# Optimization based on Loss Function

(Multi classification problems)

Menu

1 Traditional method: softmax loss ........ 2

1.1 theory ........ 2

1.2 Summary ........ 2

2 Optimization ........ 2

2.1 Weighted Softmax Loss ........ 2

2.1.1 Literature ........ 2

2.1.2 Applicable scenario ........ 2

2.1.3 Theory ........ 3

2.2 Large Margin Softmax Loss ........ 3

2.2.1 Theory ........ 3

2.2.2 Summary ........ 5

2.2.3 Legend ........ 5

2.3 Angular Softmax Loss ........ 5

2.3.1 Thesis ........ 5

2.3.2 Theory ........ 5

2.4 L2-constrained Softmax Loss ........ 6

2.4.1 Literature ........ 6

2.4.2 Theory ........ 6

2.5 Additive Margin Softmax Loss ........ 6

2.5.1 Literature ........ 6

2.5.2 Theory ........ 7

2.6 Argface Additive Angular Margin ........ 7

2.6.1 Literature ........ 7

2.6.2 Theory ........ 8

3 Summary ........ 8

# 1 Traditional method: softmax loss

## 1.1 theory

Softmax loss is a combination of softmax and cross entropy loss, which is more stable in numerical calculation.

$$f(z_k) = e^{z_k} / (\sum_j e^{z_j})$$

$$l(y, z) = -\sum_{k=0}^{C} y_c log(f(z_c))$$

$$l((y, z) = log \sum_j e^{z_j} - log e^{z_y}$$

## 1.2 Summary

The distance between optimization classes is very good, and the distance within the optimization class is relatively weak.

# 2 Optimization

## 2.1 Weighted Softmax Loss

### 2.1.1 Literature

Xie S, Tu Z.Holistically-nested edge detection[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1395-1403.

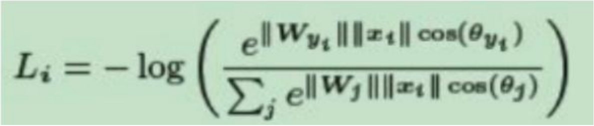
### 2.1.2 Applicable scenario

When the gap in the number of samples is hug. For example, for edge detection, at this time, the importance of obvious edge pixels is greater than that of non edge pixels. At this time, the samples can be weighted pertinently.

### 2.1.3 Theory

$$l(y, z) = -\sum_{k=0}^{C} w_c y_c log(f(z_c))$$

So Li becomes the following formula:

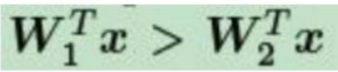


$$L_i = -\log\left(\frac{e^{\|W_{y_i}\|\|x_i\|\cos(\theta_{y_i})}}{\sum_j e^{\|W_j\|\|x_i\|\cos(\theta_j)}}\right)$$

## 2.2 Large Margin Softmax Loss

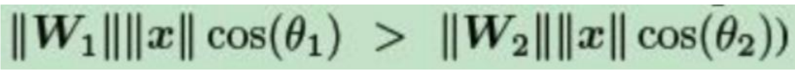
### 2.2.1 Theory

Suppose a 2-classification problem, X belongs to class 1, then the original softmax must be the hope:



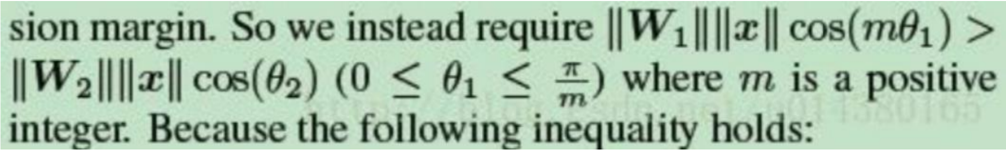
$$W_1^T x > W_2^T x$$

That is to say, the probability of category 1 is greater than that of category 2. This formula is equivalent to the following formula:



