Face Classification & Verification

11-785 Introduction to Deep Learning

Recitation 6

Fall 2019

Recap

You have seen:

✓The different Regularization methods that will be very handy with the assignment.

✓The mathematical definition of a convolutional neural network (CNN)

Logistics

You should be able to download the notebook from the course page.

- ✓**Recitation-6.ipynb:** will cover a customizable Dataset and Dataloader from PyTorch, some of the architectures discussed in this recitation and some of the Learning objective functions

Overview

1. Problem Statement

- Verification vs Classification
- Open set vs closed set
- Transfer learning

2. Model Architectures

- Experimental process
- 2D-CNN
- Alex Net
- VGG Net
- ResNet
- DenseNet
- Shuffle Net
- Mobile Net

3. Learning Objective

- Area Under the Curve (AUC)
- Transfer learning
- Contrastive loss
- Margin problem
- Center loss
- Triplet loss
- Softmax (revisited)
- Angular softmax

4. Data Explanation

- Detailed on given Dataset
- Torch Dataset (Notebook)
- Torch DataLoader (Notebook)

1. Problem Statement

Face Classification

- Classifying the person id using the image of a person’s face.

Face Verification

- Use transfer learning to design a system for Face Verification
- The images contain a lot of facial features that vary across different people
- The main task is to train a CNN to extract and represent such important features from an image.
- The extracted features will be represented in a fixed-length vector of features, known as a face embedding
- Given a pair of face embeddings, we need to identify if they belong to the same person

Classification vs Verification

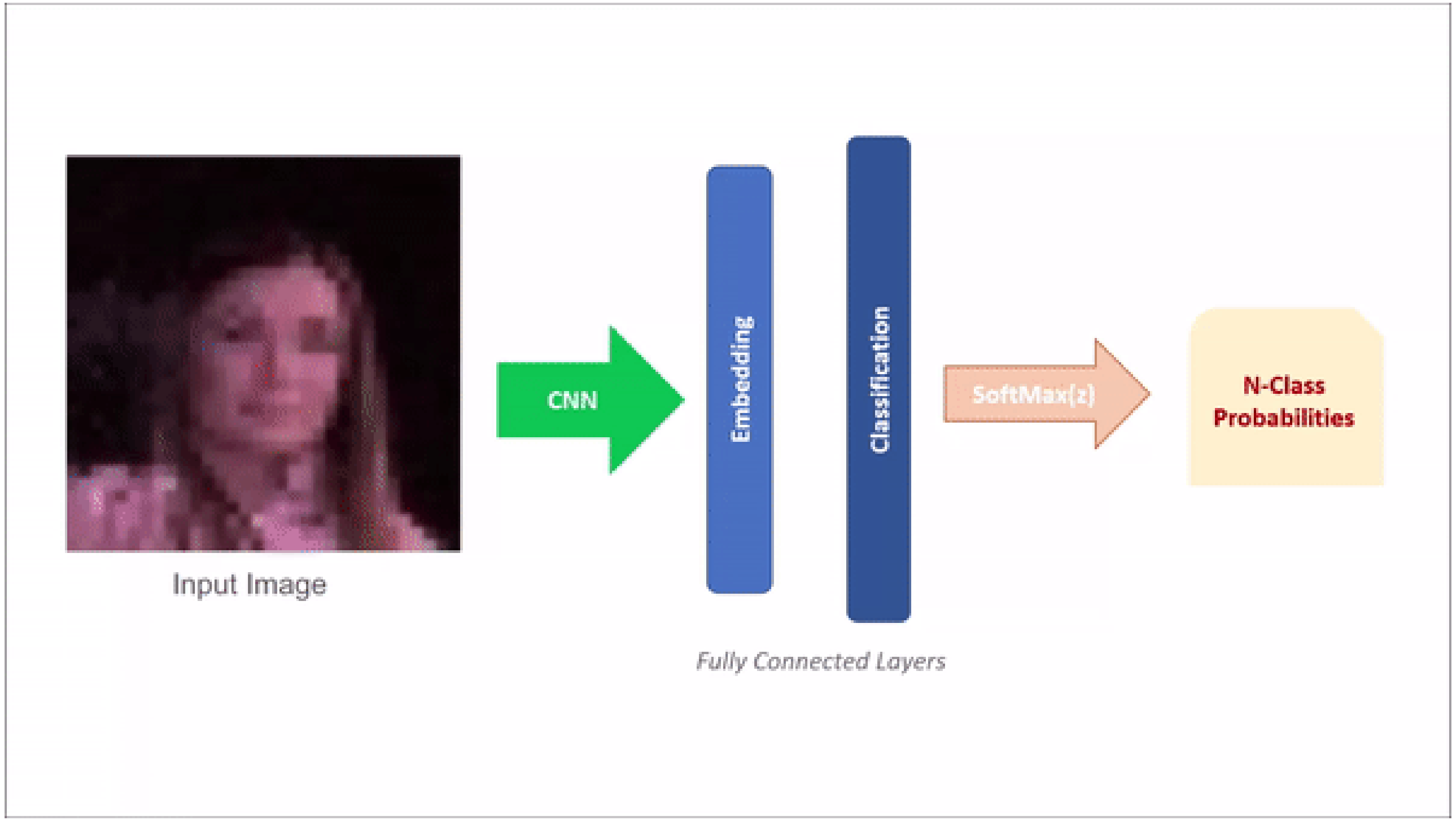
Classification

- N-way classification task. Predict from a closed set of labels.

Verification

- It's a 1-to-n matching where given a sample, you match it to the closest sample among n other samples.
- Can also be performed as a 1-to-1 task where we verify if they belong to the same class of data or not.

