

進捗報告

1 今週やったこと

- PointNet の実装 (2 クラス分類)

2 今週の収穫

2.1 PointNet の実装

先週紹介した PointNet を二次元座標バージョンで軽く実装してみた。図 1 に Point Net のアーキテクチャを再掲する。

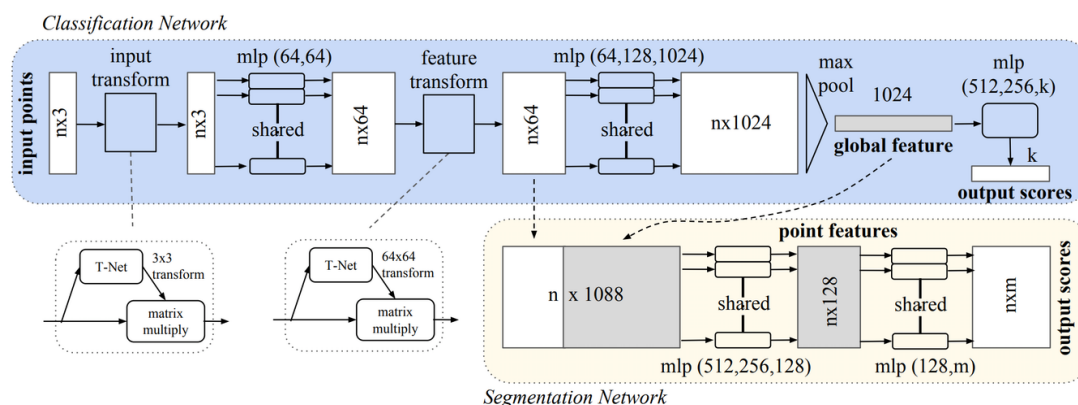


図 1: PointNet のアーキテクチャ

一様分布と正規分布から三次元データをランダムサンプリングし、PointNet を使ってどちらからサンプリングされたのかを予測する二値分類タスクを試みているコードがあったため、それを二次元座標用に変換して実装した。図 2 にモデル構造を示す。

今回は、予備実験としてオリオン座とさそり座の 2 クラス識別実験をした。使用したのはオリオン座とさそり座の星図各 288 枚 (合計 576 枚) である。データの前処理として、星図を 256×256 に整形し、画像の左上を原点として全ての星の座標を獲得した。次に、各画像につきサイズが大きい星上位 10 個の星の座標のみを抽出し、これを入力データとする。また表 1 に実験パラメータを示す。

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	1.0×10^{-3}
損失関数	binary cross entropy
エポック数	100
バッチサイズ	10

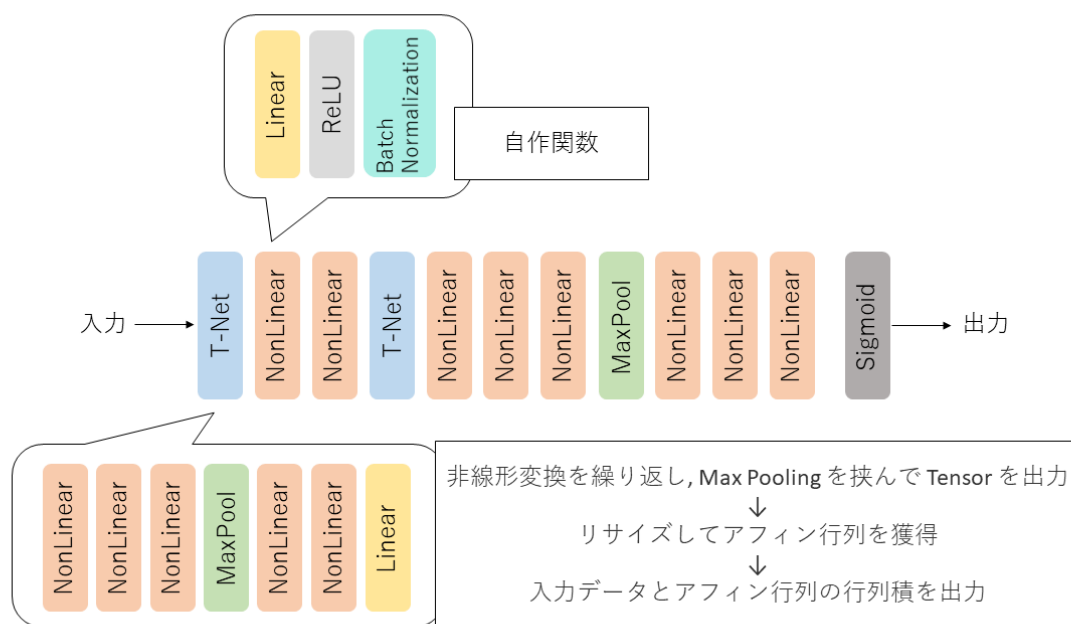


図 2: PointNet のモデル構造

図 3 に loss の推移を示す. すごく荒ぶっているが一応下がっている. またテスト結果は 0.93 (58 枚中 54 枚) であった.

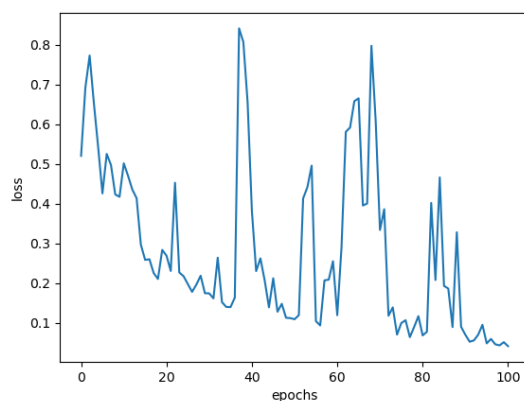


図 3: loss の推移

3 今後の課題

- ・ loss の推移が割と荒ぶっているがこれは大丈夫なのか.
- ・ 今回のモデルに畳み込み層は一切使っていない. (参考にしたコードが使っていなかったため) NonLinear 層に畳み込み層を追加したりしたほうがいいのか.

・今回は星図を識別対象としたため、星の座標の抽出をサイズが大きな上位 10 個にしたが、スマホなどで撮った写真にはそれほど写っている星の数は多くない。(かといって減らしすぎると識別が難しくなってしまう)

4 今後の方針

ある程度現在のネットワークを改良したのち、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。