华南理工大学

South China University of Technology

The Experiment Report of *Deep Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Qichen Huang

Supervisor:
Mingkui Tan

Student ID:
201920142806

Grade:
Graduate

November 9, 2019

Logistic Regression and Support Vector Machine

***Abstract*—Linear classification is to find a hyperplane that separate different classes data properly. In this experiment, we attempt to solve linear classification problem with Logistic Regression and Support Vector Machine, comparing their similarity and difference, figuring out their theory and implementation details. The final outputs of them give out a satisfying result.**

I. Introduction

CLASSIFICATION is one of the most common problem in Machine Learning, which is to make up function mapping from specific features to a class label. Linear Classification, a kind of binary classification, is the simplest classification problem. It aims to find a hyperplane in feature space that properly separate different classes data. In this experiment, we focus on linear classification problem, conquering it with Logistic Regression and Support Vector Machine. Both of them are classical solution to deal with Linear Classification. Our motivation is to 1) compare and understand the difference between gradient descent and batch random 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. We carry out this experiment on a9a data in LIBSVM.

II. Methods and Theory

The target of Linear Classification is to find a hyperplane or a linear equation that correctly separate different classes data. The data dropping in different sides of hyperplane would be marked as different classes label.

A. Logistic Regression

Logistic Regression(LR) actually utilizes Linear Regression function $y = \boldsymbol{w}^T\boldsymbol{x}$ to fit log odds $\ln \frac{y}{1-y}$ of data, where y and $1-y$ are the probability that it belongs to positive and negative class separately. In other words, LR attempts to compute the probability of positive class in the form of $y = \frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x})}}$. Thus we assume that a data with positive label should have high probability, as close to 1 as possible, otherwise have low probability, as close to 0 as possible.

Considering positive and negative labels are marked as +1 and -1, we can define a loss function $\mathcal{L} = \frac{1}{n}\sum_i \ln\left(1 + e^{-y_i\boldsymbol{w}^T\boldsymbol{x}_i}\right)$, where n is the number of samples. Now that the problem is simplified as minimization of loss function, in this experiment, we try to solve it by mini-batch stochastic gradient descent. Therefore we can compute the gradient of loss function with respect to $\boldsymbol{w}$ as $\mathcal{G} = -\frac{1}{n}\sum_i \frac{\boldsymbol{x}_i y_i}{1+e^{y_i\boldsymbol{w}^T\boldsymbol{x}_i}}$ and update $w = w - \eta\mathcal{G}$ at each iteration, where η is the learning rate.

B. Support Vector Machine

Same as LR, Support Vector Machine(SVM) also separate data with a hyperplane. However, SVM concentrates on maximize the margin of support vector instead of the probability of positive class. Support vector is the nearest sample data points from hyperplane that makes

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_i = +1, & y_i = +1 \\ \boldsymbol{w}^T\boldsymbol{x}_i = -1, & y_i = -1 \end{cases}$$

satisfied and margin is the sum of distance between different classes support vector and hyperplane, which is $\gamma = \frac{2}{||\boldsymbol{w}'||}$, where $\boldsymbol{w}'$ is feature weight vector without bias b. At the same time, a correctly separated sample should satisfy the condition that $y_i(\boldsymbol{w}^T\boldsymbol{x}_i) \geqslant 1$.

For the noise of data, we should permit some data unsatisfied the above condition, but as less as possible. So we utilize hinge loss function $l_{hinge}(z) = \max(0, 1-z)$ to design our loss function of SVM as $\mathcal{L} = \frac{||\boldsymbol{w}'||^2}{2} + \frac{C}{n}\sum_i \max(0, 1 - y_i\boldsymbol{w}^T\boldsymbol{x}_i)$. Then the gradient of loss function for i th data is computed as

$$\mathcal{G}_i = \begin{cases} \boldsymbol{w} - Cy_i\boldsymbol{x}_i & 1 - y_i\boldsymbol{w}^T\boldsymbol{x}_i > 0 \\ \boldsymbol{w} & 1 - y_i\boldsymbol{w}^T\boldsymbol{x}_i \leqslant 0 \end{cases},$$

where w is rectified by replacing the end element, bias b, with 0.

III. Experiments

A. Dataset

Experiment uses a9a of LIBSVM Data, including 32561 training samples and 16281 testing samples and each sample has 123 features.

B. Implementation

For Logistic Regression, the experiment steps are as follows:

1) Load the training set and validation set.
2) Initialize logistic regression model parameter randomly.
3) Select the loss function and calculate its derivation.
4) Determine the size of the batch_size and randomly take some samples,calculate gradient $\mathcal{G}$ toward loss function from partial samples.
5) Use the SGD optimization method to update the parametric model.
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 loss $\mathcal{L}_{validation}$.
7) Repeat step 4 to 6 for several times, and drawing graph of $\mathcal{L}_{validation}$ with the number of iterations.

In the experiment, parameter w is initialized randomly, and batch size is assigned with 100,training step with 10000, learning rate with 0.01.

TABLE I: Final Accuracy

Accuracy	training set	validation set
Linear Regression	0.833912	0.836681
Support Vector Machine	0.759190	0.763774

Final result of LR is presented in Table I. During training process, loss value and accuracy are shown in Fig.1 and Fig.2 separately.