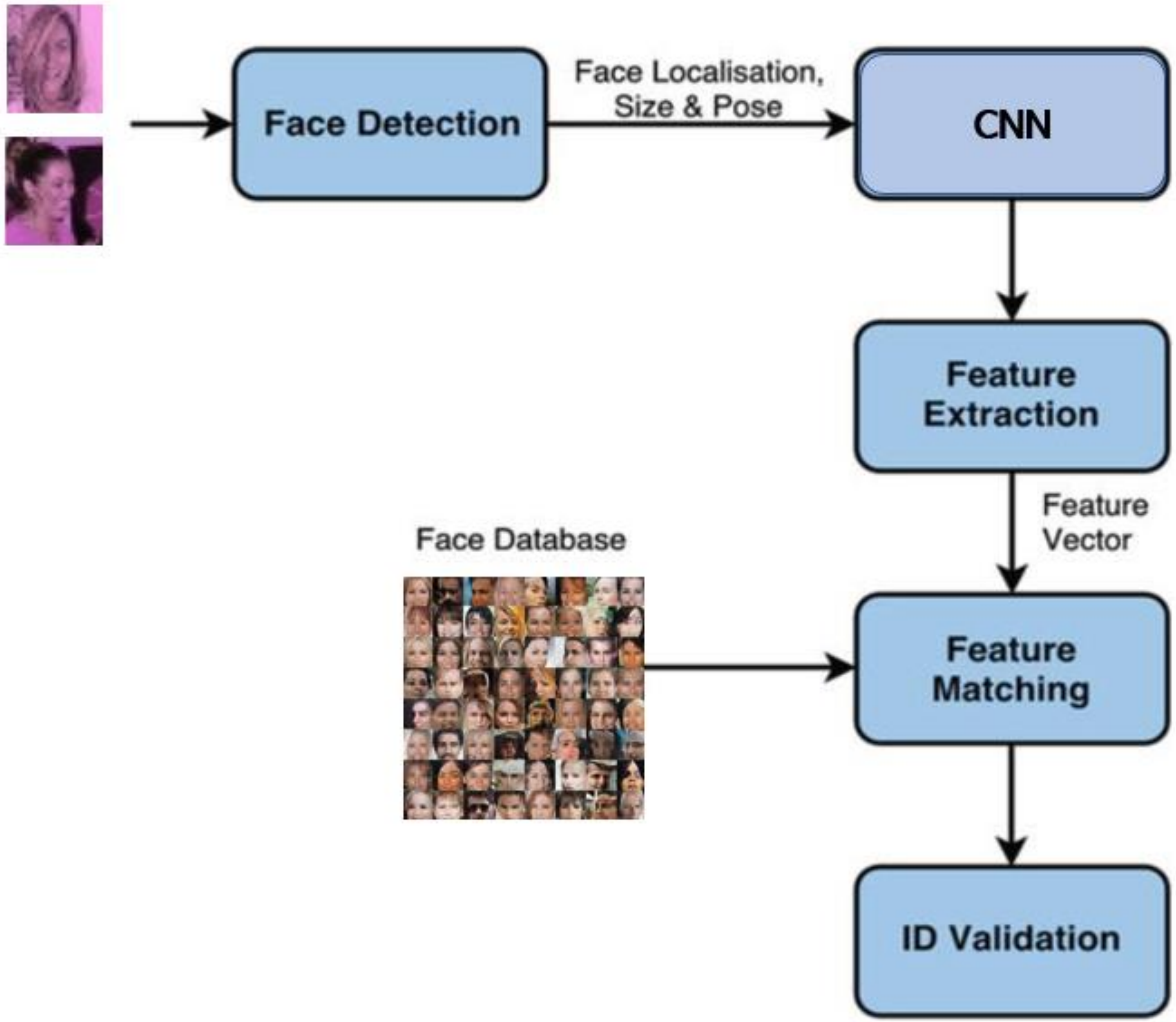
Face Classification



N-way classification where N is the number of people present in the training set

- Cross-Entropy objective
- Applied per sample

Face Verification



1.2 Open Set vs Closed Set

Open Set

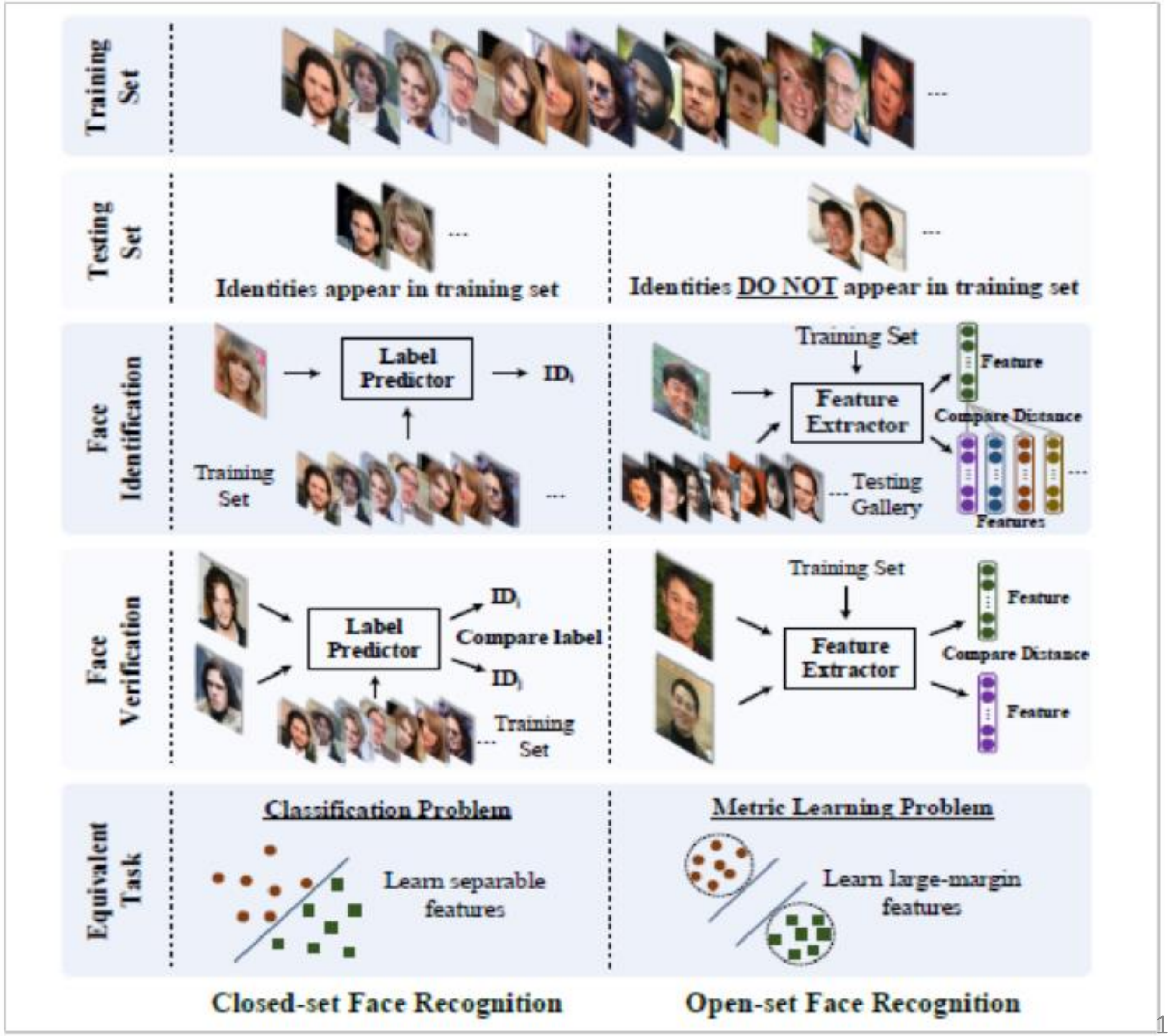
- Learn features instead of data classes
- Training dataset is not exhaustive
- Used as a metric learning problem
- Face Verification is Open Set

Closed Set

- Treated as a classification problem
- Training dataset is assumed as exhaustive
- All possible data classes are present in the training set
- Identification is among previously defined classes
- Face Classification is Closed set

Open Set vs Closed Set

Classification versus Verification & Open versus Closed set for the facial recognition problem.



1.3 Transfer Learning

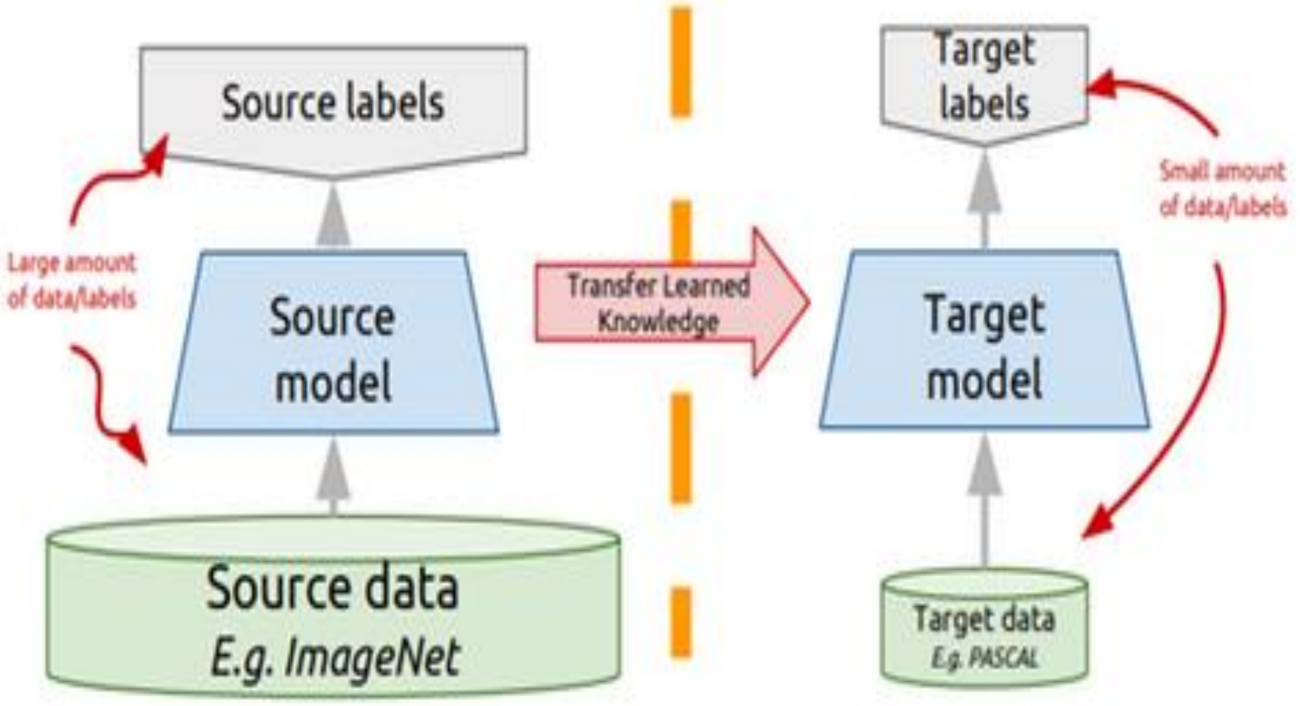
Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task

√ Develop Model Approach

- Select a source task
- Develop source model
- Reuse model
- Tune model

√Pre-trained Model Approach

- Select source model (usually pretrained)
- Reuse model
- Tune model



Overview

1. Problem Statement

- Verification vs Classification
- Open set vs closed set
- Transfer learning

2. Model Architectures

- Experimental process
- 2D-CNN
- Alex Net
- VGG Net
- ResNet
- DenseNet
- Shuffle Net
- Mobile Net

3. Learning Objective

- Area Under the Curve (AUC)
- Transfer learning
- Contrastive loss
- Margin problem
- Center loss
- Triplet loss
- SoftMax (revisited)
- Angular softMax

4. Data Explanation

- Detailed on given Dataset
- Torch Dataset (Notebook)
- Torch DataLoader (Notebook)

2. Model Architectures

CNNs in PyTorch

Widely used in ML and CV as it achieves impressive results when applying computational techniques in visual content.

