

300. 様々な研究

秋葉洋哉

2024 年 7 月 15 日

1 Metric Learning

距離学習 (Metric Learning) は、人物同定をはじめ、顔認識、画像分類、画像検索、異常検知といった幅広いタスクで用いられる技術である。その基本概念は、同じクラスに属するデータは近くに、異なるクラスに属するデータは遠くに配置するような特徴空間 (Embedding Space) を学習することである。Metric Learning は、特徴空間の学習を通じて、データの分布を変換することで、データの分類や検索を行う。

まず、特徴空間とは、特徴ベクトルを次元として持つ空間のことである。例えば、手書きの数字を入力として与え、その数字を分類するタスクを考える。この場合、 28×28 次元のデータを 10 次元に圧縮することで、特徴空間を構築することができる。学習が進んでいくと、同じ数字のデータは近くに、異なる数字のデータは遠くに配置されるようになる (図 1)。

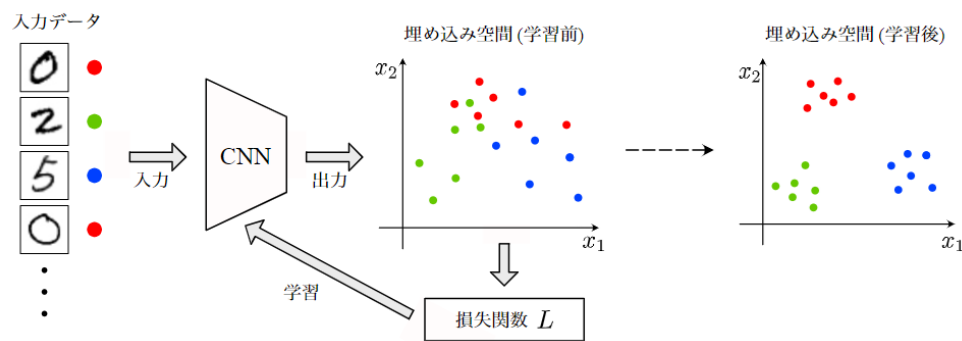


図 1: Metric Learning の概要

例えば、画像認識においては、CNN のネットワーク構造自体を工夫しなくても、特徴空間を構成できるように学習を行うだけで、精度を向上させることができるようになるという利点がある。

こうした特徴空間内のデータの配置を深層学習によって学習していくことを、深層距離学習と呼ぶ。深層距離学習は、Siamese Network や Triplet Network といった手法が存在する。

2 Siamese Network

Siamese Network (シムネットワーク) は、2つのデータを同時に入力し、それらのデータが同じクラスに属するかどうかを判定するネットワークである。入力 x_1, x_2 に対して出力 $f(x_1), f(x_2)$ であった場合、その特徴ベクトルの距離 $D(f(x_1), f(x_2))$ を計算し、損失関数 L に入力することで、最適なネットワークのパラメータを学習する。損失関数は、contrastive loss という名称があり、以下で表される。

$$L = \frac{1}{2} [yD^2 + (1 - y) \max(m - D, 0)^2] \quad (1)$$

ここで、 y は同じクラスに属するかどうかを示すラベルであり、 m はマージンを示す。 D は特徴ベクトルの距離であり、 $D = \|f(x_1) - f(x_2)\|$ (ユークリッド距離) である。

