



Triplet Lossで Metric Learning

18-B11 DJ2
千枝 睦実



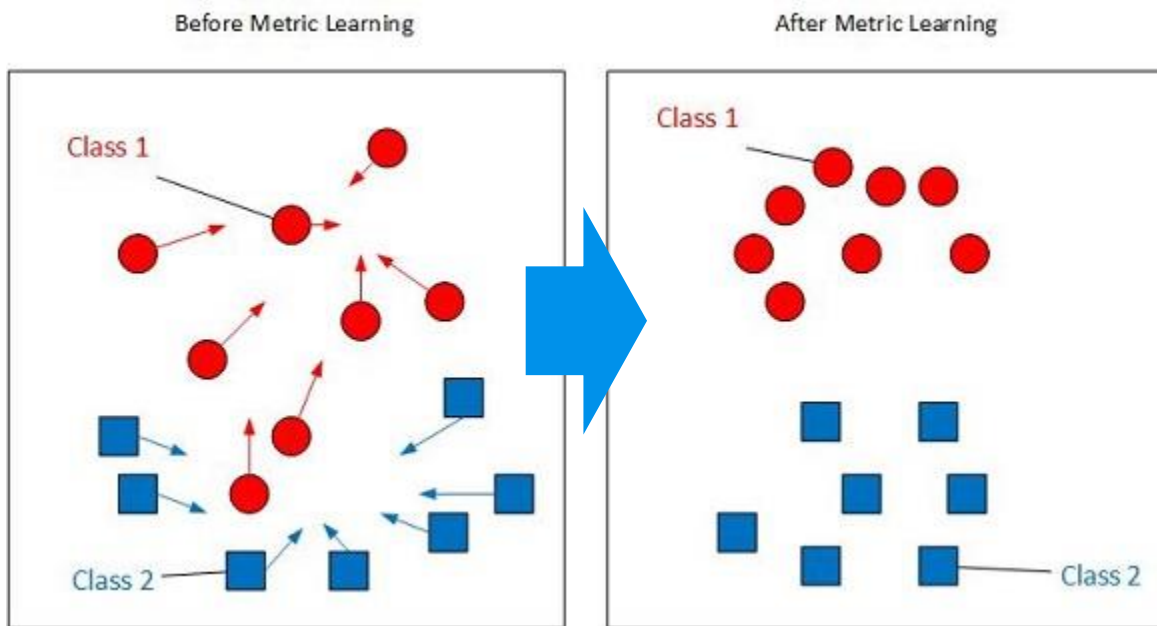
Metric Learning とは

なにをするのか

特徴量の空間で、
求めたい条件
に合わせて、
訓練データ間の距離
を修正する

分類条件に
合致するものは近くに、
してないものは
遠くに寄せる

分類が
しやすくなる！



学習前

学習後

特徴空間

役立つ場面例

Objective

スナック菓子の不良品の検知

Situation

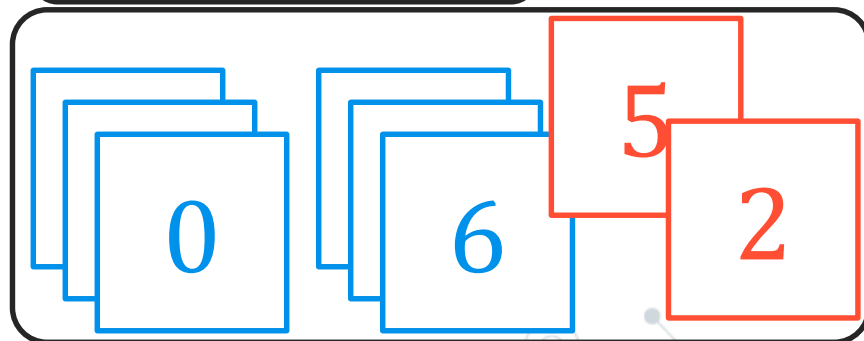
訓練データにないデータが
テストデータにたくさんある

役立つ場面例

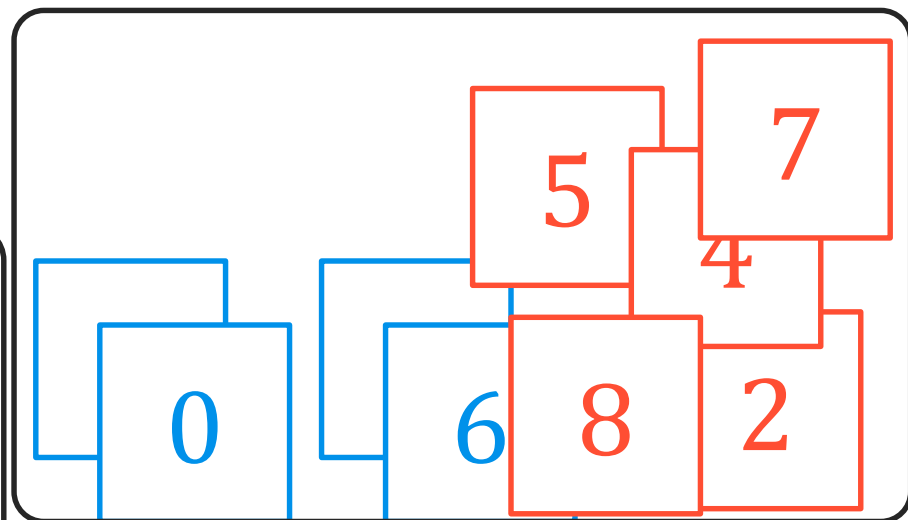
