Fuzzy Systems and Data Mining VIII
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220385

Accelerate the Convergence Speed of Perceptron Learning Algorithm with Weight

Liuyang ZHAO [a,1], Shifeng LI* and Ye TIAN [a] and Yue WANG [a]
[a] *College of Information Science and Technology, Bohai University, LiaoNing Province, China*

Abstract. The main problems of the traditional perceptron learning algorithm (PLA) is that there are too many iterations and it is difficult to generate a model quickly, and more iterations are needed when the boundary between the two classes is closed. In this paper, we improve PLA by introducing the current weight into the updating formulation, which can significantly accelerate the iteration. The experiments on different public datasets show that our proposed method can greatly improve the speed of the traditional PLA.

Keywords. PLA, Machine learning, Perceptron

1. Introduction

We cannot study statistical machine learning without considering Rosenblatt's perceptron [1]. In 1943, McCulloch, Warren, etal, a few mathematical logicians, proposed the first mathematical model of neurons – MP model. The MP model is of pioneering significance and provides a basis for the later research work. From the late 1950s to the early 1960s, Rosenblatt added learning function on the basis of MP model (Mcculloch–Pitts model) [2] and put forward the single-layer perceptron model, which put the study of neural network into practice for the first time [3].

Through the combination with PLA, many research works have made better progress. Akshi and Meghna [4] brings forward a model for the Stance Classification of Rumours on a Twitter dataset which utilizes the newly introduced Capsule Network along with Perceptron. The rule-based strategy is used to merge the output of both the networks in a way that utilizes the strength of the two networks. Shalin, Vahid distinguise between several arrhythmias by using deep neural network algorithms such as multi-layer perceptron (MLP) and convolution neural network (CNN) [5].

However, the traditional perceptron algorithm has many problems, where the convergence speed is too slow and it is difficult to avoid in the implementation of the perceptron algorithm. In this paper, the convergence speed of the perceptron algorithm is accelerated by increasing the update weight method, which can make the perceptron al-

[1]Corresponding Author: Shifeng Li, College of Information Science and Technology, Bohai University, LiaoNing Province, The China; E-mail: lishifeng@qymail.bhu.edu.cn

gorithm reach the final result faster. Finally, a new updating method is discussed so that the iteration direction is always correct.

2. Related Work

There it exists an abundance of literature on neural network. Perception Learning Algorithm (PLA) is a branch of machine Learning research field, which is a basic component of neural network model and linear classification model. It belongs to a discriminant model and has a wide range of applications in solving linear separable problems or pattern classification [6].

After the perceptron was proposed, it was severely criticized by Minsky, he argues that the perceptron model cannot be generalized to nonlinear (linearly indivisible) problems. It was not solved until 1986 that Rumelhart and Mcclelland proposed multilevel perceptrons [7], thus solving the problem of nonlinear expansion of perceptrons [8].

So far, researches on PLA thrives, Emmanuel, Shuangning et al. think Efficient Algorithms Can Find Solutions in a Rare Well-Connected Cluster [9]. Benjamin, Will et al. expounded the Storage capacity in Symmetric binary Perceptrons [10]. Jian and Nike et al. calculated the Capacity Lower Bound for the Ising Perceptron [11]. Mihailo discovered Discrete Perceptrons [12].

But the one of the main problems of the traditional perceptron learning algorithm (PLA) that the convergence speed is too slow still hasn't been solved, and more iterations are needed when the boundary between the two classes is small. In this paper, the PLA algorithm is improved to significantly accelerate the iteration speed, and the proposed method can greatly improve the speed of the traditional PLA algorithm.

3. Improved PLA Algrothirm

This section firstly introduces the principle of the perceptron algorithm, and then introduces the improvement of the perceptron algorithm—PLA-λ, and finally introduces the feasibility of the PLA-λ algorithm, which proves that it is effective.

