Solution 4. Your solution goes here.
(a) The cross entropy formula takes in two distributions, $p(x)$, the true distribution, and $q(x)$, the estimated distribution, defined over the discrete variable x and is given by

$$H(p,q) = -\sum_x p(x)log(q(x))$$

For a neural network, the equation could be rewriten as

$$L = -y \cdot log(\hat{y}),$$

where y is the ground truth vector and $\hat{y}$ is the estimate.
A cost function based on multiclass log loss for data set of size N is represented as

$$J = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot log(\hat{y}_i)).$$

For CIFAR-10 dataset, it contains 60,000 32x32 color images in 10 different classes. When we start to train a new CNN neural network, all the parameters will initialize randomly according to normal distribution. Therefore, the result of Softmax for each class will be approximately 0.1 (Since there are 10 classes and the total probability is 1, and 1/10 = 0.1). In the cost function, the ground truth y gives all probability to the correct value, and the other values are zero, so we can ignore them. Therefore, the estimated cost function is

$$J = -\frac{1}{N}\sum_{i=1}^{N}(1 * log(0.1)) \approx 2.303.$$

(b) The code for running the suggested algorithm could be found in `https://colab.research.google.com/drive/1dokdPoW2KgxIN-2gbnxM13Ve-RJ8UE4N`. The weight-decay is set to 10^3, momentum parameter is set to 0.9, and η is initialized as 10^{-5}. As shown in the following figure, I find the local minimum value of η^* as 0.3.

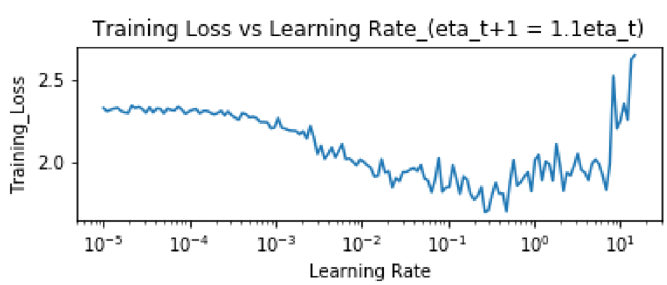
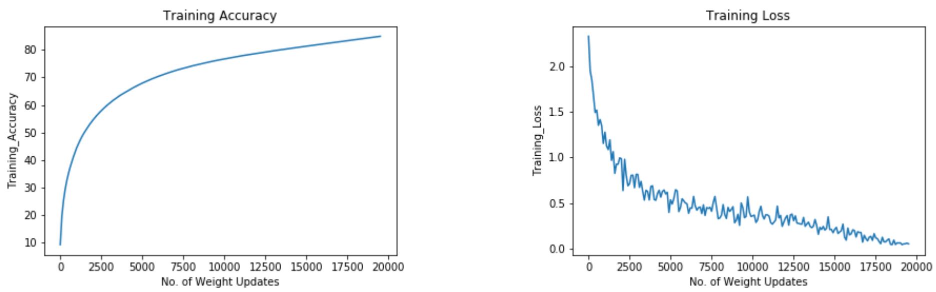


FIGURE 3. Training Loss vs Learning Rate

(c) The code for this problem could be found in `https://colab.research.google.com/drive/1U4n2eWT7Xr40G5ZLPW75F0ZSpay2cZtj`. The η_{max} is set as 0.03 as

$$\eta_{max} = \frac{1}{10} * \eta^* = \frac{1}{10} * 0.3 = 0.03$$

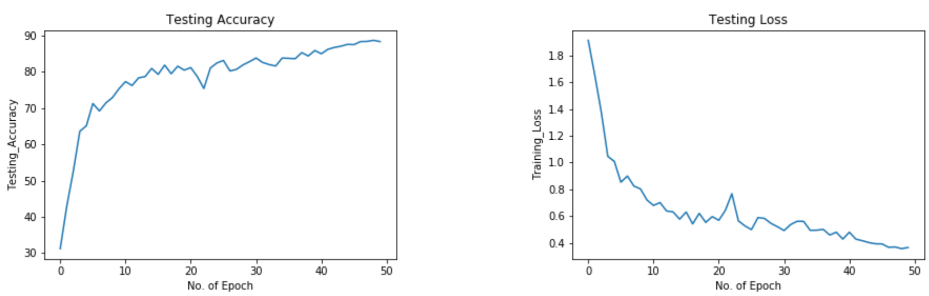
The training loss and training accuracy are shown as follows. I plot the figure per 100 weight updates.



