

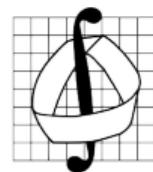
Introduction to Machine and Deep Learning Theory

FaceID: Evolution of Loss function

Aleksandr Petiushko

Lomonosov MSU
Faculty of Mechanics and Mathematics

February 15, 2023



Content

- ① Representation Learning
- ② FaceID task
- ③ Historical overview
- ④ SoftMax Loss and variations
- ⑤ Euclidian Metric Loss and variations
- ⑥ Angular Loss and variations

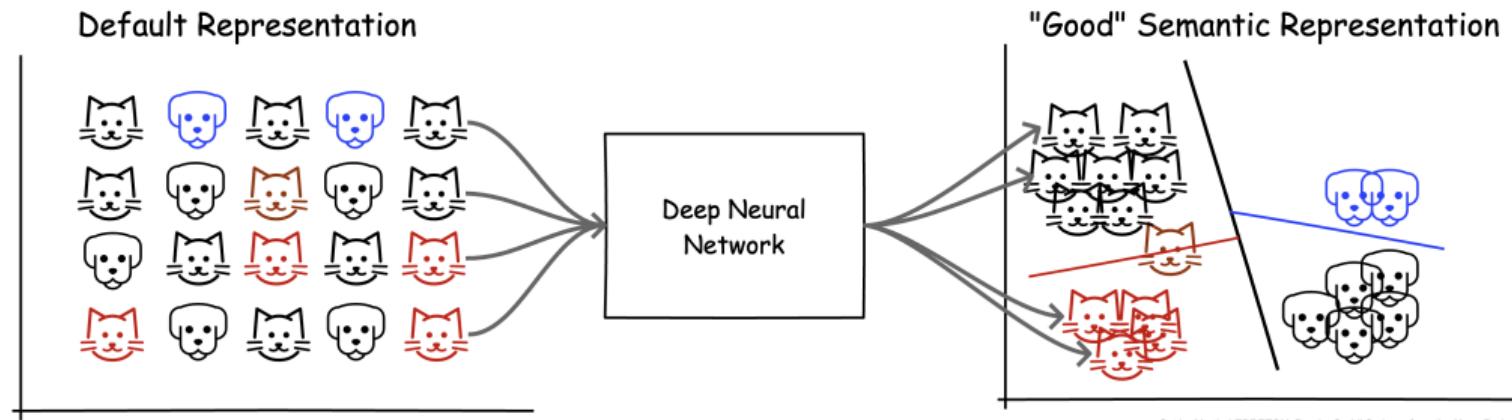
Feature engineering

- Initially the Machine Learning machinery was targeted at **feature engineering** (or extraction, or discovery)
 - ▶ Mapping f from arbitrary object $o \in O$ to its unified feature space representation $x \in X \subseteq \mathbb{R}^D: f: O \rightarrow X$
- Suppose we have m objects \Rightarrow can construct training dataset X^m
- On top of that representation (usually a *linear*) a classifier is learned (e.g., SVM):
 $a(w, X^m) = \text{sign } g(x, w) = \text{sign} \langle x, w \rangle$

Representations Learning

- Main goal: to have objects representations rich and descriptive:
 - ▶ Representations of objects from the same class are better *to be close* (according to some metric, e.g., Euclidean) in the feature space
 - ▶ Representation of objects from different classes are better to *lie far* from each other (according to some metric)
- Sometimes the initial features $x \in X$ are not suited enough for this task
- For this purpose, additional mapping h from X into some additional representation space $R \subseteq \mathbb{R}^N$: $h : X \rightarrow R$
- The learning of this additional mapping h is called **Representation Learning**

Representation spaces illustration¹



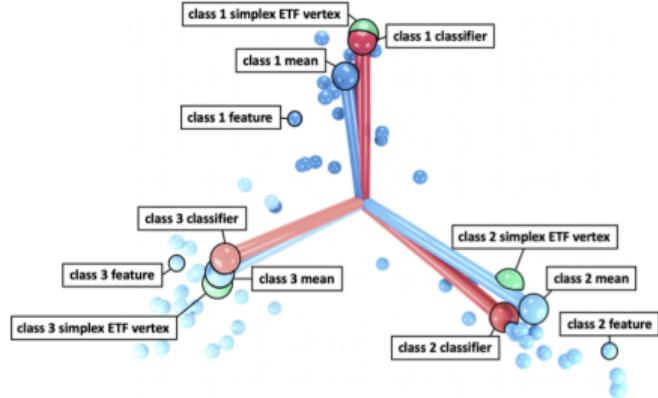
¹Image credit: blog.fastforwardlabs.com

Representation for classification

- Suppose the representation of input $x \in \mathbb{R}^D$ is done by function (e.g., Neural Net-based) $h(x) \in \mathbb{R}^N$
- Our goal is to classify x based on its representation $h(x)$ into one of K classes: $Y = \{0, \dots, K - 1\}$
- For this purpose, we need to project the (learned) representation into K *logits*, that later will be used as the inputs to SoftMax (and corresponding SoftMax loss) to produce probabilities
 - ▶ Logits are defined by $l = W * h(x) + b$, where W is the projection matrix $K \times N$, $b \in \mathbb{R}^K$ (often $b = 0$)
 - ▶ Output probabilities are $q = \text{SoftMax}(l) = \text{SoftMax}(W * h(x) + b)$
 - ▶ Linear classifier is $\arg \max_{c \in Y} \text{SoftMax}(l)_c \Leftrightarrow \arg \max_{c \in Y} \langle W_c, h(x) \rangle + b_c$, where W_i is the i -th row of W

Neural Collapse

- Recently it was found that in the terminal stage of training of any deep learning classifier (with Cross-Entropy loss) in the overparametrization regime, when ER is already zero but ERA is not, the average representations $h(x)$ tend to group around each class center
 - And even coincide up to rescaling to linear classifiers W_i
- This phenomena is called **Neural Collapse**, initially it was discovered for classifiers², and later found for regression too³



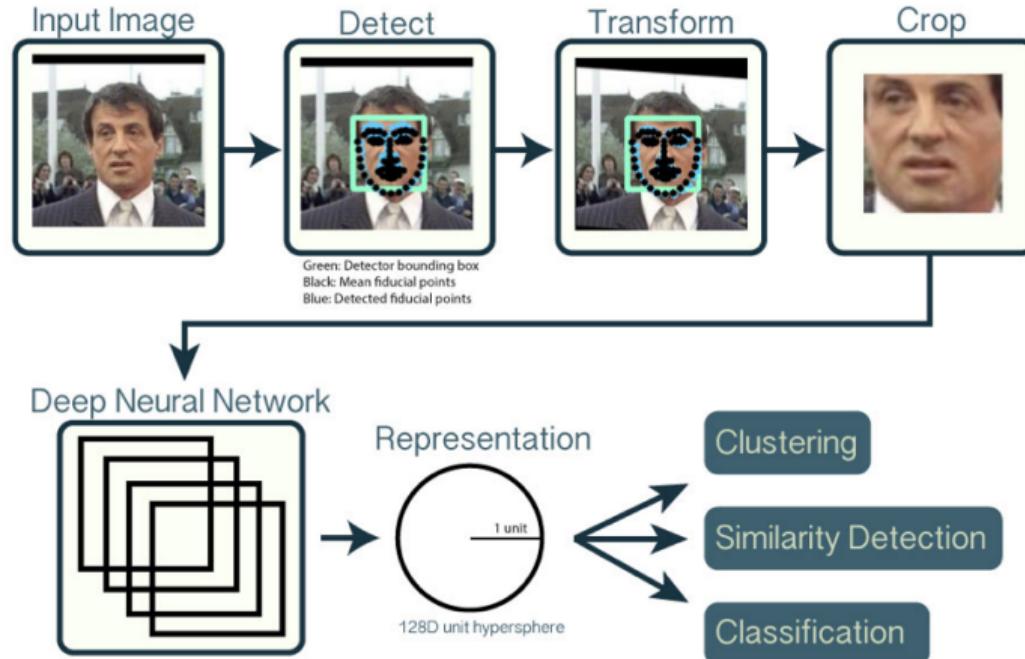
²V. Petyan et al. "Prevalence of neural collapse during the terminal phase of deep learning training." 2020.

³X. Y. Han et al. "Neural collapse under mse loss: Proximity to and dynamics on the central path." 2021.

