CNN Classification and Verification

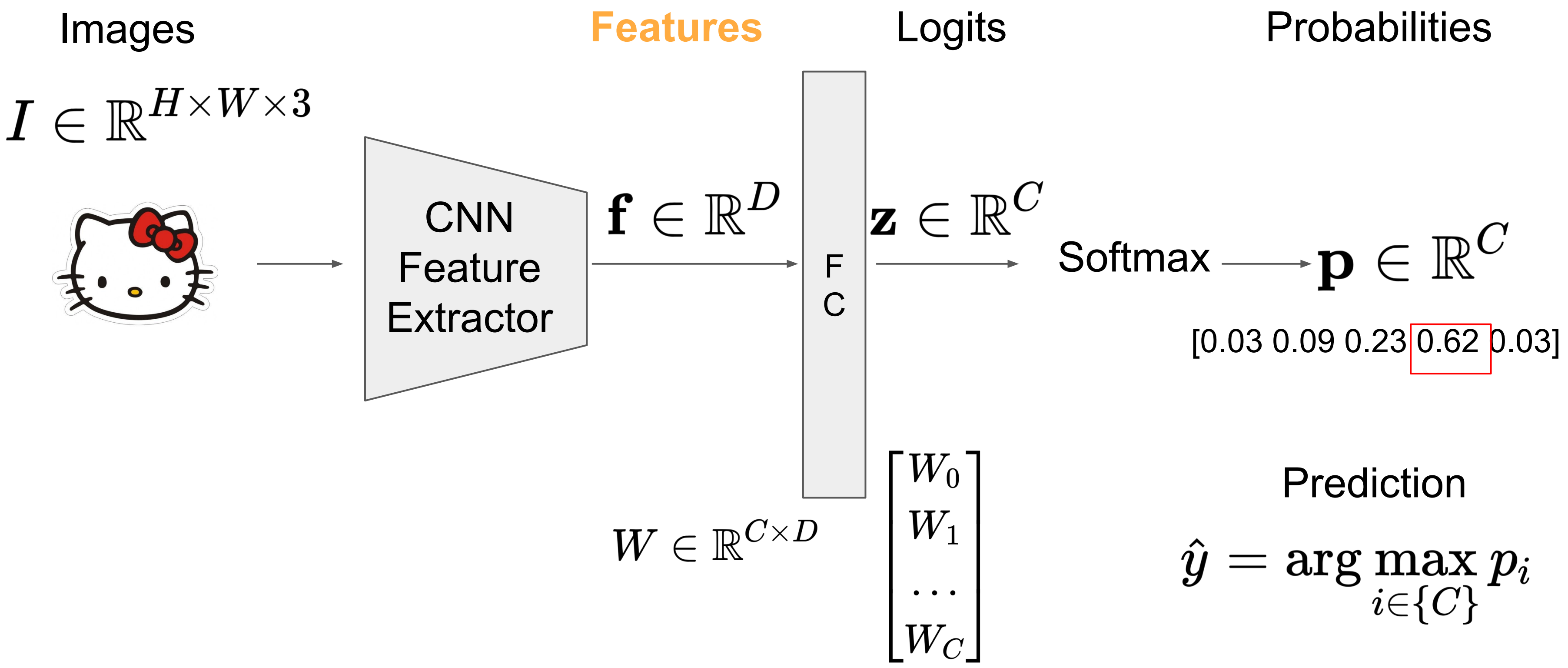
Hao Chen

Content

- Basics
- Face Classification and Verification
- How to Learn Better Features?
- Augmentation & Regularization

Basics

Classification



How to Train Classification?

Cross-Entropy $H(p,q) = -\sum_x p(x)\log q(x)$

How to Train Classification?

Cross-Entropy $H(p,q) = -\sum_x p(x) \log q(x)$

In classification, p(x) is one-hot: [0, 0, 0, 1, 0]

q(x) is predicted probability from softmax $[\frac{e^{W_0^T \mathbf{f}}}{\sum_{j=1}^{C} e^{W_j^T \mathbf{f}}}, \frac{e^{W_1^T \mathbf{f}}}{\sum_{j=1}^{C} e^{W_j^T \mathbf{f}}}, \ldots, \frac{e^{W_C^T \mathbf{f}}}{\sum_{j=1}^{C} e^{W_j^T \mathbf{f}}}]$

How to Train Classification?

Cross-Entropy $H(p,q) = -\sum_x p(x) \log q(x)$

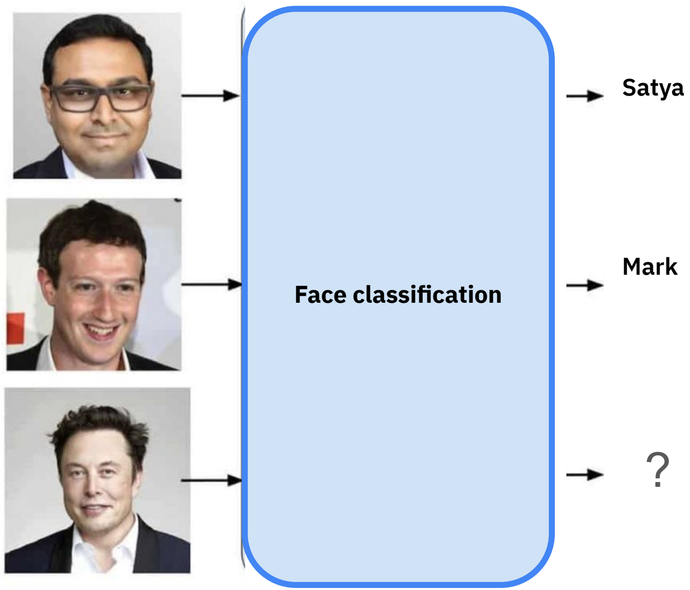
In classification, p(x) is one-hot: [0, 0, 0, 1, 0]