Objective

スナック菓子の不良品の検知

Situation



Train



Test

5

役立つ場面例

Objective

スナック菓子の不良品の検知

Problem

通常の分類手順だと、
訓練にないラベルのデータを
誤分類しやすい

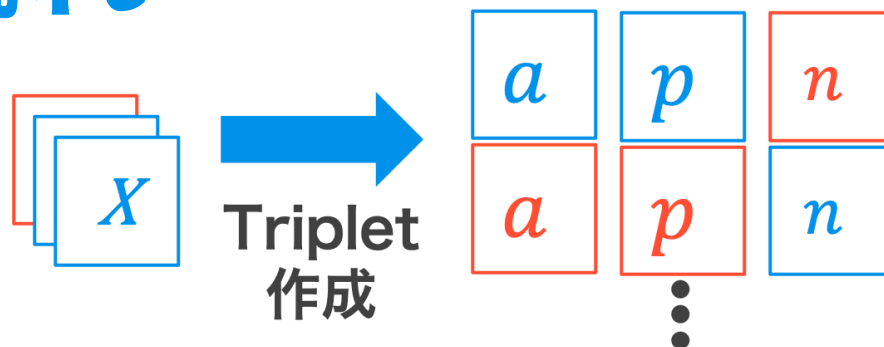


分類までの流れ



分類までの流れ

Triplet
作成



Tripletを用いて
Metric Learning

正常品の特徴の
平均を算出

そこから
離れているものは
不良品！

Triplet とはこの3枚組

各画像 a に対して…
同じラベルの画像 p
違うラベルの画像 n

Tripletの組み合わせかたによって
距離学習の目標を設定できる

