
  Max diff for grad_b:  4.072298054325074e-13
Running numeric gradient check for TwoLayerNet
  Max diff for grad_W1:  5.029613878160255e-16
  Max diff for grad_b1:  3.5258254649228604e-16
  Max diff for grad_W2:  4.206704429243757e-16
  Max diff for grad_b2:  4.440892098500626e-16
```

```
Input scores:
[[    0. -1000. -2000.]
 [-2000.     0. -1000.]
 [-1000. -2000.     0.]]
Input labels: [0 1 2]
Output loss: -0.0
It seems like your softmax_loss is numerically stable!
```

```
Checking accuracy
  Train: 46.34
  Val:   43.20
Saving plot to plot.pdf
Saving model checkpoint to checkpoint.pkl
(base) Joshs-MBP:neuralnet JOSH$ python test.py
Loading model from checkpoint.pkl
Test-set accuracy: 44.27
(base) Joshs-MBP:neuralnet JOSH$
```


Loss
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0
2
4
6
8
10
Epoch


Accuracy
train
val
45
40
35
30
25
0
2
4
6
8
10
Epoch