q(x) is predicted probability from softmax $[\frac{e^{W_0^T \mathbf{f}}}{\sum_{j=1}^{C} e^{W_j^T \mathbf{f}}}, \frac{e^{W_1^T \mathbf{f}}}{\sum_{j=1}^{C} e^{W_j^T \mathbf{f}}}, \ldots, \frac{e^{W_C^T \mathbf{f}}}{\sum_{j=1}^{C} e^{W_j^T \mathbf{f}}}]$

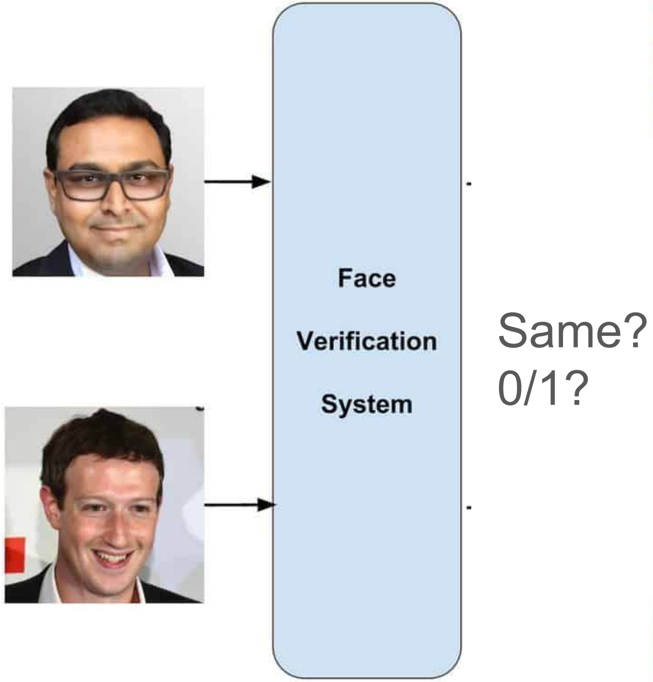
Cross-Entropy Loss: $-\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{W_{y_i}^{\top} \mathrm{f}_i}}{\sum_{j=1}^{C} e^{W_j^{\top} f_i}}$

Face Classification and Verification

Face Classification



Face Verification



This is usually achieved by **comparing features of two faces**

Distance vs Similarity between Vectors

How to compare two face features?

Distance: $\|\mathbf{f}_1 - \mathbf{f}_2\|^2 = \|\mathbf{f}_1\|^2 + \|\mathbf{f}_2\|^2 - 2\mathbf{f}_1\mathbf{f}_2^T$

Cosine Similarity: $cos(\mathbf{f}_1, \mathbf{f}_2) = \dfrac{\mathbf{f}_1\mathbf{f}_2}{\|\mathbf{f}_1\|\|\mathbf{f}_2\|}$

Why Classification Network is not Good for Verification?

Features learned by the classifier with Cross-Entropy only is **not discriminative** enough

A Closer Look at Cross-Entropy

$$-\log \frac{\exp(z^y)}{\sum_{j=1}^{C} \exp\left(z^k\right)}$$

$$= \log \frac{\sum_{j=1}^{C} \exp\left(z^k\right)}{\exp(z^y)}$$

A Closer Look at Cross-Entropy

$$
\begin{aligned}
&- \log \frac{\exp(z^y)}{\sum_{j=1}^{C} \exp\left(z^k\right)} \\
&= \log \frac{\sum_{j=1}^{C} \exp\left(z^k\right)}{\exp(z^y)} \\
&= \log\left(1 + \sum_{j=1, j \neq y}^{C} \exp\left(z^j - z^y\right)\right)
\end{aligned}
$$

A Closer Look at Cross-Entropy

$$
\begin{aligned}
& -\log \frac{\exp(z^y)}{\sum_{j=1}^{C} \exp\left(z^k\right)} \\
& = \log \frac{\sum_{j=1}^{C} \exp\left(z^k\right)}{\exp(z^y)} \\
& = \log\left(1 + \sum_{j=1, j \neq y}^{C} \exp\left(z^j - z^y\right)\right) \\
& = \log\left(1 + \exp\left(\log\left(\sum_{j=1, j \neq y}^{C} \exp\left(z^j - z^y\right)\right)\right)\right)
\end{aligned}
$$

