Large-Margin Softmax loss for Convolutional Neural Networks

M2019170

허성실

0

Abstract

- Cross-entropy loss together with softmax is one of the most common used super-vision components
- But the component does not explicitly encourage discriminative learning of features
- L-Softmax loss explicitly encourages Intra-class compactness and inter-class separability
- Not only can adjust the desired margin but also can avoid overfitting
- Can be optimized by typical stochastic gradient descent

Introduction

- CNN needs more discriminative information
- Intra-class compactness and inter-class separability should simultaneously maximized.
- Contrastive loss and triplet loss solved that problem, but needs too many training samples
- They define the softmax loss as the combination of a cross-entropy loss, a softmax function and the las fully connected layer
- The purpose of this paper is to generalize the softmax loss to a more general large-marin softmax loss
- This is done by incorporating a preset constant m multiply with the angle between sample

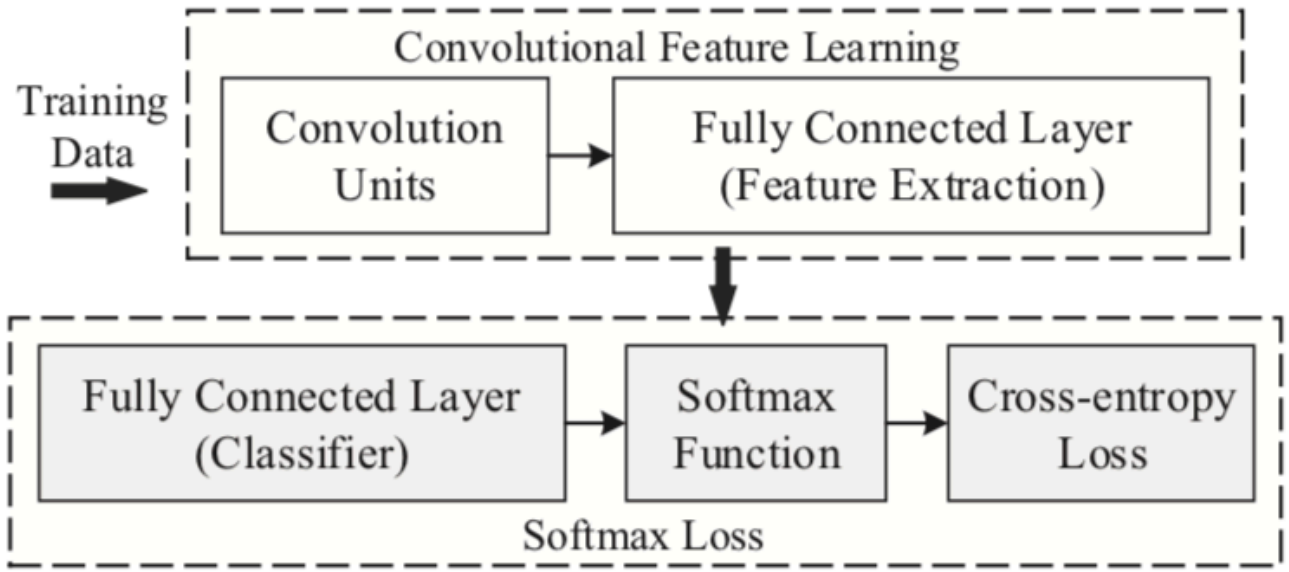


Figure 1. Standard CNNs can be viewed as convolutional feature learning machines that are supervised by the softmax loss.