この損失関数は、以下のように分解できる。

$$L = yL_1 + (1 - y)L_2 \quad (2)$$

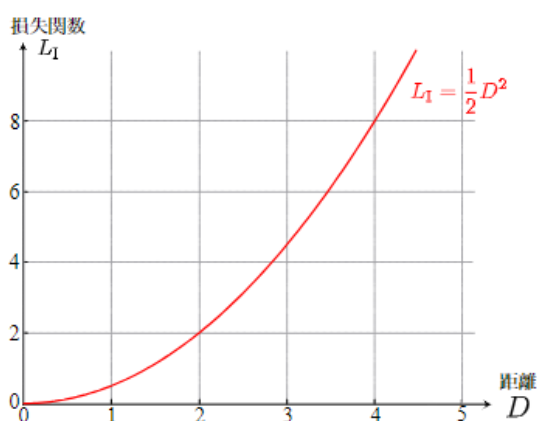
$$L_1 = \frac{1}{2} D^2 \quad (3)$$

$$L_2 = \frac{1}{2} \max(m - D, 0)^2 \quad (4)$$

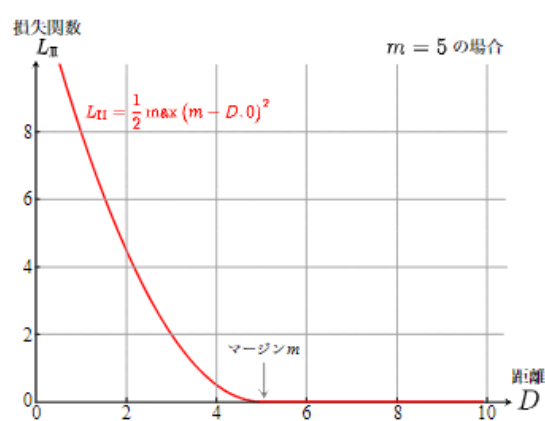
L_1 は、シンプルに距離の2乗であり、この損失関数を最適化することで、同じクラスに属するデータは近くに配置されるようになる (図 2a)。一方で、 L_2 は、マージン m よりも離れたデータに対しては、損失が発生しないようになる。つまり、

$$\begin{cases} D < m & : L_2 = \frac{1}{2}(m - D)^2 \\ D \geq m & : L_2 = 0 \end{cases} \quad (5)$$

となる (図 2b)。



(a) L_1 の損失関数



(b) L_2 の損失関数

図 2: contrastive loss

深層距離学習では、類似データは近くに、異なるデータは遠くに配置されるように学習することで、特徴空間を構築する。そのため、Siamese Network は、入力ペアが同じクラスの場合、 L_1 を、異なるクラスの場合、 L_2 を損失関数として用いることで、距離学習を行う。

識別ラベル y は、同じクラスの場合は $y = 1$ 、異なるクラスの場合は $y = 0$ となる。こうすることで、損失関数を一つの式でまとめて書いたものが、 $L(\text{contrastive loss})$ である。

この損失関数は大きな欠点が存在する。それは、異なるクラスのデータに対しては、 $D = m$ を超えた時点で最適化が終了するが、同じクラスのデータに対しては、 $D = 0$ になるまで最適化し続けてしまうことである。この問題を解決するために、Triplet Network が提案されている。

3 Triplet Network

Triplet Network (トリプレットネットワーク) は、Siamese Network の欠点を解消するために提案されたネットワークである。Triplet Network は、Anchor, Positive, Negative の3つのデータを入力とし、Anchor と Positive の距離を近づけ、Anchor と Negative の距離を遠ざけるように学習する。

Triplet Network の損失関数は、triplet loss と呼ばれ、以下で表される。

$$L = \max(D_p - D_n + m, 0) \quad (6)$$

ここで、 D_p は Anchor と Positive の距離、 D_n は Anchor と Negative の距離、 m はマージンを示す。

triplet loss は、以下のように場合分けできる。

$$\begin{cases} D_p + m > D_n & : L = D_p - D_n + m \\ D_p + m \leq D_n & : L = 0 \end{cases} \quad (7)$$

$D_p + m > D_n$ という状況は、Negative と Anchor の距離 D_n が Positive と Anchor の距離 D_p から、マージン m 分大きな領域内に入っていて、十分に類似しないサンプルの位置を話すことができていない状況を示している。逆に、 $D_p + m \leq D_n$ という状況は、Negative と Anchor の距離 D_n が Positive と Anchor の距離 D_p から、マージン m 分大きな領域の外にあるため、十分に異なるサンプル同士が離れた状況になる。

Triplet Network は、コンテキスト情報をアンカーに与えることで、Siamese Network ではできなかった、複数のクラスに対する距離学習を行うことができる。例えば、

- 画像 A : 男性, 鈴木
- 画像 B : 男性, 田中

というデータが与えられた場合、どういったタスクを行うかによって、A と B の距離を近づけるか、遠ざけるかが変わる。Triplet Network は、Anchor に画像 C: 女性, 鈴木を与えることで、A と C の距離を近づけ、B と C の距離を遠ざけるように学習するし、Anchor に画像 D: 男性, 田中を与えることで、A と D の距離を遠ざけ、B と D の距離を近づけるように学習する。

