Graph Convolutional Network 概説

非常に簡潔なまとめ

@UMU____

目次

このスライドはGraph Convolutional Networkを簡単に説明したもので，私の主観や間違いを含んでいる可能性があります．

- Graph Convolutional Networkとは？
- グラフの畳み込み演算とは？
- まとめ

Graph Convolutional Networkとは？

- Graph Convolutional Network (GCN) Is 何？

グラフに対しての畳み込み演算を用いてニューラルネットワーク(NN)を構成する手法

- グラフ Is 何？

グラフ理論のグラフのこと．棒グラフ等のことではない．

グラフはノード（頂点）とエッジ（辺）とエッジの重みで構成されている．エッジはノードを結ぶ．ノード間にエッジがあってもなくてもよい．グラフはノードとエッジと重みだけで決まる．

- グラフに対してのNNとは？

グラフの頂点にデータを入力して，グラフの頂点からデータが出てくるような，NN．

Graph Convolutional Networkとは？

- 畳み込み演算

一般的に知られている（グラフとは関係ない）畳み込み演算$*$は,
数式で書けば，次のようになる(連続,1次元の場合)

$$f * g = \int f(x-y)g(y)dy$$

と表せる． $f(x)$が入力で， $g(y)$がフィルターである．
離散的な畳み込み演算は

$$\sum_{y\in\{-c,-c+1,-c+2,\dots,c-1,c\}} f[x-y]g[y]$$

となる．ここでcはフィルターのサイズに関係する（サイズは2c+1）．一般的に入力データのサイズよりフィルターの方が小さい．つまり均一かつ局所的な演算となる．

Graph Convolutional Networkとは？

- GCNを構成する意味は？

「グラフの頂点にデータを入力して，グラフの頂点からデータが出てくるような，NN」を構成するだけであれば，グラフという事を考えずに，頂点to頂点の全結合NNを構成するのでも問題ない．

しかしながら，昨今のDNNでは，元データの性質をうまく活用することでパラメータを削減・共有することで成功を収めた．

例）CNN (画像やボリュームデータ等に対して)

例）RNN（時系列データ等に対して）

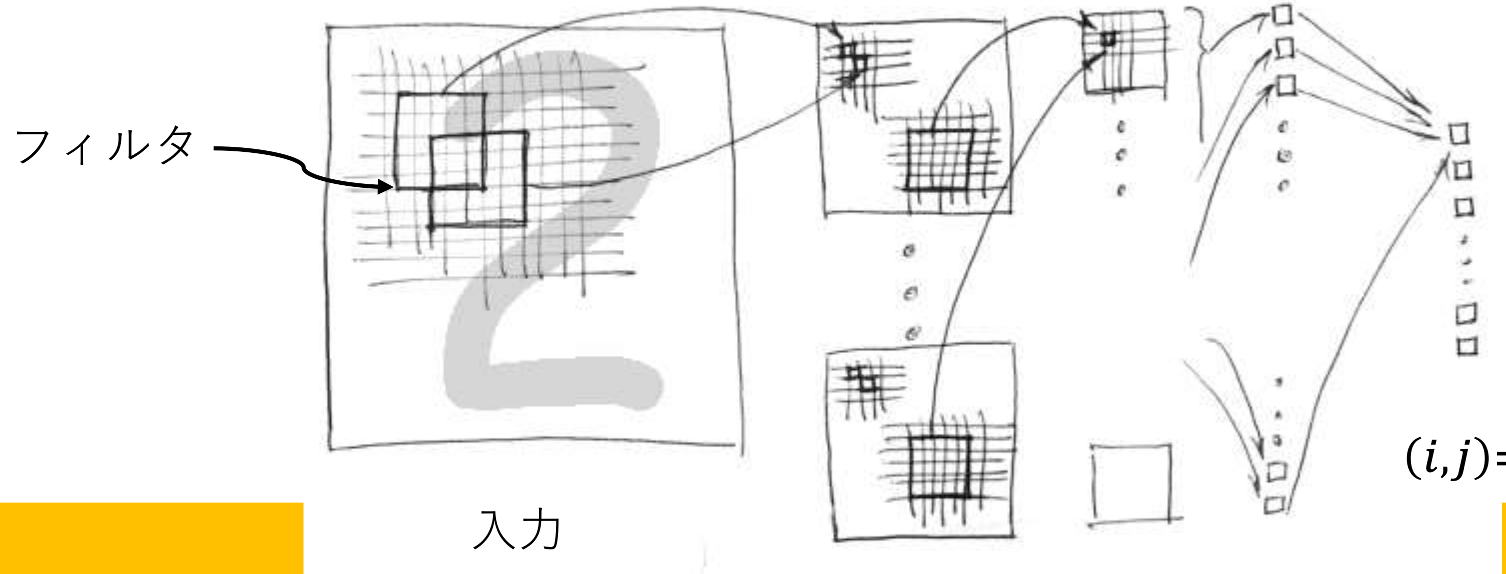
よって，ノード間の関係が既知（エッジとその重み）という性質を持ったグラフに対して，うまく構成されたＮＮがあるとよい．