Related Work

- **Softmax loss + contrastive loss**

-> need pair of training samples

-> same class requires having similar features

- **Triplet loss**

-> needs 3 training samples

-> minimize. The distance between the anchor and positive samples

- They both need a carefully designed pair selection procedure

- Softmax loss

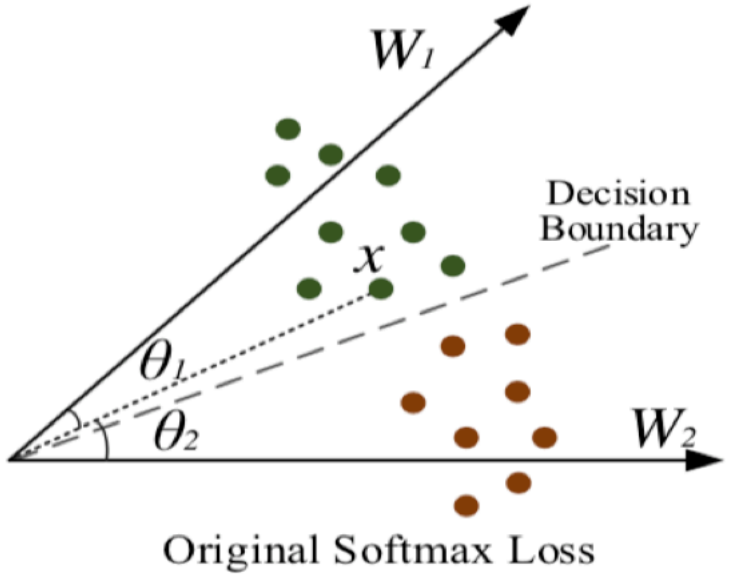
$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

=

$$L_i = -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i})}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_j)}}\right)$$