(A) Training Accuracy (B) Training Loss

FIGURE 4. Training Accuracy and Training Loss for 4(c)

The testing loss and testing accuracy are shown as follows. I plot the figure per epoch.



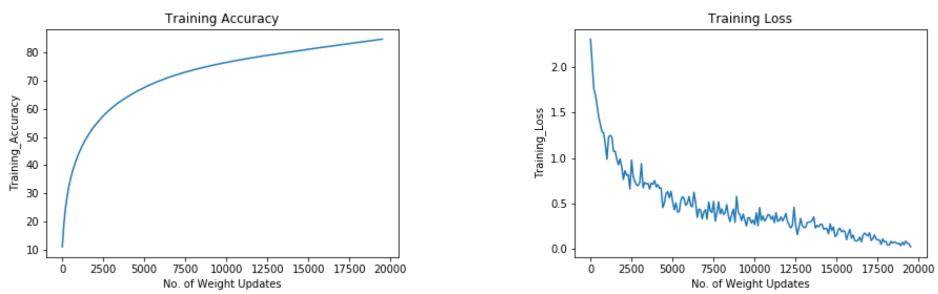
(A) Testing Accuracy (B) Testing Loss

FIGURE 5. Testing Accuracy and Training Loss for 4(c)

After training, the overall testing accuracy of the 79 batches is 88.7%.

(d) In order to validate the theorem, I run the training algorithm in 3 different setting and compare the testing accuracy. The code for running the suggested algorithm could be found in `https://colab.research.google.com/drive/1QbxOYKrPYBI0U90J7sfm1egku9E51M2Q`.

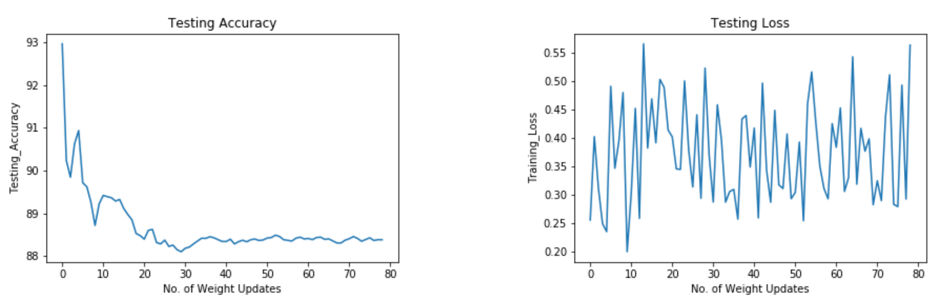
(i) $\eta_{max} = 0.03$ and momentum parameter $\rho = 0.9$ The result of this setting is shown as follows. The testing accuracy of the 79 batches is 88.7%. The training loss and training accuracy are shown as follows. I plot the figure per 100 weight updates.



(A) Training Accuracy (B) Training Loss

FIGURE 6. Training Accuracy and Training Loss for setting (i) of 4(d)

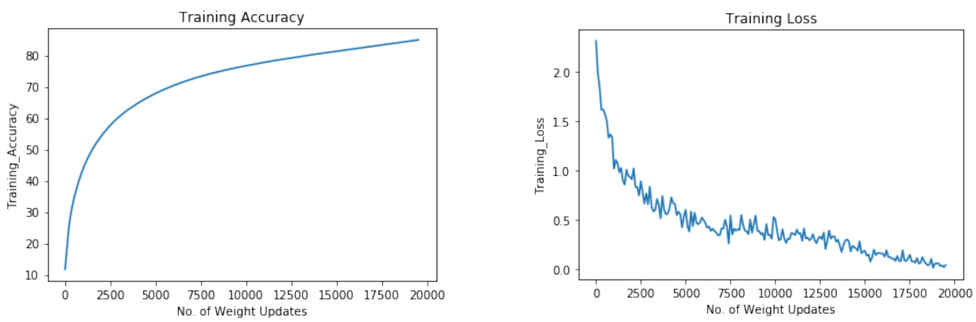
The testing loss and testing accuracy are shown as follows. I plot the figure per testing batch.



