The Experiment Report of Machine Learning

SCHOOL: SCUT

SUBJECT: SOFTWARE ENGINEERING

Author:
夏俊煊

Supervisor:
Mingkui Tan

Student ID：
201530613139

Grade:
Undergraduate

December 9, 2017

Linear Regression, Linear Classification and Gradient Descent

Abstract—
1. Compare the difference and connection between the gradient descent and the stochastic gradient descent.
2. Compare and understand the differences and connections between logistic regression and linear classification.
3. Further understand the principle of SVM and practice it on larger data.

I. INTRODUCTION

Logical regression and random gradient descent

Read the experimental training set and the validation set.
The parameter initialization of the logistic regression model can consider all zero initialization, random initialization or normal distribution initialization.
Select the Loss function and seek guidance for it.
The gradient of a partial sample to the Loss function is obtained.
Update the model parameters using different optimization methods (NAG, RMSProp, AdaDelta, and Adam).
Choosing the appropriate threshold, we will verify that the mark of the concentrated calculation is more than the threshold, and vice versa. The Loss function valuesL_{NAG}， $L_{RMSProp}$， $L_{AdaDelta}$和L_{Adam} are tested on the validation set and obtained with different optimization methods.
Repeat step 4-6 several times and draw a change diagram of L_{NAG}， $L_{RMSProp}$， $L_{AdaDelta}$和L_{Adam} with the number of iterations.

Linear classification and random gradient descent

Read the experimental training set and the validation set.
The support vector machine model parameter initialization can consider all zero initialization, random initialization or normal distribution initialization.
Select the Loss function and seek guidance for it.
The gradient of a partial sample to the Loss function is obtained.
Update the model parameters using different optimization methods (NAG, RMSProp, AdaDelta, and Adam).
Choosing the appropriate threshold, we will verify that the mark of the concentrated calculation is more than the threshold, and vice versa. The Loss function values L_{NAG}， $L_{RMSProp}$， $L_{AdaDelta}$和L_{Adam} are tested on the validation set and obtained with different optimization methods.
Repeat step 4-6 several times and draw a change diagram of L_{NAG}， $L_{RMSProp}$， $L_{AdaDelta}$和L_{Adam} with the number of iterations.

II. METHODS AND THEORY

Logistic Regression:

The labels are binary: $y_i \in \{0,1\}$

$$h_w(x) = g(w^T x) = \frac{1}{1+e^{-w^T x}}$$

Probability:

$$P = \begin{cases} h_w(x) & y_i = 1 \\ 1 - h_w(x) & y_i = 0 \end{cases}$$

$$\begin{aligned} \max \prod_{i=1}^{n} P(y_i|\mathbf{x}_i) &\Leftrightarrow \max \log \left(\prod_{i=1}^{n} P(y_i|\mathbf{x}_i) \right) \\ &\equiv \max \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i) \\ &\Leftrightarrow \min -\frac{1}{n} \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i) \end{aligned}$$

$$P(y_i|\mathbf{x}_i) = h_{\mathbf{w}}(\mathbf{x}_i)^{y_i} \cdot (1 - h_{\mathbf{w}}(\mathbf{x}_i))^{(1-y_i)}$$

$$J(\mathbf{w}) = -\frac{1}{n}\left[\sum_{i=1}^{n} y_i \log\, h_{\mathbf{w}}(\mathbf{x}_i) + (1-y_i)\log\,(1-h_{\mathbf{w}}(\mathbf{x}_i))\right]$$

For all samples:

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

$$\mathbf{w} := \mathbf{w} - \frac{1}{n}\sum_{i=1}^{n} \alpha \left(h_{\mathbf{w}}(\mathbf{x}_i) - y_i\right)\mathbf{x}_i$$

Linear Classification:

The hinge loss is $\xi_i = \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))$

Let $g_{\mathbf{w}}(\mathbf{x}_i) = \frac{\partial \xi_i}{\partial \mathbf{w}}$

if $1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) >= 0$:

$$\begin{aligned} g_{\mathbf{w}}(\mathbf{x}_i) &= \frac{\partial(-y_i(\mathbf{w}^\top \mathbf{x}_i + b))}{\partial \mathbf{w}} \\ &= -\frac{\partial(y_i \mathbf{w}^\top \mathbf{x}_i)}{\partial \mathbf{w}} \\ &= -y_i \mathbf{x}_i \end{aligned}$$

if $1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 0$:

$$g_{\mathbf{w}}(\mathbf{x}_i) = 0$$

so we have:

$$g_{\mathbf{w}}(\mathbf{x}_i) = \begin{cases} -y_i\mathbf{x}_i & 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) >= 0 \\ 0 & 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 0 \end{cases}$$

Let $g_b(\mathbf{x}_i) = \frac{\partial \xi_i}{\partial b}$

$$g_b(\mathbf{x}_i) = \begin{cases} -y_i & 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) >= 0 \\ 0 & 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 0 \end{cases}$$

Optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad L(\mathbf{w}, b) &= \frac{\|\mathbf{w}\|^2}{2} + \frac{C}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\|\mathbf{w}\|^2}{2} + C\max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))\right) \\ &= \frac{1}{n}\sum_{i=1}^{n} L_i(\mathbf{w}, b) \end{aligned}$$