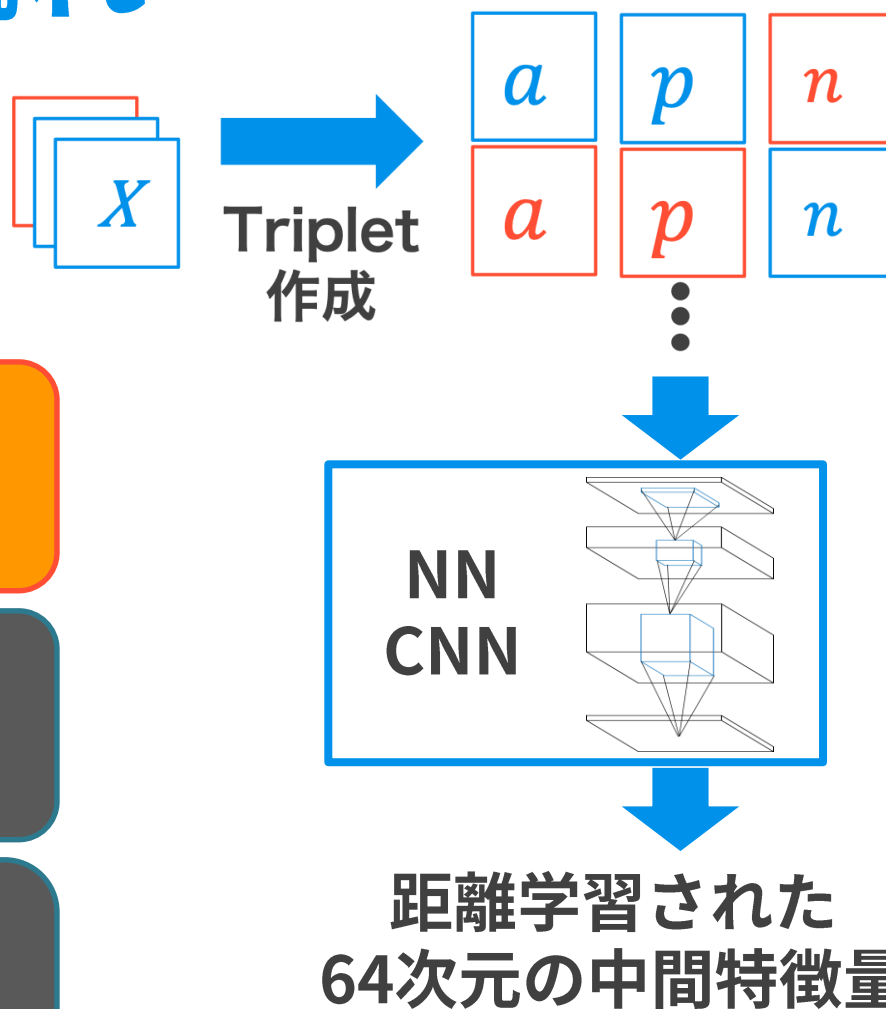
分類までの流れ

Triplet
作成

Tripletを用いて
Metric Learning

正常品の特徴の
平均を算出

そこから
離れているものは
不良品！



分類の流れ

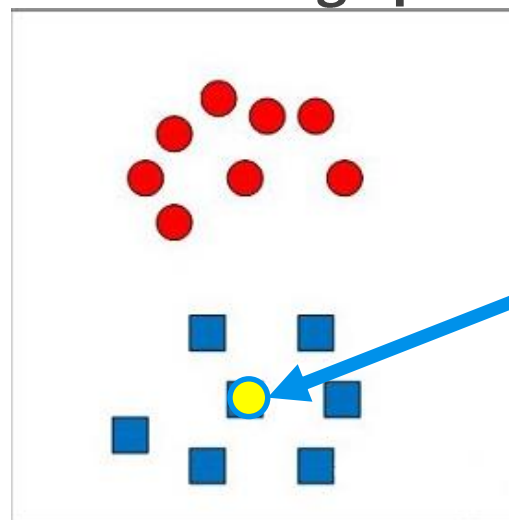
Triplet
作成

Tripletを用いて
Metric Learning

正常品の特徴の
平均を算出

そこから
離れているものは
不良品！

Embedding Space



平均

距離学習された
64次元の中間特徴量

10

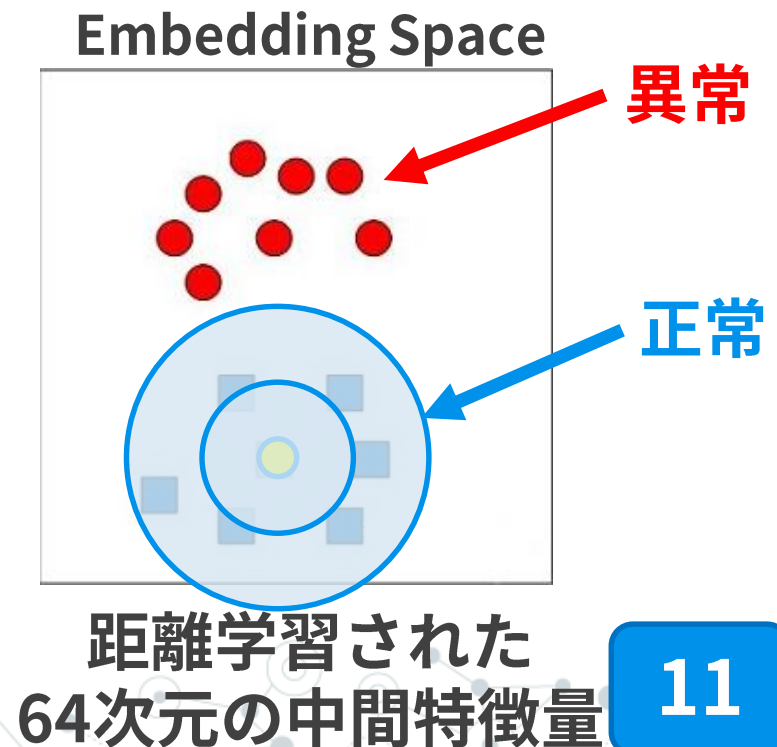
分類の流れ

Triplet
作成

Tripletを用いて
Metric Learning