✓Conv2d

- Set number of channels to 3 (RGB)
- While average pooling, average features across both dimension (H, W) to get a 2D vector

✓Tapering

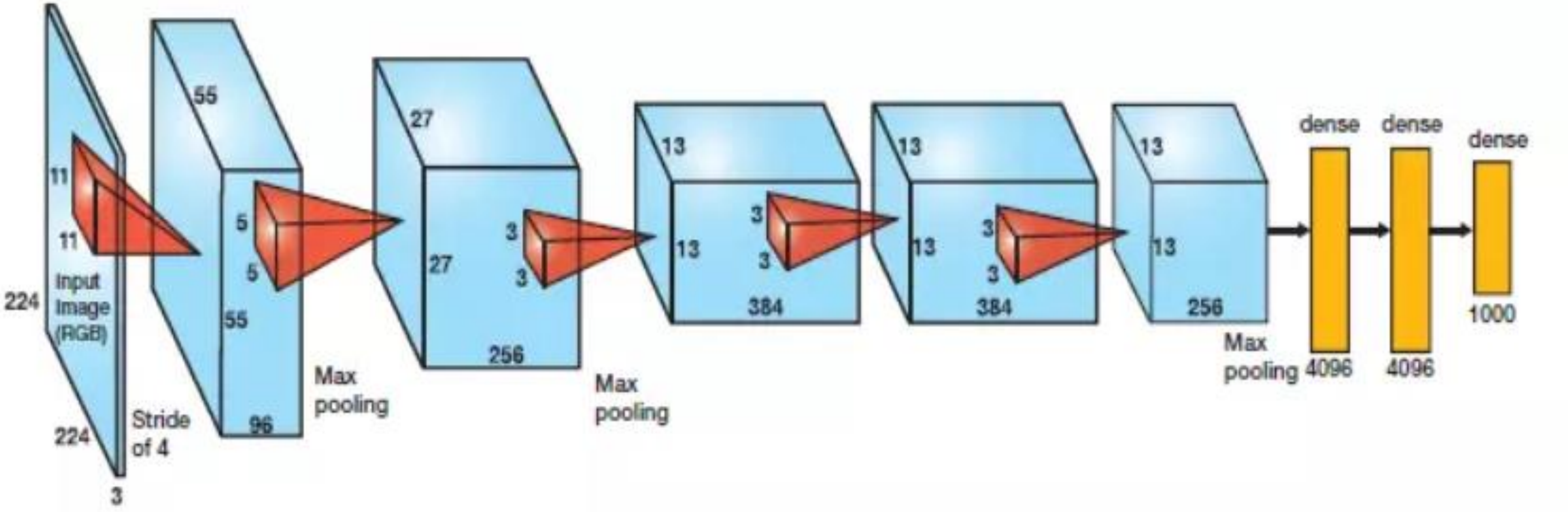
- Memory Optimization Technique: Layers near the input can have higher filter sizes and larger strides

- Reduce filter sizes and strides as the network goes deeper

CNN Architectures: AlexNet

AlexNet Published on 2012 for which it won ImageNet LSVRC-2012

- ✓ Uses ReLU as it decreases training time.
- ✓ Uses data augmentation techniques.
- ✓ Implements dropout layers.
- ✓ Trains the model using batch SGD, with specific values for momentum and weight decay.

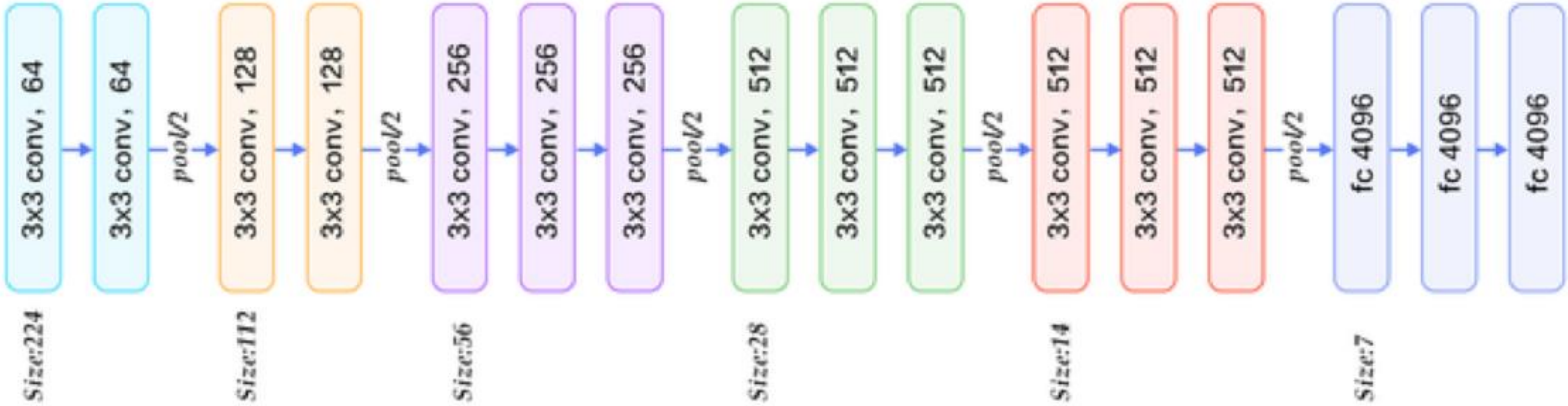


https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

VGGNet

VGG (2014) architecture reduces the size of each layer yet increases the overall depth of it. reinforces the idea that CNNs must be deep in order to work well on visual data.

- ✓ Convolutions layers (used only 3*3 size)
- ✓ Max pooling layers (used only 2*2 size)
- ✓ Fully connected layers at end
- ✓ Total 16 layers

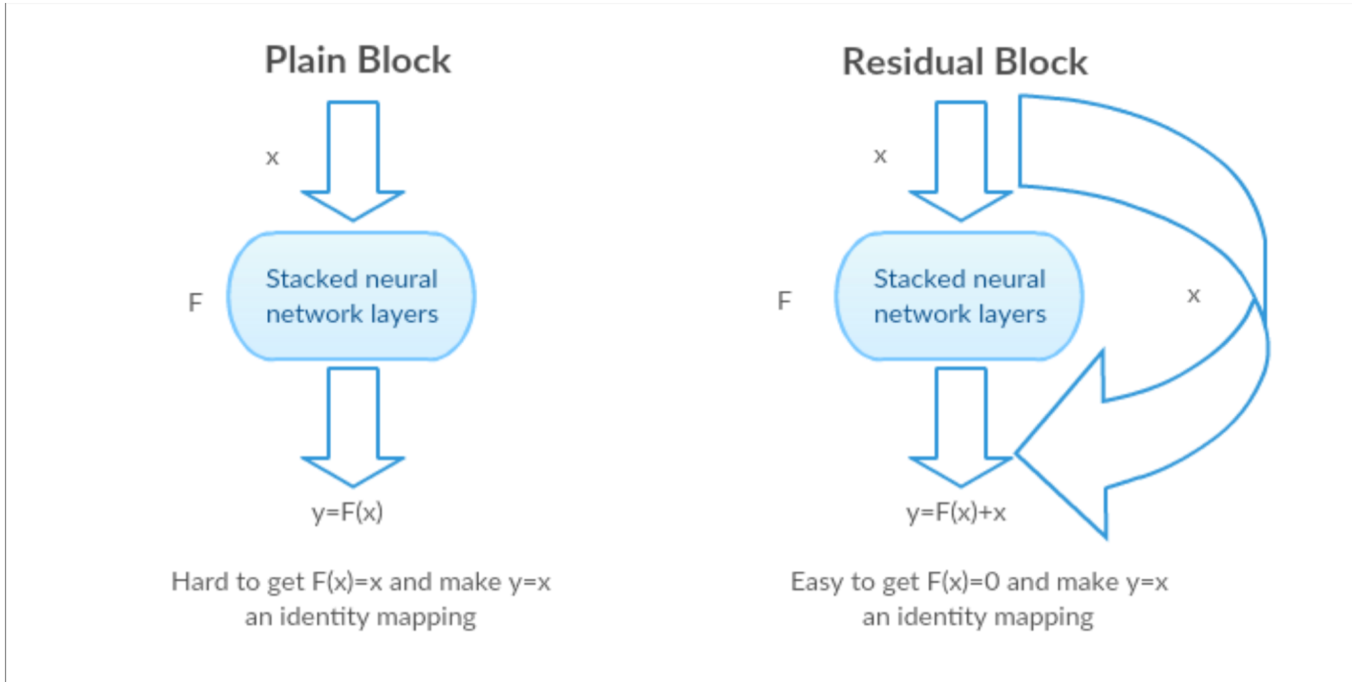


https://arxiv.org/pdf/1409.1556.pdf

ResNets

Published on 2015, utilizes efficient bottleneck structures and learns them as residual functions with reference to the layer inputs, instead of learning unreferenced functions.

- ✓ They are easier to optimize than traditional networks and can gain accuracy from considerably increased depth


Plain Block
x
F
Stacked neural network layers
y=F(x)
Hard to get F(x)=x and make y=x an identity mapping
Residual Block
x
F
Stacked neural network layers
x
y=F(x)+x
Easy to get F(x)=0 and make y=x an identity mapping

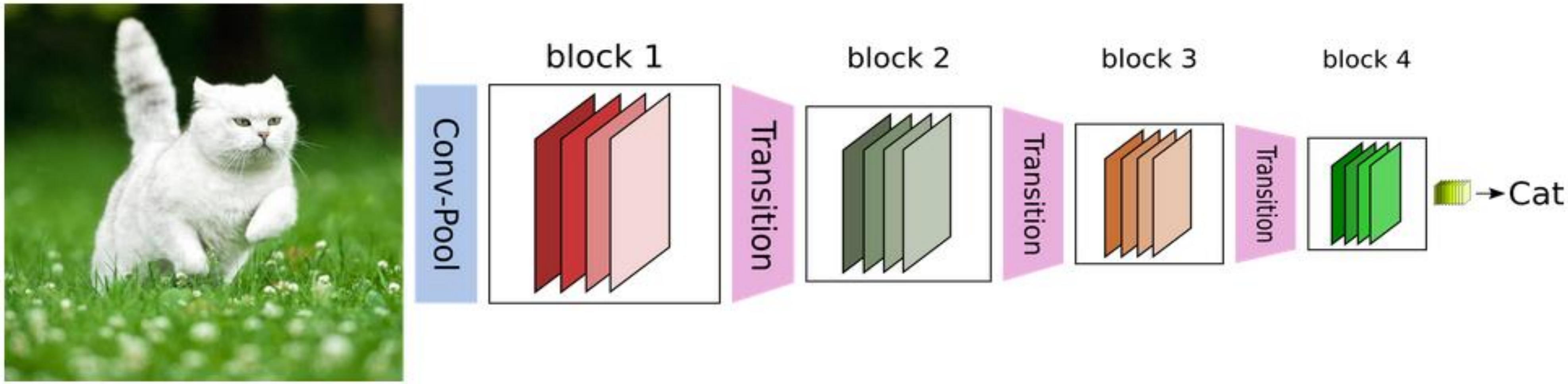
Identity mapping in Residual blocks

https://arxiv.org/pdf/1512.03385.pdf

DenseNets

Published on 2016, it is an extension of ResNets

- A DenseNet architecture can be described as a stack of dense blocks followed by transition layers.
- Each block consists of a series of units. Each unit packs two convolutions, each preceded by Batch Normalization and ReLU activations.
- Strengthen feature propagation, encourage feature reuse, and uses fewer parameters.