So we have:

$$\nabla_{\mathbf{w}} L_i(\mathbf{w}, b) = \mathbf{w} + C g_{\mathbf{w}}(\mathbf{x}_i)$$
$$\nabla_b L_i(\mathbf{w}, b) = C g_b(\mathbf{x}_i)$$

Different algorithms:

SGD:

$$\mathbf{g}_t \leftarrow \nabla J_i(\boldsymbol{\theta}_{t-1})$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta \mathbf{g}_t$$

η=0.01

NAG:

$$\mathbf{g}_t \leftarrow \nabla J(\boldsymbol{\theta}_{t-1} - \gamma \mathbf{v}_{t-1})$$
$$\mathbf{v}_t \leftarrow \gamma \mathbf{v}_{t-1} + \eta \mathbf{g}_t$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \mathbf{v}_t$$

η=0.01 $\gamma = 0.9$

RMSProp:

$$\mathbf{g}_t \leftarrow \nabla J(\boldsymbol{\theta}_{t-1})$$
$$G_t \leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot \mathbf{g}_t$$

η=0.001 $\gamma = 0.9$ $\epsilon = 10^{-8}$

AdaDelta:

$$\mathbf{g}_t \leftarrow \nabla J(\boldsymbol{\theta}_{t-1})$$
$$G_t \leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t$$
$$\Delta\boldsymbol{\theta}_t \leftarrow -\frac{\sqrt{\Delta_{t-1} + \epsilon}}{\sqrt{G_t + \epsilon}} \odot \mathbf{g}_t$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} + \Delta\boldsymbol{\theta}_t$$
$$\Delta_t \leftarrow \gamma \Delta_{t-1} + (1-\gamma)\Delta\boldsymbol{\theta}_t \odot \Delta\boldsymbol{\theta}_t$$

$\gamma = 0.9$ $\epsilon = 10^{-8}$

Adam:

$$\mathbf{g}_t \leftarrow \nabla J(\boldsymbol{\theta}_{t-1})$$
$$\mathbf{m}_t \leftarrow \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$$
$$G_t \leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t$$
$$\alpha \leftarrow \eta \frac{\sqrt{1-\gamma^t}}{1-\beta^t}$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha \frac{\mathbf{m}_t}{\sqrt{G_t + \epsilon}}$$

$\gamma = 0.999$ $\beta = 0.9$ $\epsilon = 10^{-8}$

III. EXPERIMENT

Data set:

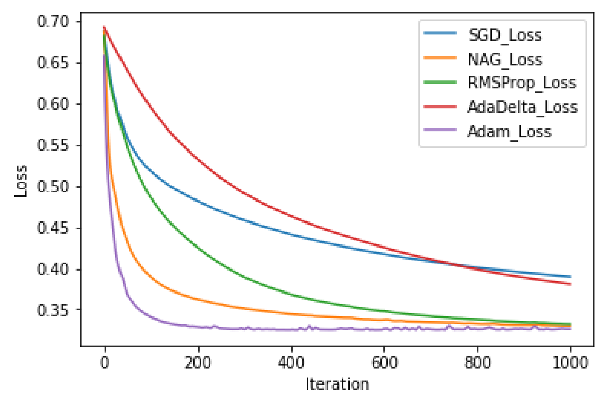
The experiment uses the a9a data in LIBSVM Data, which contains 32561 / 16281 (testing) samples with 123/123 (testing) attributes per sample.

Experimental environment:

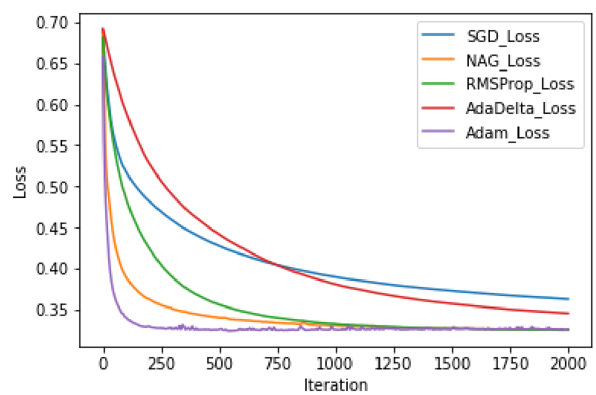
Anaconda3, which contains at least the following Python packages: sklearn, numpy, jupyter, Matplotlib.

IV. CONCLUSION

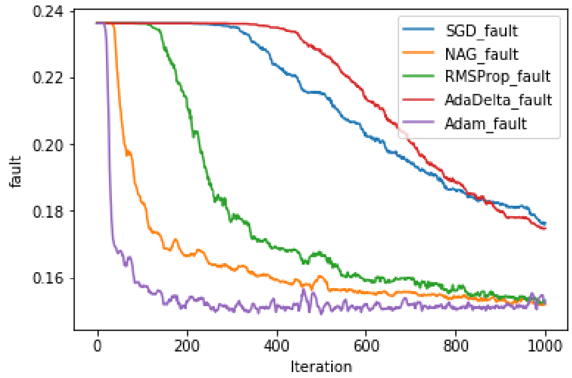
Experiment screenshot:



Loss-Logistic Classification



Loss-Logistic Classification



Fault- Logistic Classification

This is the fault rate of Logistic Classification, so it will shake

Conclusion:

AdaDelta algorithm converge more slowly than SGD at the first time because its learning rate is not given manually. It converge faster and faster.

Adam work very well.