$$\|W_1\|\|x\|\cos(\theta_1) \ > \ \|W_2\|\|x\|\cos(\theta_2))$$

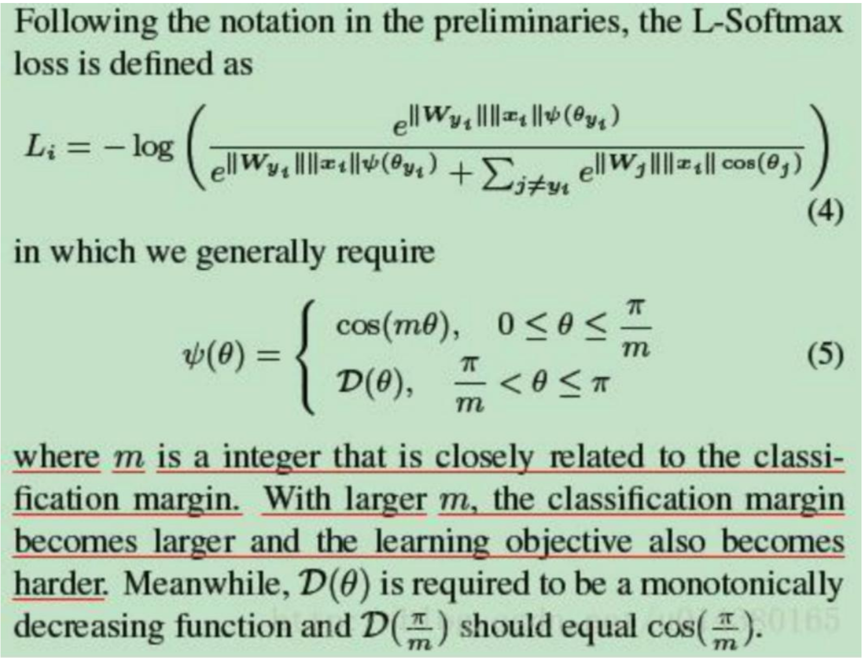
So large margin softmax is to replace the above inequality with the following formula:



sion margin. So we instead require $\|W_1\|\|x\|\cos(m\theta_1) > \|W_2\|\|x\|\cos(\theta_2)$ $(0 \leq \theta_1 \leq \frac{\pi}{m})$ where $m$ is a positive integer. Because the following inequality holds:

Because m is a positive integer and COS function is monotonically decreasing from 0 to π, cos (m x) is smaller than cos (x). The higher the m value is, the more difficult it is to learn. That is to say, the graphs in Figure 2 at the beginning represent different m values. Therefore, defining the loss in this way will force the model to learn the features with larger distance between classes and smaller distance within classes.

In this way, the Li formula of l-softmax loss can be modified on the Li formula of the original softmax loss to get:



Following the notation in the preliminaries, the L-Softmax loss is defined as

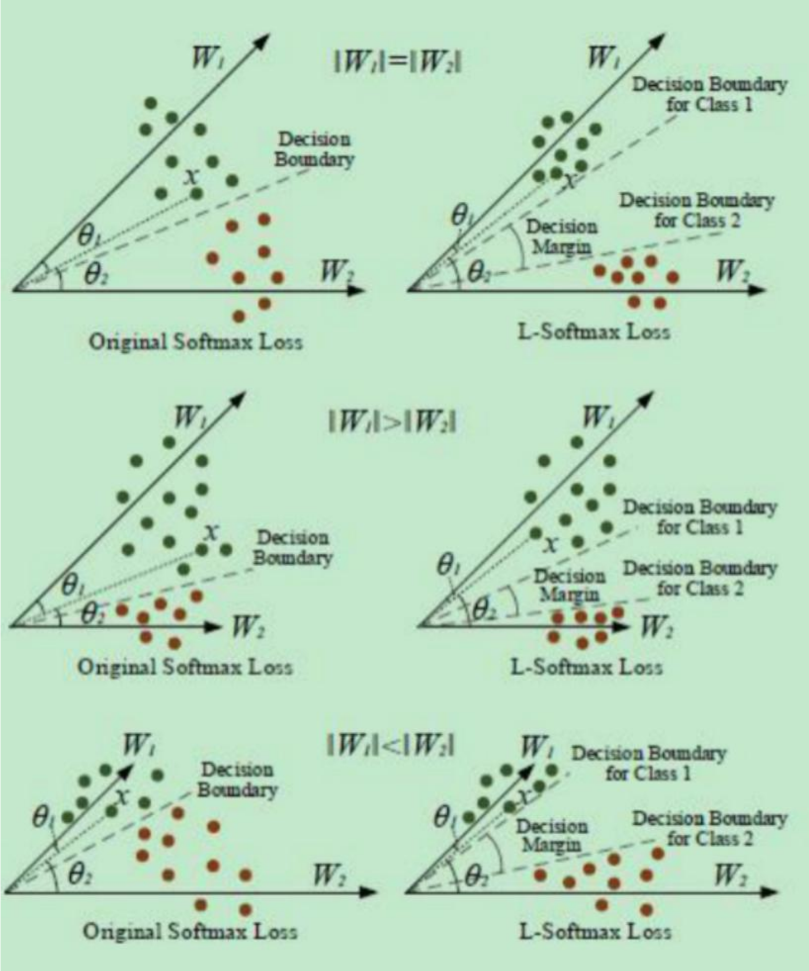
$$L_i = -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\psi(\theta_{y_i})}}{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\psi(\theta_{y_i})}+\sum_{j\neq y_i} e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_j)}}\right) \tag{4}$$