→その１手法として，畳み込みを利用した：ＧＣＮ

Graph Convolutional Networkとは？

- 畳み込みニューラルネットワーク(CNN)とは？

（一般的な）畳み込み演算を用いたNN.
入力とフィルターが畳み込み演算されて出力される.
CNNではフィルターを決定付けるパラメータを学習すべき量とする.
CNNは画像の特性を生かしたNNの構造である.



二次元での畳み込み演算

$$\sum_{(i,j)=(0,0)\text{の周り}} f[x-i,y-j]\,g[i,j]$$

グラフの畳み込み演算とは？

2つの違うやり方があるよ

- グラフの畳み込み演算

グラフに対して(一般的な)畳み込み演算を行うことはできない.

なぜなら，畳み込み演算はグラフに対しては定義されていないからである.

→そこで，GCNでは，

手法①数学的に妥当な畳み込み演算を定義し，それを用いる.

「数学的に妥当」という言葉が数学的でない件については見逃してください.

手法②畳み込みに似たことをする演算を用いる.（複数手法があるが，原理的な考えを紹介する.）

畳み込みをするという意味で

によってグラフに畳み込み演算を導入し，CNNと同様，グラフのノードに対する入力から，畳み込み演算を通して，グラフのノードに対する出力を得る．学習するのは畳み込み演算のパラメータである.

グラフの畳み込み演算とは？：手法①

手法①は，畳み込みをグラフ上で数学的に定義したいという考えに基づく．これをするために，一般的な畳み込み演算が満たしている条件と同じような条件を満たす演算を持ち込み，それをグラフ上の畳み込み演算とする手法である．

- その演算の性質とは，畳み込み演算はフーリエ係数では要素積となる性質である．

→グラフフーリエ変換を行って，そこで（学習すべきパラメータとなる）値を要素積し，逆グラフフーリエ変換を施す，という一連の作業を，グラフ畳み込み演算とする．

グラフの畳み込み演算とは？：手法①

- グラフフーリエ変換とは？

（一般的な）フーリエ変換は，ラプラシアンの固有関数によって関数を展開した形：

$$F(\omega) = \langle f(t), p_\omega(t) \rangle$$
$$\mathcal{L} p_\omega = \lambda_\omega p_\omega$$
$$\downarrow$$
$$p_\omega(t) = e^{i\omega t}, \lambda_\omega = -\omega^2$$

となっている．

→類推により，グラフラプラシアンの固有ベクトルでグラフ信号を展開した形をグラフフーリエ変換とする

グラフの畳み込み演算とは？：手法①

グラフラプラシアンを，グラフの隣接行列Aと次数行列Dによって

$$L = D - A$$

と定義する．また，これは

$$x^T L x = \sum_{(i,j)\in edge} \overbrace{W_{ij}}^{\text{辺ijの重み}} \big(x(i) - x(j)\big)^2$$

と定義するのと同じである．（よってLは半正定値行列）

グラフラプラシアンの固有ベクトルu_kは，次式より得られる．

$$L u_k = \lambda_k u_k$$

グラフラプラシアン自体にもとても面白い性質があるが割愛．

この固有ベクトルによって，グラフ信号Xを

$$X = \sum \beta_k u_k = UB$$

と展開したとき，Bがグラフフーリエ変換によって得られる係数である．

グラフの畳み込み演算とは？：手法①

グラフフーリエ変換を$\mathcal{F}[X]$と書くことにする.
ここでグラフラプラシアンは(グラフが無向の場合)実対称行列であるから，Uは直交行列とできる．すると，

$$\mathcal{F}[X] = U^T X = B, \mathcal{F}^{-1}[B] = UB = X$$

を導く.

