進捗報告

1　今週やったこと

- 点集合の深層学習手法のサーベイ

2　今週の収穫

2.1　点集合の深層学習手法のサーベイ

点集合の深層学習手法には, Deep Sets と PointNet が挙げられる.

2.1.1　Deep Sets

Deep Sets では, 順不変性が保証される. つまり, 点の順列を入れ替えて機械学習モデルに入力したとしても出力が不変である, というものである. 論文内においては, Permutation Invariant と Permutation Equivariant の二種類のモデルが載せられていた. 図 1 に二つの違いを載せる.

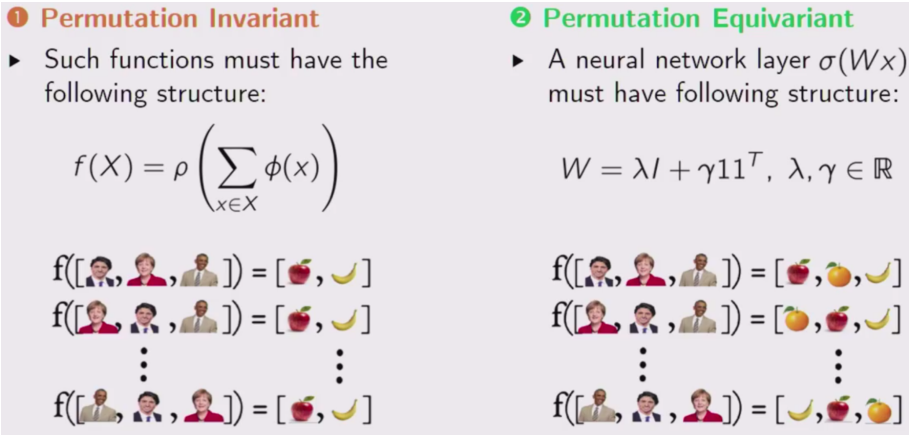

図 1: モデルの違い

・Permutation Invariant:
入力を $\boldsymbol{X} = [x_1, x_2, ..., x_M]$, 出力を $\boldsymbol{Y} = [y_1, y_2, ..., y_M]$ としたとき, 入れ替えに対して不変な関数 f ($f : 2^{\mathfrak{X}} \to \mathcal{Y}$) は任意の入れ替え π に対して以下の (1) 式を満たす.

$$f(\{x_1, x_2, ..., x_M]\}) = f(\{x_{\pi(1)}, x_{\pi(2)}, ..., x_{\pi(M)}]\}) \tag{1}$$

また, 一般に入れ替えに対して等価な関数は以下の (2) 式のように書ける.

$$f(X) = \rho(\sum_{x \in X} \phi(x)) \tag{2}$$

なお $\rho()$, $\phi()$ は適当な関数である. 図 2 にモデルのアーキテクチャを示す.

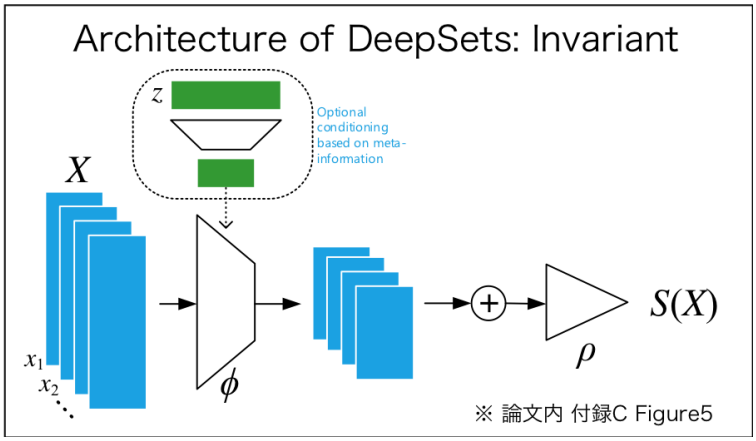

図 2: モデルのアーキテクチャ (Invariant)