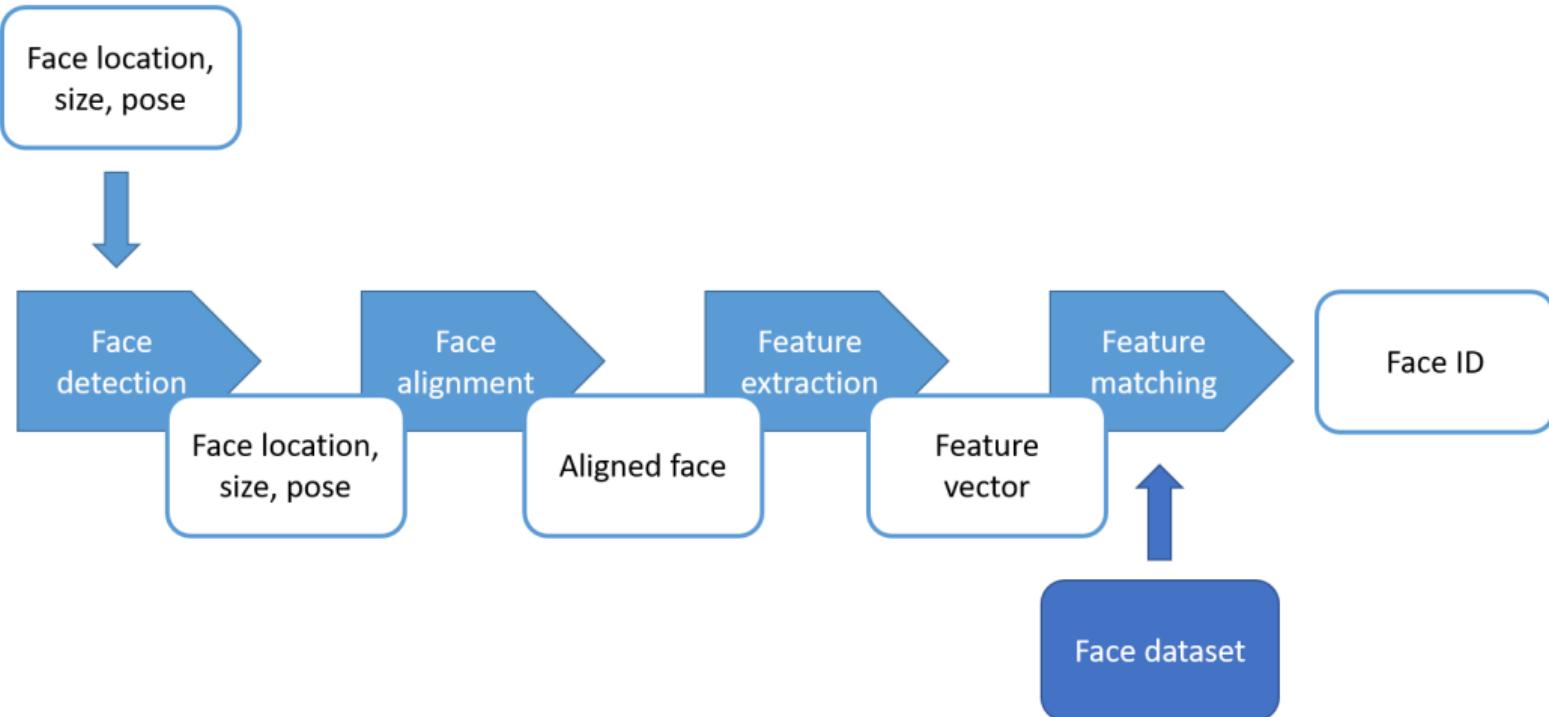
Face Recognition: Verification vs Identification

- ➊ Two main Face Recognition tasks: Verification and Identification (FaceID)
- ➋ **Verification:** check if the input face is the same as the original one
- ➌ **Identification:** find the identity from the ID dataset, or report the absence
- ➍ "FaceID" on phones is actually Verification
- ➎ FaceID is much harder than Verification
- ➏ Nowadays the number of IDs in the ID dataset is $\sim 1B$, and the training dataset can contain $\sim 10M$ of IDs
- ➐ It makes FaceID one of the most challenging applications of Representation Learning techniques