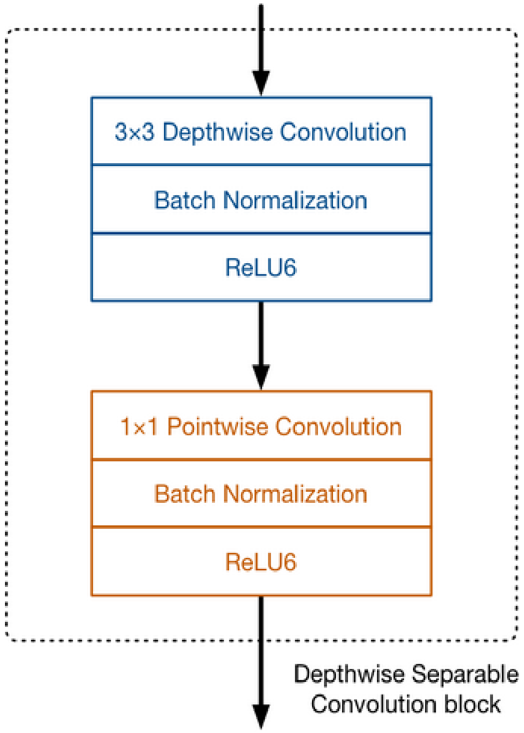
https://arxiv.org/pdf/1608.06993.pdf

Mobile Nets

Initially published on 2017, intended to adapt neural networks to less powerful architectures such as mobile devices, leading to the creation of Mobile Nets with an optimized module for mobility.

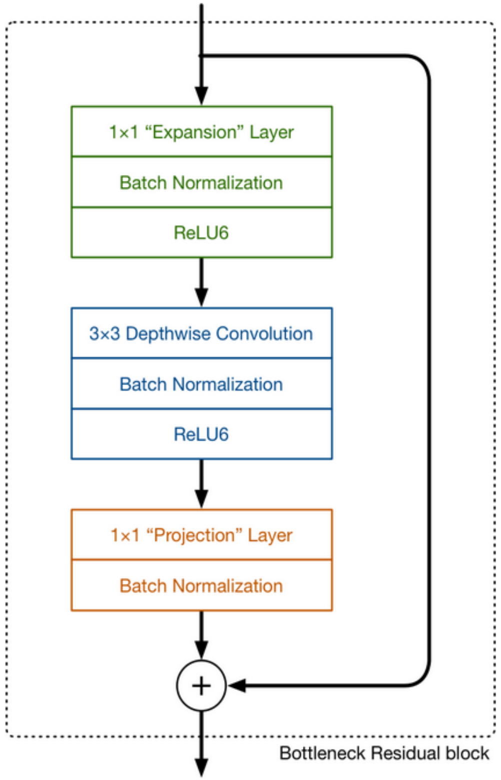
Mobile Net v1

The main idea **Mobile Net V1** is that instead of using convolutional (which are computationally expensive) they are replaced by depth wise separable convolutions.


3×3 Depthwise Convolution
Batch Normalization
ReLU6
1×1 Pointwise Convolution
Batch Normalization
ReLU6
Depthwise Separable Convolution block

Mobile Net v2

Mobile Net V2 still uses depth wise separable convolutions, but it makes the number of channels smaller

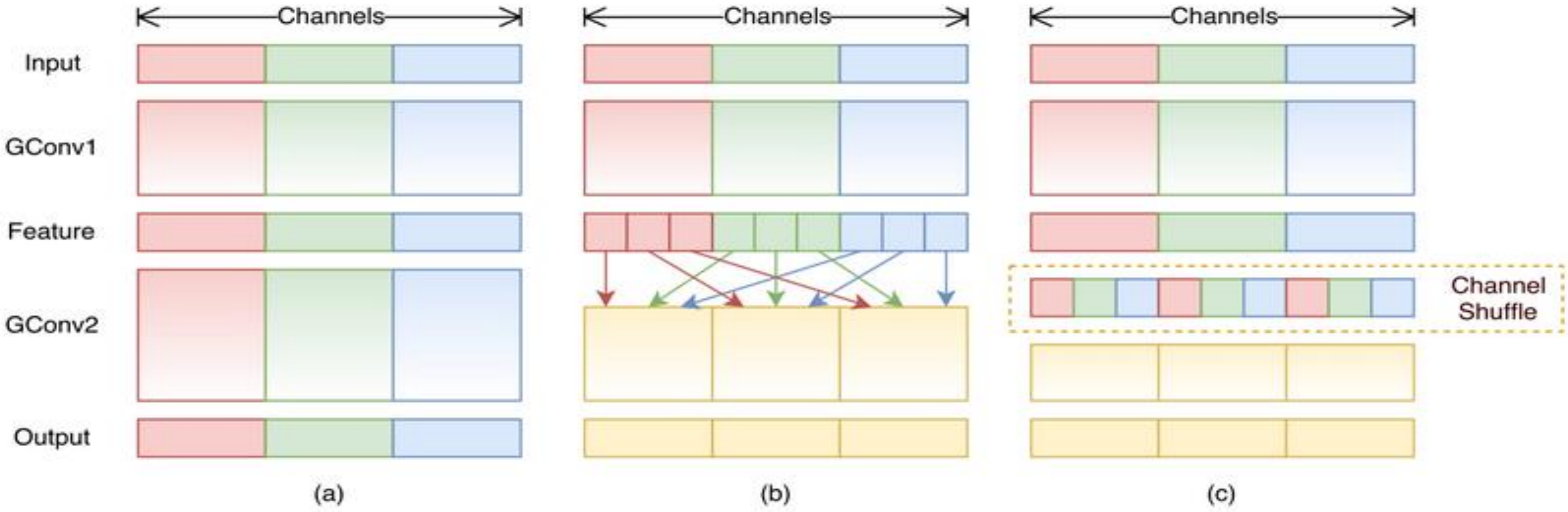


Newer Architectures to Explore…

https://arxiv.org/pdf/1801.04381.pdf

Shuffle Nets

Published on 2017, goes one step further by replacing the pointwise group convolutions that become the main cost. With pointwise group convolutions shuffles the output of group convolutions, so that information can flow from each group of the next group convolution.



Newer Architectures to Explore…

https://arxiv.org/pdf/1707.01083.pdf

Overview

1. Problem Statement

- Verification vs Classification
- Open set vs closed set
- Transfer learning

2. Model Architectures

- Experimental process
- 2D-CNN
- Alex Net,
- VGG Net
- ResNet,
- DenseNet
- Shuffle Net
- Mobile Net

3. Learning Objective

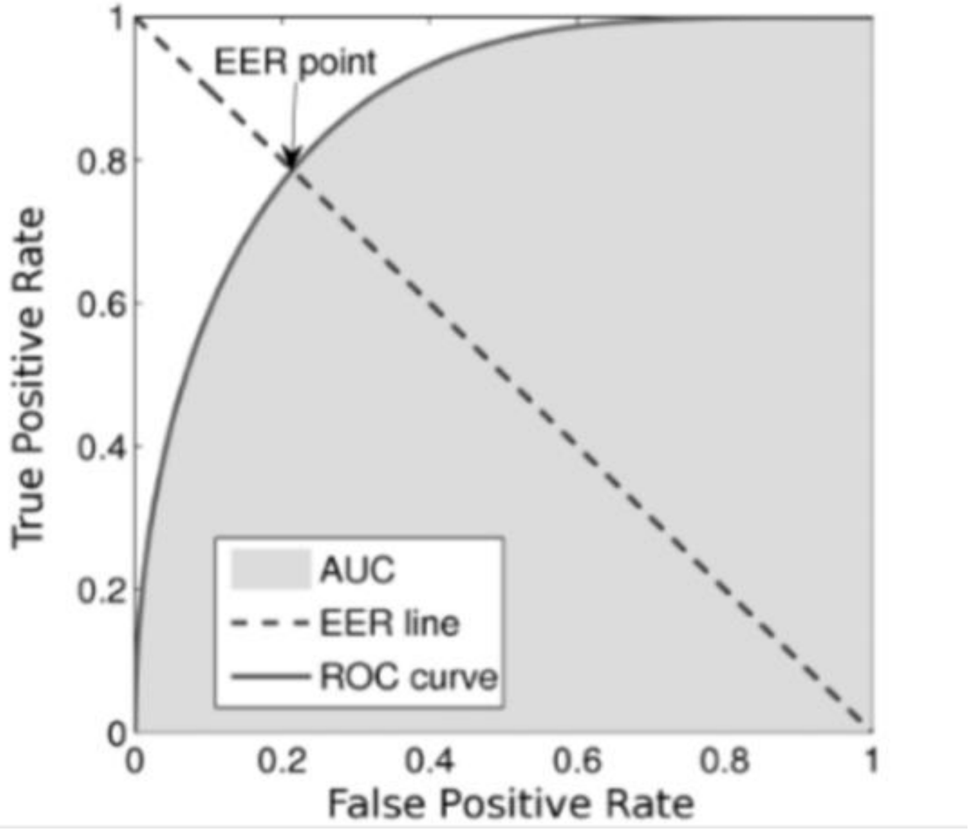
- Area Under the Curve (AUC)
- Transfer learning
- Contrastive loss
- Margin problem
- Center loss
- Triplet loss
- Softmax (revisited)
- Angular softmax