For Support Vector Machine, the experiment steps are as follows:

1) Load the training set and validation set.
2) Initialize SVM model parameters randomly.
3) Select the loss function and calculate its derivation.
4) Determine the size of the batch_size and randomly take some samples,calculate gradient G toward loss function from partial samples.
5) Use the SGD optimization method to update the parametric model.
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 loss $\mathcal{L}_{validation}$.
7) Repeat step 4 to 6 for several times, and drawing graph of $\mathcal{L}_{validation}$ with the number of iterations.

In the experiment, parameter w is initialized randomly, and batch size is assigned with 100,training step with 5000, learning rate with 0.01.

Final result of SVM is presented in Table I. During training process, loss value and accuracy are shown in Fig.3 and Fig.4 separately.

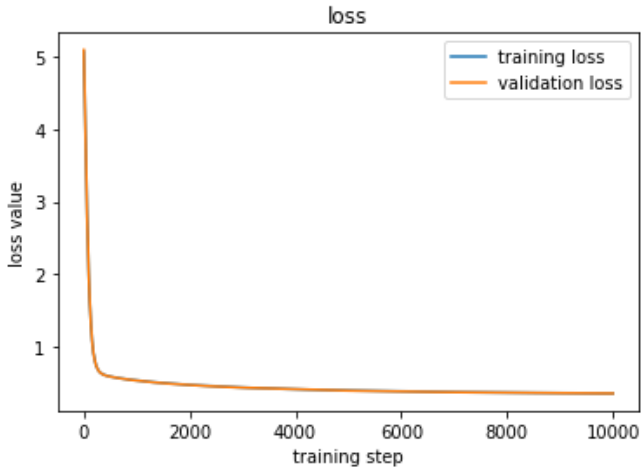

Fig. 1: Loss of Linear Regression

IV. Conclusion

In this experiment, we explored linear classification problem with Logistic Regression and Support Vector Machine. Both of them aim to find a hyperplane that properly separate different classes data, but concentrate on different way, one on probability and the other on margin of support vector. From the final Accuracy, we can tell that LR is much better than SVM, maybe because the margin term influences the performance of SVM.

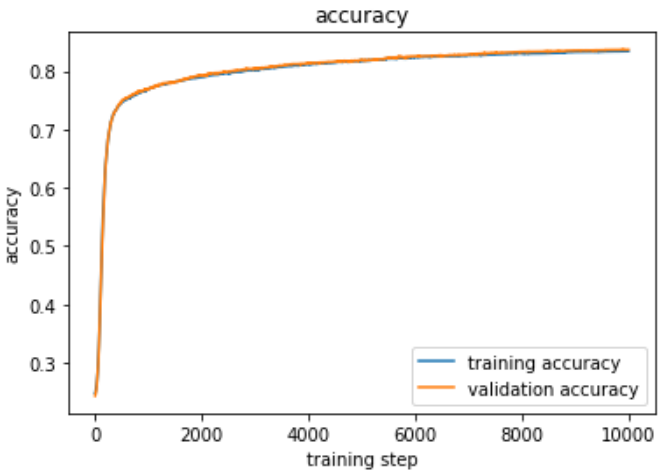

Fig. 2: Accuracy of Linear Regression

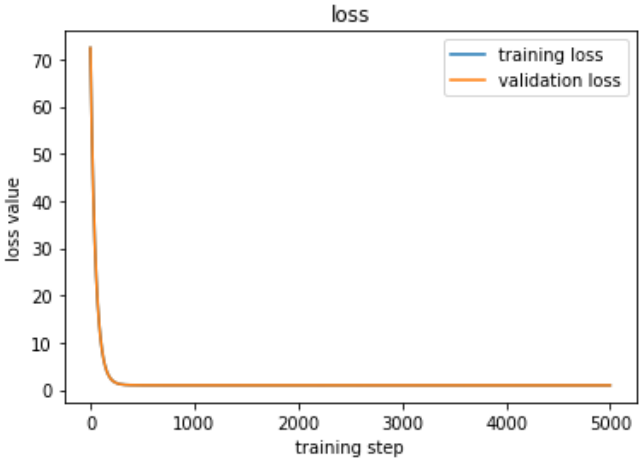

Fig. 3: Loss of Support Vector Machine

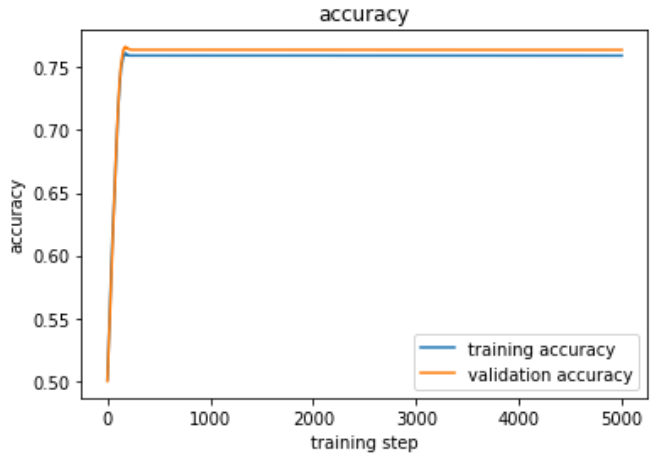

Fig. 4: Accuracy of Support Vector Machine