- グラフ上の畳み込み演算を，次のように定義する.
- Θは畳み込みの**フィルタ**（ベクトル）．⊙は要素積.

Θ_iを対角要素にもつ対角行列

$$\mathrm{X} * \Theta = \mathcal{F}^{-1}\left[\Theta \odot \mathcal{F}[X]\right]$$

または， diag(Θ)を用いて，

$$\mathrm{X} * \Theta = \mathrm{U}\,\mathrm{diag}(\Theta)\,\mathrm{U}^{\mathrm{T}}\,\mathrm{X}$$

となる．手法①終.
手法①は，数学的にきれいだが理論的・計算量的にかなり複雑.

グラフの畳み込み演算とは？：手法②

- 手法②

手法①は，数学的にきれいだが理論的・計算量的にかなり複雑であった．手法②では，手法①を近似し，かつ直感的な計算法を導きたい．

ここで，畳み込み演算

$$\mathrm{X} * \Theta = \mathrm{U}\ \mathrm{diag}(\Theta)\ \mathrm{U}^{\mathrm{T}}\ \mathrm{X}$$

のフィルタΘ_kが，対応する固有値λ_kのp次多項式であるとき，畳み込み演算は，頂点からp近傍内(頂点から辺をp回までたどったときにたどり着ける頂点全体)に入力されているデータの線形和で表せることが分かっている．（参考文献３に詳しくある）

このような議論の末，畳み込み演算を，頂点の1近傍内(自分自身とその周り）だけの線形和で近似する計算法を導入し，それを用いる．

グラフの畳み込み演算とは？：手法②

数式で表現すれば,

$$AXW$$

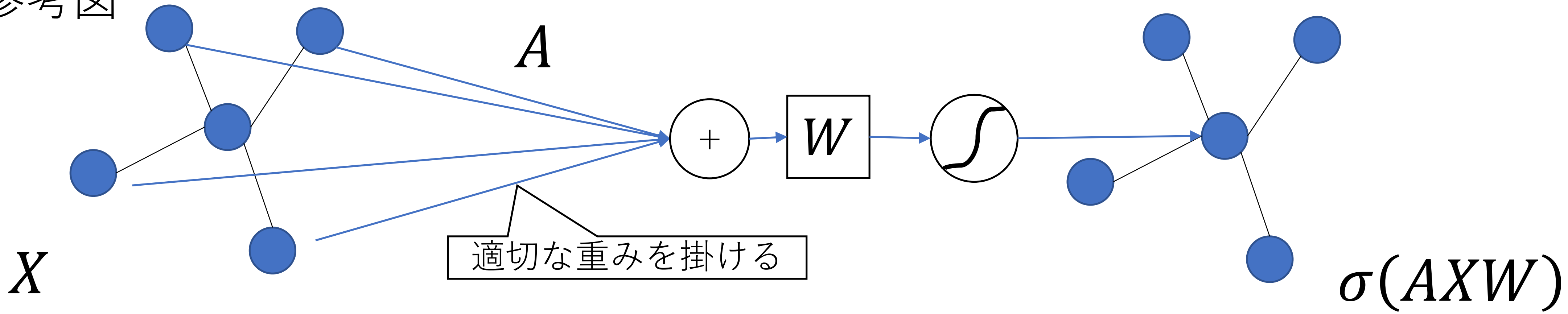
となる．Aが隣接行列，Wがフィルターである． A:N*N,X:N*D,W:D*F

GCNは，これに対して非線形項(ReLu等)を加えたもの

$$\sigma(AXW)$$

であり，これを適当に改変した物がGCNとして使われている．手法②終.

参考図



まとめ

- グラフにとって「良い」NN構造を作るために，グラフ上の畳み込み演算の定義という形でGCNは作られた．

主な参考文献

- １ ：How powerful are Graph Convolutions? (review of Kipf & Welling, 2016)

http://www.inference.vc/how-powerful-are-graph-convolutions-review-of-kipf-welling-2016-2/

- ２ ：GRAPH CONVOLUTIONAL NETWORKS

https://tkipf.github.io/graph-convolutional-networks/

- ３ ：グラフ信号処理のすゝめ

https://www.jstage.jst.go.jp/article/essfr/8/1/8_15/_pdf