A Closer Look at Cross-Entropy

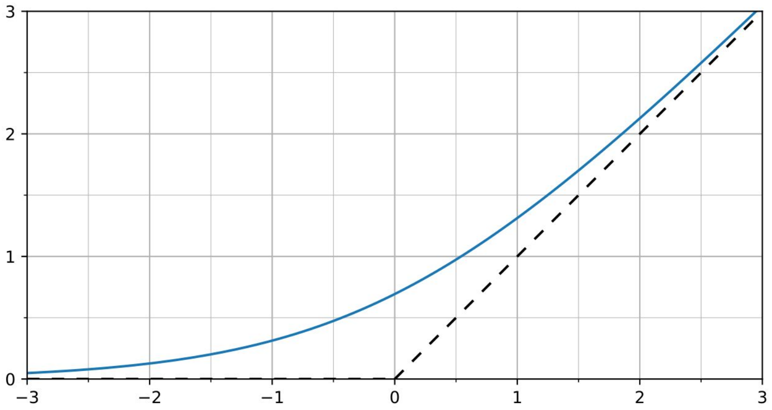
$$-\log \frac{\exp(z^y)}{\sum_{j=1}^{C} \exp\left(z^k\right)}$$

$$= \log \frac{\sum_{j=1}^{C} \exp\left(z^k\right)}{\exp(z^y)}$$

$$= \log\left(1 + \sum_{j=1, j \neq y}^{C} \exp\left(z^j - z^y\right)\right)$$

$$= \log\left(1 + \exp\left(\log\left(\sum_{j=1, j \neq y}^{C} \exp\left(z^j - z^y\right)\right)\right)\right)$$

$$\geq \max\left(\log\left(\sum_{j=1, j \neq y}^{C} \exp z^j - z^y\right), 0\right)$$



Softplus $log(1 + \exp(x)) \geq max(x, 0)$

A Closer Look at Cross-Entropy

$$
\begin{aligned}
&- \log \frac{\exp(z^y)}{\sum_{j=1}^{C} \exp\left(z^k\right)} \\
&= \log \frac{\sum_{j=1}^{C} \exp\left(z^k\right)}{\exp(z^y)} \\
&= \log\left(1 + \sum_{j=1,j\neq y}^{C} \exp\left(z^j - z^y\right)\right) \\
&= \log\left(1 + \exp\left(\log\left(\sum_{j=1,j\neq y}^{C} \exp\left(z^j - z^y\right)\right)\right)\right) \\
&\geq \max\left(\log\left(\sum_{j=1,j\neq y}^{C} \exp z^j - z^y\right), 0\right)
\end{aligned}
$$

Softplus $log(1 + \exp(x)) \geq max(x, 0)$

$$
\geq \log \sum_{j=1,j\neq y}^{C} \exp\left(z^j\right) - z^y
$$

A Closer Look at Cross-Entropy

$$
\begin{aligned}
&-\log \frac{\exp(z^y)}{\sum_{j=1}^{C} \exp\left(z^k\right)} \\
&= \log \frac{\sum_{j=1}^{C} \exp\left(z^k\right)}{\exp(z^y)} \\
&= \log\left(1 + \sum_{j=1, j\neq y}^{C} \exp\left(z^j - z^y\right)\right) \\
&= \log\left(1 + \exp\left(\log\left(\sum_{j=1, j\neq y}^{C} \exp\left(z^j - z^y\right)\right)\right)\right) \\
&\geq \max\left(\log\left(\sum_{j=1, j\neq y}^{C} \exp z^j - z^y\right), 0\right) \quad \text{Softplus } log(1+\exp(x)) \geq max(x, 0) \\
&\geq \log \sum_{j=1, j\neq y}^{C} \exp\left(z^j\right) - z^y \\
&\approx \max_{j\in[C], j\neq y} \left(z^j\right) - z^y \quad \text{LogSumExp } \log \sum \exp(x) \approx \max(x)
\end{aligned}
$$