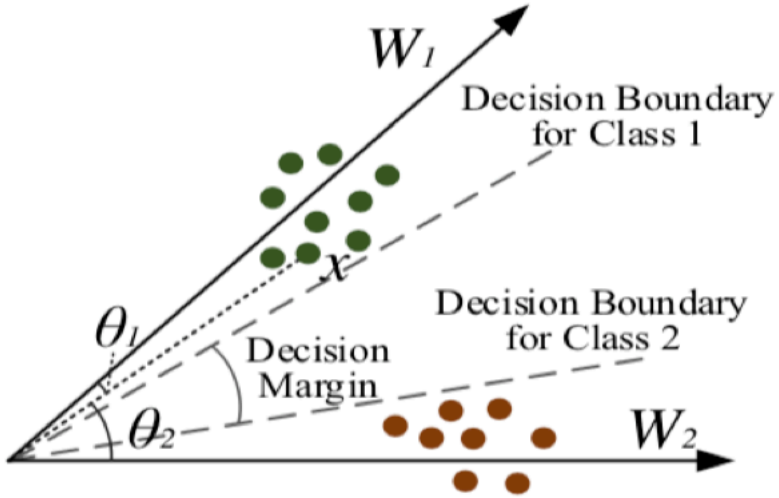
3

Large-Margin Softmax Loss



Original Softmax Loss

$$\|W_1\| = \|W_2\|$$



L-Softmax Loss

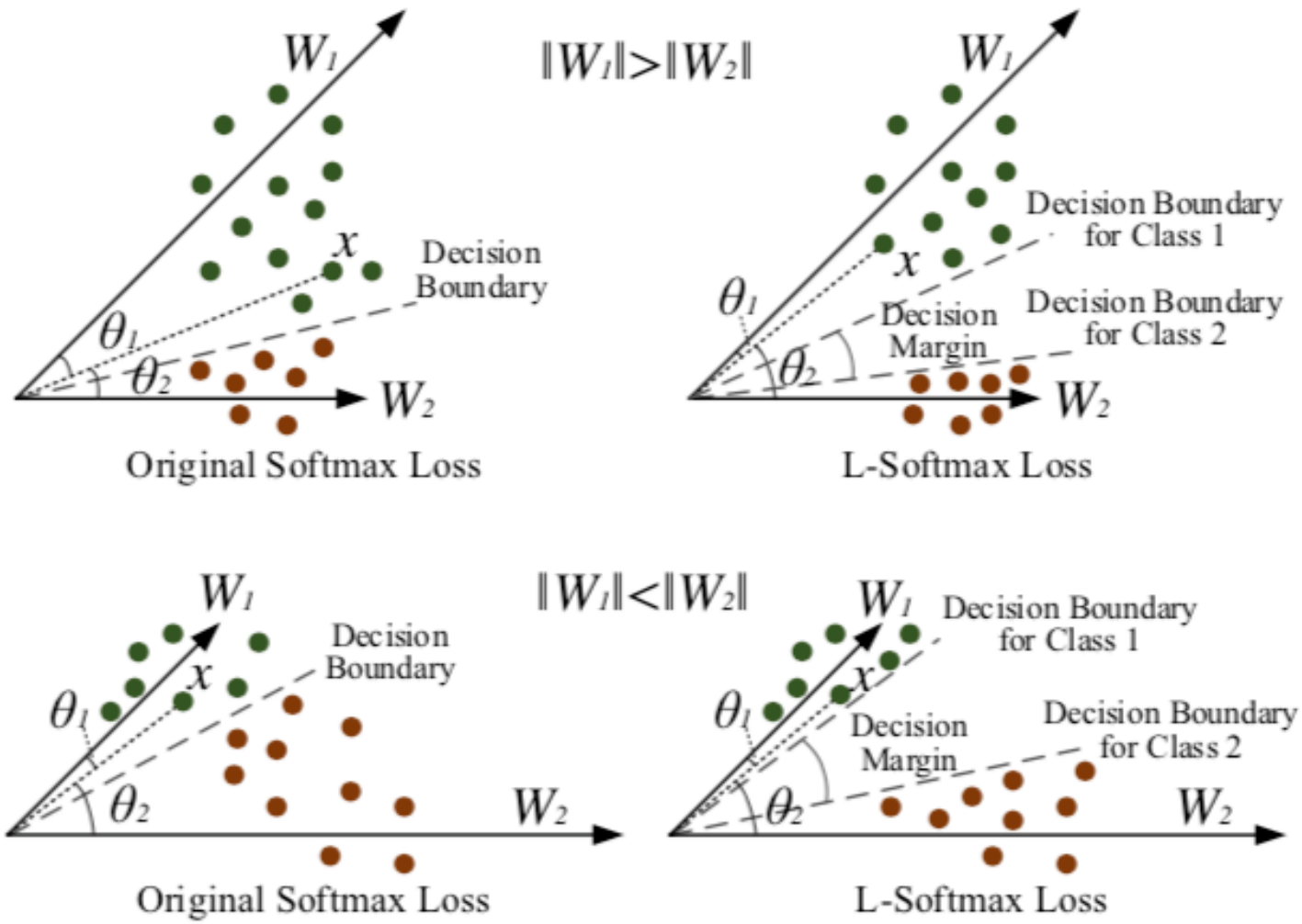
$$\boldsymbol{W}_1^T \boldsymbol{x} > \boldsymbol{W}_2^T \boldsymbol{x}$$

$$\|\boldsymbol{W}_1\|\|\boldsymbol{x}\| \cos(\theta_1) > \|\boldsymbol{W}_2\|\|\boldsymbol{x}\| \cos(\theta_2))$$

$$\|\boldsymbol{W}_1\|\|\boldsymbol{x}\| \cos(m\theta_1) > \|\boldsymbol{W}_2\|\|\boldsymbol{x}\| \cos(\theta_2) \; (0 \leq \theta_1 \leq \tfrac{\pi}{m})$$

$$\|\boldsymbol{W}_1\|\|\boldsymbol{x}\| \cos(\theta_1) \geq \|\boldsymbol{W}_1\|\|\boldsymbol{x}\| \cos(m\theta_1) > \|\boldsymbol{W}_2\|\|\boldsymbol{x}\| \cos(\theta_2).$$

- The original softmax loss requires θ1 < θ2 to classify the sample x as class 1, while the L-Softmax loss requires mθ1 < θ2 to make the same decision.



* The larger W is, the larger the feasible angle of its corresponding class is.