3.1. Perception Learning Algorithm

PLA is a linear classification model and belongs to the discriminant model. The perceptron model provides the mapping from input space to output space. Perceptron which is also known as neuron is the basic processing unit of artificial neural network. Generally, it has one output unit and multiple input units. Its structural model is shown in **Figure 1**:

If there is an input vector set $\mathbf{X}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \cdots, \mathbf{x_n})$ and its weight vector $\mathbf{w}(w_1, w_2, w_3, \cdots, w_n)$. In order to simplify the symbol, $\mathbf{w} = [b, w_1, w_2, \cdots, w_n]$, $\mathbf{x_i} = [1, x_1, x_2, \cdots, x_n]$, then:

$$f(\mathbf{x_i}) = \text{sign}(\mathbf{w^T x_i}) \tag{1}$$

Where $\mathbf{x_i}$ is the occurrence frequency or other parameters of each record, and the combination constitutes the n-dimensional input vector; $\mathbf{w}$ is the n-dimensional data

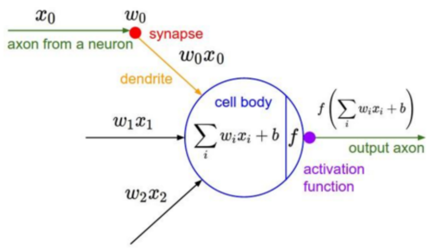
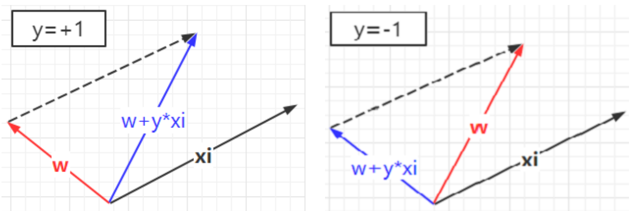


Figure 1. Neuron model.

vector entered by the weight coefficient obtained from the real-time feature evaluation model, and the combination constitutes the n-dimensional weight vector.

In addition, there is a one-to-one correspondence relationship between $\mathbf{x_i}$ and $\mathbf{w}$.Simply put, it is to find a classification hyperplane that separates the positive examples from the negative examples in the data set.The perceptron learning algorithm is to find $\mathbf{w}$ and b to determine the classification hyperplane.

The first dimensional value w_0 of vector $\mathbf{w}$ represents the degree to which the classification hyperplane deviates from the origin of data in the n-dimensional coordinate system. The perceptron learning algorithm is driven by misclassification, as shown in **Figure 2**:



(a) y=+1 (b) y=-1

Figure 2. **w** iteration mode. (a)The vector **w** iterates with y=+1. (b)The vector **w** iterates with y=-1.

For the misclassification point, $\mathbf{w}$ is modified as follows:

$$\mathbf{w} = \mathbf{w} + y\mathbf{x_i} \tag{2}$$

To sum up, set the initial value of $\mathbf{w} = \mathbf{0}$, and then select one misclassified point at a time and update $\mathbf{w} = \mathbf{w} + y\mathbf{x_i}$ until all points are correctly classified.

3.2. PLA-λ Detection

We set $\mathbf{w}$ after T iterations as $\mathbf{w_t}$, and the vector that exists in the theory that can completely divide the linearly separable data set as $\mathbf{w_f}$. Although $\mathbf{w_t}$ are gradually approaching to $\mathbf{w_t}$ in the general direction, the speed is too slow. If the weight λ is added and appropriate parameters are set, the convergence speed will be accelerated through comparison, because the unitarization operation of $\mathbf{w_t}$ makes the weight of wrong data greater, which accelerates the convergence effect.

The specific algorithm is as follows:

Algorithm 1 PLA-λ

Input: Linearly separable datasets X, label set Y, parameter λ.
Output: Weight vector $\mathbf{w}$.

```
Initialize weight vector w=0.
while ture do
    if ∃x_i ∈ X, w · x_i · y ≤ 0 then
        Update w with w + λ · y · w · x_i.
    else
        return w.
    end if
end while
```

This algorithm is called the λ-based perceptron learning algorithm(PLA-λ). Compared with the traditional Perceptron Learning Algorithm, PLA-λ mainly increases the calculation of the weighted parameters of features and generally the weighting coefficient is greater than 1.

By introducing this coefficient, the iteration number of perceptron learning algorithm is improved obviously.