正常品の特徴の
平均を算出

そこから
離れているものは
不良品！





学习方法

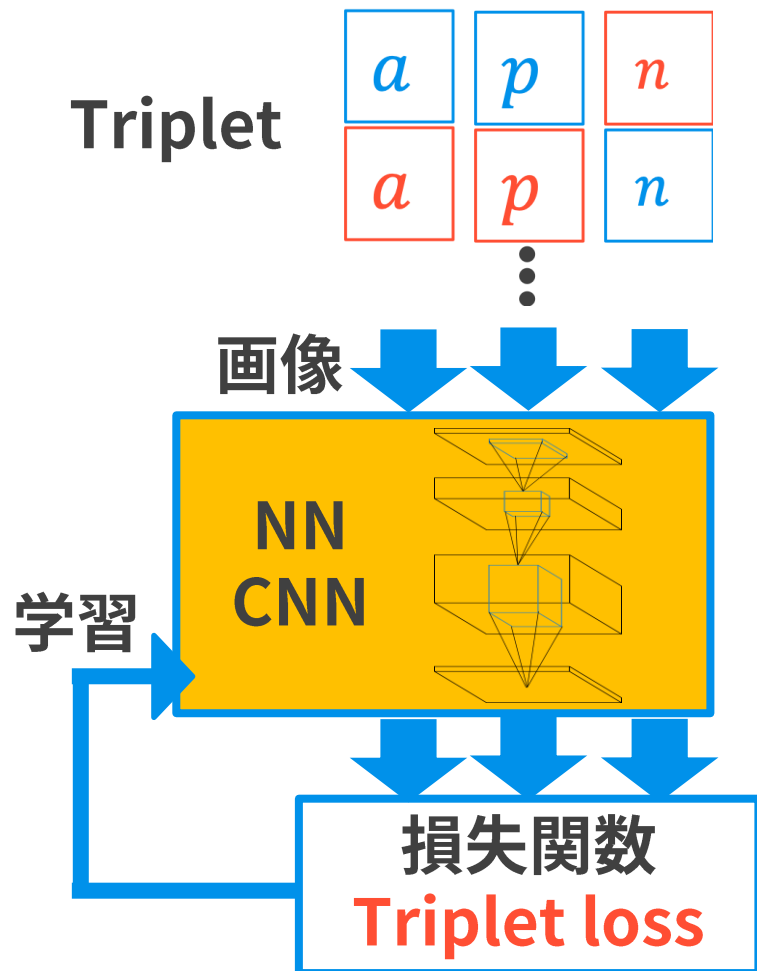
学習方法

Triplet
作成

Tripletを用いて
Metric Learning

正常品の特徴の
平均を算出

そこから
離れているものは
不良品！



損失関数

Triplet Loss

$$L_{tri} = \sum_{i=1}^N \max(0, \|x_i^a - x_i^p\|_2^2 - \|x_i^a - x_i^n\|_2^2 + \alpha)$$

a : アンカー画像, p : 同ラベル画像, n : 異ラベル画像
 α : マージン, N : Tripletのサンプル数(今回は全画像を使用)
 x : 画像をNNに通した中間特徴量