ただし、Triplet Network も、問題点がいくつか存在する。例えば、学習が停滞しやすい、クラス内距離がクラス間距離より小さくなることを保証しない、といった問題がある。これらの問題を解決するために、Quadruplet loss と呼ばれる損失関数が提案されている。

4 MAML(メタ学習)

MAML(Model-Agnostic Meta-Learning) は、少ないデータによる学習を行うための手法である。MAML は、メタ学習と呼ばれる手法の一つであり、メタ学習は、学習の学習を行う手法である。

深層学習モデル開発に必要なデータ量は膨大であり、特に画像認識などのタスクにおいては、数十万枚の画像データが必要とされる。例えば、MNIST は手書き数字のデータセットであり、7 万枚の画像データが用意されている。また、ImageNet なら約 120 万枚、Open Image Dataset V6 は、約 900 万枚の画像データが用意されている。これらのデータに対して、ラベル付けを行うためには、人手でラベル付けを行う必要があり、膨大な時間とコストがかかる。

こうした問題から、少ないデータで如何にして学習を行うかが重要となる。

少ないデータで学習を行う手法として、事前学習、転移学習、ファインチューニングという一連の手法は比較的古くから存在している。この手法は、大規模なデータセットで学習したモデルを、新しいタスクに適用することで、少ないデータで学習を行うことができる一方で、必ずしも最適な重みを得ることができるとは限らないという問題があった。

MAML は、この問題を解決するために提案された手法であり、様々なタスクに合わせて、ファインチューニングを行うことで、あらゆるタスクに対して有効な汎用的なモデルを学習することができる。

MAML は、以下のような手法で学習を行う。

1. タスク集合 τ から、タスク τ_i をランダムに選択する (Inner Loop)
2. タスク τ_i に重み θ を最適化し、新しい重み θ'_i を得る (Inner Loop)
3. すべてのタスクに対して重みを更新し、重みの集合 $\Theta = (\theta'_1, \theta'_2, \dots)$ を得る (Outer Loop)
4. 集めた重みで共通重み θ を更新する (Outer Loop)
5. 1-4 を繰り返す

この手法により、少ないデータで効果的に学習を行うことができる。しかし、MAML は、計算コストが高いという問題がある。この問題を解決するために、First-order MAML(2 次以上の勾配を無視)、Reptile(InnerLoop の逆伝播を行わない) という手法が提案されている。

5 グラフ畳み込み (GCN)

グラフ畳み込み (Graph Convolutional Network, GCN) は、グラフデータに対して畳み込み演算を行うニューラルネットワークである。グラフデータは、ノードとエッジから構成されるデータであり、例えば、ソーシャルネットワーク、化合物の構造、交通ネットワーク、新型コロナの感染経路などが挙げられる。

グラフ畳み込みでは、まず対象のノードに対して、その近傍のノードとの情報を集約する演算を行う。その後、次に近いノードとの情報を集約する演算を行い、これを繰り返すことで、グラフ全体の情報を集約する。

グラフ畳み込みの手法には、Spatial GCN, Spectral GCN, ChebNet, GraphSAGE, GAT, GraphWavelet, Graph Isomorphism Network (GIN) などが存在する。これらの手法は、グラフデータの特性に合わせて、畳み込み演算を行う手法が異なる。

6 Spatial GCN

Spatial GCN は、グラフデータに対して、クラスタリングを繰り返し、空間的な畳み込み演算を行う手法である。Spatial GCN は、グラフデータの隣接行列を用いて、ノードの特徴量を更新する。Spatial GCN は規則性が弱いものであっても、規則性を見出すことができるかもしれないという強みがある。一方で、次元が低く近傍のある場所に限られる場合、広い範囲で重みを持たせにくいという弱点がある。

7 Spectral GCN

Spectral GCN は、グラフデータに対して、フーリエ変換を用いて畳み込み演算を行う手法である。Spectral GCN は、グラフデータのラプラシアン行列を用いて、ノードの特徴量を更新する。Spectral GCN は、グラフデータの特性を捉えることができるが、グラフデータのラプラシアン行列の固有値分解を行うため、計算コストが高いという問題がある。また、グラフ間でサイズや構造が変わるとパラメータの使いまわしができないという問題もある。

■参考文献

1. 岡谷貴之/深層学習 改訂第 2 版 [機械学習プロフェッショナルシリーズ]/ 講談社サイエンティフィク/
2022-01-17