・Permutation Equivariant:

入力を $\boldsymbol{X} = [x_1, x_2, ..., x_M]$, 出力を $\boldsymbol{Y} = [y_1, y_2, ..., y_M]$ としたとき, 入れ替えに対して等価な関数 $\boldsymbol{f}$ ($\boldsymbol{f}$: $\mathfrak{X}^M \to \mathcal{Y}^M$) は任意の入れ替え π に対して以下の (3) 式を満たす.

$$\boldsymbol{f}([x_{\pi(1)}, x_{\pi(2)}, ..., x_{\pi(M)}]) = [f_{\pi(1)}(\boldsymbol{X}), f_{\pi(2)}(\boldsymbol{X}), ..., f_{\pi(M)}(\boldsymbol{X})] \tag{3}$$

また, 一般に入れ替えに対して等価な関数は以下の (4) 式のように書ける.

$$f(\boldsymbol{X}) = \sigma(\boldsymbol{X}\Lambda - \mathbf{1}\mathrm{maxpool}(\boldsymbol{X})\Gamma) \tag{4}$$

なお σ は要素ごとに作用する活性化関数, Λ, Γ は学習で最適化するパラメータで, 入力チャネル数 × 出力チャネル数の行列, $\mathbf{1}$ は 1 が M 個並んだベクトル $[1, 1, ..., 1]^T$ である. 図 3 にモデルのアーキテクチャを示す.

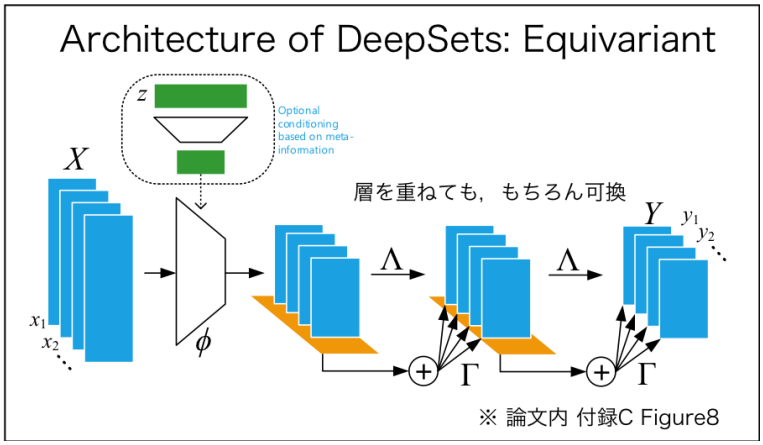

図 3: モデルのアーキテクチャ (Equivariant)

ただ Deep Sets の論文内では分類問題は行っていたが検出やセマンティックセグメンテーションは行われていなかった. 複数の星座が写っている場合にはあまり向かないかもしれない.

2.1.2 PointNet

PointNet は Deep Sets とほぼ同時期に発表された, 点群データを対象とした深層学習モデルである. (直接比較は行われていないためどちらが優れているかについてはなんとも言えないが, Deep Sets は点群に限らずジェネラルな問題で入れ替え不変の問題を解いているのに対し, PointNet は点群に特化しているイメージ)

PointNet には以下の 3 つの性質が保証される.

- 順不変性: (1) 式
- 移動不変性: 平行移動や回転移動を作用させた点群データを機械学習モデルに入力したとしても出力は不変.

  平行移動不変性は以下の (5) 式を満たす.

$$f(x_1 + \boldsymbol{r}, x_2 + \boldsymbol{r}, ..., x_M + \boldsymbol{r}) = f(x_1, x_2, ..., x_M) \tag{5}$$

  なお, $\boldsymbol{r}$ は平行移動させるベクトル量である.