L-softmax

$$L_{softmax} = -log\frac{\exp(W_y^T \boldsymbol{x})}{\sum_{j=1}^{C}\exp(W_j^T \boldsymbol{x})} = -log\frac{\exp(\|W_y\|\|\boldsymbol{x}\|\cos(\theta_y)))}{\sum_{j=1}^{C}\exp(\|W_j\|\|\boldsymbol{x}\|\cos(\theta_j))}$$

$$L_{L-softmax} = -log\frac{\exp(\|W_y\|\|\boldsymbol{x}\|\psi(\theta_y)))}{\exp(\|W_y\|\|\boldsymbol{x}\|\psi(\theta_y))) + \sum_{j\neq y}\exp(\|W_j\|\|\boldsymbol{x}\|\cos(\theta_j)}$$

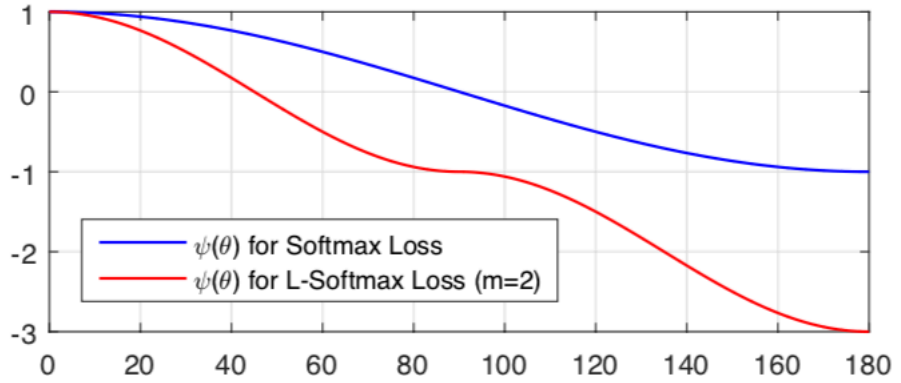


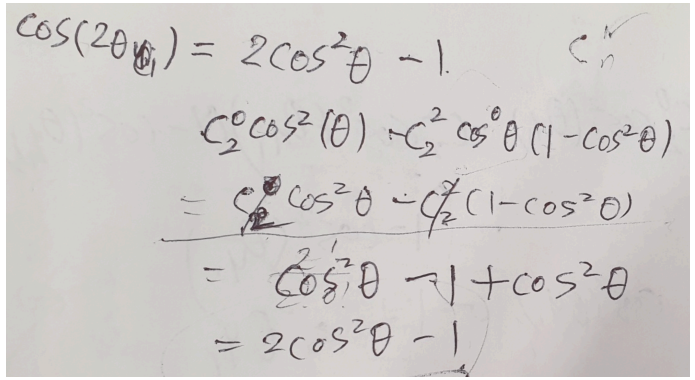
Figure 3. $\psi(\theta)$ for softmax loss and L-Softmax loss.

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

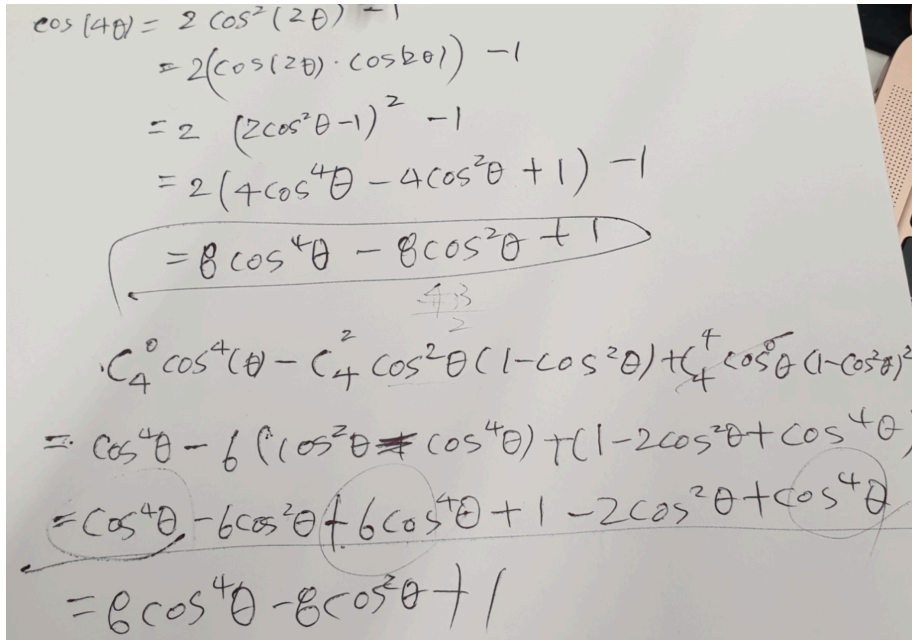
$$\psi(\theta) = (-1)^k \cos(m\theta) - 2k, \quad \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$$