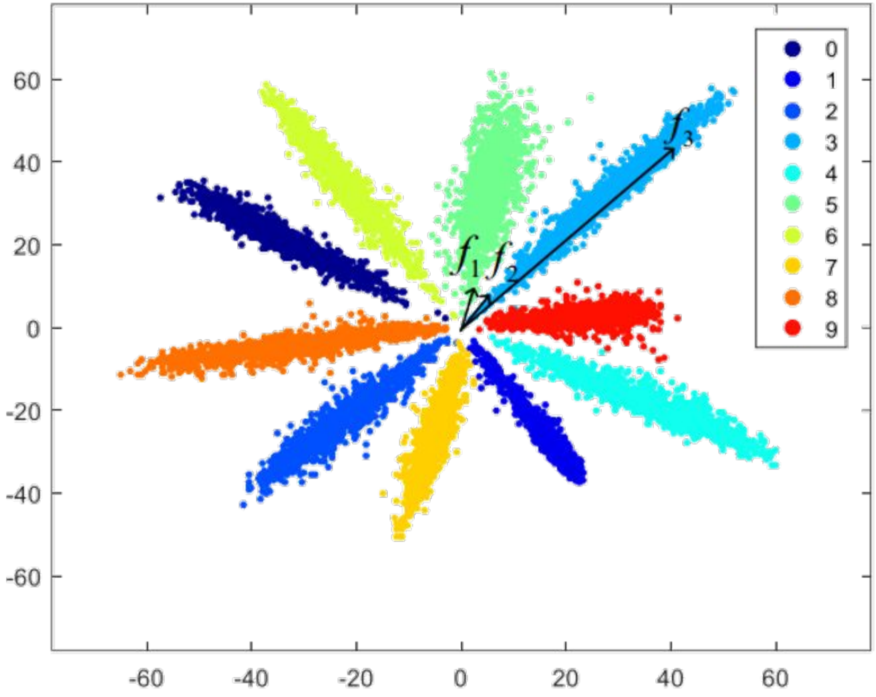
A Closer Look at Cross-Entropy

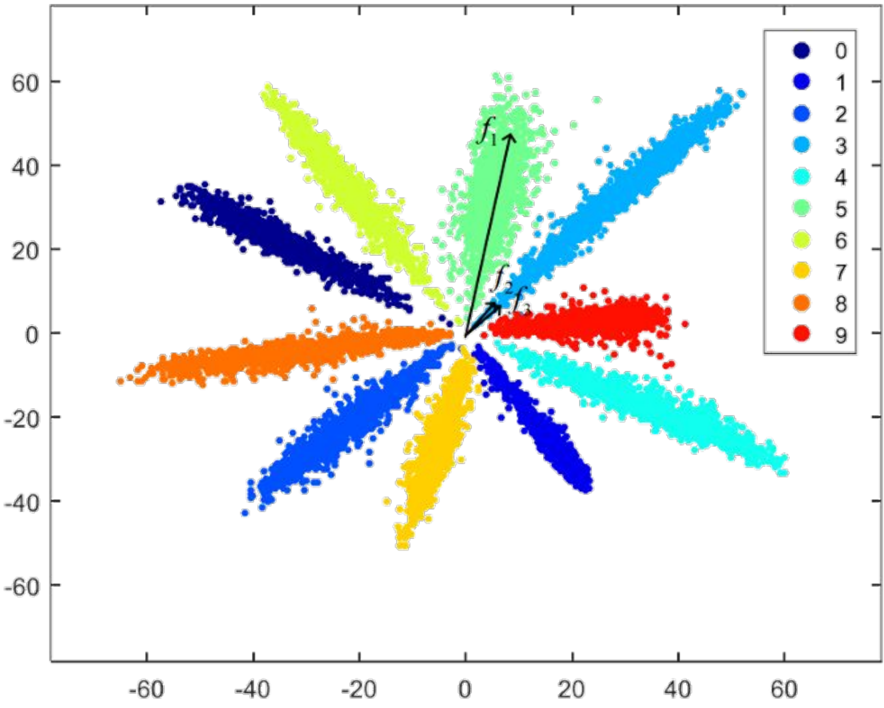
$$
\begin{aligned}
&-\log \frac{\exp\left(z^y\right)}{\sum_{j=1}^{C} \exp\left(z^k\right)} \\
&= \log \frac{\sum_{j=1}^{C} \exp\left(z^k\right)}{\exp\left(z^y\right)} \\
&= \log\left(1 + \sum_{j=1, j\neq y}^{C} \exp\left(z^j - z^y\right)\right) \\
&= \log\left(1 + \exp\left(\log\left(\sum_{j=1, j\neq y}^{C} \exp\left(z^j - z^y\right)\right)\right)\right) \\
&\geq \max\left(\log\left(\sum_{j=1, j\neq y}^{C} \exp z^j - z^y\right), 0\right) \quad \text{Softplus } log(1+\exp(x)) \geq max(x, 0) \\
&\geq \log \sum_{j=1, j\neq y}^{C} \exp\left(z^j\right) - z^y \\
&\approx \max_{j\in[C], j\neq y}\left(z^j\right) - z^y \quad \text{LogSumExp } \log \sum \exp(x) \approx \max(x)
\end{aligned}
$$

