schulzdLab7Report

May 3, 2021

1 Lab 7 - Completing the From-Scratch Library

David Schulz

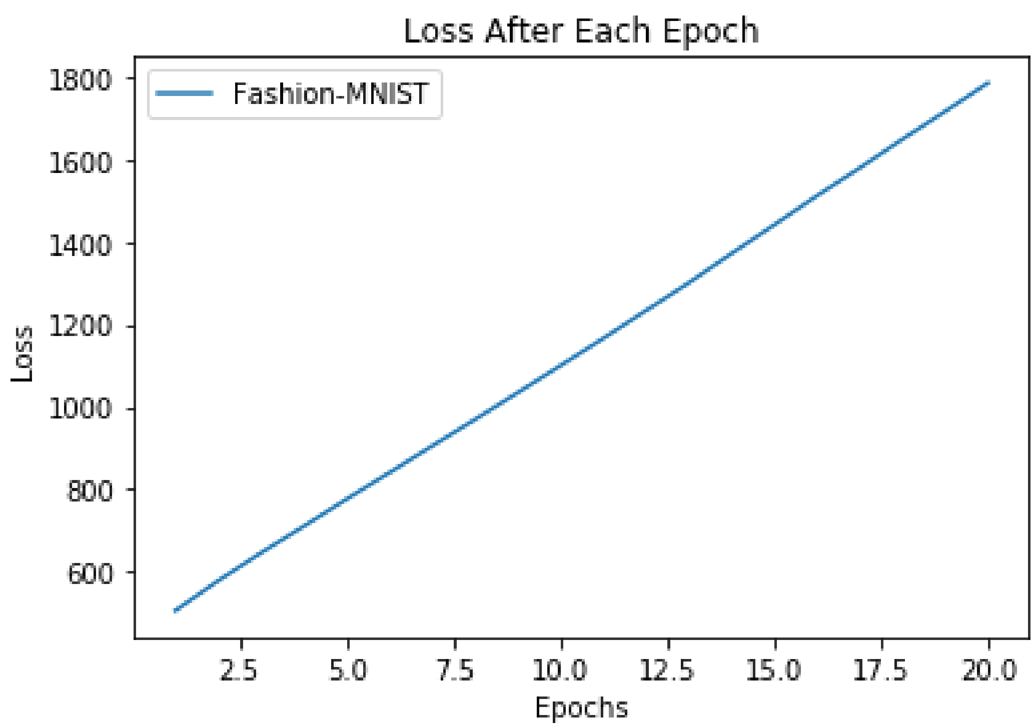
1.1 Training Curves

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
```

[2]:
```
epochs = np.array(range(1,21))

fashion_losses = [505.24212646484375, 577.6492919921875, 645.2394409179688, 709.
 ↪7667846679688,
                  775.5195922851562, 839.787841796875, 904.2780151367188, 968.
 ↪4905395507812,
                  1033.58056640625, 1099.2720947265625, 1165.2178955078125,␣
 ↪1232.8719482421875,
                  1300.3104248046875, 1371.1170654296875, 1440.698486328125,␣
 ↪1511.46484375,
                  1578.900146484375, 1648.78369140625, 1716.6722412109375, 1785.
 ↪1988525390625]
fashion_losses = np.array(fashion_losses)