in which we generally require

$$\psi(\theta) = \begin{cases} \cos(m\theta), & 0 \le \theta \le \frac{\pi}{m} \\ \mathcal{D}(\theta), & \frac{\pi}{m} < \theta \le \pi \end{cases} \tag{5}$$

where $m$ is a integer that is closely related to the classification margin. With larger $m$, the classification margin becomes larger and the learning objective also becomes harder. Meanwhile, $\mathcal{D}(\theta)$ is required to be a monotonically decreasing function and $\mathcal{D}(\frac{\pi}{m})$ should equal $\cos(\frac{\pi}{m})$.

The following figure shows the difference between the two kinds of losses from a geometric perspective. The parameters learned by l-softmax loss can increase the distance between the two types of samples. Through comparison, we can see that the degree of separation between the features learned by l-softmax loss is much more obvious than the original.



### 2.2.2 Summary

The idea of l-softmax loss is to increase the learning difficulty of the original softmax loss. If the original softmax loss is that as long as the distance between the support vector and the classification surface is greater than h, the classification effect is better, then l-softmax loss is that the distance needs to reach MH (M is a positive integer).

### 2.2.3 Legend

The upper line represents training set, and the lower line represents testing set. The first one of each line is the traditional softmax, and the last three are l-softmax with different parameters. Look at the distance between and within classes!

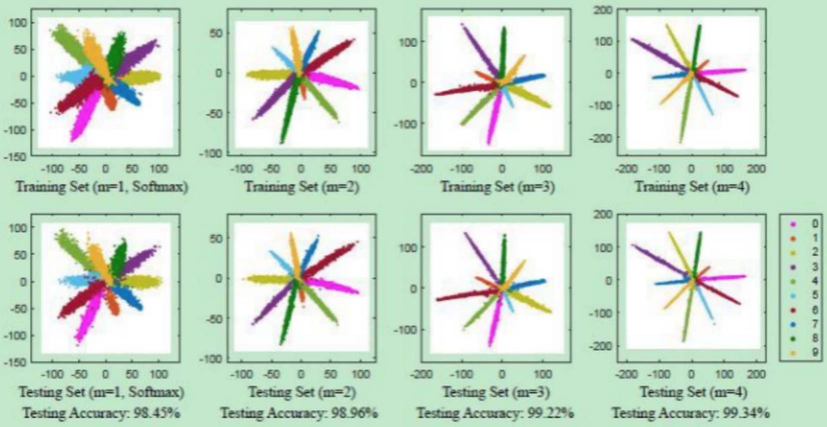


*Figure 2.* CNN-learned features visualization (Softmax Loss (m=1) vs. L-Softmax loss (m=2,3,4)) in MNIST dataset. Specifically, we set the feature (input of the L-Softmax loss) dimension as 2, and then plot them by class. We omit the constant term in the fully connected layer, since it just complicates our analysis and nearly does not affect the performance. Note that, the reason why the testing accuracy is not as good as in Table. 2 is that we only use 2D features to classify the digits here.

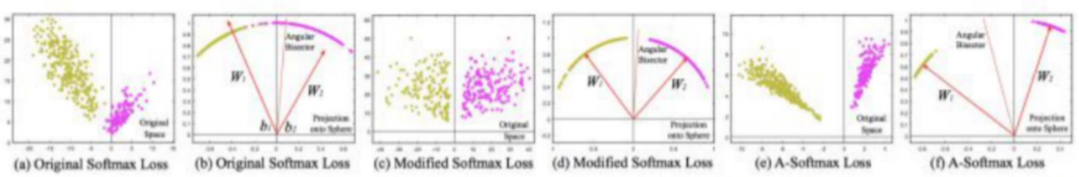
## 2.3 Angular Softmax Loss

### 2.3.1 Thesis

Liu W, Wen Y,Yu Z, et al. Sphereface: Deep hypersphere embedding for face recognition[C]//The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, 1.