Cross-Entropy optimizes

the target logit score to be larger than the maximum of remaining logit scores

Feature Space

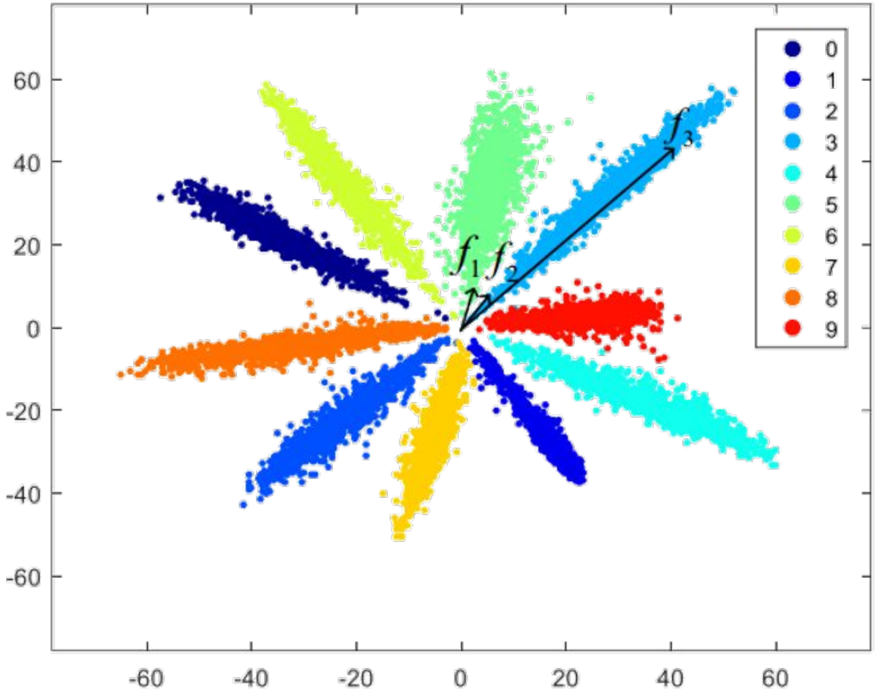


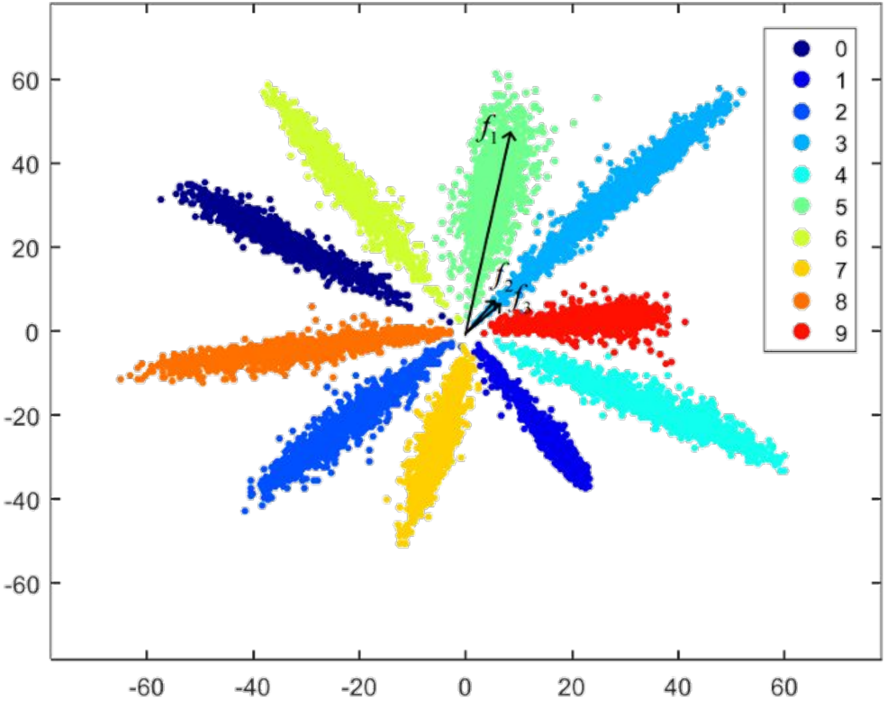


Cross-Entropy leads to radius feature representations:

- It strengthens the feature length/magnitude of correctly classified samples

Problem of Radius Feature Space





Left: Distance of features from different classes are smaller than that of same class

Right: Cosine similarity of features from different classes are larger than that of the same class

How to Learn Better Features?

What are Good Features for Verification?

- Features of the same classes have larger similarity than other classes
  - Ideally with some margin

What are Good Features for Verification?

- Features of the same classes have larger similarity than other classes
  - Ideally with some margin

- Optimization objectives:
  - Maximize the intra-class (same class) similarity
  - Minimize the inter-class (different class) similarity
  - Encourage a margin between intra-class and inter-class similarity

What are Good Features for Verification?

- Features of the same classes have larger similarity than other classes
  - Ideally with some margin

- Optimization objectives:
  - Maximize the intra-class (same class) similarity
  - Minimize the inter-class (different class) similarity
  - Encourage a margin between intra-class and inter-class similarity

- Loss objective: $z^n - z^p + m$
- But…how to define z^n and z^p

Two Paradigms

- Margin-based Softmax Losses
  - Sphereface
  - Cosface
  - Arcface
  - …
- Metric/Pair-based Losses
  - Triplet Loss
  - N-Pair Loss
  - …

Margin-based Softmax

In Cross-Entropy, Weight vector W_j of Weight matrix W can be viewed as anchor for each class

Margin-based Softmax

In Cross-Entropy, Weight vector W_j of Weight matrix W can be viewed as anchor for each class

Intra-class logit score: $z^y = W_y^T \mathbf{f}$

Inter-class logit score: $z^j = W_j^T \mathbf{f}, j \neq y$

Margin-based Softmax

In Cross-Entropy, Weight vector W_j of Weight matrix W can be viewed as anchor for each class

Intra-class logit score: $z^y = \boldsymbol{W_y^T}\mathbf{f}$

Inter-class logit score: $z^j = \boldsymbol{W_j^T}\mathbf{f}, \boldsymbol{j \neq y}$

Cross-Entropy with normal Softmax is optimizing $\max_{j\in[C],j\neq y}\left(z^j\right) - z^y$

What is missing?

Margin-based Softmax

In Cross-Entropy, Weight vector W_j of Weight matrix W can be viewed as anchor for each class

Intra-class logit score: $z^y = W_y^T \mathbf{f}$

Inter-class logit score: $z^j = W_j^T \mathbf{f}, j \neq y$

Cross-Entropy with normal Softmax is optimizing $\max\limits_{j \in [C], j \neq y} \left(z^j\right) - z^y$

What is missing? **Margin!**

How to Introduce Margin in Softmax

- Normalize the feature and weight vectors -> dot product becomes cosine angle

$$\frac{e^{W_y^\top f}}{\sum_{j=1}^{C} e^{W_j^\top f}} = \frac{e^{\cos\theta_y}}{\sum_{j=1}^{C} e^{\cos\theta_j}}$$