- Transform cos($m\theta$) into combinations of cos(θ):

$$\cos(m\theta_{y_i}) = C_m^0 \cos^m(\theta_{y_i}) - C_m^2 \cos^{m-2}(\theta_{y_i})(1-\cos^2(\theta_{y_i})) + C_m^4 \cos^{m-4}(\theta_{y_i})(1-\cos^2(\theta_{y_i}))^2 + \cdots (-1)^n C_m^{2n} \cos^{m-2n}(\theta_{y_i})(1-\cos^2(\theta_{y_i}))^n + \cdots$$



- Represent cos(θ) as $\frac{\boldsymbol{W}_j^T \boldsymbol{x}_i}{\|\boldsymbol{W}_j\| \|\boldsymbol{x}_i\|}$



- The only difference between the original loss and the L-softmax loss lies in f_{y_i}

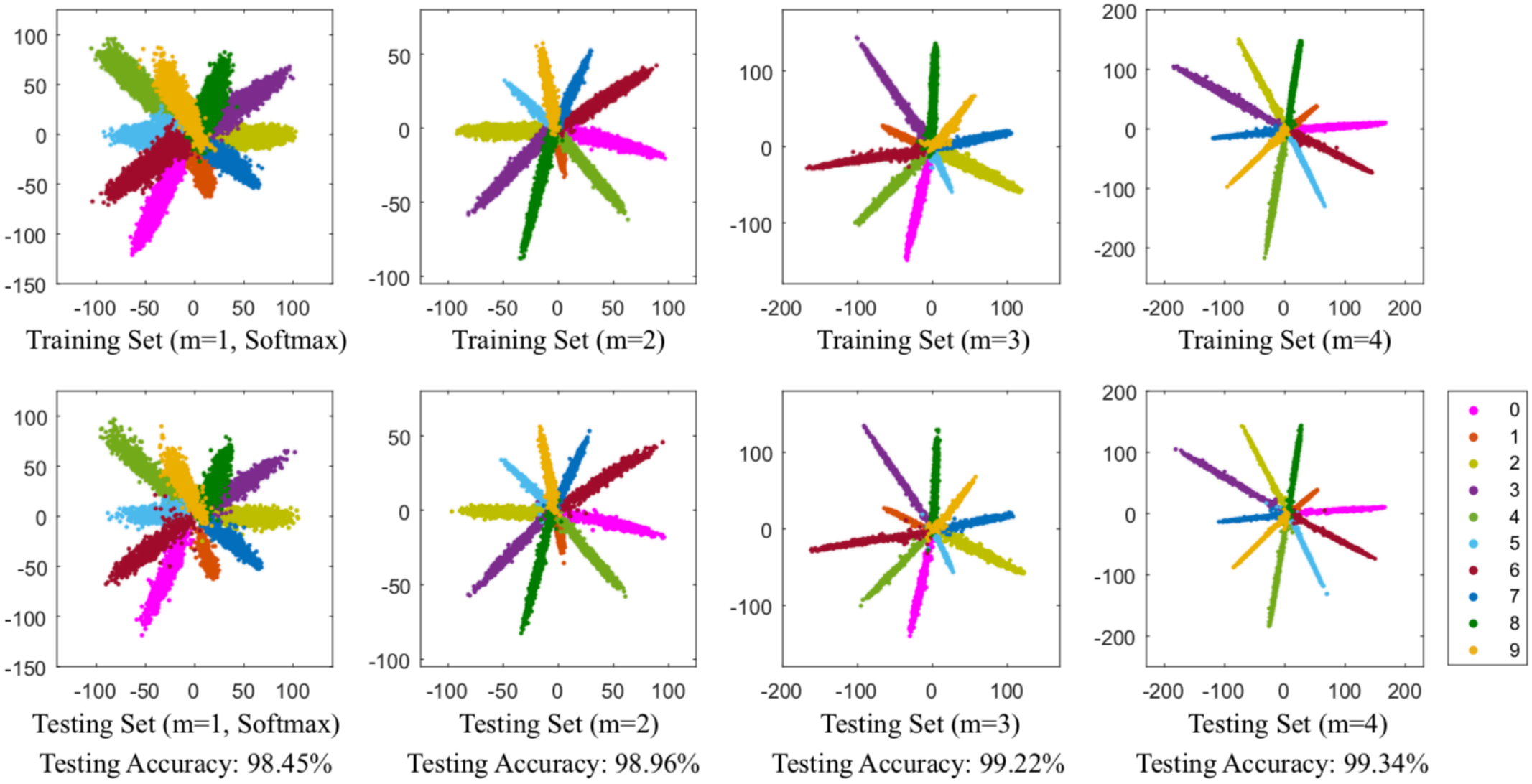
$$
\begin{aligned}
f_{y_i} =& (-1)^k \cdot \|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\| \cos(m\theta_i) - 2k \cdot \|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\| \\
=& (-1)^k \cdot \|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\| \Big(C_m^0 \big(\frac{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i}{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|}\big)^m - \\
& C_m^2 \big(\frac{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i}{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|}\big)^{m-2} \big(1 - \big(\frac{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i}{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|}\big)^2\big) + \cdots \Big) \\
& - 2k \cdot \|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|
\end{aligned}
$$