### 2.3.2 Theory

Angular softmax loss is also called a-softmax loss. On the basis of large margin softmax loss, it adds two restrictions, i.e. w| = 1 and B = 0, so that the prediction only depends on the angle θ between W and X.



(a) Original Softmax Loss (b) Original Softmax Loss (c) Modified Softmax Loss (d) Modified Softmax Loss (e) A-Softmax Loss (f) A-Softmax Loss

Figure 2: Comparison among softmax loss, modified softmax loss and A-Softmax loss. In this toy experiment, we construct a CNN to learn 2-D features on a subset of the CASIA face dataset. In specific, we set the output dimension of FC1 layer as 2 and visualize the learned features. Yellow dots represent the first class face features, while purple dots represent the second class face features. One can see that features learned by the original softmax loss can not be classified simply via angles, while modified softmax loss can. Our A-Softmax loss can further increase the angular margin of learned features.

The above figure compares the results of the original softmax loss by adding the constraint of | w| = 1 and adding the constraint of | = 1 on the basis of l-softmax loss Why add a constraint of|w| = 1?

The author makes two explanations: one is the feature learned by softmax loss, which has strong differentiation according to the angle; the other is that the face is a manifold, which maps its feature to the hyperplane surface

## 2.4 L2-constrained Softmax Loss

### 2.4.1 Literature

Ranjan R, Castillo C D, Chellappa R. L2-constrained softmax loss for discriminative face verification[J]. arXiv preprint arXiv: 1703.09507, 2017.

### 2.4.2 Theory

The learning feature x is normalized In this paper, the author observes that a good face has a large L2 norm of features, while a face with no obvious features has a small L2 norm of corresponding features. Therefore, this constraint is proposed to enhance the differentiation of features.

$$\begin{aligned}
&\text{minimize} \quad -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{W_{y_i}^T f(\mathbf{x}_i)+b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^T f(\mathbf{x}_i)+b_j}} \\
&\text{subject to} \quad \|f(\mathbf{x}_i)\|_2 = \alpha, \ \ \forall i = 1,2,\ldots M,
\end{aligned}$$

## 2.5 Additive Margin Softmax Loss

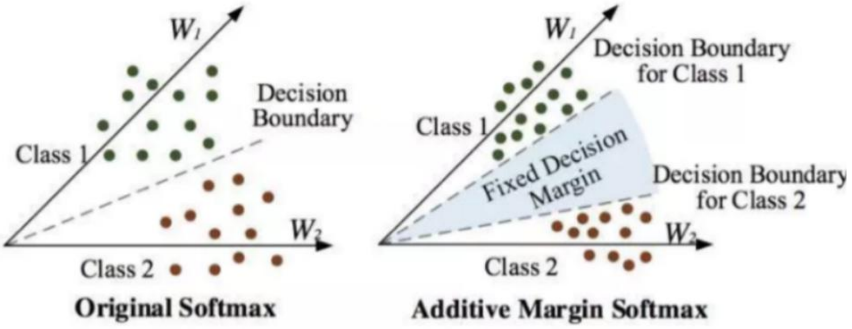
### 2.5.1 Literature

Wang F, Liu W,Liu H, et al. Additive Margin Softmax for Face Verification[J]. arXiv preprint arXiv:1801.05599, 2018.

### 2.5.2 Theory

The multiplication of l-softmax is changed to subtraction, and scale factor s is added After this change, the author's forward and backward propagation becomes more simple. Where W and F are normalized, the author sets m as 0.35 in the paper

$$\begin{aligned}\mathcal{L}_{AMS} &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cdot\left(cos\theta_{y_i}-m\right)}}{e^{s\cdot\left(cos\theta_{y_i}-m\right)}+\sum_{j=1,j\neq y_i}^{c} e^{s\cdot cos\theta_j}} \\ &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cdot\left(W_{y_i}^T \boldsymbol{f}_i-m\right)}}{e^{s\cdot\left(W_{y_i}^T \boldsymbol{f}_i-m\right)}+\sum_{j=1,j\neq y_i}^{c} e^{sW_j^T \boldsymbol{f}_i}}.\end{aligned}$$



**Original Softmax** **Additive Margin Softmax**

Figure 1. Comparison between the original softmax loss and the additive margin softmax loss. Note that, the angular softmax [9] can only impose unfixed angular margin, while the additive margin softmax incorporates the fixed hard angular margin.