4. Data Explanation

- Detailed on given Dataset
- Torch Dataset (Notebook)
- Torch DataLoader (Notebook)

3. Area Under the Curve (AUC)

- The Receiver Operating Characteristic (ROC) curve is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings
- The Area Under the Curve (AUC) for the ROC curve is equal to the probability that a classifier will rank a randomly chosen positive pair (images of same people) higher than a randomly chosen negative one (images from two different people) (assuming ’similar’ ranks higher than ’dissimilar’ in terms of similarity scores)



https://www.webopedia.com/TERM/E/equal_error_rate.html

Area Under the Curve (AUC) - Cont.

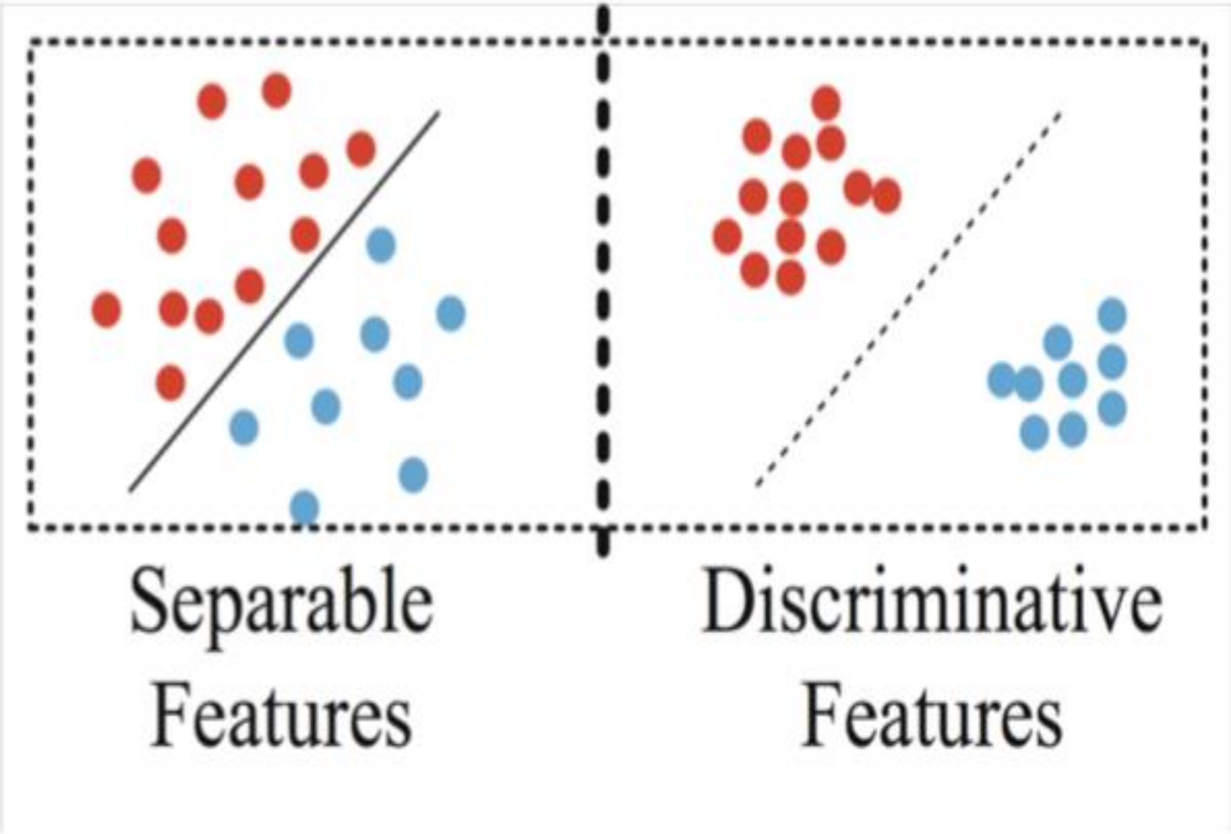
- ✓ AUC of 0.0 means that the model ranks all negatives above all positives. This is the worst model possible. In fact if you flip the scores, you will be better off.
- ✓ AUC between 0 and 0.5, means the model ranks a random positive example higher than a random negative example less than 50% of the time.
- ✓ AUC of 0.5, means it ranks a random positive example higher than a random negative example 50% of the time. It can be said that the predictive ability of the model is no better than random guessing.

Aim for AUC between 0.5 and 1.0, means it ranks a random positive example higher than a random negative example more than 50% of the time.

AUC of 1.0 means the model ranks all positives above all negatives.

Discriminative Features

- ✓ Classification task optimizes learning separable features
- ✓ Optimally we wish to learn discriminative features
  - ◆ **Maximal** *intra* class distance
  - ◆ **Minimal** *intra* class distance
- ✓ First define a measure of distance
- ✓ Most often consider Euclidean distance



Center Loss

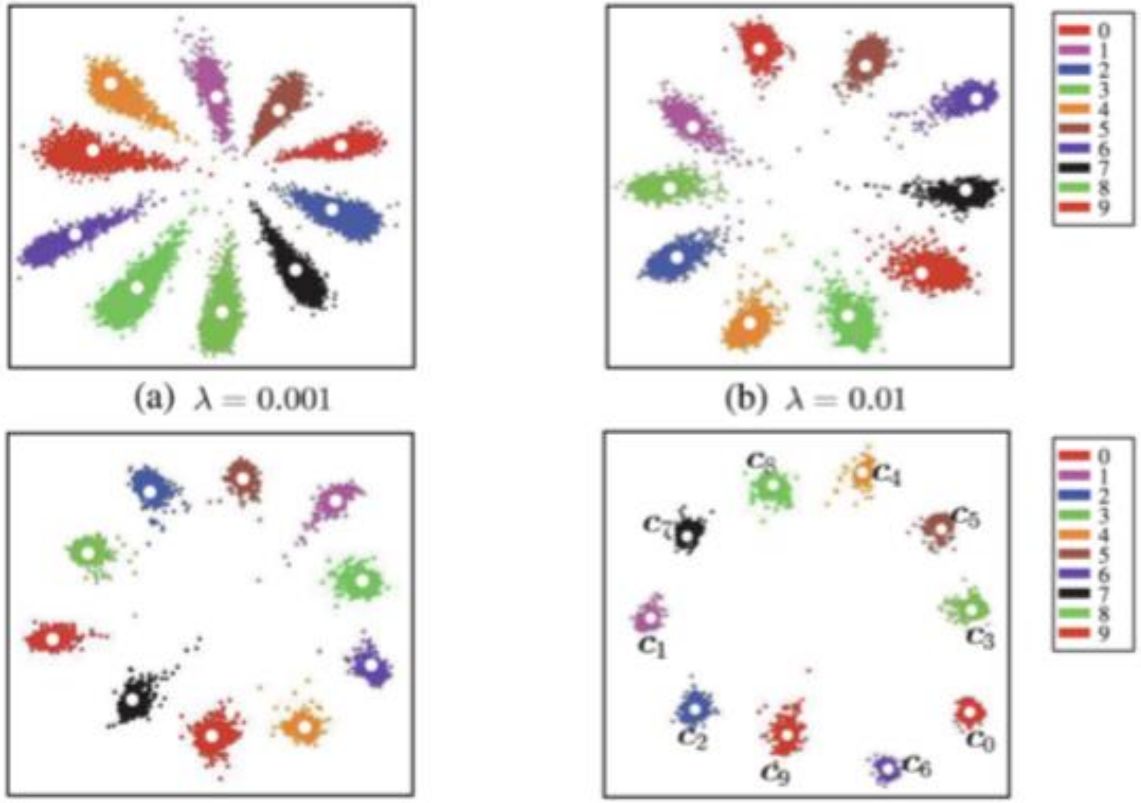
- ✓ Define a criterion that promotes minim intra- class distance while maintaining inter-class separability
- ✓ Does **not** explicitly try to maximize inte class class distance

Center Loss: $$\mathcal{L}_C = \frac{1}{2}\sum_{i=1} \|x_i - c_{y_i}\|_2^2$$

Joint Objective: $$\mathcal{L} = \mathcal{L}_S + \lambda\mathcal{L}_C$$

c_{y_i} feature center for class label

$\mathcal{L}_S$ softmax loss (softmax + xent)