  回転移動不変性は以下の (6) 式を満たす.

$$f(Rx_1, Rx_2, ..., Rx_M) = f(x_1, x_2, ..., x_M) \tag{6}$$

- 局所性: 空間的に近い点同士は何らかの密接な関係性を有していて, 空間的に遠い点同士は関連性が小さい. (実際は PointNet では満足できていない)

図 4 に PointNet のアーキテクチャを示す.

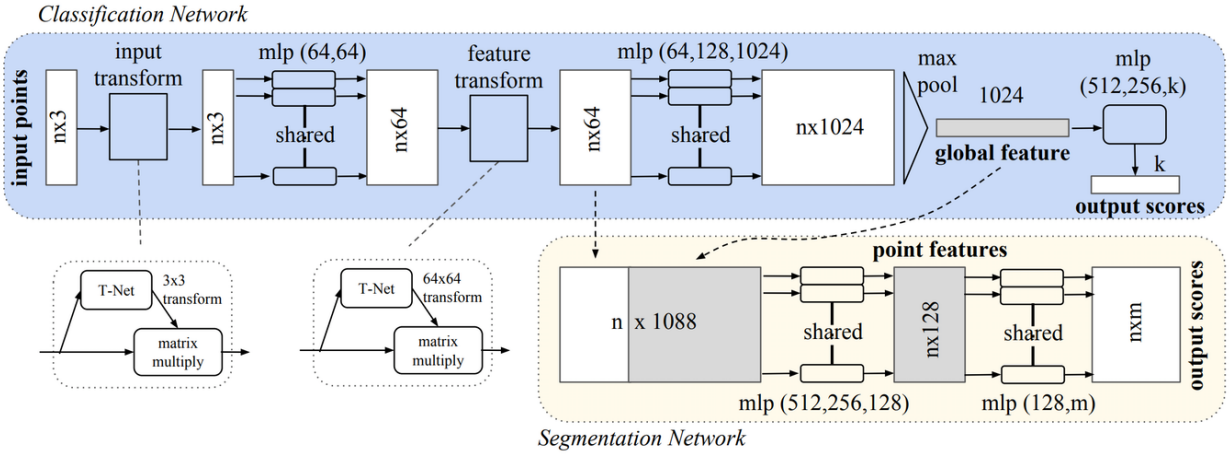

図 4: モデルのアーキテクチャ (PointNet)

分類問題の場合, まず input transform で入力点群に対してアフィン変換を施し, 移動不変性を近似的に獲得する. 次にアフィン変換後の点群に対してニューラルネットワークで処理を行い, feature transform で再びアフィン変換を施す. そしてニューラルネットワークで処理し, 最後に max pooling で順不変性を獲得して出力を得る.

なお input transform では入力点群に対してアフィン行列を作用させる. このアフィン行列は T-Net の出力として得られる. T-Net はミニ PointNet のような構造となっており, ニューラルネットワークと max pooling の組み合わせからなる. この T-Net に三次元点群を入力することで出力としてアフィン行列が得られる.

また PointNet は Tensorflow, Pytorch などいくつかの種類で実装例が存在するため実装面における心配はなさそう. (Deep Sets はあまり実装例が見つからなかった)

さらに PointNet を拡張したネットワークもいくつか見つかった. 以下に PointNet の拡張例を示す.

- PointNet++: PointNet では満足できていなかった局所性を満たすように改良されたもの
- VoxelNet: PointNet + CNN
- PointPillars: PointNet + CNN(SSD)

2.2 点群ネットワークの問題点

点群ネットワークで使用されていたデータセットは, CAD のデータであるためそれに合わせたデータが必要かもしれない. もともと 3D 物体を想定してのネットワークなので二次元座標でうまく対応できるかは分からない.

3 今後の方針

やりたいこと: 点集合の中から特定の点集合の特定 (検出またはセマンティックセグメンテーション)
Deep Sets の考え方を基にこれに適したネットワークを作成したほうがいいのかもしれない.