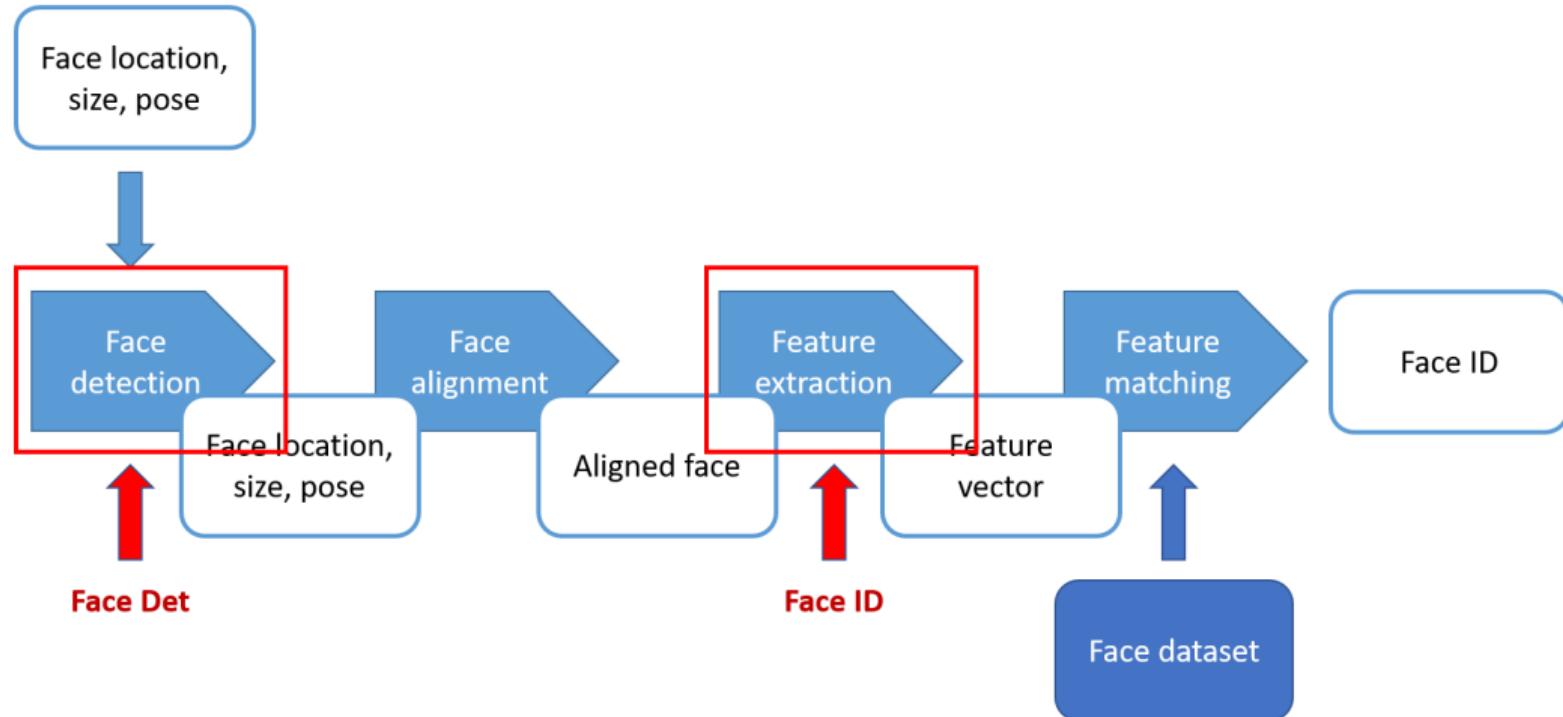
FaceID common pipeline



FaceDet and FaceID



FaceDet and FaceID



FaceID task

FaceID

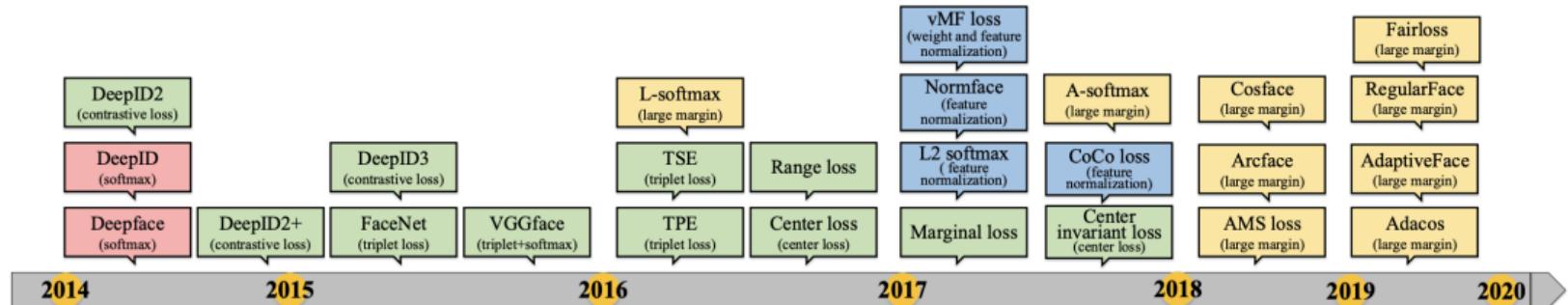
- Initially task formulation – Face Classification: assigning an input to a predefined class
- Later the task evolved to few-shot learning: very similar, but classes are known only during inference (not during training)
- Anyway, both are based on a simple classification task

FaceID task

- **Classification:** Given K classes (IDs) and input face x , find the closest class
$$\arg \max_{i \in \{0, \dots, K-1\}} \langle W_i, h(x) \rangle + b_i$$
- **Identification:** Given the dataset of m ID representations $H = (h_0, \dots, h_{m-1})$ input face x , find the closest ID based on some similarity function $s : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$:
$$\arg \max_{i \in \{0, \dots, m-1\}} s(h(x), h_i)$$
 or report the absence if $\max_{i \in \{0, \dots, m-1\}} s(x, h_i) < \theta$

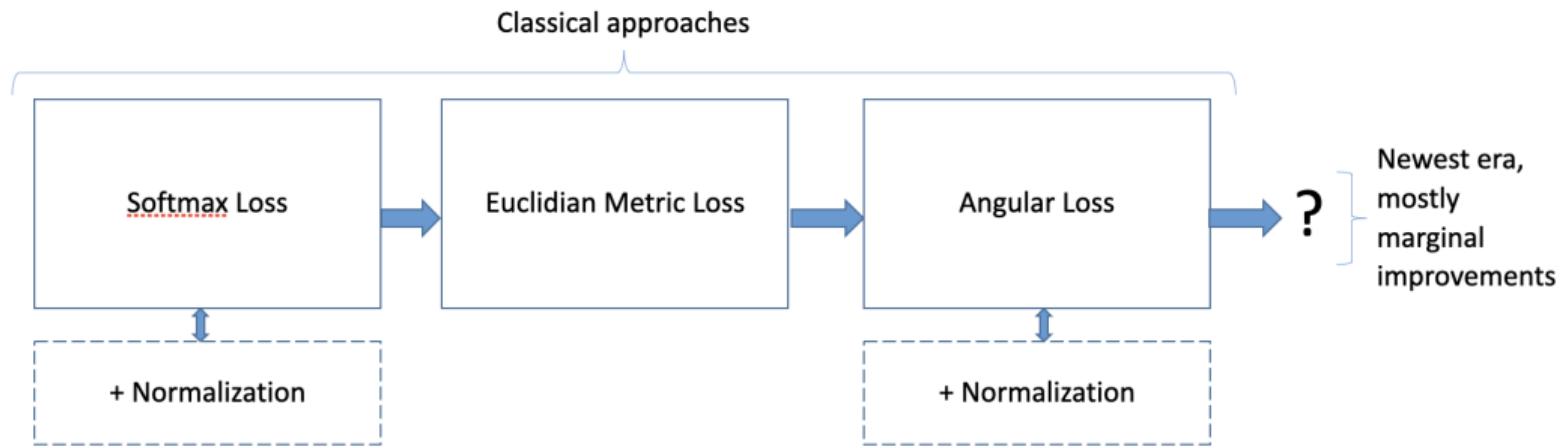
Historical overview

Evolution of losses to overview⁴:



⁴M. Wang et al. "Deep face recognition: A survey." 2018.

FaceID loss evolution: lecture scope



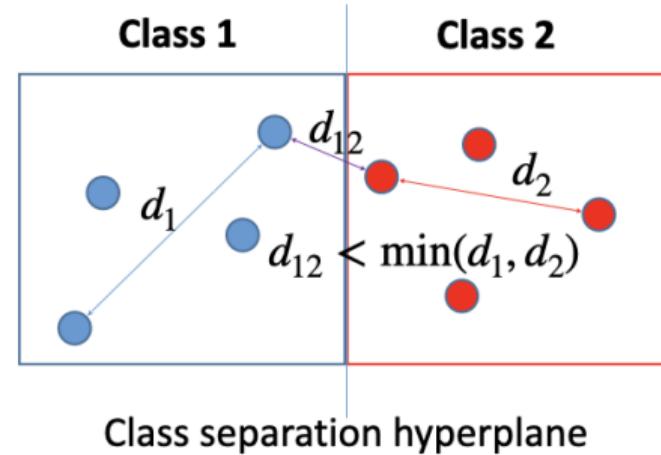
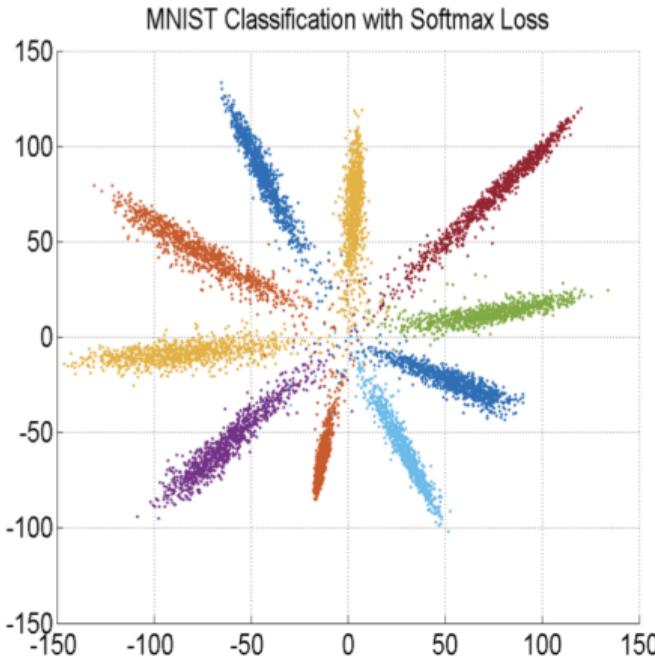
Softmax Loss

- In FaceID community, “*SoftMax (SM) Loss*” == Cross-entropy
- SM loss was the first loss type used for the FaceID task
 - ▶ Some examples are: DeepFace⁵, DeepID⁶
- In comparison to usual classification, here we have a larger number of classes
 - ▶ Class == person ID
 - ▶ Usually it is apprx 10-100K classes and even more (e.g., in surveillance applications)
- Intra-class variation can be bigger than inter-class difference

⁵Y. Taigman et al. "Deepface: Closing the gap to human-level performance in face verification." 2014.

⁶Yi Sun et al. "Deep learning face representation from predicting 10,000 classes." 2014.

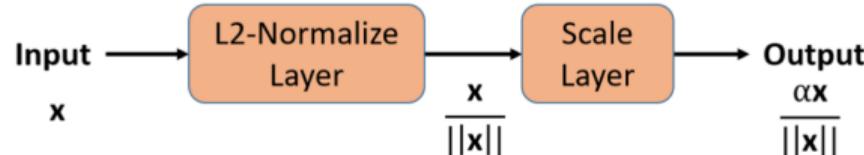
Softmax Loss: illustration



L_2 -SoftMax⁷

- Often cosine similarity is used: $s(x, y) = \frac{\langle h(x), h(y) \rangle}{\|h(x)\|_2 \|h(y)\|_2}$
- It causes some discrepancy between training loss (SM) and testing procedure, and it is empirically shown that best results are for the representations with high L_2 -norm
- Idea: introduce scale layer h' on top of representations $h(x)$ with one learnable parameter α so as: $h'(x) = \frac{\alpha h(x)}{\|h(x)\|_2}$
- The training loss:

$$L_{SM} = -\frac{1}{m} \sum_{i=0}^{m-1} \log \frac{e^{\langle W_{y_i}, h(x_i) \rangle + b_{y_i}}}{\sum_{j=1}^K e^{\langle W_j, h(x_i) \rangle + b_j}} \quad s.t. \|h(x_i)\|_2 = \alpha \quad \forall i = 0, \dots, m-1$$



⁷Rajeev R. et al. "L2-constrained softmax loss for discriminative face verification" 2017.