(a) $\lambda = 0.001$ (b) $\lambda = 0.01$ (c) $\lambda = 0.1$ (d) $\lambda = 1$

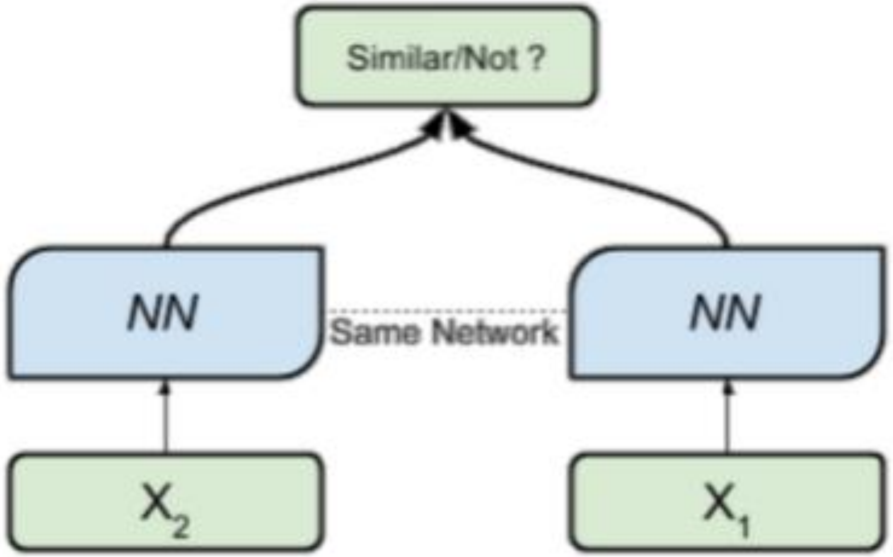
https://ydwen.github.io/papers/WenECCV16.pdf

Contrastive Loss

- For every pair of training instances, try to push the features for intra-class samples together and maintain some margin *m* between inter-class ones.
- Feedforward two samples *i, j* and apply the contrastive loss function provided some distance metric of choice *D.*
- Generally only useful for fine tuning. Not an information rich criterion

  1 bit of information per sample pair

- Tricky to balance sample pairs and choose the margin



$$\mathcal{L}_{CT}(\mathbf{z}_i, \mathbf{z}_j) = 1(y_i = y_j)\left(\frac{1}{2}D(\mathbf{z}_i, \mathbf{z}_j)^2\right) + 1(y_i \neq y_j)\frac{1}{2}(\max(0, m - D(\mathbf{z}_i, \mathbf{z}_j)^2))$$

y_k class label of sample *k* $\quad$ $\mathbf{z}_k$ embedding of sample *k* $\quad$ m margin

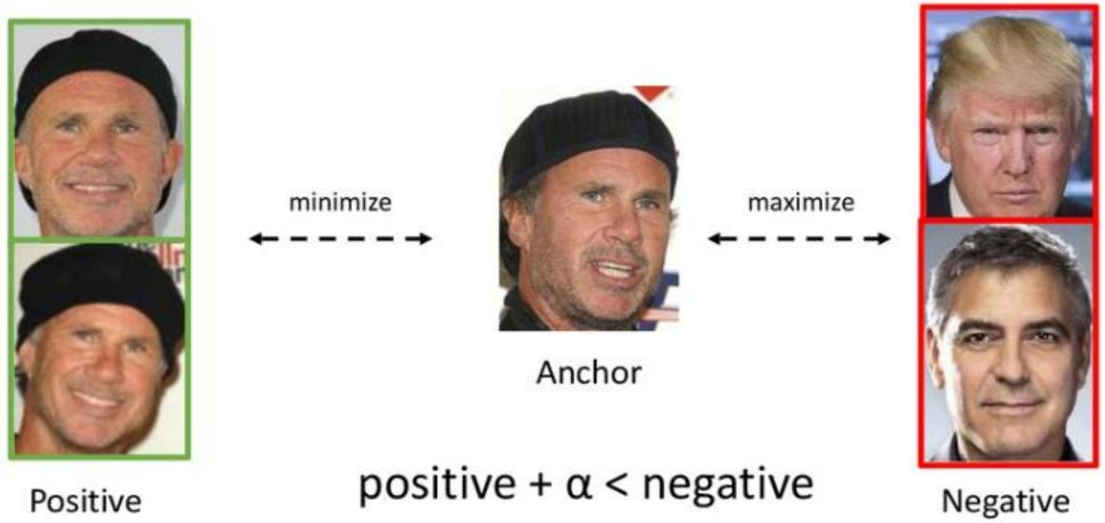
Triplet Loss

Motivated by nearest neighbor classification:

- ✓ Try to ensure that the face embedding of a face *i* is closer to all the embeddings of that same face than it is to any other embedding that belongs to a face *i != j*

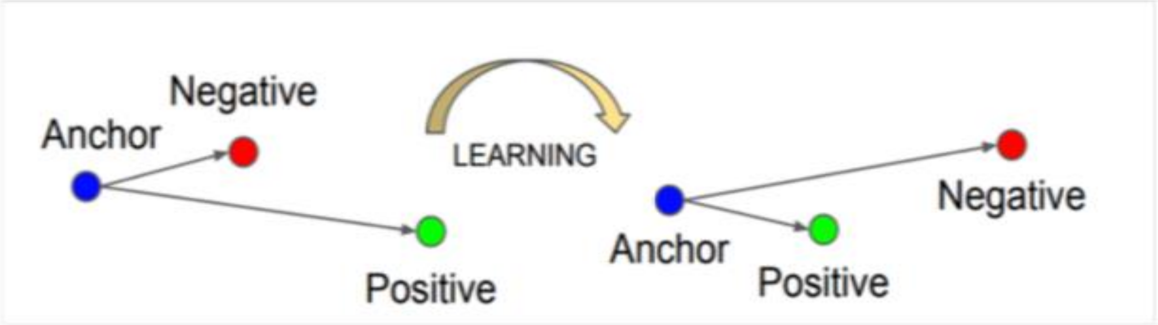
- ✓ Share parameters among 3 networks (like contrastive loss scenario)

- ✓ Good results when sampling triplets in an intelligent manner (sample mining)



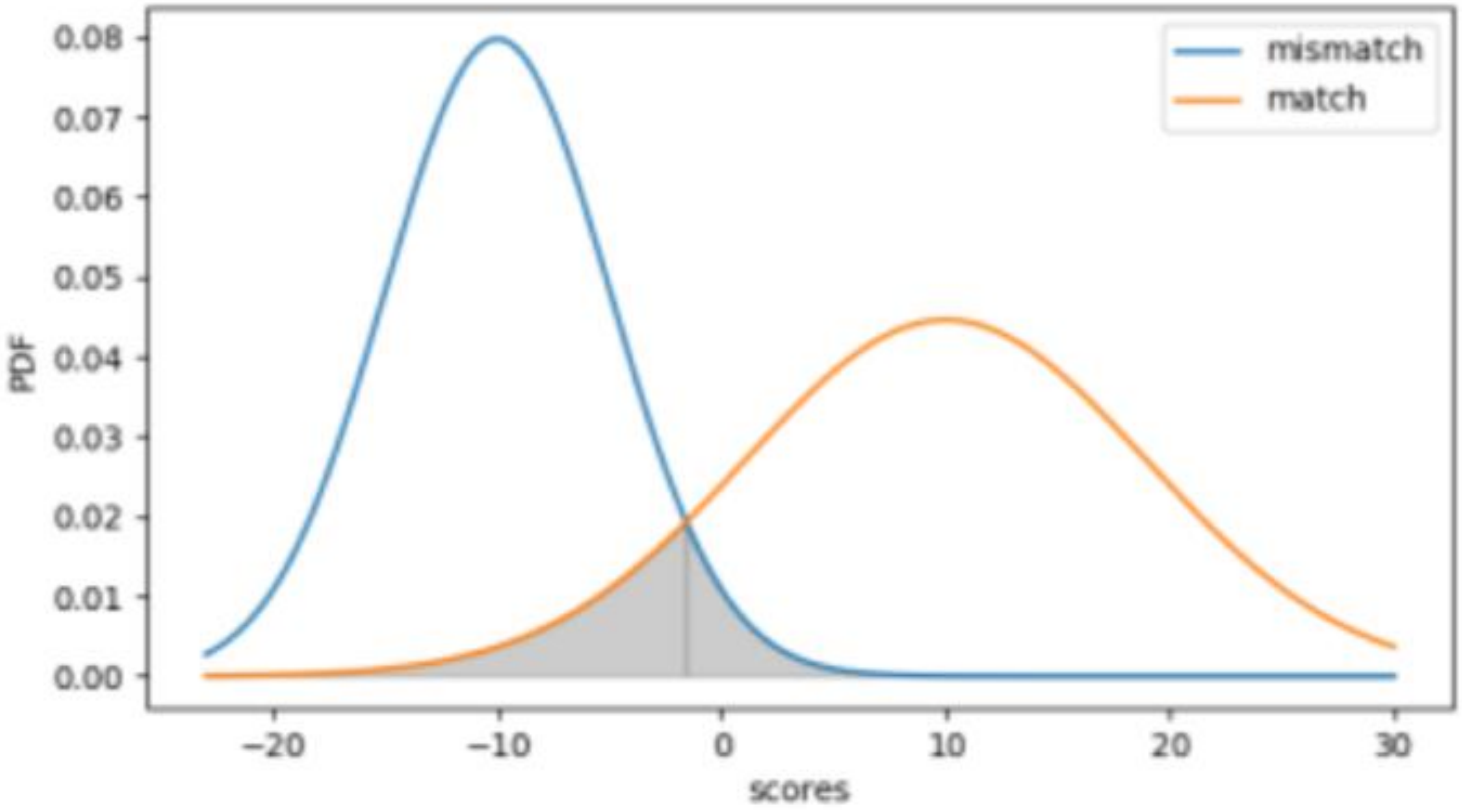
Features

- Identity -> single point
- Enforces a margin between persons



Pair-Wise Loss

For verification task, this function explicitly tries to separate the distributions of similarity scores under similar pairs and dis-similar pairs.