cifar10_losses = [7795.271484375, 7858.578125, 7921.3212890625, 7990.
 ↪33642578125,
                  8063.974609375, 8142.40185546875, 8227.3623046875, 8321.
 ↪05859375,
                  8417.5205078125, 8520.2119140625, 8626.1533203125, 8738.
 ↪3310546875,
                  8852.8837890625, 8973.73828125, 9100.08203125, 9224.
 ↪8408203125,
                  9358.5224609375, 9499.5966796875, 9638.525390625, 9781.
 ↪3505859375]
cifar10_losses = np.array(cifar10_losses)
```

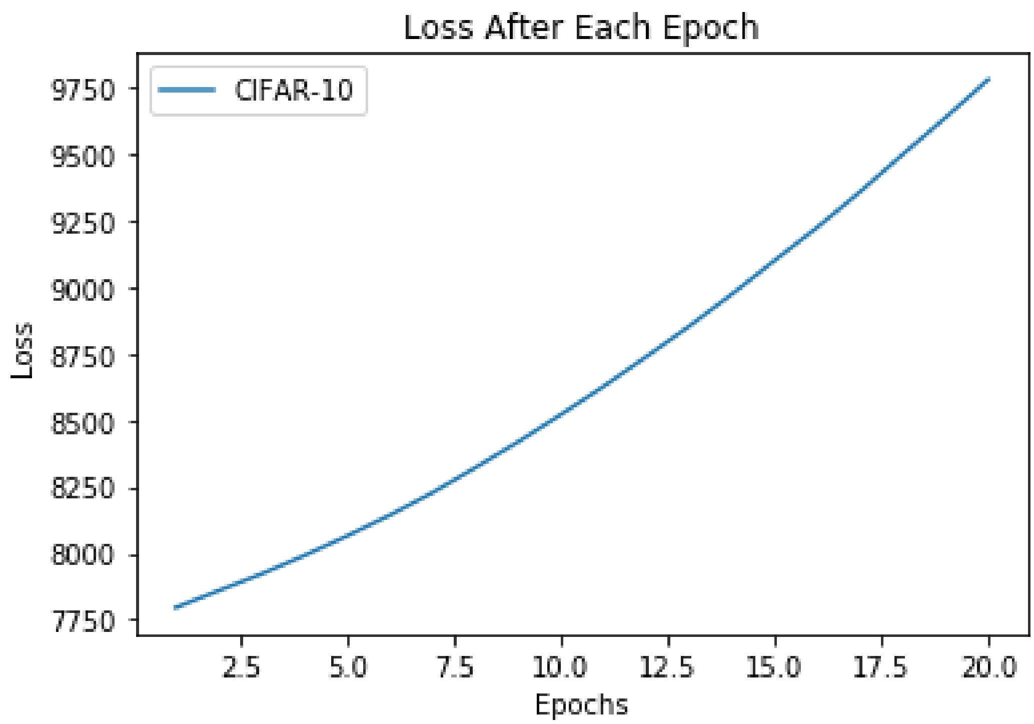
```
cifar100_losses = [15924.796875, 15999.2294921875, 16099.1171875, 16215.
 ↪767578125,
                   16362.2314453125, 16530.259765625, 16722.853515625, 16935.
 ↪900390625,
                   17169.70703125, 17421.0703125, 17686.396484375, 17966.
 ↪830078125,
                   18265.9140625, 18566.6484375, 18875.8515625, 19200.875,
                   19542.400390625, 19888.662109375, 20250.076171875, 20619.
 ↪205078125]
cifar100_losses = np.array(cifar100_losses)