3.3. *Feasibility of PLA-λ*

Previous studies show that for linearly separable data sets, $\mathbf{w_t}$ and $\mathbf{w_f}$ are approached gradually as the number of iterations(T) increases(See Annex 1 for details):

$$T \leq \frac{\max_n \|\mathbf{x_n}\|^2 \cdot \left\|\mathbf{w_f^T}\right\|^2}{\min^2 y_n \mathbf{w_f^T} \mathbf{x_n}} = \frac{R^2}{\rho^2} \tag{3}$$

It indicates that the iteration number T is upper bound, that is, after a certain number of iterations, the PLA algorithm used for the linearly fractionable data set must be able to stop.

4. Experiments

In order to verify the classification effect of PLA-λ algorithm, four real data sets, The Iris Data Set [13], The Cervical Cancer Behavior Risk Data Set [14], The Personal Information Form and Divorce Predictors Scale in the Divorce Predictors Data Set [15] and The Wireless Indoor Localization Data Set [16], were selected for numerical experiments.Among them, the data set is derived from UCI of Machine Learning Repository (http://archive.ics.uci.edu/ml/index.php), which is a commonly used in the literature on machine learning and classification.

Wherein, the value range of λ is 0.4-1.8, and the weight of each update is marked as an update, which terminates when the data set is completely divided.

The experiment was run on Windows 10 with an Intel i7 processor and 512MB of RAM. PLA algorithm was used to learn and verify all data sets. The experiment compares the influence of different weight coefficients λ on the number of iterations of PLA algorithm. The results are shown in the Table 1.

Table 1. The experimental results

Iterations \ λ / Dataset	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
Iris	5	8	6	4	4	9	11	9
Cancer	826	828	840	847	841	833	831	831
Divorce	24	26	31	33	34	38	41	32
Wireless	36	35	59	116	34	140	67	67

We can see from the experimental results in Table 1 that in the Iris data set, the number of iterations is the least when the coefficient of λ is 1.0 or 1.2, but no obvious effect can be seen in such a data set with a small number of iterations.

In the Cancer dataset, the change of λ has less effect on the number of iterations, but still has some effect, slightly better than PLA.

In the Divorce data set, the coefficient of λ has a good effect when it is 0.4-0.6, and the number of iterations is reduced by about 30% compared with PLA.

In the Wireless data set, the coefficient of λ works best when the coefficient is 0.4, 0.6, and 1.2, and the number of iterations is greatly reduced, which is about 60% lower than that of PLA.

Since there are too few linearly separable datasets in reality, it is difficult to effectively explain the experimental results, so we simulated some linearly separable datasets in the experiment to strengthen the persuasiveness. The algorithm of create analog data is as follows:

Algorithm 2 create analog data

Input: The number of cycles N.
Output: Analog dataset **X**, corresponding label set **Y**.
1: Randomly generate a vector $\mathbf{w_f}$ that can fully classify the dataset.
2: **for** N **do**
3: Randomly generate a data point $\mathbf{x_i}$.
4: Calculate the label value $\mathbf{y_i}$ corresponding to $\mathbf{x_i}$ according to $\mathbf{w_f}$.
5: Store $\mathbf{x_i}$ and $\mathbf{y_i}$ in dataset **X** and label set **Y** respectively.
6: **end for**
7: **return X**, **Y**.

Our experimental results on the simulated dataset are shown in **Figure 3**.

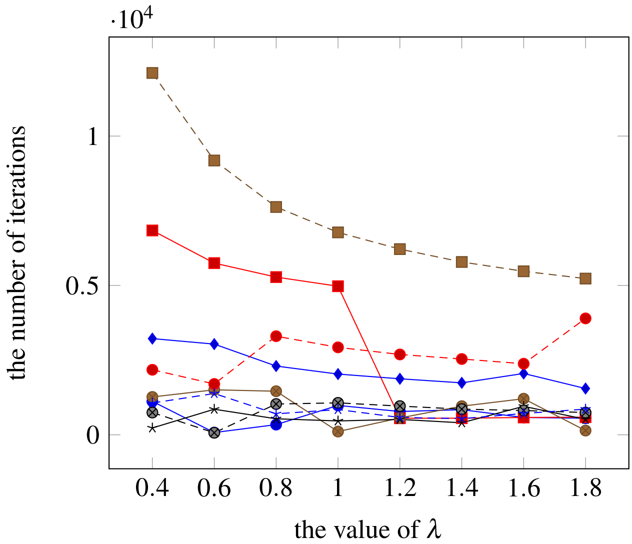


Figure 3.: The experimental results of analog data.