L_2 -SoftMax: results

- Learned α usually tends to be quite large ($\Rightarrow L_2$ -constraint is relaxed) \Rightarrow use it as an upper bound
- Fixing α to a small value leaves the hypersphere a small radius and less surface to spread the centroids
- Assuming the number of classes K to be lower than twice the representation dimension N , we can distribute the classes on a hypersphere of dimension N such that any two class centers are at least $\frac{\pi}{2}$ apart
- It leads to $p = \frac{e^\alpha}{e^\alpha + \sum_{j=1}^{K-2} e^0 + e^{-\alpha}} \approx \frac{e^\alpha}{e^\alpha + K - 2} \Rightarrow$ the lower bound to have allowable accuracy p is $\alpha_{low} = \log \frac{p(K-2)}{1-p}$
- In practice, fixed α closer to α_{low} is preferred for smaller datasets, and more close to learned α for larger ones (e.g., 10-100)

NormFace⁸: further normalization

- Consider the case of no bias term: $\arg \max_{i \in \{0, \dots, K-1\}} \langle W_i, h(x) \rangle$
- Simple Proposition: for any $s > 1$,

$$\max_{i \in \{0, \dots, K-1\}} \frac{e^{\langle W_i, sh(x) \rangle}}{\sum_{j=0}^{K-1} e^{\langle W_j, sh(x) \rangle}} \geq \max_{i \in \{0, \dots, K-1\}} \frac{e^{\langle W_i, h(x) \rangle}}{\sum_{j=0}^{K-1} e^{\langle W_j, h(x) \rangle}}$$

- It means that scaling the dot product $\langle W_i, h(x) \rangle$ makes the probability of class bigger
- Let's normalize both projection W_i **and** representation $h(x)$ by scaling factor l :
$$h'(x) = l \frac{h(x)}{\|h(x)\|_2}, W'_i(x) = l \frac{W_i(x)}{\|W_i(x)\|_2}$$
- At the same time, the following Proposition is true: for the Neural Collapse case, the lower bound on SM loss with normalized projections and representations is
$$\log(1 + (K - 1)e^{-\frac{K}{K-1}l^2})$$

Exercise: Prove these Propositions.

⁸F. Wang et al. "Normface: L2 hypersphere embedding for face verification." 2017

NormFace: training

- Taking into account the above Propositions, it makes sense to put a large l
- Also, omitting bias term b makes the training procedure as close as possible to the testing procedure with similarity function $s(x, W_i) = \frac{\langle h(x), W_i \rangle}{\|h(x)\|_2 \|W_i\|_2}$
- And the final proposal is not to fix a large l , but to have the single parameter $s = l^2$ a learnable one so as the final loss is the following:

$$L_{SM'} = -\frac{1}{m} \sum_{i=0}^{m-1} \log \frac{e^{s\langle W'_{y_i}, h'(x_i) \rangle}}{\sum_{j=1}^K e^{s\langle W'_j, h'(x_i) \rangle}}$$

CoCo Loss⁹: centroids

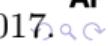
- Suppose we are using scaled representations: $h'(x) = \alpha \frac{h(x)}{\|h(x)\|_2}$
- Idea: instead of using learned W_i in the SoftMax, let's use the corresponding normalized centroids:

$$c_j = \frac{\sum_{i=0}^{m-1} \delta_j(y_i) h'(x_i)}{\sum_{i=0}^{m-1} \delta_j(y_i)}, \quad c'_j = \frac{c_j}{\|c_j\|_2}, \quad j = 0, \dots, K-1$$

- CoCo loss:

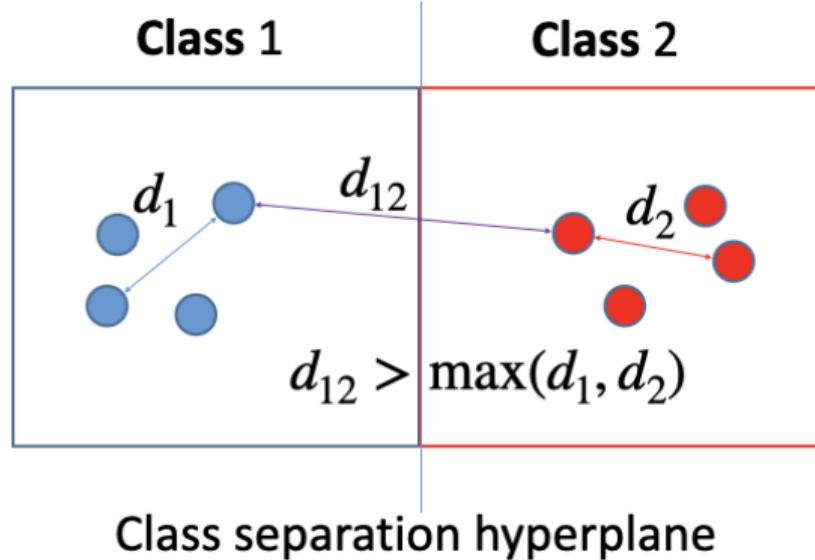
$$L_{CoCo} = -\frac{1}{m} \sum_{i=0}^{m-1} \log \frac{e^{s \langle c'_{y_i}, h'(x_i) \rangle}}{\sum_{j=0}^{K-1} e^{s \langle c'_j, h'(x_i) \rangle}}$$

Exercise. Given the upper bound for $L_{CoCo} < \epsilon$, prove the lower bound on $\alpha > \frac{1}{2} \log \frac{K-1}{e^\epsilon - 1}$

⁹Yu Liu et al. "Rethinking feature discrimination and polymerization for large-scale recognition." 2017 

Euclidean Metric Loss

- **Goal:** to increase inter-class difference and to decrease intra-class variance based not on probabilities, but on embeddings itself
 - ▶ Some examples are: DeepID2 (contrastive)¹⁰, FaceNet (triplet)¹¹



¹⁰Yi Sun et al. "Deep learning face representation by joint identification-verification." 2014.

¹¹F. Schroff et al. "Facenet: A unified embedding for face recognition and clustering." 2015.

Contrastive Learning¹³

- Setting: Face Verification
- Labeling: $y_{ij} = 0$ means that pair of faces (x_i, x_j) from different persons, $y_{ij} = 1$ means that pairs of faces from the same person
- Measure of 'closeness': Euclidean distance on top of face representations
 $s(x_i, x_j) = \|h(x_i) - h(x_j)\|_2$
- Empirical observations:
 - ▶ Need a sort of Hinge Loss for different IDs (no need to make their representations going to infinity)
 - ▶ No need in Hinge Loss for the same IDs (otherwise the representations will be concentrated near the margin boundary)
 - ▶ Need a margin to have non-trivial representations (w/o collapse)¹²

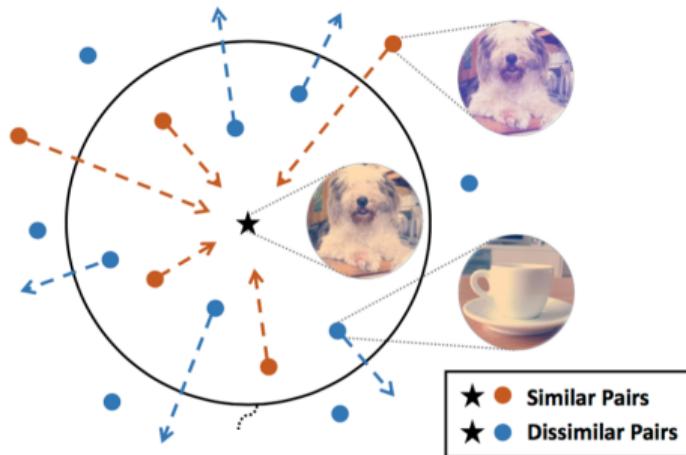
¹²Y. LeCun et al. "A tutorial on energy-based learning." 2006.

