

The Experiment Report of

Deep Learning

**College Software College**

**Subject Software Engineering**

**Members**   **陈奕男**

**Student ID 201710106543**

**E-mail easyhard@qq.com**

**Tutor**   **Mingkui Tan**

**Date submitted** **2017.12.15**

# Topic: Logistic Regression, Linear Classification and Stochastic Gradient Descent

# Time: 2017-12-15 12:00 AM

# Reporter: 陈奕男

# Purposes:

1. Compare and understand the difference between gradient descent and stochastic gradient descent.
2. Compare and understand the differences and relationships between Logistic regression and linear classification.
3. Further understand the principles of SVM and practice on larger data.

# Data sets and data analysis:

Experiment uses [a9a](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a) of [LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/), including 32561/16281(testing) samples and each sample has 123/123 (testing) features. Please download the training set and validation set.

# Experimental steps:

The experimental code and drawing are completed on jupyter.

*Logistic Regression and Stochastic Gradient Descent*

1. Load the training set and validation set.
2. Initalize logistic regression model parameters, you can consider initalizing zeros, random numbers or normal distribution.
3. Select the loss function and calculate its derivation, find more detail in PPT.
4. Calculate gradient  toward loss function from **partial samples**.
5. **Update model parameters using different optimized methods(NAG，RMSProp，AdaDelta and Adam).**
6. Select the appropriate threshold, mark the sample whose predict scores **greater than the threshold as positive, on the contrary as negative**. Predict under validation set and get the different optimized method loss ，， and .
7. Repeate step 4 to 6 for several times, and **drawing graph of ，，and  with the number of iterations**.

*Linear Classification and Stochastic Gradient Descent*

1. Load the training set and validation set.
2. Initalize SVM model parameters, you can consider initalizing zeros, random numbers or normal distribution.
3. Select the loss function and calculate its derivation, find more detail in PPT.
4. Calculate gradient  toward loss function from **partial samples**.
5. **Update model parameters using different optimized methods(NAG，RMSProp，AdaDelta and Adam).**
6. Select the appropriate threshold, mark the sample whose predict scores **greater than the threshold as positive, on the contrary as negative**. Predict under validation set and get the different optimized method loss ，， and .
7. Repeate step 4 to 6 for several times, and **drawing graph of ，，and  with the number of iterations**.

# Code:

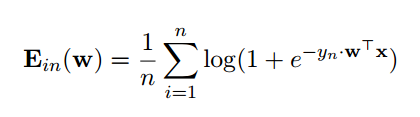
**(see ClassificationExperiment.ipynb and RegressionExperiment.ipynb on github:** <https://github.com/easyhard007/ML2017-lab-02.git>**)**

(Fill in the contents of 8-12 respectively for linear regression and linear classification)

Logistic Regression

# Selection of validation (hold-out, cross-validation, k-folds cross-validation, etc.):

Use loss function on validation set to validate:

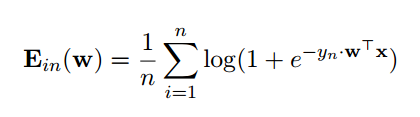


# The initialization method of model parameters:

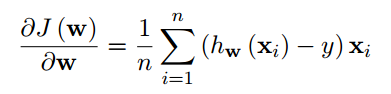
All-zero Initialization

# The selected loss function and its derivatives:

loss function:



Derivatives:



# Experimental results and curve:

## Hyper-parameter selection (η, epoch, etc.):

eta = 0.02 # Learning Rate η

iter = 4000 # Iteration times

epsilon = 0.00001 #using in 4 optimalization methods to prevent the denominator become 0

mini\_batch\_percent = 0.2 #using 20% of data in gradient descent

beta1 = 0.9 #used in adam

beta2 = 0.999 #used in adam

mu = 0.9 #used in NAG

## Predicted Results (Best Results):

Use an iteration of 4000 and eta=0.02

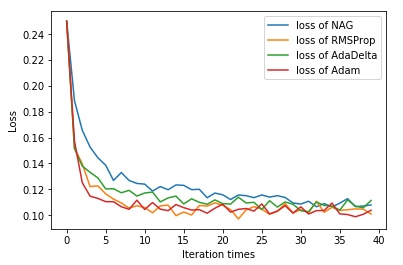
Loss on validation datasets of LNAG converge to around: 0.1197

Loss on validation datasets of LRMSProp converge to around: 0.1046

Loss on validation datasets of LAdaDelta converge to around: 0.1139

Loss on validation datasets of LAdam converge to around: 0.1016

## Loss curve of LNAG, LRMSProp, LAdaDelta, LAdam: (Iteration times \* 10)



# Results analysis:

The results of the experiment is consistent with expected. The loss curve descent down like “J”. For using mini batch gradient descent, the curves still shakes after it converged.

The speed of converge for 4 method in my experiment is: NAG < AdaDelta < RMSProp < Adam. But since they are using same hyper-parameters like eta and epsilon, this comparing seemed not perfect.

Linear Classification

# Selection of validation (hold-out, cross-validation, k-folds cross-validation, etc.):

Use loss function and accuracy rate on validation set to validate:

# The initialization method of model parameters:

All-zero Initialization

# The selected loss function and its derivatives:

loss function:

Derivatives:

# Experimental results and curve:

## Hyper-parameter selection (η, epoch, etc.):

eta = 0.001 # Learning Rate η

epsilon = 0.00001

iter = 50 \* 1000 # Iteration times

mini\_batch\_percent = 0.05 #5% of dataset is used in loss function

#used in adam

beta1 = 0.9

beta2 = 0.999

eta = 0.1

#used in NAG

miu = 0.9

## Predicted Results (Best Results):

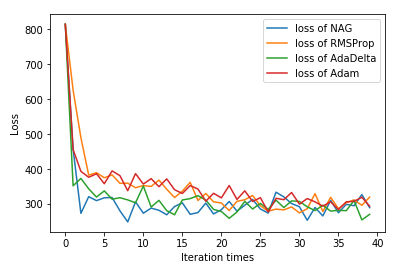
Loss on validation datasets of LNAG converge to around: 290.41

Loss on validation datasets of LRMSProp converge to around: 286.89

Loss on validation datasets of LAdaDelta converge to around: 281.51

Loss on validation datasets of LAdam converge to around: 306.54

## Loss curve of LNAG, LRMSProp, LAdaDelta, LAdam: (Iteration times \* 10)



# Results analysis:

The results of the experiment is consistent with expected. The loss curve descent down like “J”.

# Similarities and differences between logistic regression and linear classification:

Similarities: The procedure of the two experiment is similar, including the building-up of Loss Function, the iteration and gradient descent of the loss function.

Differences:

1. The loss function and derived function chosen for two experiments are different.

# Summary:

Through this project, we mastered the basic steps of linear regression and linear classification, and also have a deeper understand of machine learning. The most interesting things is that, during the logistic regression or linear classification, we don’t need to know what the meaning of the data-set is, what the meaning of each feature is. All we need is to throw the data set in, and get the best weight of features.