How to Introduce Margin in Softmax

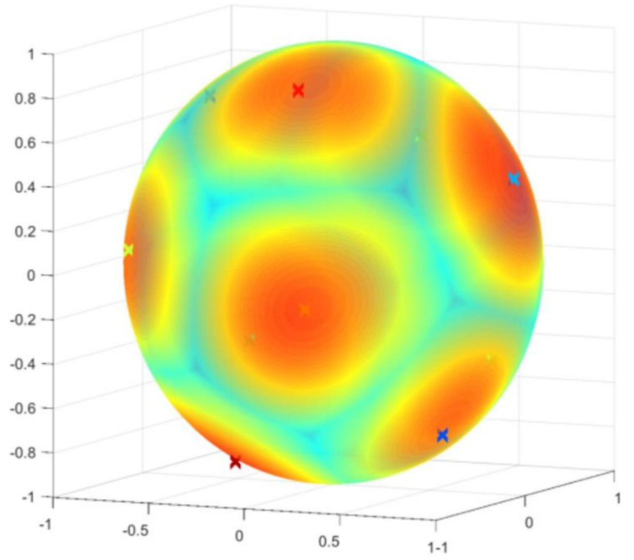
- Normalize the feature and weight vectors -> dot product becomes cosine angle

$$\frac{e^{W_y^\top f}}{\sum_{j=1}^{C} e^{W_j^\top f}} = \frac{e^{\cos\theta_y}}{\sum_{j=1}^{C} e^{\cos\theta_j}}$$

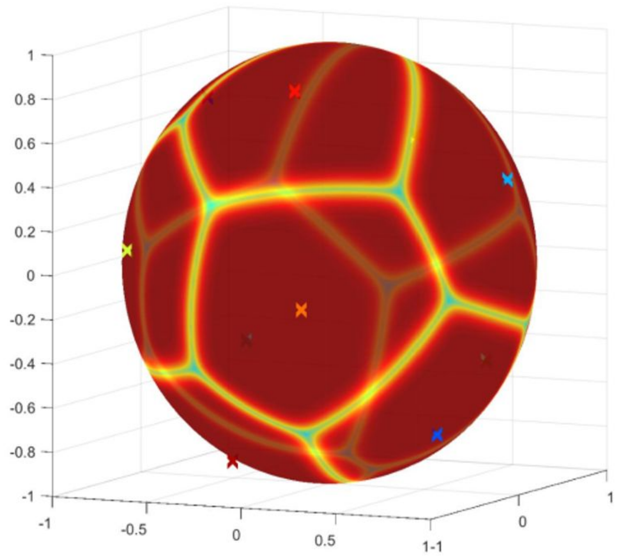
- Scale the feature to overcome the optimization problem with logit score [-1, 1]

$$\frac{e^{s\cos\theta_y}}{\sum_{j=1}^{C} e^{s\cos\theta_j}}$$

Effect of Scale s



Softmax s=5



Softmax s=30

How to Introduce Margin in Softmax

- Normalize the feature and weight vectors -> dot product becomes cosine angle

$$\frac{e^{W_y^\top f}}{\sum_{j=1}^{C} e^{W_j^\top f}} = \frac{e^{\cos\theta_y}}{\sum_{j=1}^{C} e^{\cos\theta_j}}$$

- Scale the feature to overcome the optimization problem with logit score [-1, 1]

$$\frac{e^{s\cos\theta_y}}{\sum_{j=1}^{C} e^{s\cos\theta_j}}$$

- Introduce margin…at where?

CosFace Softmax Loss

Additive Margin

$$-log \frac{e^{s(\cos\theta_y - m)}}{e^{s(\cos\theta_y - m)} + \sum_{j\neq y}^{C} e^{s\cos\theta_j}}$$

SphereFace Softmax Loss

Angular Margin

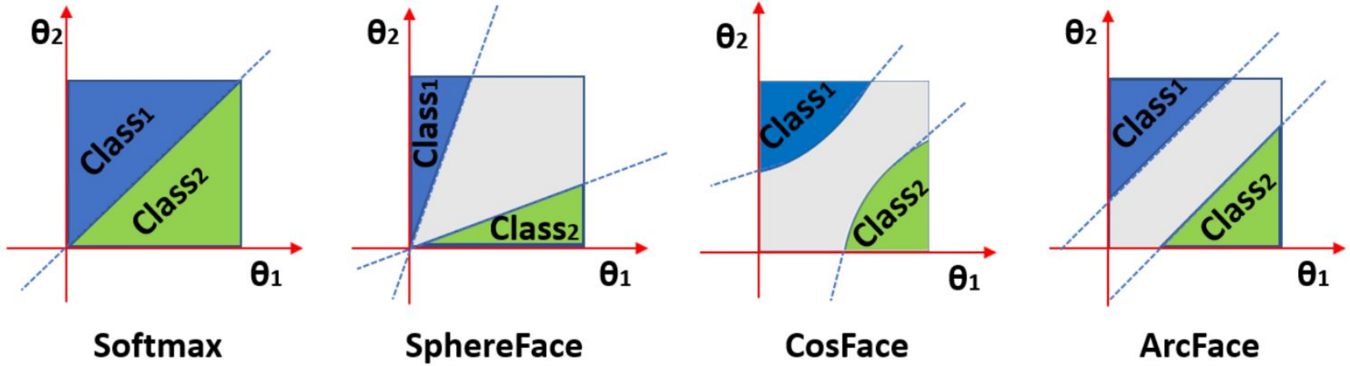
$$-log\frac{e^{s\cos m\theta_y}}{e^{s\cos m\theta_y}+\sum_{j\neq y}^{C}e^{s\cos\theta_j}}$$