¹³S. Chopra et al. "Learning a similarity metric discriminatively, with application to face verification." 2005.

Max Margin Contrastive Loss

- Denote as $m > 0$ the margin for 'negative' (e.g. from different ID) pair of faces
- The final loss for the pair (x_i, x_j) is:

$$L_{MMC}(x_i, x_j) = \frac{1}{2}y_{ij} \cdot s(x_i, x_j)^2 + \frac{1}{2}(1 - y_{ij}) \cdot \max(0, m - s(x_i, x_j))^2$$



Marginal Loss¹⁴: one non-linearity with two margins

- Let's concentrate on the farthest intra-class samples and the nearest inter-class samples to compute a margin loss:

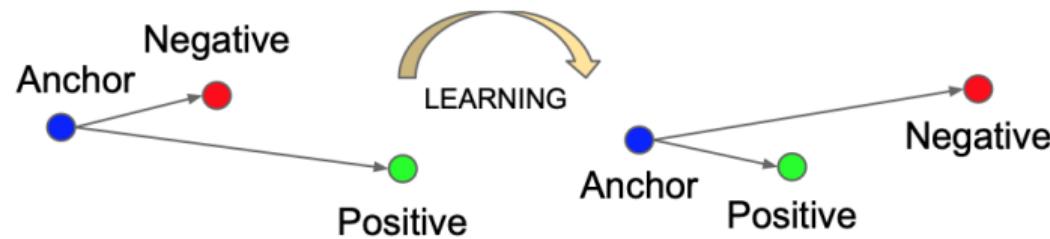
$$L_M = \frac{1}{m^2 - m} \sum_{i < j} \max \left(0, \xi - y_{ij} \left(\theta - \left\| \left\| h(x_i) \right\|_2 - \frac{h(x_j)}{\|h(x_j)\|_2} \right\|_2^2 \right) \right)$$

- where $y_{ij} = +1$ if representations from the same ID and $y_{ij} = -1$ vice versa
- θ : threshold for distinguishing whether representations are from the same class
- ξ : error margin for the classification hyperplane

¹⁴J. Deng et al. "Marginal loss for deep face recognition." 2017.

Triplets instead of pairs

- For contrastive learning, we concentrated on pairs: 'positive' representations should be as close as possible and 'negative' should be not closer than a predefined threshold
- And these distances (as well as thresholds) are absolute ones
- But actually we need for every input representation (called 'anchor') to be closer to the positive embedding than to the negative one: relative difference of distances



Triplet Loss

- In FaceNet¹⁵ the concept of triplets was used to define the Triplet Loss:

$$L_T = \frac{1}{m} \sum_{i=0}^{m-1} \max \left(0, \|h(x_i^a) - h(x_i^p)\|_2^2 - \|h(x_i^a) - h(x_i^n)\|_2^2 + \alpha \right)$$

- ▶ $x_i = x_i^a$: anchor, input face
- ▶ x_i^p : positive face (of the same ID as x_i^a)
- ▶ x_i^n : negative face (of the different ID from x_i^a)
- ▶ all representations are normalized: $\|h(x)\|_2 = 1$
- ▶ α : hyperparameter for margin (0.2 used)

¹⁵F. Schroff et al. "Facenet: A unified embedding for face recognition and clustering." 2015.

Triplet selection process

- Crucial how to construct triplets (mostly negative examples) for Triplet loss

Easy triplets

Triplets with zero loss, i.e.

$$\|h(x_i^a) - h(x_i^p)\|_2^2 + \alpha < \|h(x_i^a) - h(x_i^n)\|_2^2$$

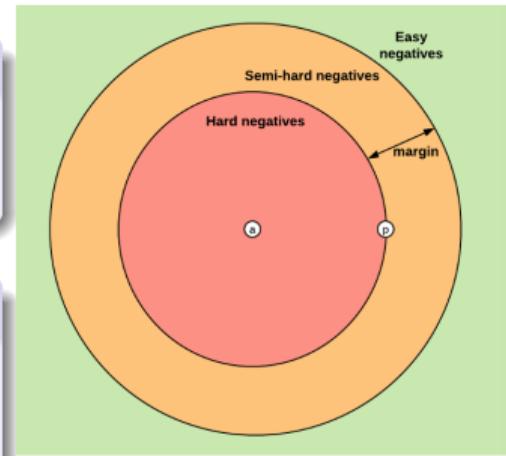
Hard triplets (used for FaceNet)

Triplets where the negative is closer to the anchor than the positive, i.e. $\|h(x_i^a) - h(x_i^n)\|_2^2 < \|h(x_i^a) - h(x_i^p)\|_2^2$

Semi-hard triplets

Triplets where the negative is not closer to the anchor than the positive, but still having the non-zero loss, i.e.

$$\|h(x_i^a) - h(x_i^p)\|_2^2 < \|h(x_i^a) - h(x_i^n)\|_2^2 < \|h(x_i^a) - h(x_i^p)\|_2^2 + \alpha$$



Better triplets

- Idea: to project the representations (even as a post-processing!) from arbitrary FaceID NN and to learn these projection matrices T : $h'(x) = T * h(x)$
 - TSE¹⁶ approach:

$$L_{TSE} = \frac{1}{m} \sum_{i=0}^{m-1} \max \left(0, \alpha + \langle h'(x_i^a), h'(x_i^n) \rangle - \langle h'(x_i^a), h'(x_i^p) \rangle \right)$$

- TPE¹⁷ approach:

$$L_{TPE} = -\frac{1}{m} \sum_{i=0}^{m-1} \log \frac{e^{\langle h'(x_i^a), h'(x_i^p) \rangle}}{e^{\langle h'(x_i^a), h'(x_i^p) \rangle} + e^{\langle h'(x_i^a), h'(x_i^n) \rangle}}$$

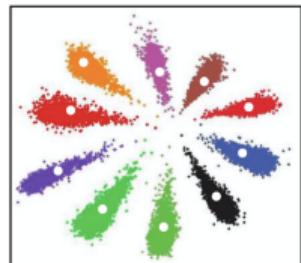
- Empirical observation: very unstable

¹⁶S. Sankaranarayanan et al. "Triplet similarity embedding for face verification." 2016.

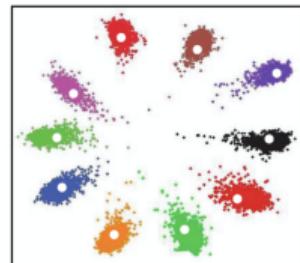
¹⁷S. Sankaranarayanan et al. "Triplet probabilistic embedding for face verification and clustering." 2016.

Center loss¹⁸

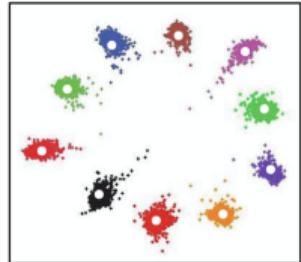
- Idea: to reduce intra-variance
- Additional loss term:
$$L_C = \frac{1}{2} \sum_{i=0}^{m-1} \|h(x_i) - c_{y_i}\|_2^2$$
- Centers $c_i, i = 0, \dots, K - 1$ can be initialized with W_i and updated later with the fine-tuning process
- Final loss is $L_{SM} + \lambda L_c$



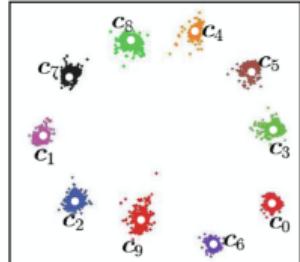
(a) $\lambda = 0.001$



(b) $\lambda = 0.01$



(c) $\lambda = 0.1$



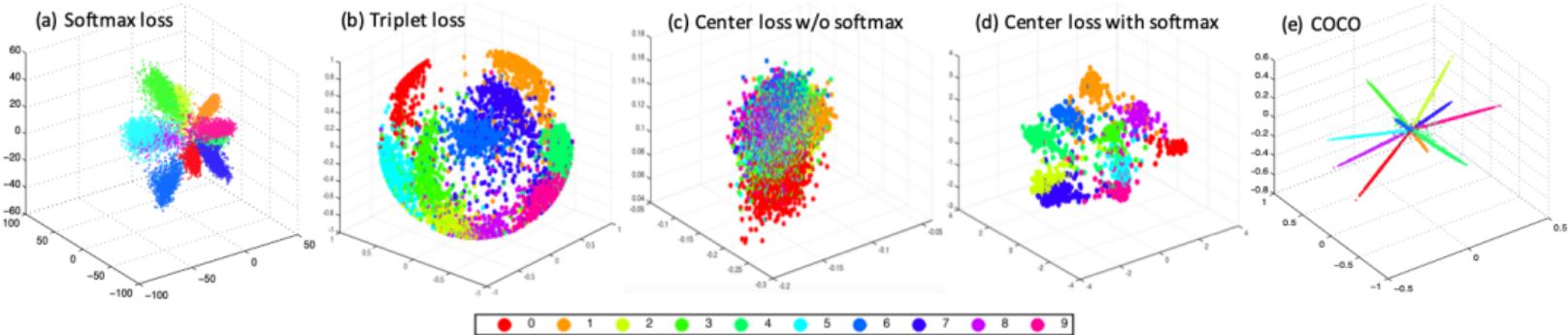
(d) $\lambda = 1$

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

¹⁸Y. Wen et al. "A discriminative feature learning approach for deep face recognition." 2016.