Normalization is the guarantee of convergence to a good point. At the same time, scale layer must be added. Scale scale is set to 30 in this paper When is normalization needed and when is it not? This actually depends on the quality of the picture.

The formula is as follows

$$y = \frac{x}{\alpha} \quad \Rightarrow \quad \frac{dy}{dx} = \frac{1}{\alpha}.$$

Among them, α is the module of vector x, which means that if the module value is relatively small, there will be a larger gradient back propagation error coefficient, which is actually equivalent to difficult sample mining. However, we should also pay attention to the problem that those with very poor quality and too small modulus may cause gradient explosion.

## 2.6 Argface Additive Angular Margin

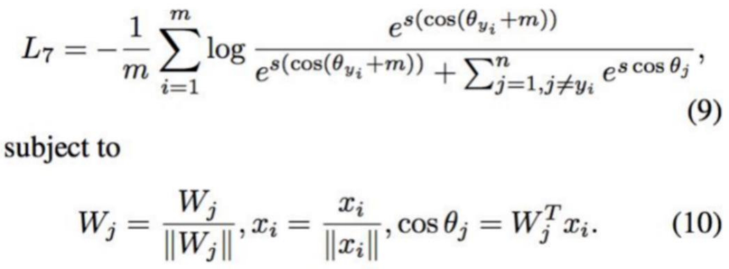
### 2.6.1 Literature

Deng J, Guo J, Zafeiriou S. ArcFace: Additive Angular Margin Loss for Deep Face

Recognition[J]. arXiv preprint arXiv:1801.07698, 2018.

### 2.6.2 Theory

Definition



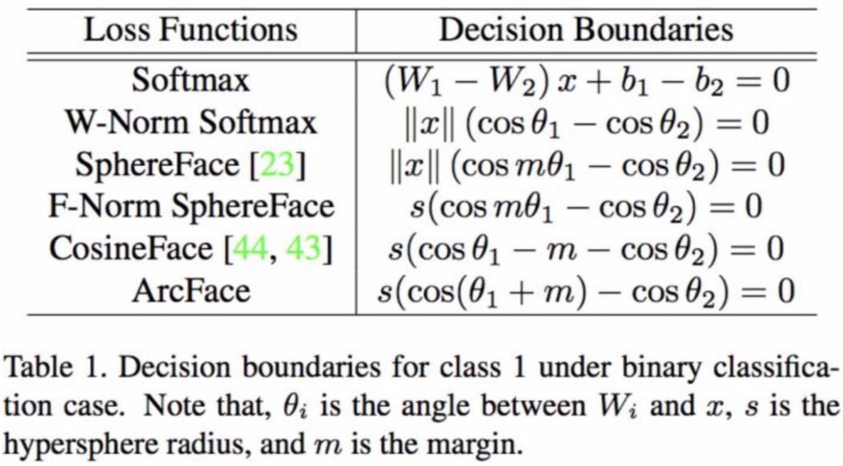
$$L_7 = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}, \tag{9}$$

subject to

$$W_j = \frac{W_j}{\|W_j\|}, x_i = \frac{x_i}{\|x_i\|}, \cos\theta_j = W_j^T x_i. \tag{10}$$

## 3 Summary

Binary classification decision boundary conditions based on softmax loss and its improved loss:



| Loss Functions | Decision Boundaries |
|---|---|
| Softmax | $(W_1 - W_2)\,x + b_1 - b_2 = 0$ |
| W-Norm Softmax | $\|x\|\,(\cos\theta_1 - \cos\theta_2) = 0$ |
| SphereFace [23] | $\|x\|\,(\cos m\theta_1 - \cos\theta_2) = 0$ |
| F-Norm SphereFace | $s(\cos m\theta_1 - \cos\theta_2) = 0$ |
| CosineFace [44, 43] | $s(\cos\theta_1 - m - \cos\theta_2) = 0$ |
| ArcFace | $s(\cos(\theta_1 + m) - \cos\theta_2) = 0$ |

Table 1. Decision boundaries for class 1 under binary classification case. Note that, $\theta_i$ is the angle between $W_i$ and $x$, $s$ is the hypersphere radius, and $m$ is the margin.

Reference:
https://www.jianshu.com/p/3c8fc9dc5ab1
https://blog.csdn.net/u014380165/article/details/76946358