ArcFace Softmax Loss

Additive angular Margin

$$-\log \frac{e^{s\cos(\theta_y - m)}}{e^{s\cos(\theta_y - m)} + \sum_{j\neq y}^{C} e^{s\cos\theta_j}}$$

Margin Visualization



Different types of margin can be combined -> CombinedMarginFace

CombinedMargin Face

Metric/Pair-based Loss

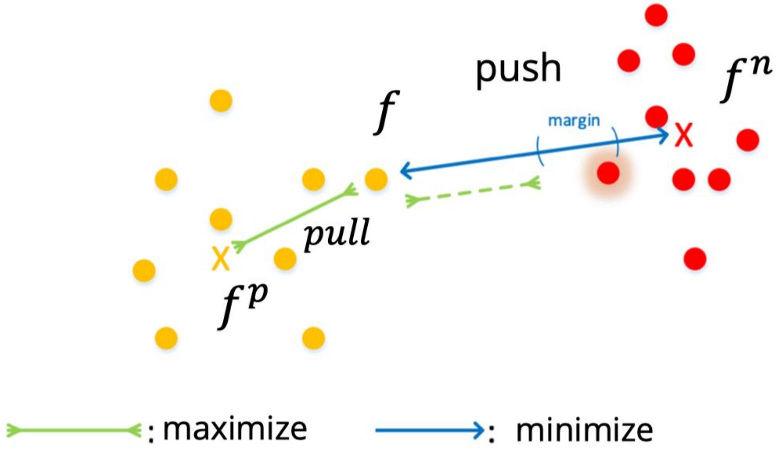
Based on pairs…with **Data Sampler**

Intra-class logit score: $z^p = \mathbf{f}_p^T \mathbf{f}$

Inter-class logit score: $z^n = \mathbf{f}_n^T \mathbf{f}$

Triplet Loss

$$\mathcal{L}_{\text{Triplet}} = \max(\mathbf{f}_n^T \mathbf{f} - \mathbf{f}_p^T \mathbf{f} + m, 0)$$



How to Construct Positive and Negative Pairs?

- Simple way: construct from data batch
    - Might not have samples from the same classes

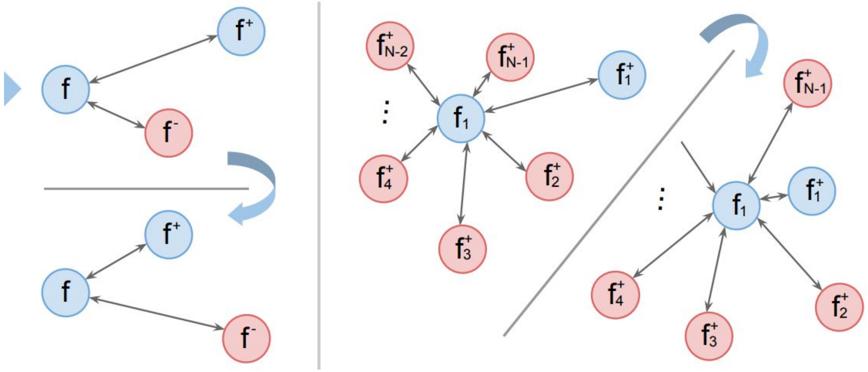
- Use Data Sampler to construct data batch with both samples of positive classes and negative classes

- **Hard mining**: find the most difficult pairs

N-Pair Loss/Contrastive Loss

Why Triplet Loss is not good enough?

Only one positive pair and one negative pair



$$\log\left(1+\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^{T}\mathbf{f}-\mathbf{f}_{p}^{T}\mathbf{f}\right)\right)$$

More on N-Pair/Contrastive Loss

$$\log\left(1+\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}-\mathbf{f}_p^T\mathbf{f}\right)\right)$$

$$=\log\left(1+\exp\log\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}-\mathbf{f}_p^T\mathbf{f}\right)\right)$$

More on N-Pair/Contrastive Loss

$$\log\left(1+\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}-\mathbf{f}_p^T\mathbf{f}\right)\right)$$

$$=\log\left(1+\exp\log\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}-\mathbf{f}_p^T\mathbf{f}\right)\right)$$

$$\geq \max(\log\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}-\mathbf{f}_p^T\mathbf{f}\right),0)$$

More on N-Pair/Contrastive Loss

$$\log\left(1 + \sum_{i=1}^{N-1} \exp\left(\mathbf{f}_{n_i}^T\mathbf{f} - \mathbf{f}_p^T\mathbf{f}\right)\right)$$

$$= \log\left(1 + \exp\log\sum_{i=1}^{N-1} \exp\left(\mathbf{f}_{n_i}^T\mathbf{f} - \mathbf{f}_p^T\mathbf{f}\right)\right)$$

$$\geq \max(\log\sum_{i=1}^{N-1} \exp\left(\mathbf{f}_{n_i}^T\mathbf{f} - \mathbf{f}_p^T\mathbf{f}\right), 0)$$

$$\geq \log\sum_{i=1}^{N-1} \exp\left(\mathbf{f}_{n_i}^T\mathbf{f}\right) - \mathbf{f}_p^T\mathbf{f}$$