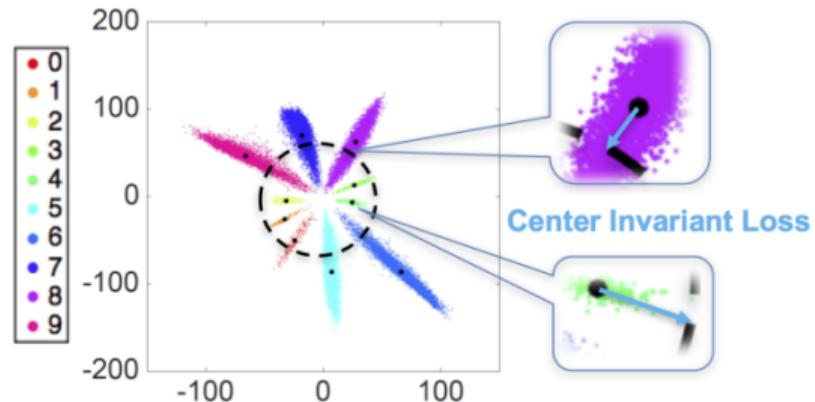
SM-based and some metric-based losses comparison



Imbalance problem: Center Invariant (CI) Loss¹⁹, Range (R) Loss²⁰

- CI Idea: normalize (implicitly) class centers
- Done by additional loss term to Center Loss
- R Idea: minimize intra-class volume **and** maximize inter-class distance
- $L_{R_{intra}} = \sum_{i=0}^{K-1} \frac{r}{\sum_{j=1}^r \frac{1}{D_i^j}}$, where D_i^j is the j -th largest pairwise representation distance inside class i
- $L_{R_{inter}} = \max(0, m - \min_{i < j} \|c_i - c_j\|_2^2)$

$$L_{CI} = \frac{1}{4} \sum_{i=0}^{m-1} \left(\|c_{y_i}\|_2^2 - \frac{1}{K} \sum_{j=0}^{K-1} \|c_j\|_2^2 \right)$$

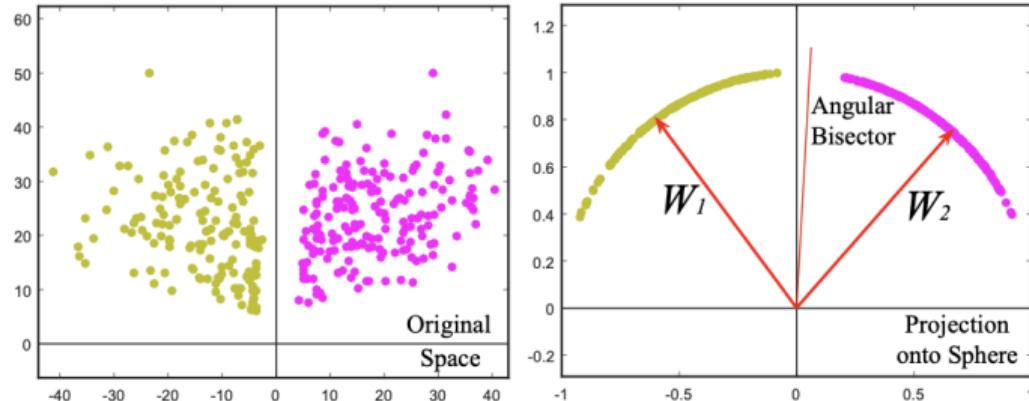


¹⁹Yue Wu et al. "Deep face recognition with center invariant loss." 2017.

²⁰X. Zhang et al. "Range loss for deep face recognition with long-tailed training data." 2017.

Angular Loss

- **Goal:** to perform the separation not in Euclidean space of embeddings, but in the space of embedding angles θ
 - ▶ Some examples are: L-Softmax²¹, A-Softmax (SphereFace)²², AM-Softmax²³ (CosFace²⁴)



²¹W. Liu et al. "Large-margin softmax loss for convolutional neural networks." 2016.

²²W. Liu et al. "Sphereface: Deep hypersphere embedding for face recognition." 2017.

²³F. Wang et al. "Additive margin softmax for face verification." 2018.

²⁴H. Wang et al. "Cosface: Large margin cosine loss for deep face recognition." 2018.

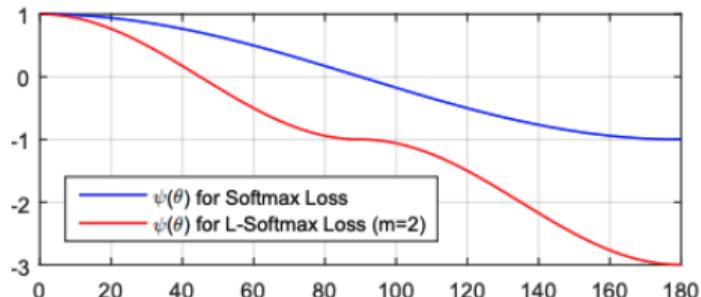
L-Softmax²⁵ idea

- Let's omit bias term b and reformulate the logit l : $l_i = \langle W_i, h(x) \rangle = \|W_i\| \|h(x)\| \cos \theta_i$, where $\theta_i \in [0, \pi]$ is the angle between projection vector W_i and representation $h(x)$
- So SM loss becomes $L_{SM}(x_i) = -\log \frac{e^{\|W_{y_i}\| \|h(x_i)\| \cos \theta_{y_i}}}{\sum_j e^{\|W_j\| \|h(x_i)\| \cos \theta_j}}$
- For binary classification with the correct class 1, the original SM forces $\langle W_1, h(x) \rangle > \langle W_2, h(x) \rangle$, i.e. $\|W_1\| \|h(x)\| \cos \theta_1 > \|W_2\| \|h(x)\| \cos \theta_2$
- Idea: let's do the task even harder: $\|W_1\| \|h(x)\| \cos m\theta_1 > \|W_2\| \|h(x)\| \cos \theta_2$, $\theta_1 \in [0, \frac{\pi}{m}]$, $m \in \mathbb{N}$
- In this case we'll automatically get the needed $\|W_1\| \|h(x)\| \cos \theta_1 > \|W_2\| \|h(x)\| \cos \theta_2$
 - just because $\|W_1\| \|h(x)\| \cos \theta_1 \geq \|W_1\| \|h(x)\| \cos m\theta_1 > \|W_2\| \|h(x)\| \cos \theta_2$

²⁵W. Liu et al. "Large-margin softmax loss for convolutional neural networks." 2016.

