

株式会社 ALBERT 入社課題レポート

千枝睦実

2019/03/07

1 課題

出題された課題は以下の通りである。本レポートでは、課題5に該当するモデルの構造やハイパーパラメータ、精度について考察とともに述べる。

1. Linear 層, ReLU 関数からなる 3 層のニューラルネットワークを定義せよ。ニューラルネットワークの出力は 64 次元のベクトルとなるようすること。
2. Triplet loss を用いて 1 で定義したモデルを訓練せよ。その際、正常品と正常品および不良品と不良品との距離は小さく、正常品と不良品の間の距離は大きくなるようすること。同時に、異なる不良タイプ（数字）間の距離も大きくなるようにせよ。Triplet loss の margin は適当に調節すること。
3. 訓練されたモデルを使い、評価セット中のそれぞれのサンプルについて不良度を推定せよ。学習データ中の正常品を用いて基準となるベクトルを作成し、その基準ベクトルからの距離を不良度とすること。
4. モデルを評価せよ。評価指標としては、Recall 0.999 を満たす検出閾値を設定した場合の True Negative Rate（真陰性率）を用いること。
5. 作成したモデルの構造やハイパーパラメータ、精度等をレポートにまとめよ。また、上記設問に従って作成したモデルの欠点を指摘し、改善案を自由に記述せよ。

2 開発環境

開発に使用した環境を表 1 に示す。このうち、matplotlib は本レポートにある図表の作成にのみ使用した。

表 1: 開発環境			
種別	名称	バージョン	備考
OS	Windows	10 ビルド 17134.590	-
CPU	Intel Core i7-4790 CPU 8GB	-	3.6GHz
GPU	GeForce GTX 1060 6GB	-	-
言語	Python	3.7.1	-
モジュール	CUDA	v10.1	-
	cupy	5.3.0	ビルド 100010
	numpy	1.16.2	-
	chainer	5.3.0	-
	matplotlib	3.0.3	-

3 モデル構造

作成したモデルの構造を図3に示す．課題1に示された通り，3層の Linear 層でニューラルネットワークを形成し，ユニットによる次の層への出力には ReLU 関数を用いた．

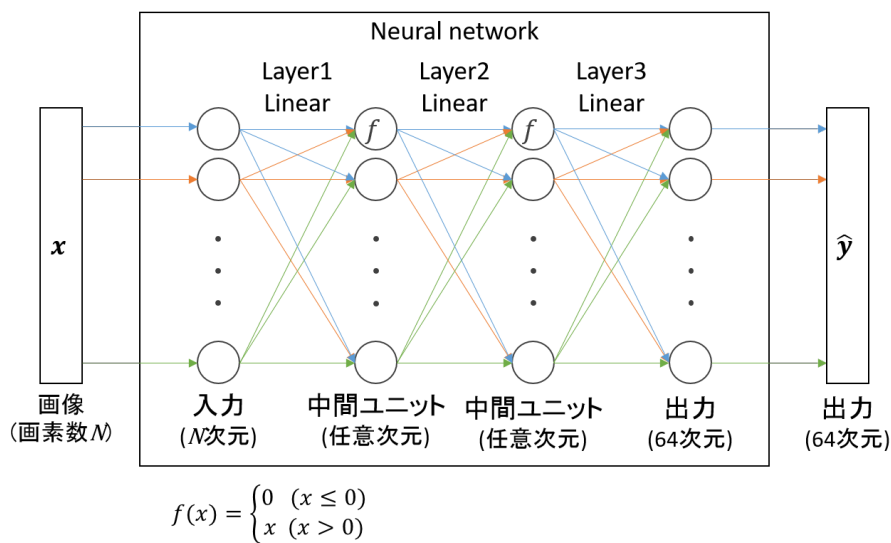


図 1: 作成したモデルの構造

今回の課題では，テストケースにある未知の値に対応するため Triplet loss 関数を用いた距離学習を行う．そこで，各サンプルに対するトリプレットを予め作成しておき，訓練データとした．したがって，作成したモデルは訓練