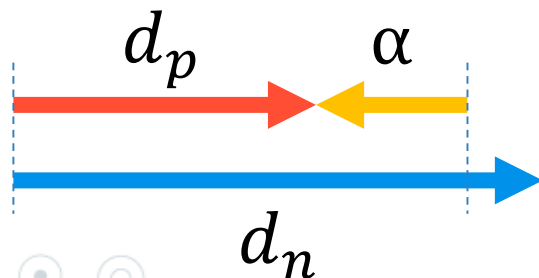
これを最小化しよう
NNを調整

損失関数

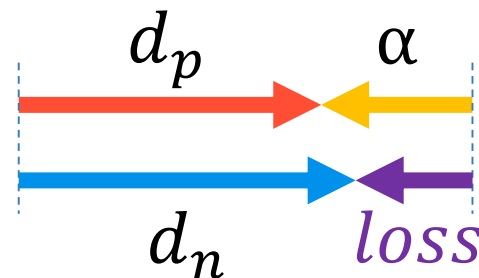
Triplet Loss

$$L_{tri} = \sum_{i=1}^N \max(0, d_p - d_n + \alpha)$$

a : アンカー画像, p : 同ラベル画像, n : 異ラベル画像
 α : マージン, N : Tripletのサンプル数(今回は全画像を使用)



$$L_{tri} = 0$$



$$L_{tri} = d_p - d_n + \alpha$$

改善点

Tripletの選択

基本は全てのデータに対して、
全ての組み合わせを作成

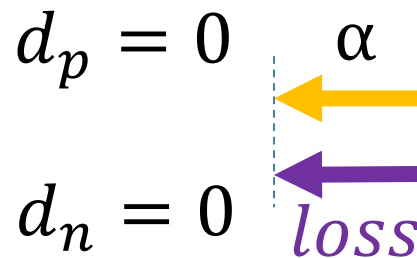
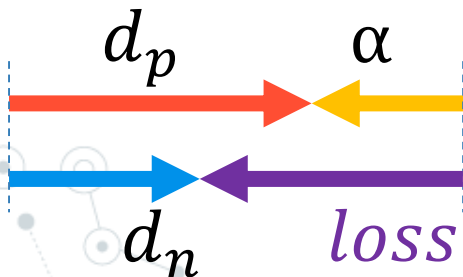
組合せの数は膨大
(N^3 オーダーで増える)

Tripletの選択に工夫が必要

Tripletの選択

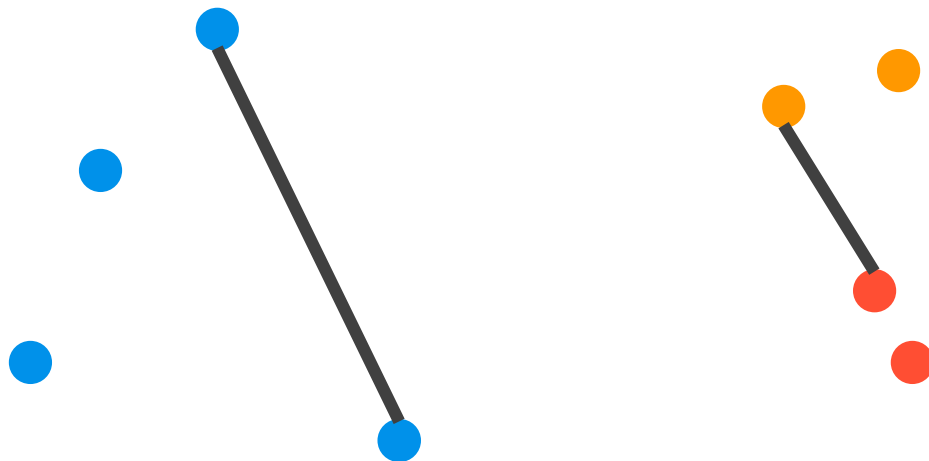
FaceNetでの例

- ミニバッチ法の適用
- 異ラベルの画像は1通りのみとする
-> データ数を N^2 オーダーに削減
- $d_n > d_p$ なTripletを使わない
(Loss = マージンとなる局所解を避ける)



Triplet Lossの問題点

クラス内距離 > クラス間距離
となることがある



距離学習後の識別方法
によっては不都合

Triplet Lossの問題点

Quadruplet Loss

$$L_{tri} = \sum_{i=1}^N \max(0, d_p - d_{n1} + \alpha) + \max(0, d_p - d_{n2} + \alpha)$$

d_p : 同ラベル画像との距離, d_n : 異ラベル1画像との距離

d_n : 異ラベル2画像との距離

α : マージン