SoftMax Loss Revisited

$$p_1 = \frac{\exp(\mathbf{W}_1^T x + b_1)}{\exp(\mathbf{W}_1^T x + b_1) + \exp(\mathbf{W}_2^T x + b_2)}$$

$$p_2 = \frac{\exp(\mathbf{W}_2^T x + b_2)}{\exp(\mathbf{W}_1^T x + b_1) + \exp(\mathbf{W}_2^T x + b_2)}$$

$$(W_1 - W_2)x + b_1 - b_2 = 0$$

Resulting decision boundary
(binary case)

For this example, assume two classes (binary classification).

- Define SoftMax *loss* as a combination of the final layer parameters, the SoftMax operator and cross-entropy loss.

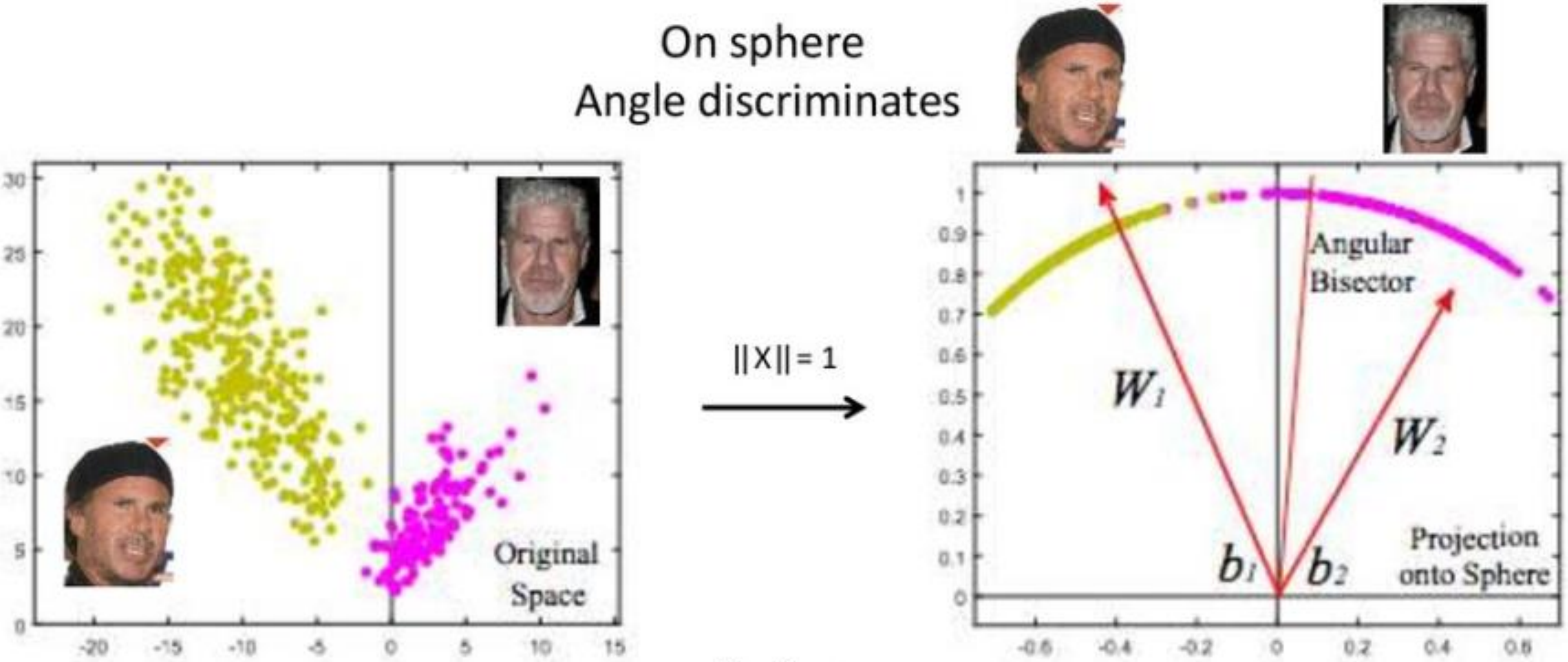
Angular SoftMax Loss

Idea: Combine Euclidean margin constraints with the SoftMax loss

- ✓ Formulate an angular interpretation to SoftMax that can be projected onto a hyper-spheric manifold
- ✓ No need for sample mining when identifying pairs / triplets for contrastive / triplet losses
- ✓ Interpretable as learning features that are discriminative

Angular SoftMax Loss

Recall binary classification with SoftMax decision boundary:



On sphere
Angle discriminates

||X|| = 1

$$(\boldsymbol{W}_1 - \boldsymbol{W}_2)\boldsymbol{x} + b_1 - b_2 = 0 \xrightarrow[\text{b=0}]{\text{||W|| = 1}} \|\boldsymbol{x}\|(\cos(\theta_1) - \cos(\theta_2)) = 0$$

Note that: Where θ_i is the angle between the weight and feature vector.

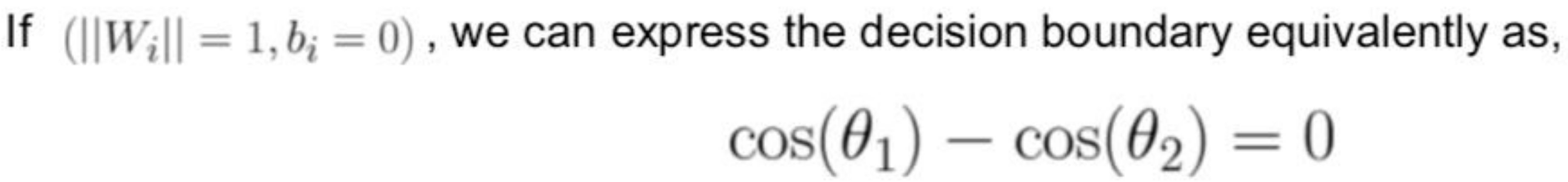
Angular SoftMax Loss

Recall binary classification with softmax decision boundary:

$$(W_1 - W_2)x + b_1 - b_2 = 0$$

Note that

$$W_i^T x_i + b_i = \|W_i^T\| \, \|x_i\| \cos(\theta_i)$$



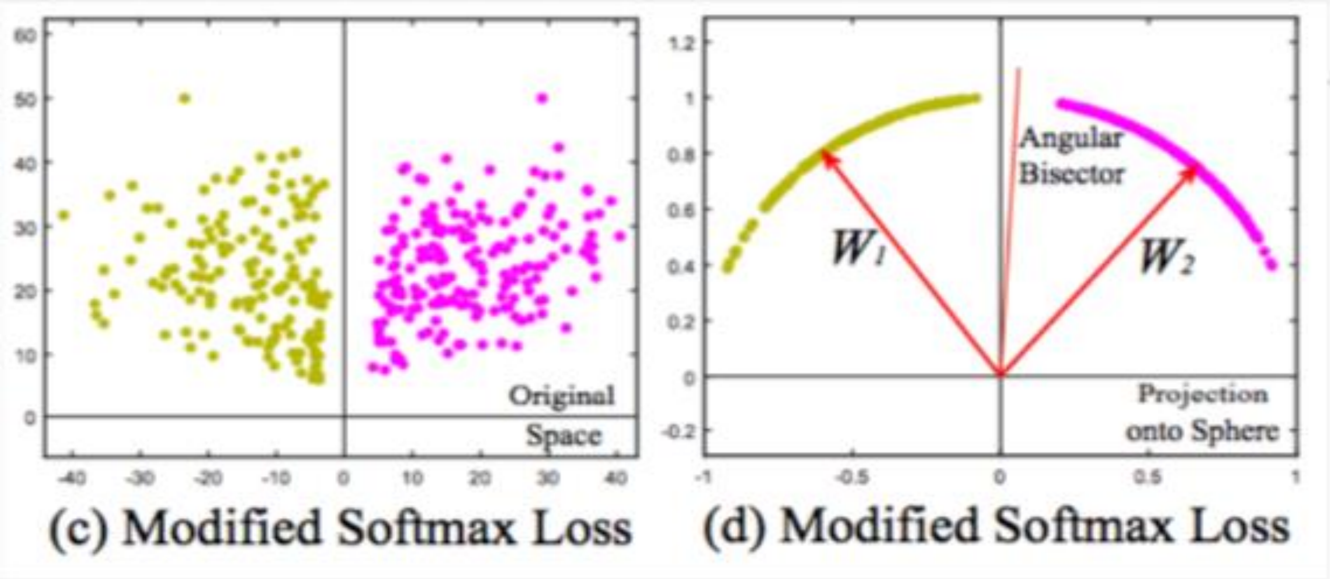
If $(||W_i|| = 1, b_i = 0)$, we can express the decision boundary equivalently as,