データのトリプレットロスを最小化するように各サンプルに対して距離学習する。

その後、正常画像 (0,6,9) の基準ベクトルを作成し、基準ベクトルとの距離が検知閾値以上のものを異常画像とした。検知閾値は 0 から大きくしていき、 $\text{Recall} \geq 0.999$ を満たさなくなった時点で最も大きい真陰性率 (selectivity) を出した閾値を採用し、最終的な予測を出力する。

4 モデルの評価

モデルのハイパーパラメータの初期値を表 2 に示す。なお、エポック数は 50 に設定しているが、Early stopping を導入しているため、実際はこれよりもエポック数は少なくなる。

表 2: 各ハイパーパラメータの初期値

ハイパーパラメータ	値
最適化アルゴリズム	SGD
中間層ユニット数	256
ミニバッチ数	100
Triplet loss 関数のマージン	0.2

このパラメータで学習したモデルによる $\text{Recall} \geq 0.999$ 時の最大真陰性率は、0.06719 であり、エポック数は 46 だった。

ここで、検出閾値ごとの Recall と真陰性率を図 4 に示す。 $\text{Recall} \geq 0.99$ だった場合の真陰性率は 0.32440、 $\text{Recall} \geq 0.9$ だった場合の真陰性率は 0.78792 であった。そのため、学習・予測自体は正常に行われていると判断し、ここからはハイパーパラメータ探索で改善する。

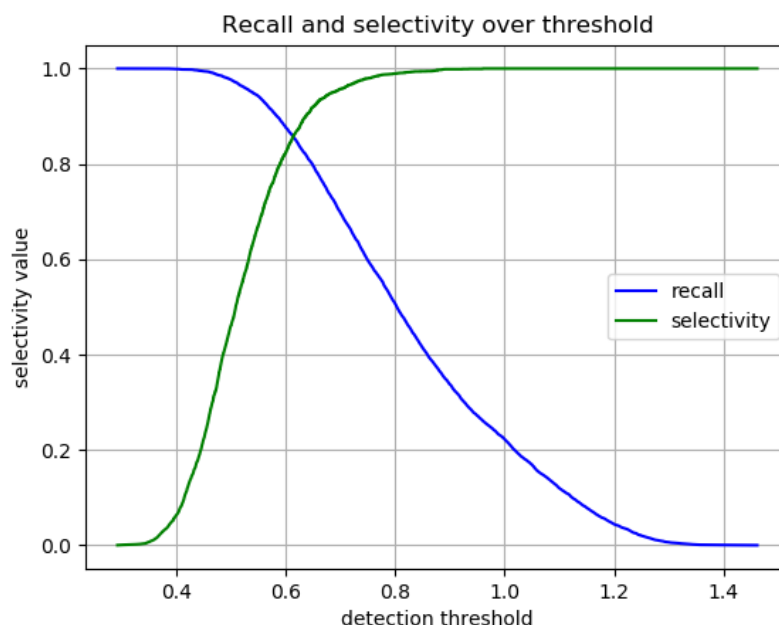


図 2: 検出閾値ごとの Recall と真陰性率

4.1 アルゴリズム比較

まず、最適化アルゴリズムの検討を行った。各アルゴリズムを用いたときの、 $\text{Recall} \geq 0.999$ における真陰性率を表 3 に示す。この中で、CorrectedMomentumSGD アルゴリズムを用いたモデルが、最も高い真陰性率 (0.19206) を出した。そこで、その他のパラメータはこのアルゴリズムを用いたモデルで探索する。また、このうち、AdaDelta, Adam, RMSprop, SMORMS3 アルゴリズムでは、テストケースでの Triplet loss がエポックごとに増大していったため、どれも 4 回のエポックで Early stopping が行われた。原因については、期間の関係で検証できなかった。

表 3: 最適化アルゴリズムと真陰性率

アルゴリズム	真陰性率
AdaDelta	0.05327
AdaGrad	0.08042
Adam	0.05836
CorrectedMomentumSGD	0.19206
NesterovAG	0.12080
RMSprop	0.02681
RMSpropGraves	0.09874
SGD	0.06719
SMORMS3	0.04309

