

## 進捗報告

### 1 今週やったこと

- 先週の修正
- PointNet 追加実験

### 2 今週の収穫

今週は先週の修正をいくつかした。

#### 2.1 先週の修正

先週使用していたデータセットは, CNN の画像認識用でありフィルタ処理 (縮小処理) をかけただけのものも含まれていたため PointNet では簡単すぎた. (PointNet ではデータを座標で扱うため, フィルタ処理をかけたものでも元と同じデータになる)

そこで, データセットの中身を星座が [真ん中, 右下, 左下, 右上, 左上] に写っているもの (5 種類)  $\times$  回転画像 (12 種類)  $\times$  切り取る平面の大きさを変えたもの (4 種類) = 240 枚に変更した.

また PyTorch で初めて実装したので loss の書き方を間違っていることに気が付いた. 修正して実験パラメータを以下の表 1 に変更して再実験した.

表 1: 実験パラメータ

訓練 : テスト	9 : 1
最適化関数	Adam
学習率	$1.0 \times 10^{-4}$
損失関数	binary cross entropy
エポック数	500
バッチサイズ	10

図 1 に loss の推移を示す. これに関しては解決した気はする.

#### 2.2 PointNet の追加実験

一応, 座標を 3 次元にしたバージョン ( $z$  座標を 0 にしたもの) と畳み込み層を追加したものも追加で実験した. 図 2 に 3 次元バージョンの, 図 3 に畳み込み層を追加したバージョンの loss の推移を示す.

なお, 表 2 にそれぞれのテスト識別率を示す. 予想通りではあるが畳み込み層を追加したものが最も識別率が高かった.

本当は SVM でもやってみたかったのだが SVM での集合の扱いがわからなかったのでできなかった. (ちなみに画像として VGG19 で識別した場合には全ての識別に成功していた.)

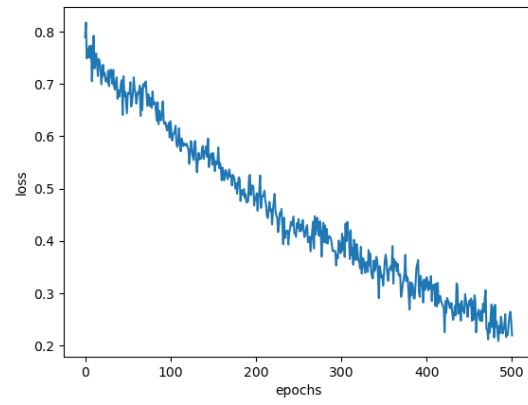


図 1: loss の推移

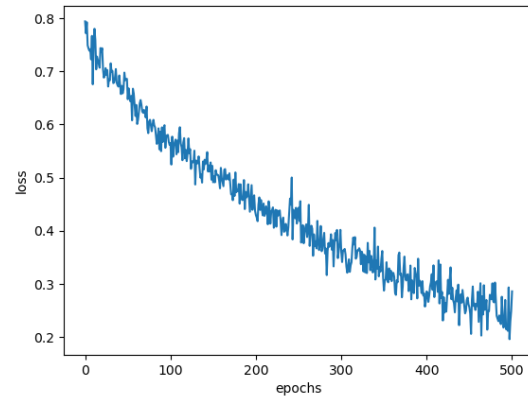


図 2: loss の推移 (3 次元)

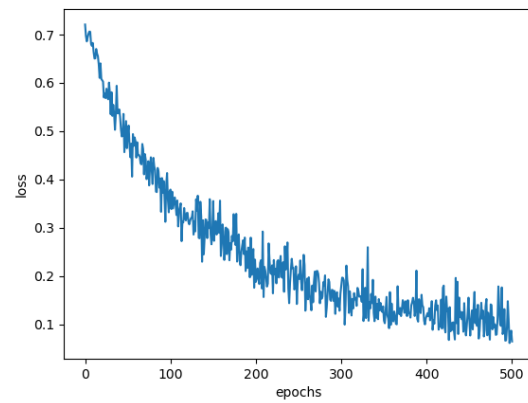


図 3: loss の推移 (畳み込み層追加)

表 2: テスト識別率

	2 次元	3 次元	畳み込み (3 次元)
識別率	0.792	0.646	0.854

### 3 今後の方針

ネットワークを識別用ではなくセグメンテーション用に拡張していきたい. また来週は JSAI の発表練習ができるように準備しておく.