$$\cos(\theta_1) - \cos(\theta_2) = 0$$

This is simply the angle bisector of the two parameter vectors

Angular SoftMax Loss

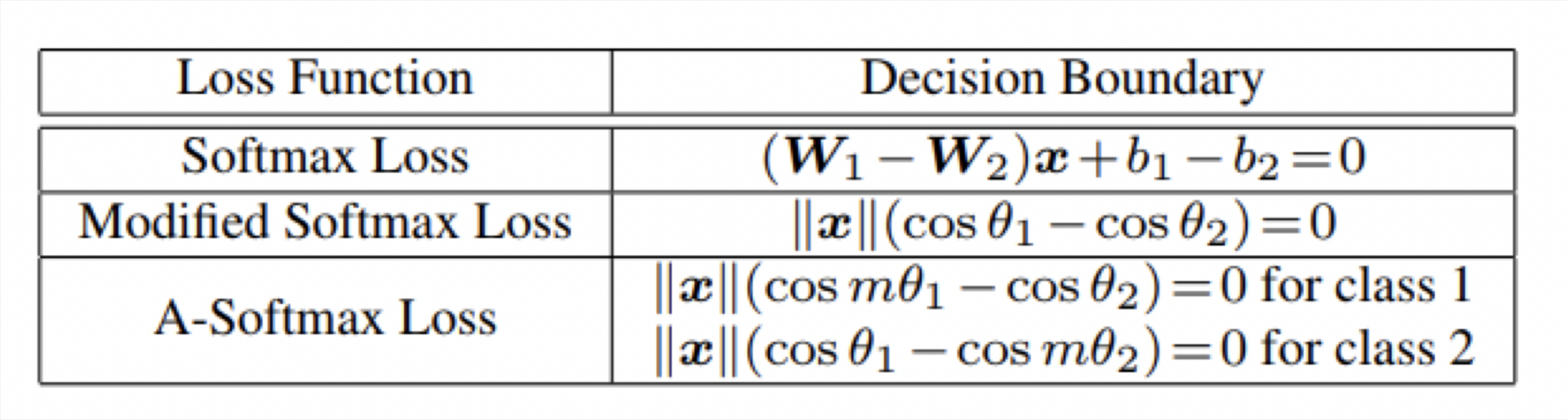
Generalize to multi-class



(c) Modified Softmax Loss (d) Modified Softmax Loss

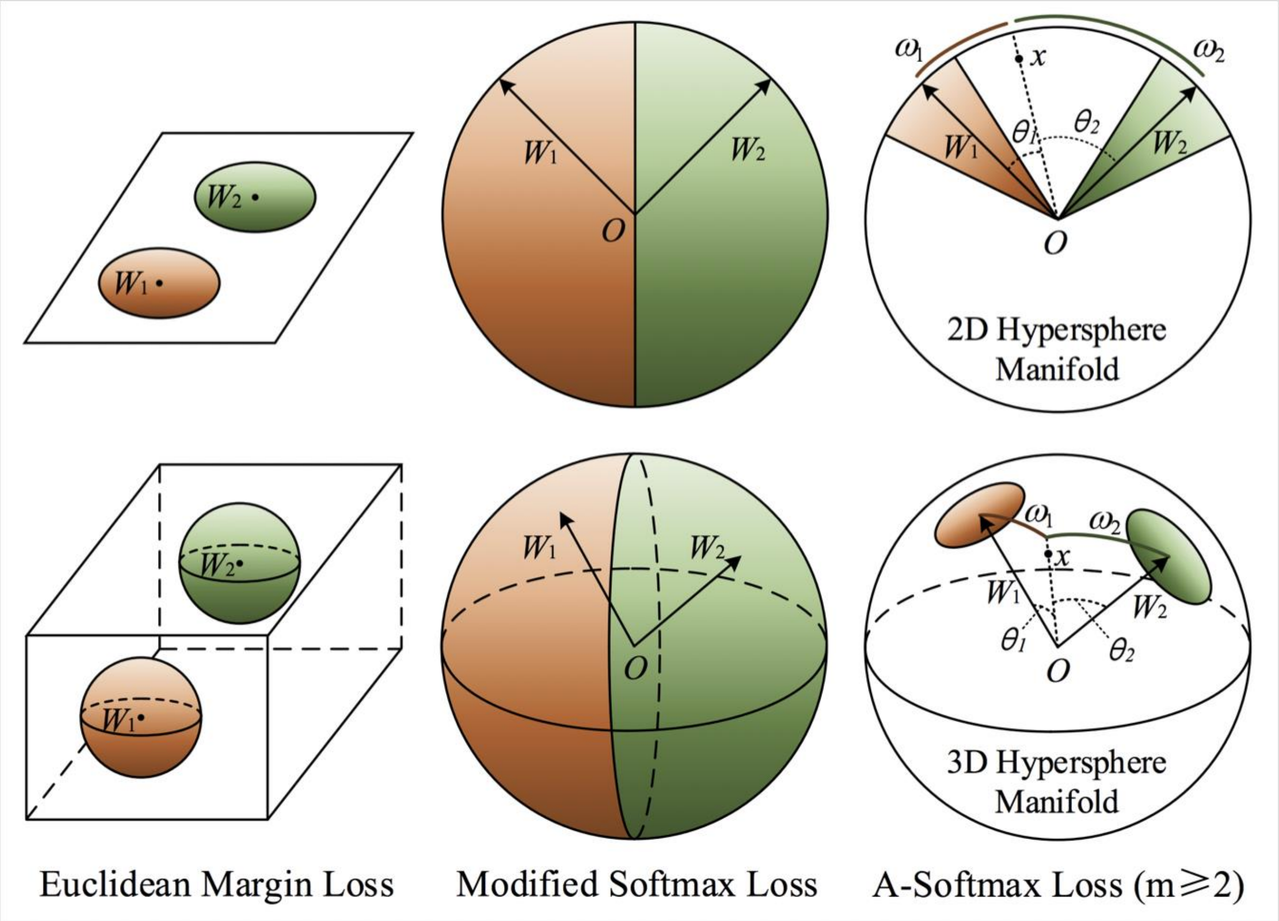
$$\theta_{j,i}\ (0 \leq \theta_{j,i} \leq \pi), \text{ for } W_j, x_i$$

Angular SoftMax Loss



Loss Function	Decision Boundary
Softmax Loss	$(\boldsymbol{W}_1 - \boldsymbol{W}_2)\boldsymbol{x} + b_1 - b_2 = 0$
Modified Softmax Loss	$\|\boldsymbol{x}\|(\cos\theta_1 - \cos\theta_2) = 0$
A-Softmax Loss	$\|\boldsymbol{x}\|(\cos m\theta_1 - \cos\theta_2) = 0$ for class 1 $\|\boldsymbol{x}\|(\cos\theta_1 - \cos m\theta_2) = 0$ for class 2

Angular Softmax loss



A geometric interpretation when weights are normalized, and biases set to zero (in the last fully-connected layer).

Overview

1. Problem Statement

- Verification vs Classification
- Open set vs closed set
- Transfer learning

2. Model Architectures

- Experimental process
- 2D-CNN
- Alex Net, VGG Net
- ResNet, DenseNet
- Shuffle Net and Mobile Net

3. Learning Objective

- Area Under the Curve (AUC)
- Transfer learning
- Contrastive loss
- Margin problem
- Center loss
- Triplet loss
- SoftMax (revisited)
- Angular SoftMax

4. Data Explanation

- Detailed on given Dataset
- Torch Dataset (Notebook)
- Torch DataLoader (Notebook)

Putting it All Together…

For Homework 2

- You will be using Transfer Learning for face verification
- We follow Develop Model Approach
- Train a convolutional network to perform N-way face classification
- Extract intermediate representations of face classifier
- Verify if two intermediate samples belong to same person
- Fine-tune the network to learn more discriminative features

4. Datasets for Verification

Overview of the datasets given

Training	• Medium (2,300 Face IDs) • Large (2,000 Face IDs)
Validation	• Validation_verification • Validation_trials_verification.txt 100,000 total trials True Label give {0,1}
Testing	• Test_verification • Test_trials_verification_student.txt • 900,000 total trials

Format of Trial File

The trial file contains 2 columns:

- 1st column containing a pair of file

- 2nd is the label. 0 meaning they the pair is images of different people. 1 meaning otherwise

File_name_1	File_name_2	Label
fid_373/0111_01.jpg	fid_423/0170_01.jpg	0

Torch DataSet and DataLoader
(Jupyter Notebooks)

Relevant Sources

https://arxiv.org/pdf/1512.03385.pdf

https://arxiv.org/pdf/1608.06993v3.pdf

https://arxiv.org/pdf/1409.1556.pdf

https://arxiv.org/pdf/1704.08063.pdf

https://arxiv.org/pdf/1503.03832v3.pdf

http://ydwen.github.io/papers/WenECCV16.pdf

https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf

https://towardsdatascience.com/densenet-2810936aeebb

https://www.ece.rice.edu/~dhj/courses/elec241/spectrogram.html

http://yann.lecun.com/exdb/publis/pdf/hadsell-chopra-lecun-06.pdf

https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624