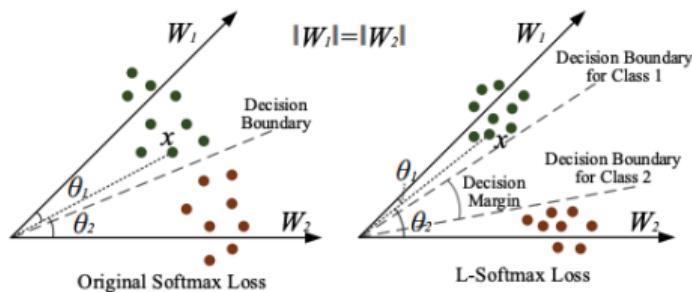
L-Softmax

- The new L-SM: $L_{L-SM}(x_i) = -\log \frac{e^{\|W_{y_i}\| \|h(x_i)\| \psi(\theta_{y_i})}}{e^{\|W_{y_i}\| \|h(x_i)\| \psi(\theta_{y_i})} + \sum_{j \neq y_i} e^{\|W_j\| \|h(x_i)\| \cos \theta_j}}$
 - Where $\psi(\theta) = \cos m\theta, \theta \in [0, \frac{\pi}{m}]$,
 - $\psi(\theta) = D(\theta), \theta \in (\frac{\pi}{m}, \pi], D(\frac{\pi}{m}) = \cos m \frac{\pi}{m}$, $D(\theta)$ – monotonically decreasing function
 - Let's use $\psi(\theta) = (-1)^k \cos m\theta - 2k, \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}], k \in [0, m-1]$
 - In practice authors provided the relaxed strategy ($SM \rightarrow L-SM$) gradually reducing parameter λ from a large value (1000) to a small value (5):
$$\psi(\theta) = \frac{(-1)^k \cos m\theta - 2k + \lambda \cos \theta}{\lambda + 1}, \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}], k \in [0, m-1]$$



L-Softmax: decision rule

- Suppose the norms of projection vectors are the same: $\|W_1\| = \|W_2\| \Rightarrow$ the initial SM rule is $\theta_1 < \theta_2$ (case $\|W_1\| \neq \|W_2\|$ is more complicated, but generally similar)
- For the L-SM case, it means $m\theta_1 < \theta_2$
 - ▶ And the same for the second class: $m\theta_2 < \theta_1$
- Suppose α is the angle between W_1 and W_2 . It leads to the following:
 - ▶ Decision boundary for the class 1: $m\theta_1 = \theta_2, \theta_1 + \theta_2 = \alpha \Rightarrow \theta_1 = \alpha \frac{1}{m+1}$
 - ▶ Decision boundary for the class 2: $\theta_2 = \alpha \frac{1}{m+1}$
- Note: decision boundaries are different for classes 1 and 2 (cf. to SM)
- Decision margin is $\alpha_M = \alpha - (\theta_1 + \theta_2) = \alpha \frac{m-1}{m+1}$ (e.g. if $\alpha = 180^\circ, m = 2 \Rightarrow \alpha_M = 60^\circ$)
- Recall: SVM margin



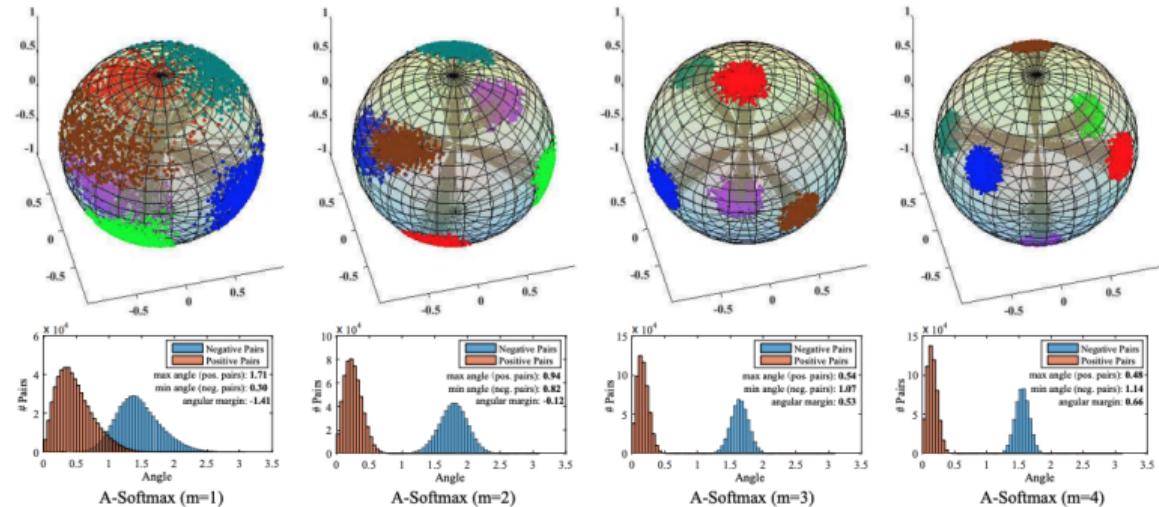
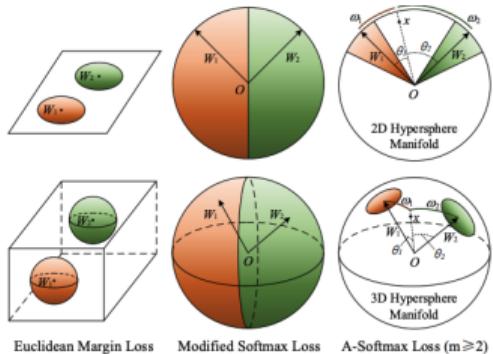
A-Softmax²⁷: normalization

- Explicitly normalize projection vectors $\|W_i\| = 1$ and zero biases $b_i = 0 \quad \forall i$: it is called '**Modified SM Loss**'
- Introduce angular margin ($m\theta$): '**A-SM Loss**'
- Binary case:
 - ▶ Max intra-class angle is $\alpha_{intra} = \frac{\alpha}{m-1} + \frac{\alpha}{m+1}$ (depending on whether the representation is lying inside or outside angle α)
 - ▶ Min inter-class angle is $\alpha_{inter} = \alpha \frac{m-1}{m+1}$
 - ▶ $\alpha_{intra} \leq \alpha_{inter} \Rightarrow m \geq 2 + \sqrt{3}$
- For multi-class case $m \geq 3$ ²⁶
 - ▶ For experiments makes sense to use $m = 4$

²⁶Prove it (use assumption of W_i uniform distribution)

²⁷W. Liu et al. "Sphereface: Deep hypersphere embedding for face recognition." 2017.

A-Softmax: illustration

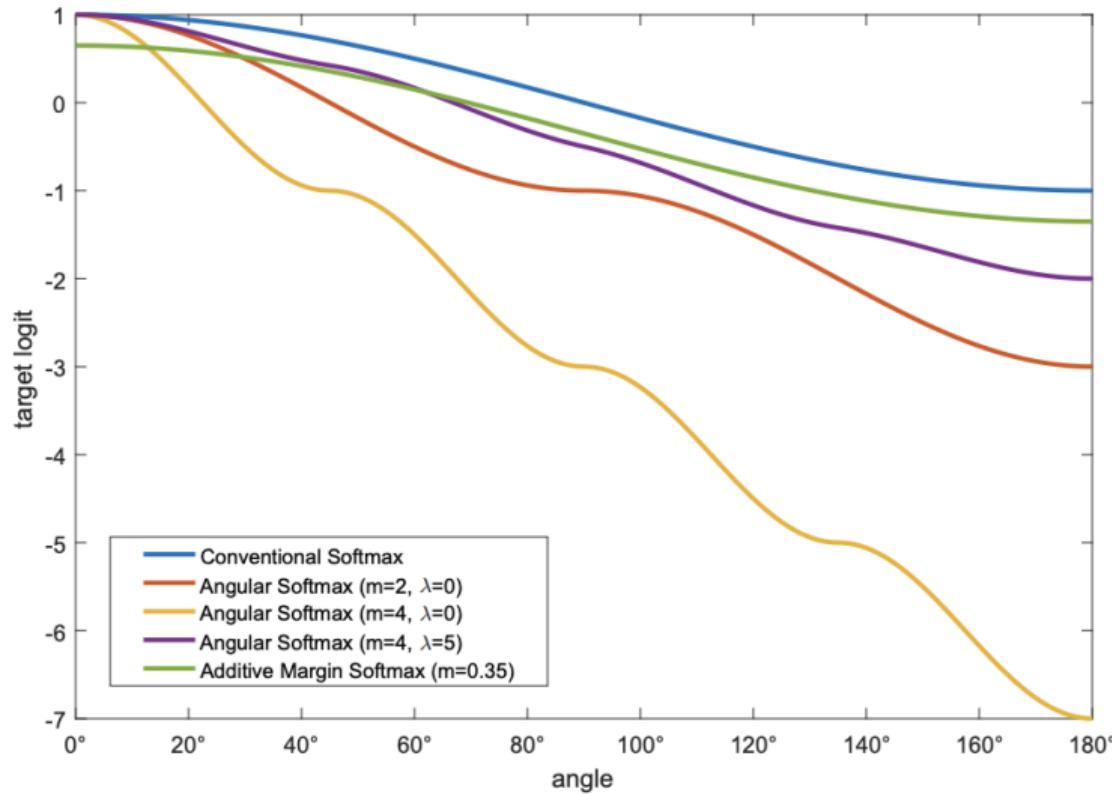


AM-Softmax²⁸

- Additive Margin SM idea: add margin, but additive (not multiplicative) and outside of $\cos \theta$: $\psi(\theta) = \cos \theta - m$
- Normalization: like NormFace, normalize both representations $h(x)$ and projection W_i and scale both of them to l , so as $s = l^2 = \|W'_i\| \|h'(x)\|$
- AM Loss: $L_{AM-SM}(x_i) = -\log \frac{e^{s\psi(\theta y_i)}}{e^{s\psi(\theta y_i)} + \sum_{j \neq y_i} e^{s \cos \theta_j}}$
- In practice: use $s = 30, m = 0.4$, no any relaxation strategy (no hyperparam λ)