4.2 Triplet loss のマージン

次に，アルゴリズムを真陰性率の高かった CorrectedMomentum アルゴリズムに変更し，最適な Triplet loss のマージンを探索する．マージンの，各値での $\text{Recall} \geq 0.999$ における真陰性率を表 4 に示す．マージンが 0.3 の時に，最も高い真陰性率 (0.20835) を出した．

表 4: Triplet loss 関数のマージンと真陰性率

マージン	真陰性率
0.01	0.10757
0.1	0.19206
0.2	0.19206
0.25	0.16050
0.3	0.20835
0.35	0.11503
0.4	0.14388

4.3 中間層のユニット数

マージンを 0.3 に変更し，中間層のユニット数を探索する．中間層のユニット数の，各値での $\text{Recall} \geq 0.999$ における真陰性率を表 5 に示す．中間層のユニット数が 256 の時に，最も高い真陰性率 (0.20835) を出した．

表 5: 中間層のユニット数と真陰性率

ユニット数	真陰性率
64	0.05701
128	0.00441
200	0.06481
255	0.08551
256	0.20835
257	0.08483
300	0.04208
400	0.06820
512	0.06108
1024	0.09773

4.4 ミニバッチサイズ

中間層のユニット数を 256 に変更し，ミニバッチサイズを探索する．ミニバッチサイズの各値での $\text{Recall} \geq 0.999$ における真陰性率を表 6 に示す．ミニバッチサイズが 100 の時に，最も高い真陰性率 (0.20835) を出した．

表 6: ミニバッチサイズと真陰性率

ユニット数	真陰性率
10	0.002850
50	0.19545
95	0.16661
99	0.18681
100	0.20835
101	0.18867
105	0.12182
120	0.18697
150	0.16254
200	0.19104
300	0.14048
400	0.06820
500	0.16966
1000	0.17374

4.5 探索結果

今回のハイパーパラメータ探索で得られた結果を表7に示す。このパラメータで得られた $\text{Recall} \geq 0.999$ における真陰性率は 0.20835 であった。本レポートでは、各パラメータを順番に調整したが、本来であれば、グリッドサーチや、optuna ライブラリによるパラメータ最適化の方法を用いるべきであると考えられる。

表 7: 探索で得られたハイパーパラメータ

ハイパーパラメータ	値
最適化アルゴリズム	CorrectedMomentumSGD
中間層ユニット数	256
Triplet loss 関数のマージン	0.3
ミニバッチ数	100

5 考察

5.1 モデルの欠点

今回の課題では、使用するニューラルネットワークは全結合層が3層の構成であった。

全結合層の問題点として、情報を1次元のベクトルとして入出力するため、画像を認識する際、空間的な位置情報が無視される点が挙げられる。たとえば、64画素の画像で、 $(x, y) = (1, 2)$ の画素と $(x, y) = (1, 3)$ の画素は隣りあっているが、全結合層では $x = 9$ の画素と $x = 17$ の画素として扱われ、隣接していたという情報が失われている。

また、全結合層は、接続数が多く、必要な計算量や勾配消失の観点から、今日よく使われる高画質な画像には対応できない点も挙げられる。

5.2 改善案

1. 画像認識では、特徴抽出を行う畳み込み層と、物体の位置の違いを吸収するプーリング層とを交互に配置した畳み込みニューラルネットワーク (CNN) が適するとされている。その理由として、以下の点が挙げられる。

- 畳み込み層の複数のフィルタによって、画像のサイズごと、部分ごとの特徴を保持できる。

- プーリング層による圧縮によって、物体の位置の微少なずれに影響されにくい。

そこで、今回のモデルの代わりに CNN を使用して学習するという案が考えられる。

2. 異常とする画像が訓練データに少なく、十分に学習できなかった可能性が考えられる。そこで、異常なサンプルをアフィン変換や射影変換などで変換したものを訓練データに加えて学習する方法が考えられる。
3. ハイパーパラメータの探索が不十分だった可能性が考えられる。そこで、optuna などのライブラリを用いて最適なパラメータを求めた上で学習する方法が考えられる。