CS F425: Deep Learning

13

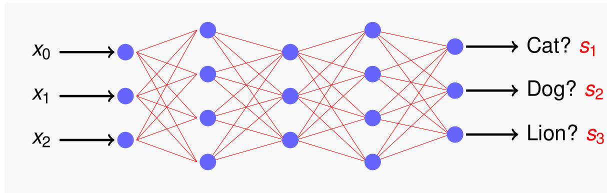
Loss Function For Neural NW

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Feb 17, 2023 ON-CAMPUS Campus @ BITS-Pilani [Jan-May 2023]

http://ktiwari.in/dl

Training Loss for Multi-Class classification



- **Softmax** can help make summation one (for being probalility)

$$f(s_i) = \frac{e^{s_i}}{\sum_{j=1}^{C} e^{s_j}}$$

- **Loss** can tell how good these values are
Cross Entropy is the right choice:

$$L = \frac{1}{m}\sum_m \sum_{i=1}^{C} -y_i \log(f(s_i))$$

Lets look closer

- Consider two class classification and a single example
- **Cross Entropy** loss is:

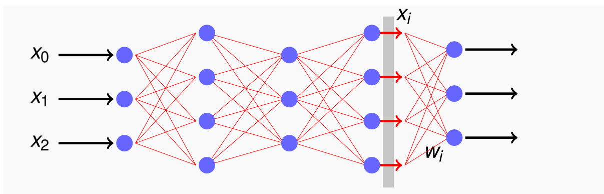
$$\begin{aligned} L &= \frac{1}{m}\sum_m \sum_{i=1}^{C} -y_i \log(f(s_i)) \\ &= \sum_{i=1}^{2} -y_i \log(f(s_i)) = -y_1 \log(f(s_1)) - y_2 \log(f(s_2)) \\ &= -y_1 \log(f(s_1)) - (1-y_1)\log(1-f(s_1)) \end{aligned}$$

- Output vector is *one-hot* for the multi-class classification, so most of the y_i are zero except a single target positive one, leading to

$$L = -\log\left(\frac{e^{s_p}}{\sum_{j=1}^{C} e^{s_j}}\right)$$

here s_p is output with respect to the positive class of the input.

The Cross Entropy loss



$$L = -\log\left(\frac{e^{s_p}}{\sum_{j=1}^{C} e^{s_j}}\right) = -\frac{1}{m}\sum_m \log\left(\frac{e^{W_{y_i}^T x_i + b_i}}{\sum_{j=1}^{C} e^{W_{y_j}^T x_j + b_j}}\right) \quad (1)$$

- where x_i is deep feature of ith sample belonging to class y_i

Softmax loss is separable but not discriminative enough

Centre Loss

- Calculate center[1] for each class, let c_{y_i} is centre of deep features belonging to y_i^{th} class
- Then move inter-correlated features closer to the centre of the class

$$L_c = \frac{1}{2}\sum_m ||x_i - c_{y_i}||_2^2$$

Issues

- Determining center is expensive
- Centre is not accurate while using batch
- Euclidean distance is not the best measure for the feature similarity

[1] average of features of a class

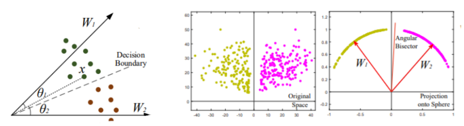
Cosine Formulation of the Softmax Loss

- $W_{y_i}^T x_i + b_i$ can be taken as $W_{y_i}^T x_i$ which is equal to

$$||W_{y_i}^T|| \,.\, ||x_i|| \;\cos\theta_i$$

- If we normalize the weights, making $||W_{y_i}^T|| = 1$
Modified loss function is represented as

$$L = -\frac{1}{m}\sum_i \log\left(\frac{e^{||x_i||\cos\theta_{y_i,i}}}{\sum_j e^{||x_i||\cos\theta_{j,i}}}\right)$$



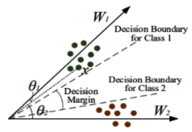
Loss is angularly distributed

ShpereFace: angular softmax

Incorporate multiplicative angular margin $m \geq 1$

$$L = -\frac{1}{m}\sum_i \log\left(\frac{e^{||x_i||\psi(\theta_{y_i,i})}}{e^{||x_i||\psi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{||x_i||\cos(\theta_{j,i})}}\right)$$

- $\psi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k$
- $\theta_{y_i,i} \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$
- $k \in [0, m]$



Margin enforce compression for intra class feature distribution and expands inter-class margin.

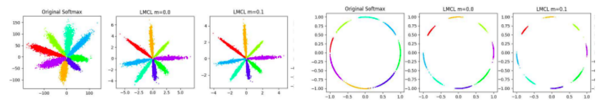
Issues: different margin for different classes, difficult to train.

Hypersphere

1. With feature similarity as cosine, $||x_i||$ does not contribute to score
2. Let us fix it $||x_i|| = s$
3. And use loss as

$$L = -\frac{1}{m}\sum_i \log\left(\frac{e^{s.\cos\theta_{y_i,i}}}{\sum_j e^{s.\cos\theta_{j,i}}}\right)$$

4. Feature vectors are distributed on hypersphere of radious s.



CosFace: Large Margin Cosine Loss [2]

1. Introduces margin $m \geq 0$ in cosine difference between classes

$$L = -\frac{1}{m}\sum_i \log\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{s.\cos(\theta_{j,i})}}$$

2. Defines a decision margin in cosine space rather than the angle space
3. **Issue:** nonlinear angular margin

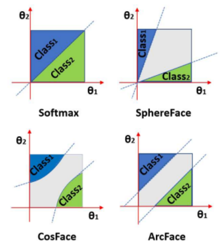
[2] Wang, Hao, et al. "Cosface: Large margin cosine loss for deep face recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

ArcFace: [3]

Introduces margin $m \geq 1$ to the classification boundary

$$L = -\frac{1}{m}\sum_i \log\frac{e^{s\,.\cos(\theta_{y_i}+m)}}{e^{s\,.\cos(\theta_{y_i}+m)} + \sum_{j\neq y_i} e^{s\,.\cos(\theta_j)}}$$

$||w_j|| = 1$, and x_i is L_2 normalized and scaled to s



Method	#Image	LFW	YTF
DeepID [30]	0.2M	99.47	93.20
Deep Face [31]	4.4M	97.35	91.4
VGG Face [22]	2.6M	98.95	97.30
FaceNet [27]	200M	99.63	95.10
Baidu [13]	1.3M	99.13	-
Center Loss [36]	0.7M	99.28	94.9
Range Loss [43]	5M	99.52	93.70
Marginal Loss [6]	3.8M	99.48	95.98
SphereFace [15]	0.5M	99.42	95.0
SphereFace+ [14]	0.5M	99.47	-
CosFace [35]	5M	99.73	97.6
MS1MV2, R100, ArcFace	5.8M	**99.83**	**98.02**

[3] Deng, Jiankang, et al. "Arcface: Additive angular margin loss for deep face recognition." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

Thank You!

Thank you very much for your attention! [4]

[4] **Adopted from:** https://towardsdatascience.com/additive-margin-softmax-loss-am-softmax-912e11ce1c6b