华南理工大学

South China University of Technology

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

1. **Topic: Logistic Regression, Linear Classification and Stochastic Gradient Descent**
2. **Time: 2017-12-15 12:00 AM**
3. **Reporter: 陈奕男**
4. **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.

5. **Data sets and data analysis:**

Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features. Please download the training set and validation set.

6. **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**.

7. 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

8. Selection of validation (hold-out, cross-validation, k-folds

cross-validation, etc.):

Use loss function on validation set to validate:

$$\mathbf{E}_{in}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\log(1 + e^{-y_n \cdot \mathbf{w}^\top \mathbf{x}})$$

9. The initialization method of model parameters:

All-zero Initialization

10. The selected loss function and its derivatives:

loss function:

$$\mathbf{E}_{in}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\log(1 + e^{-y_n \cdot \mathbf{w}^\top \mathbf{x}})$$

Derivatives:

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{n}\sum_{i=1}^{n}\left(h_{\mathbf{w}}(\mathbf{x}_i) - y\right)\mathbf{x}_i$$

11. Experimental results and curve:

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

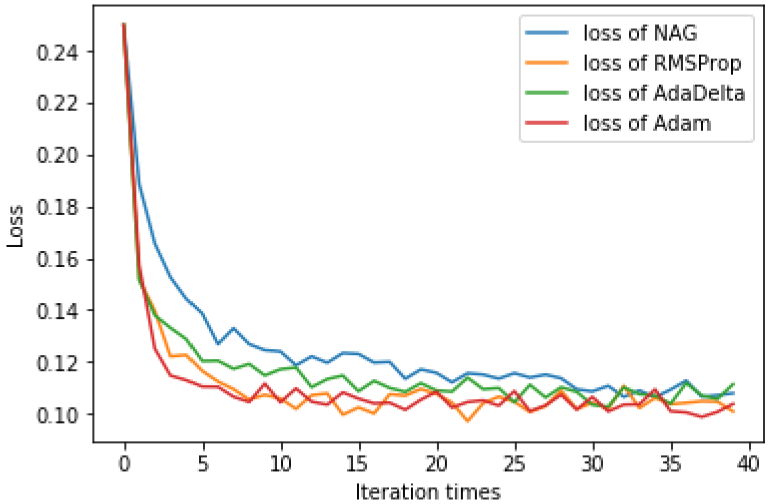
eta = 0.02 # Learning Rate η
iter = 4000 # Iteration times
epsilon = 0.00001 #using in 4 optimalization methods to prevent the denominator become 0
mini_batch_percent = 0.2 #using 20% of data in gradient descent
beta1 = 0.9 #used in adam
beta2 = 0.999 #used in adam
mu = 0.9 #used in NAG

Predicted Results (Best Results):

Use an iteration of 4000 and eta=0.02
Loss on validation datasets of L_{NAG} converge to around: 0.1197

Loss on validation datasets of $L_{RMSProp}$ converge to around: 0.1046
Loss on validation datasets of $L_{AdaDelta}$ converge to around: 0.1139
Loss on validation datasets of L_{Adam} converge to around: 0.1016

Loss curve of L_{NAG}, $L_{RMSProp}$, $L_{AdaDelta}$, L_{Adam}: (Iteration times * 10)



12. 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

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

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

$$L = \sum_{i}^{m} \left(1 - y^{(i)} \cdot \omega X^{(i)}\right) + \frac{1}{2}\lambda\omega^2$$

9. The initialization method of model parameters:

All-zero Initialization

10. The selected loss function and its derivatives:

loss function:

$$L = \sum_{i}^{m} \left(1 - y^{(i)} \cdot \omega X^{(i)}\right) + \frac{1}{2} \lambda \omega^2$$

Derivatives:

$$\nabla L_{\omega} = |\lambda \omega| - y^{(i)} \cdot X^{(i)}$$

11. 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

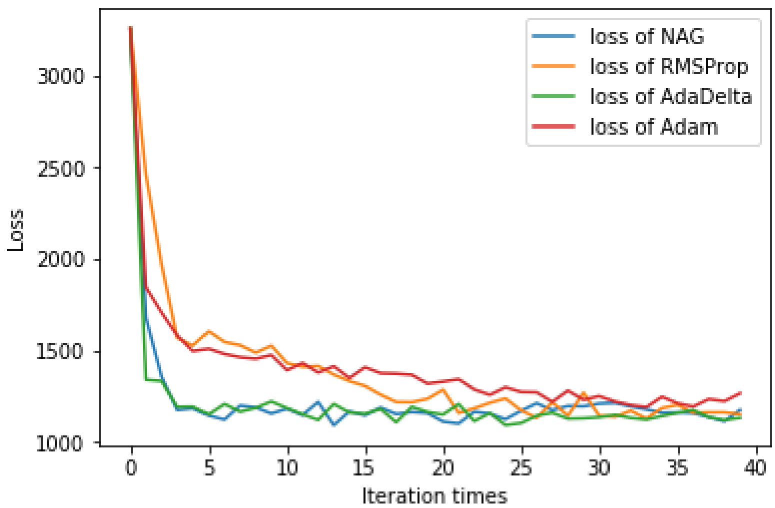
#used in adam
beta1 = 0.9
beta2 = 0.999
eta = 0.1

#used in NAG
mu = 0.9

Predicted Results (Best Results):

Loss on validation datasets of L_{NAG} converge to around: 1163.03
Loss on validation datasets of $L_{RMSProp}$ converge to around: 1172.39
Loss on validation datasets of $L_{AdaDelta}$ converge to around: 1143.30
Loss on validation datasets of L_{Adam} converge to around: 1221.94

Loss curve of L_{NAG}, $L_{RMSProp}$, $L_{AdaDelta}$, L_{Adam}: (Iteration times * 10)



12. Results analysis:

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

In this experiment, NAG and AdaDelta converges more faster than the other 2. The speed of converge for 4 method in my experiment is: Adam < RMSProp < NAG < AdaDelta. But since they are using same hyper-parameters like eta and epsilon, this comparing seemed not perfect.

13. 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.
2. The result of the two experiments is different , since the speed of 4 methods varies and different methods gets the best in the two experiments.

14. 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.

Different optimization methods is used in the two experiments, but it seemed that we cannot choose a “best” one, for they represent differently on different hyper-parameters.