- In practice, they seek to mininmize:

$$
f_{y_i} = \frac{\lambda \|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\| \cos(\theta_{y_i}) + \|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\| \psi(\theta_{y_i})}{1+\lambda}
$$

- Start with large λ(0.1) and gradually reduce to a very small value.(0.01, 0.001…)

MNIST example



Testing Accuracy: 98.45%

Testing Accuracy: 98.96%

Testing Accuracy: 99.22%

Testing Accuracy: 99.34%

Less overfitting problem compared to original Softmax

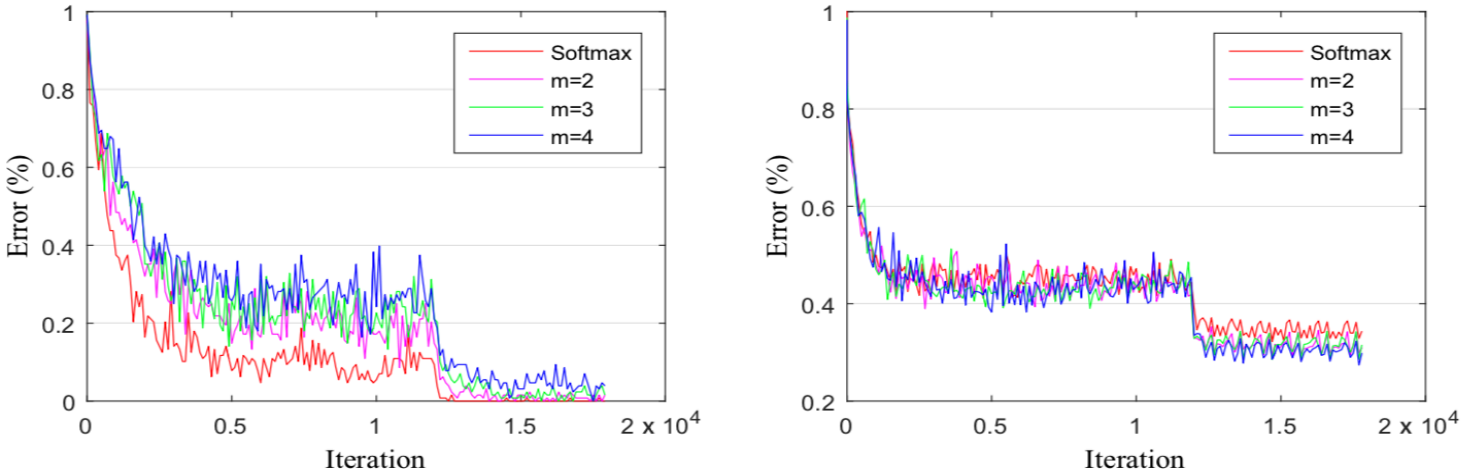


Figure 6. Error vs. iteration with different value of m on CIFAR100. The left shows training error and the right shows testing error.

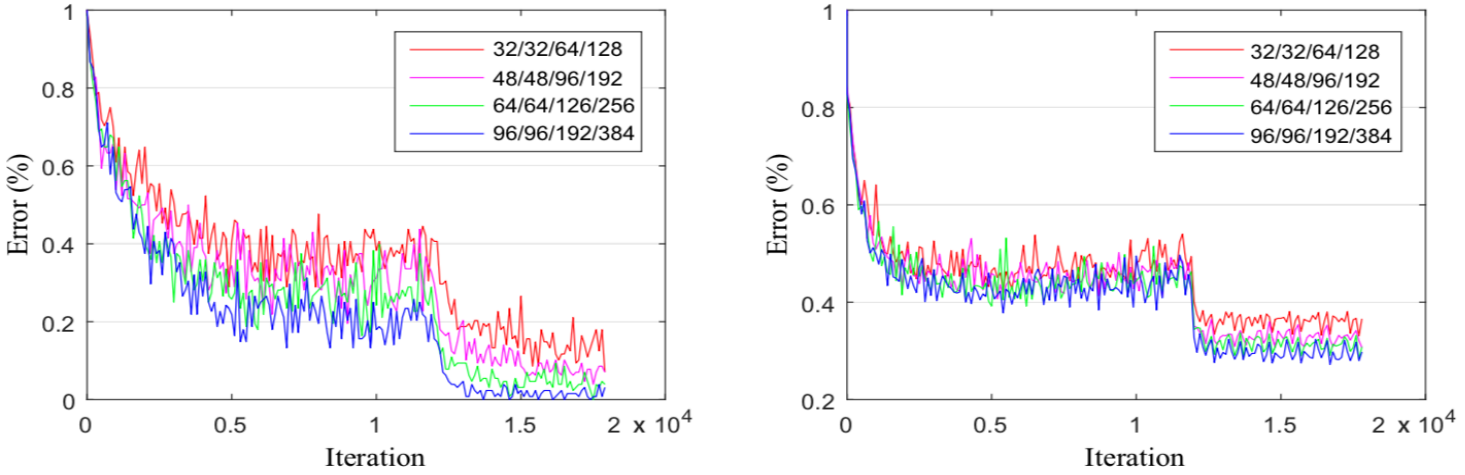


Figure 7. Error vs. iteration (m=4) with different number of filters on CIFAR100. The left (right) presents training (testing) error.

Method	Outside Data	Accuracy
FaceNet (Schroff et al., 2015)	200M*	**99.65**
Deep FR (Parkhi et al., 2015)	2.6M	98.95
DeepID2+ (Sun et al., 2015)	300K*	98.70
(Yi et al., 2014)	WebFace	97.73
(Ding & Tao, 2015)	WebFace	98.43
Softmax	WebFace	96.53
Softmax + Contrastive	WebFace	97.31
L-Softmax (m=2)	WebFace	97.81
L-Softmax (m=3)	WebFace	98.27
L-Softmax (m=4)	WebFace	**98.71**

Table 5. Verification performance (%) on LFW dataset. * denotes the outside data is private (not publicly available).

5

Conclusion

- Adjustable margin with parameter m
- Clear intuition and geometric interpretation
- Show clear advantages over current state-of-the-art CNNs