plt.title("Loss After Each Epoch")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.plot(epochs, fashion_losses, label="Fashion-MNIST")
plt.legend()
```

[2]: <matplotlib.legend.Legend at 0x7f1978c75940>



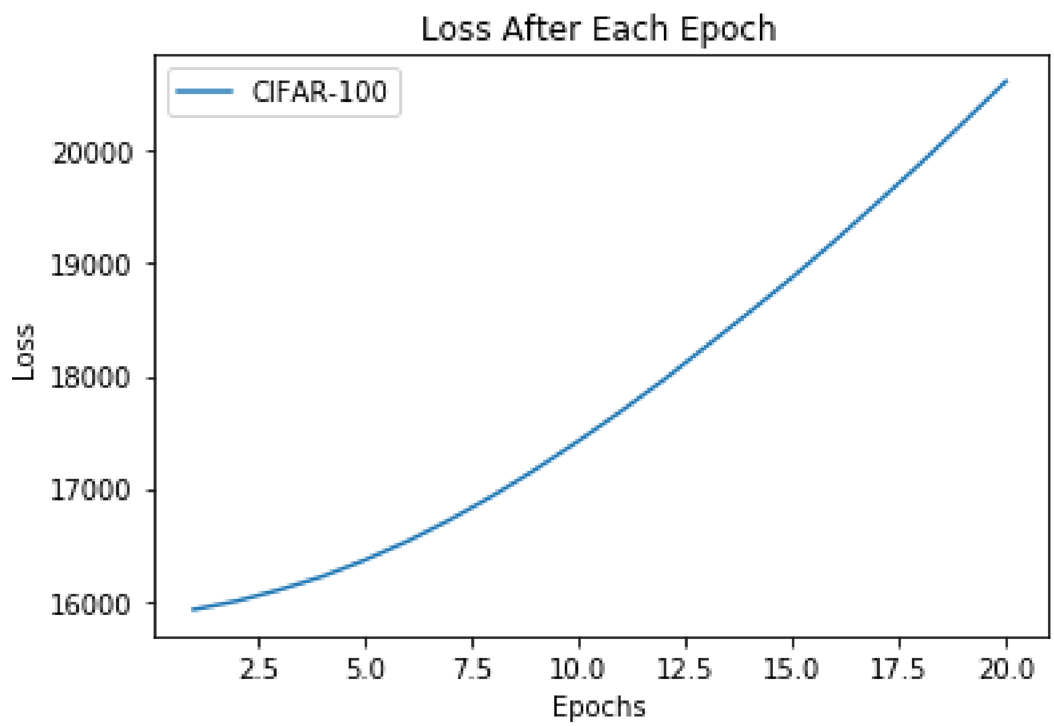
[3]:
```
plt.title("Loss After Each Epoch")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.plot(epochs, cifar10_losses, label="CIFAR-10")
plt.legend()
```

[3]: <matplotlib.legend.Legend at 0x7f1978bb6be0>



[4]:
```
plt.title("Loss After Each Epoch")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.plot(epochs, cifar100_losses, label="CIFAR-100")
plt.legend()
```

[4]: <matplotlib.legend.Legend at 0x7f1978b324a8>



1.2 Summary Tables

1.2.1 Fashion-MNIST

	Final Loss	Accuracy
Training	1785.199	0.921
Testing	1785.199	0.861

1.2.2 CIFAR-10

	Final Loss	Accuracy
Training	9781.351	0.423
Testing	9780.689	0.343

1.2.3 CIFAR-100

	Final Loss	Accuracy
Training	20619.205	0.200
Testing	20621.480	0.095

1.3 Reflection

When deriving the backpropagation equations, I used the tutorial on DIDL 4.7 as my main guide. It wasn't identical to the network we are creating in the lab, however, so I used the recorded lectures to figure out how to derive the parts of the network that are not in the tutorial, such as the biases, separate weight regularizations, and softmax/cross-entropy loss. When writing all of this by hand, I made a lot of mistakes. After seeing what the values were supposed to be from the code itself, I was able to guide myself in correcting those mistakes.

When writing the unit tests and, in conjunction, debugging the software, the biggest help was probably the shape assertions in every layer. When going through the network by hand, I was able to figure out rules for the shapes of tensors that had to be followed for the network to work properly. When shapes failed the assertions, I was more easily able to find exactly where, when, and how the error occurred. Writing the unit tests also allowed me to split the problem into different pieces, such as each layer, and make sure they worked individually before I put them together.

The training curves and final accuracy reveal that, even though the loss may be increasing, it doesn't necessarily mean that the network's training is getting worse. With all three datasets, the loss was increasing at an almost constant rate, but the accuracy for both CIFAR datasets was somewhat or significantly better than a random guess. This means that the loss isn't the only performance metric that needs to be kept track of when training.