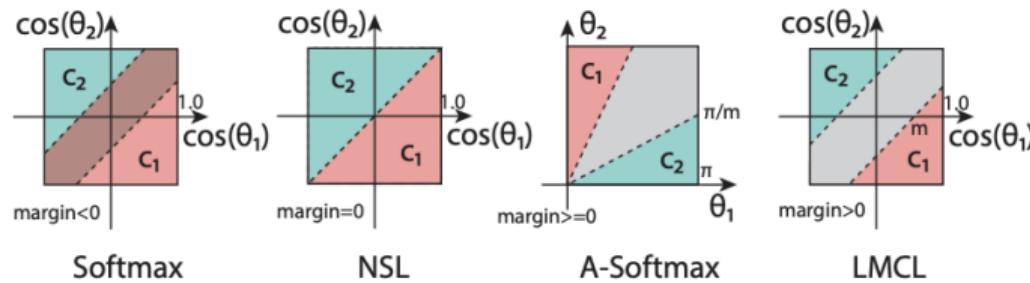
²⁸F. Wang et al. "Additive margin softmax for face verification." 2018.

AM-Softmax: visualization



CosFace³⁰

- The very same idea as for AM-SoftMax, but with a grain of theory
- Lower bound of SM Loss is $-\log P \geq \log(1 + (K - 1)e^{-\frac{K}{K-1}s})$ (refer to NormFace) $\Rightarrow s \geq \frac{K}{K-1} \log \frac{P(K-1)}{1-P}$
- Bounds for cosine margin: $0 \leq m \leq \frac{K}{K-1}$ ²⁹ (in practice $m = 0.35$ was used)

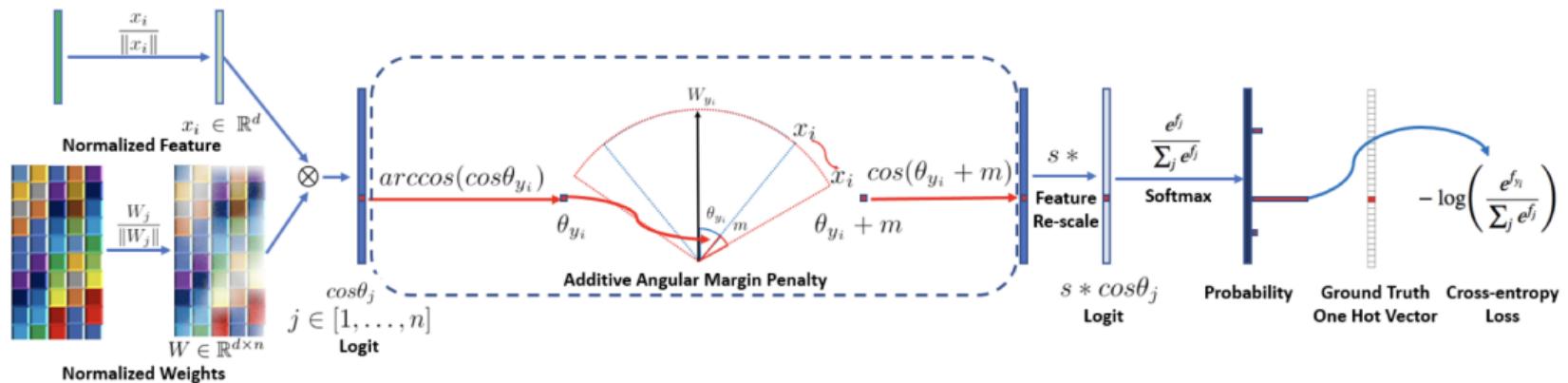


²⁹Prove it. Hint: use $0 \leq m \leq 1 - \max_{i \neq j} W_i W_j$

³⁰H. Wang et al. "Cosface: Large margin cosine loss for deep face recognition." 2018.

Arcface: best of Angular Losses

- Arcface³¹ is most probably the most successful implementation of Angular Loss idea

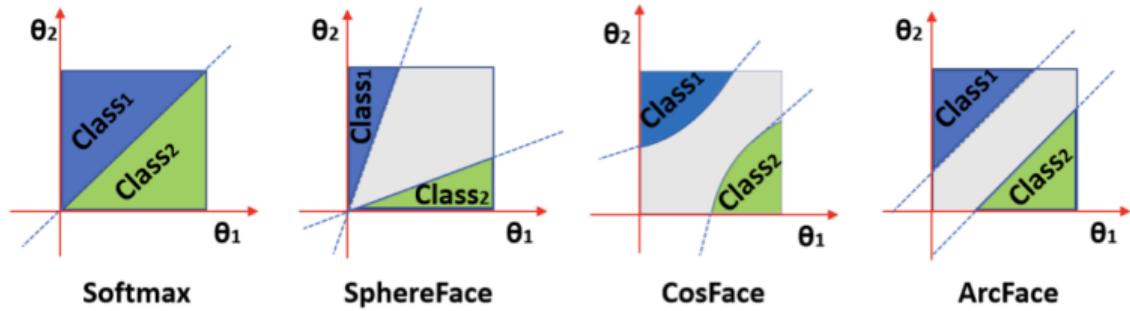
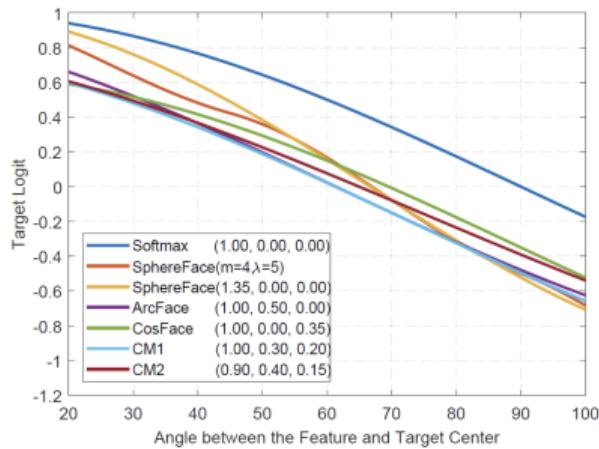


³¹J. Deng et al. "Arcface: Additive angular margin loss for deep face recognition." 2019.

Arcface: details

- Idea: additive margin **inside** $\cos \theta$: $\psi(\theta) = \cos(\theta + m)$ (in practice $m = 0.5$ is used)
- **ArcFace Loss**: $L_{Arcface}(x_i) = -\log \frac{e^{s\psi(\theta_{y_i})}}{e^{s\psi(\theta_{y_i})} + \sum_{j \neq y_i} e^{s \cos \theta_j}}$ (in practice $s = 64$ is used)
- And the final step: let's combine all margins!
- The function for **Combined Margin Loss**: $\psi(\theta) = \cos(m_1\theta + m_2) - m_3$

Final Loss comparison



Takeaway notes

- ➊ Representations learned are one of the most valuable outcomes of Deep Learning and Neural Nets in particular
- ➋ Given the correct representation, the final classifier is (surprise!) just a usual linear discriminate function
- ➌ Normalization of representation and projection vector is important
- ➍ Metric learning is unstable, but can be really inevitable (for Contrastive learning w/o labels)
- ➎ Angular-based margins now are the major and dominant forms of separation for representations (angular analogy of SVM)
- ➏ Overviewing the historical perspective, we can better understand the reasoning behind the design choices

Thank you!