(A) Testing Accuracy (B) Testing Loss

FIGURE 7. Testing Accuracy and Testing Loss for setting (i) of 4(d)

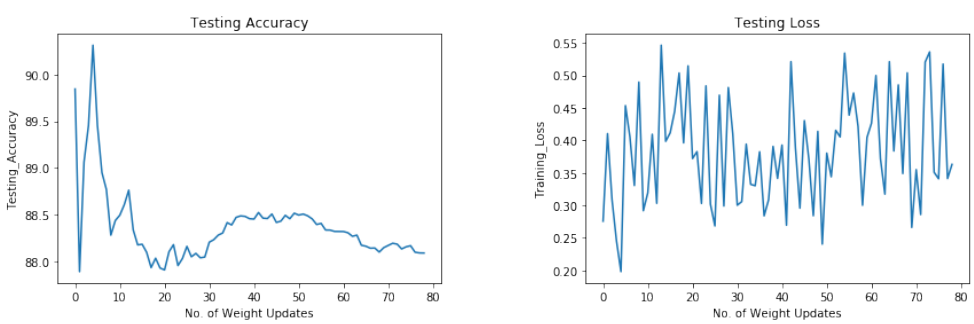
(ii)$\eta_{max} = 0.03 * 5 = 0.15$ and momentum parameter $\rho = 0.5$. The result of this setting is shown as follows. The testing accuracy of the 79 batches is 88.4%.
The training loss and training accuracy are shown as follows. I plot the figure per 100 weight updates.



(A) Training Accuracy (B) Training Loss

FIGURE 8. Training Accuracy and Training Loss for setting (ii) of 4(d)

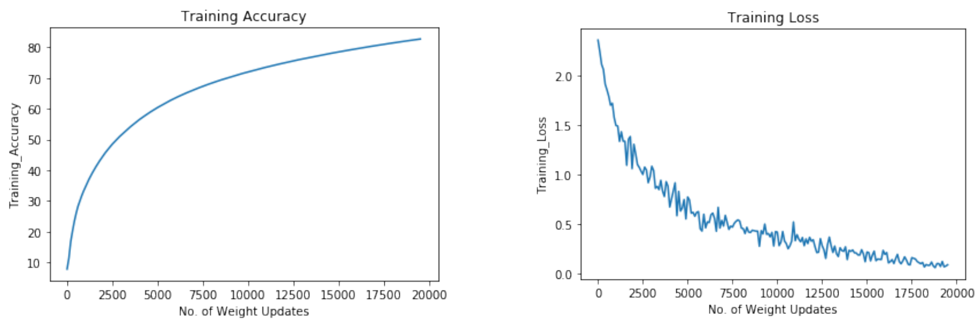
The testing loss and testing accuracy are shown as follows.



(A) Testing Accuracy (B) Testing Loss

FIGURE 9. Testing Accuracy and Testing Loss for setting (ii) of 4(d)

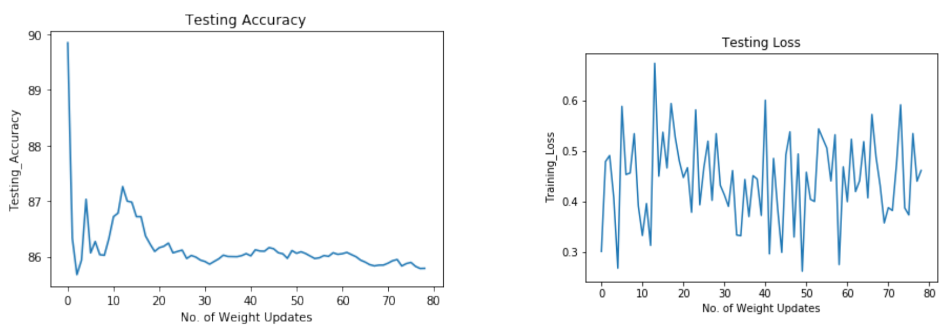
(iii)$\eta_{max} = 0.03$ and momentum parameter $\rho = 0.5$. The result of this setting is shown as follows. The testing accuracy of the 79 batches is 86.2%.
The training loss and training accuracy are shown as follows. I plot the figure per 100 weight updates.



(A) Training Accuracy (B) Training Loss

FIGURE 10. Training Accuracy and Training Loss for setting (iii) of 4(d)

The testing loss and testing accuracy are shown as follows.



(A) Testing Accuracy (B) Testing Loss

FIGURE 11. Testing Accuracy and Testing Loss for setting (iii) of 4(d)

The testing accuracy of 3 settings per testing batch is shown as follows. We could observe validation error is about the same for the first two but increases for the third setting. The data and code is included in *17999524 4d plot.zip*.

FIGURE 12. Validation Error of 3 settings