# Machine Learning, 2023 Spring
# Assignment 3

**WeiLiang Sun 2020533010**

## Notice

Plagiarizer will get 0 points.
LaTeXis highly recommended. Otherwise you should write as legibly as possible.

Requirement Provide a numerical experiment for SGD algorithm, and submit the results in a report. Both code and report need to be submit. Contents of the report:

1. Background: Introduce the purpose, background of the experiment.
2. Data acquisition: How the data is generated.
3. Algorithm: The optimization model and the applied alogrithm, especially the alogrithm parameters.
4. Experiment result for the following taskes:
   (a) SGD with fixed step length. (At least 3 different step length.)
   (b) SGD with decresing step length.
   (c) Demonstrate early termination. (If the result is not perfect, please demonstrate it in discussion.)
   (d) (optional) Compare GD and SGD under large dataset.

**Tips**

(a) The programing language is not restricted.(Python / Matlab)

(b) The optimization model should be suitable for GD/SGD algorithm. e.g. Logistic regression / Least square with about 10 variable.

(c) Choose enough training data $N.N$ should be no less than $10^3$. You can also approximate the distribution, and out of sample error with more data generated. (First genreate $\approx 10 * N$ data points, to approximate the distribution, and sample $N$ training data in it. The out of sample error can also be evluated with the approximate distribution.)

**Experiment Result**

### 1. Background

The purpose of this experiment is to train a Softmax regression model using the stochastic gradient descent (SGD) algorithm. I choose a famous dataset CIFAR-10 as the training data and also use its test data to evaluate the performance of my model.
To have a brief explanation of the dataset, CIFAR-10 is a dataset of 50000 32x32 color images in 10 classes as training data, and 10000 32x32 color images in 10 classes as test data. The 10 classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.
The goal of the experiment is to classify the images into 10 classes.

### 2. Data Acquisition

The dataset is downloaded from the official website of CIFAR-10. The training data is divided into 5 batches, each batch contains 10000 images. The test data contains 10000 images in a single test

batch.
The images are stored in binary format, and the labels are stored in a separate file. The images are stored in the order of the labels. I used the "unpickle" function to convert the binary data into grayscale images and applying the Histogram of Oriented Gradients (HOG) feature extraction method to extract the features of the images. The HOG features are used as the input of the Softmax regression model.

### 3. Algorithm

The Softmax regression model is a linear classifier that uses the Softmax function to predict the probability of each class. And the SGD algorithm is used to optimize the model parameters by updating the weights in the direction of the negative gradient of the loss function.

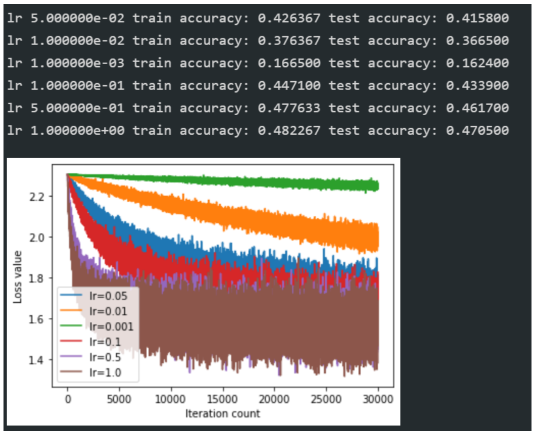
In my experiment, the SGD algorithm is implemented in the train function of the Softmax regression model. The function takes the input data and labels, as well as the hyperparameters and returns the previous loss of each epoch. I also include the parameters to control the learning rate decay and early stop to improve the performance of the algorithm. My explanation of the parameters are as follows:

(1) Learning rate:
The learning rate should be chosen based on the specific dataset. In my experiment I use the dataset CIFAR-10, and the learning rate of this dataset can't be too small or it will lead to bad performance. So to have a more detailed comparison I choose some different learning rate and these will be introduced in the following part.
(2) Regularization parameter:
It is known as a parameter called reg which is chosen to prevent overfitting. I choose a small Regularization parameter.
(3) Batch size:
The batch size determines the number of samples used in each iteration of the SGD algorithm. A large batch size will lead to a more stable gradient, but it will also lead to a slower convergence. A smaller batch size will lead to a faster convergence, but it will also lead to a more unstable gradient. In my experiment, I choose a batch size of 200 to balance stability and efficiency.
(4) Number of iterations:
The number of iterations determines the number of times the SGD algorithm will be run. In my experiment, because the complexity of the dataset CIFAR-10 is high, so I choose a bigger number of iterations to ensure the convergence of the algorithm. When the iteration is small I find the performance terrible and it can't converge well, so I choose a big value 30000.
(5) Learning rate decay:
To satisfy the Requirement and also have a comparison between the fixed learning rate and the dacaying ones, I add a bool parameter called decay in my training function, and it is set to false initially. When it is true, we will let the learning rate decay by multiplying a decay rate to the learning rate. Otherwise when it is false, we only use the fixed learning rate as usual.
(6) Early stop:
The early stop parameter is same as the learning rate decay and is set to false initially. When it is true, we will stop the training process when the abstract value of the loss function is less than a threshold. I set the threshold to 1e-6. Otherwise when it is false, we will train the model until the number of iterations is reached.

### 4. Experiment Result

#### 4.1. SGD with fixed step length

I choose six different learning rates to compare the performance of the SGD algorithm. The learning rates are 5e-2, 1e-2, 1e-3, 0.1, 0.5, 1.0. The reason why I choose these learning rate is that since the dataset of CIFAR-10 needs larger learning rate, so I choose the 0.1, 0.5, 1.0 as the learning rate to compare the performance of the SGD. Meanwhile, to have a better comparison, I also choose some smaller learning rate to compare the performance of the SGD such as 5e-2, 1e-2, 1e-3.

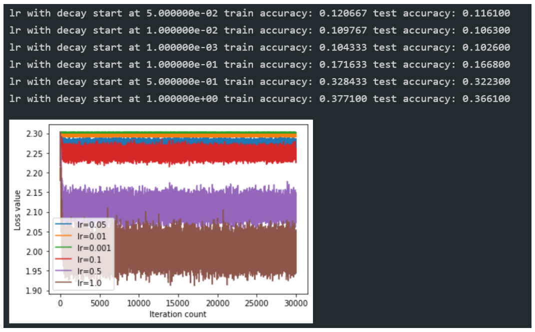


```
lr 5.000000e-02 train accuracy: 0.426367 test accuracy: 0.415800
lr 1.000000e-02 train accuracy: 0.376367 test accuracy: 0.366500
lr 1.000000e-03 train accuracy: 0.166500 test accuracy: 0.162400
lr 1.000000e-01 train accuracy: 0.447100 test accuracy: 0.433900
lr 5.000000e-01 train accuracy: 0.477633 test accuracy: 0.461700
lr 1.000000e+00 train accuracy: 0.482267 test accuracy: 0.470500
```

From the result picture we can see that when the learning rate is larger, its loss will be small, which proves what I have said before. A larger learning rate on the CIFAR-10 dataset can be better. When it is small as the green line, the performance seems to be terrible.

### 4.2. SGD with decreasing step length

I choose the same learning rate as the previous part, and set the decay parameter to true. The decay rate is 0.95. The result is as follows:

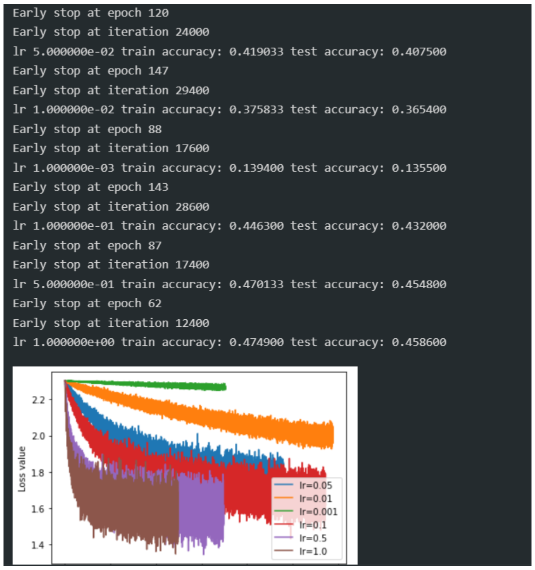


```
lr with decay start at 5.000000e-02 train accuracy: 0.120667 test accuracy: 0.116100
lr with decay start at 1.000000e-02 train accuracy: 0.109767 test accuracy: 0.106300
lr with decay start at 1.000000e-03 train accuracy: 0.104333 test accuracy: 0.102600
lr with decay start at 1.000000e-01 train accuracy: 0.171633 test accuracy: 0.166800
lr with decay start at 5.000000e-01 train accuracy: 0.328433 test accuracy: 0.322300
lr with decay start at 1.000000e+00 train accuracy: 0.377100 test accuracy: 0.366100
```

From the result picture we can see that when the learning rate is larger, its loss will be small, and with the decay learning rate, its performance becomes worse. It is because that our learning rate is set to a small one and the biggest one is 1.0 which I think is not enough large for the dataset of CIFAR-10, so in my experiment when the learning rate becomes smaller using decay technique, its performance becomes worse.

### 4.3. Early termination

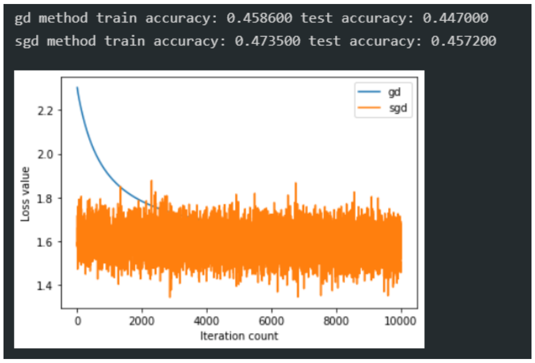
When the abstract value of the difference between the present loss and the previous loss is smaller than a fixed value 1e-6, we will stop the iterations. And our result is as follows:



```
Early stop at epoch 120
Early stop at iteration 24000
lr 5.000000e-02 train accuracy: 0.419033 test accuracy: 0.407500
Early stop at epoch 147
Early stop at iteration 29400
lr 1.000000e-02 train accuracy: 0.375833 test accuracy: 0.365400
Early stop at epoch 88
Early stop at iteration 17600
lr 1.000000e-03 train accuracy: 0.139400 test accuracy: 0.135500
Early stop at epoch 143
Early stop at iteration 28600
lr 1.000000e-01 train accuracy: 0.446300 test accuracy: 0.432000
Early stop at epoch 87
Early stop at iteration 17400
lr 5.000000e-01 train accuracy: 0.470133 test accuracy: 0.454800
Early stop at epoch 62
Early stop at iteration 12400
lr 1.000000e+00 train accuracy: 0.474900 test accuracy: 0.458600
```

I find the result seems good enough. The early termination means preventing the overfitting which is displayed in the picture as the plane line. So the line will stop when the line becomes plane.

**4.4. Comparison between GD and SGD algorithm** Since our dataset CIFAR-10 is already large enough, we don't need to change the dataset and we only need to do our comparison based on our dataset. The result is as follows:



```
gd method train accuracy: 0.458600 test accuracy: 0.447000
sgd method train accuracy: 0.473500 test accuracy: 0.457200
```

From the result picture we can see that the SGD algorithm is better than the GD algorithm. And with the iterations increasing, the performance of the GD algorithm will be better and better and more stable, approaching the performance of SGD algorithm.

**5. Conclusion** In this experiment, we find that the SGD algorithm is better than the GD algorithm. However, There is something we need to discuss. We find that when the learning rate is small, the performance is worse than large learning rate. I think it is because the iteration I set which is 30000 is still too small to let the model converge. Because the dataset of CIFAR-10 is too complex, so it should need further more iterations to make the model converge. When the learning rate is large, the 30000 should be enough, so it get a better performance than the small ones.