In the simulation data set, we compared the optimal number of iterations after the introduction of λ with the number of iterations without the introduction. The specific results are shown in Table 2.

Table 2. Iteration speed optimization rate

Dataset	original	Join λ	Increase rate(%)
Simulated data set 1	984	76	92.28
Simulated data set 2	4974	555	88.84
Simulated data set 3	110	143	-30.00
Simulated data set 4	465	230	50.54
Simulated data set 5	2034	1550	23.80
Simulated data set 6	2928	1703	41.84
Simulated data set 7	6776	5230	22.82
Simulated data set 8	1070	746	30.28
Simulated data set 9	846	543	35.82

If the weight coefficient is 1, it is the original PLA algorithm. It can be found that the number of iterations of data has a close relationship with λ. As can be seen from the Fig. 3, adding a certain weight coefficient has a good performance, which is obviously better than the original PLA algorithm and significantly increases the iteration speed.

Through theoretical analysis, it is known that the computational complexity of PLA algorithm is a linear function of the data amount, and the computational time complexity is also a linear function of the data amount.

It should be noted that the implementation of PLA-λ did not carry out any necessary coding optimization, but when we used the most trivial way to code, it can still be seen that the iteration speed of PLA-λ is much faster than that of PLA.

5. Conclusion and Future Work

In this paper, a feature weighted perceptron learning algorithm PLA-λ is proposed. By increasing the feature weighted coefficient, the weighted coefficient can be completed in linear time. Therefore, the computational efficiency is better than the original PLA algorithm, and it takes up less resources. Therefore, it is a better perceptron algorithm. Future work is to determine the specific value or certain range of λ according to the data set, so that the number of iterations is always in the minimum range.

6. Acknowledgements

This work was jointly supported by the National Natural Science Foundation of China (61402049), Science and Technology Research Project of the Department of Education of Liaoning Province (LJKZ1019) and Social Science Planning Fund of Liaoning Province (L21BGL002).

References

[1] Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review. 1958;65:386-408.
[2] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics. 1943;5(4):115-33.
[3] White B. Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. JSTOR; 1963.
[4] Kumar A, et al. Rumour stance classification using A hybrid of capsule network and multi-layer perceptron. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 2021;12(13):4110-20.
[5] Savalia S, Emamian V. Cardiac Arrhythmia Classification by Multi-Layer Perceptron and Convolution Neural Networks. Bioengineering. 2018;5(2).
[6] Sagheer A, Zidan M, Abdelsamea MM. A novel autonomous perceptron model for pattern classification applications. Entropy. 2019;21(8):763.
[7] Rumelhart DE, Mcclelland JL. Parallel Distributed Processing. Encyclopedia of Database Systems. 1986:45 76.
[8] Taud H, Mas J. Multilayer perceptron (MLP). In: Geomatic Approaches for Modeling Land Change Scenarios. Springer; 2018. p. 451-5.
[9] Abbe E, Li S, Sly A. Binary perceptron: efficient algorithms can find solutions in a rare well-connected cluster. In: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing; 2022. p. 860-73.
[10] Aubin B, Perkins W, Zdeborová L. Storage capacity in symmetric binary perceptrons. Journal of Physics A Mathematical and Theoretical. 2019;52(29).
[11] Ding J, Sun N. Capacity lower bound for the Ising perceptron. In: the 51st Annual ACM SIGACT Symposium; 2019. .
[12] Stojnic M. Discrete perceptrons. Statistics. 2013.
[13] Dua D, Graff C. UCI Machine Learning Repository; 2017.
[14] Machmud R, Wijaya A, et al. Behavior determinant based cervical cancer early detection with machine learning algorithm. Advanced Science Letters. 2016;22(10):3120-3.
[15] Yöntem MK, Kemal A, Ilhan T, KILIÇARSLAN S. Divorce prediction using correlation based feature selection and artificial neural networks. Nevşehir Hacı Bektaş Veli Üniversitesi SBE Dergisi. 2019;9(1):259-73.
[16] Rohra JG, Perumal B, Narayanan SJ, Thakur P, Bhatt RB. User Localization in an Indoor Environment using Fuzzy Hybrid of Particle Swarm Optimization and Gravitational Search Algorithm with Neural Networks. 2017.