More on N-Pair/Contrastive Loss

$$\log\left(1+\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}-\mathbf{f}_p^T\mathbf{f}\right)\right)$$

$$=\log\left(1+\exp\log\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}-\mathbf{f}_p^T\mathbf{f}\right)\right)$$

$$\geq \max(\log\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}-\mathbf{f}_p^T\mathbf{f}\right),0)$$

$$\geq \log\sum_{i=1}^{N-1}\exp\left(\mathbf{f}_{n_i}^T\mathbf{f}\right)-\mathbf{f}_p^T\mathbf{f}$$

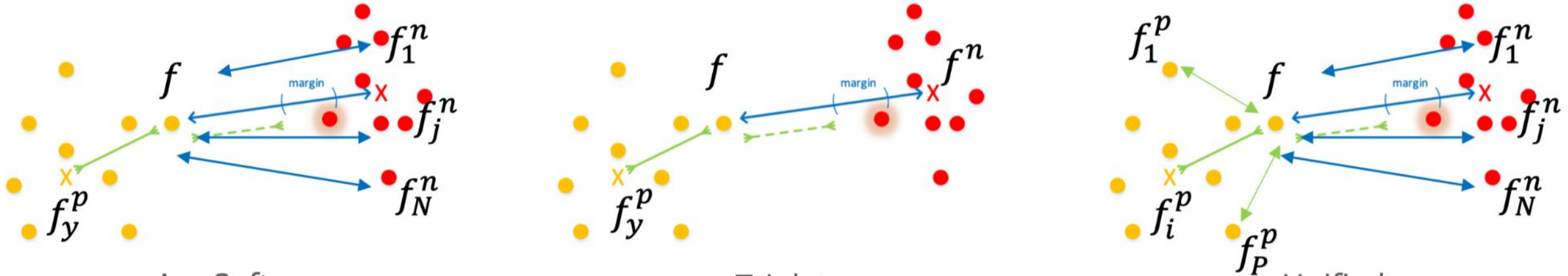
$$\geq \max(\mathbf{f}_{n_i}^T\mathbf{f})-\mathbf{f}_p^T\mathbf{f}$$

- Optimization goal the same as Cross-Entropy
- But negative pairs are different
- Margin can also be added

A Unified View of Both Paradigm

Can we introduce multiple positive pairs and multiple negative pairs?

$$\log\left(1+\sum_{i}^{P}\sum_{j}^{N}e^{\gamma\left(z_{n_j}-z_{p_i}+m\right)}\right)$$



Am-Softmax
1 positive pair
(C-1) negative pairs

Triplet
1 positive pair
1negative pair

Unified
P positive pairs
N negative pairs

Unified Formulation

- This is a unified formulation

$$\log\left(1+\sum_{i}^{P}\sum_{j}^{N}e^{\gamma\left(z_{n_j}-z_{p_i}+m\right)}\right)$$

- You can derive softmax, arcface, triplet, n-pair losses from this formulation
  - By setting the number of positives/negatives and how to construct them

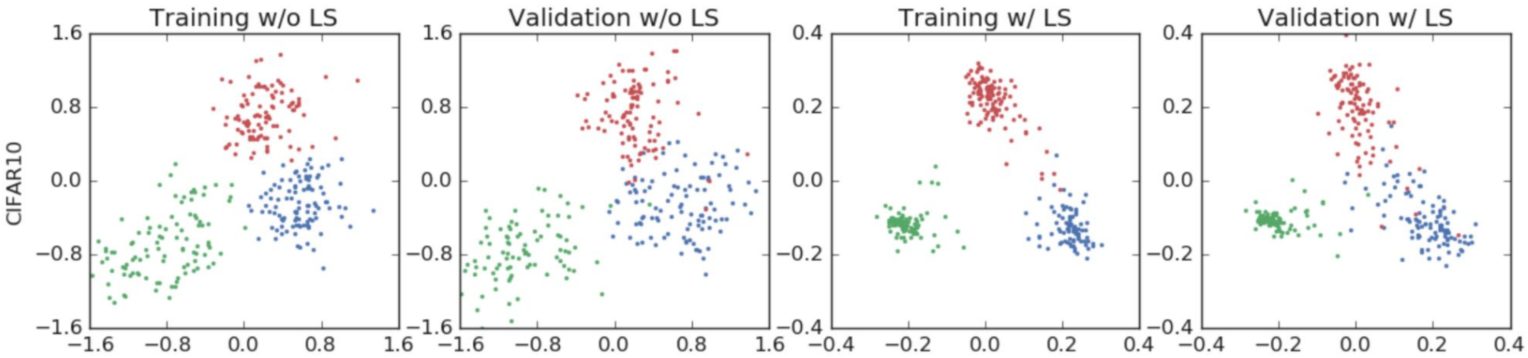
- **Circle loss** can also be derived from this function
- **Supervised Contrastive loss** is another loss function with multiple p/ns

Augmentation & Regularization

Label Smoothing

One-hot target: [0, 0, 0, 1, 0]

Smoothed (ls=0.1) target: [0.025, 0.025, 0.025, 0.9, 0.025]



Label smoothing learns more separable features with smaller feature norm

Similar to the effect of s in Softmax

Mixup/Cutout/Cutmix