

Q3, What does this tell you about designing deep learning architectures for processing spatial or spatio-temporal data (like images and videos)?

空間的 時空間的

DL Book 9.2 Motivation (Convolution)

① sparse interaction (疎な結合) ←

② parameter sharing

③ equivalent representation

④ 小さな kernel を利用することで実現できる
画像処理では保管すべき param
の数を減らせる

行列の掛け算 $O(m \times n)$ k 個の積を
↓ 行ったり
 $O(k \times n)$ 制約を
↑ 加える

[2] Parameter sharing

例 H^2 上の vv に対する Kernel の基底は、 $\frac{1}{\sqrt{2}}, \frac{3}{\sqrt{2}}, \dots$ の

↑

入力の全2の位置で1回、5分

よって、全ての入力位置に対して別々に重みの集合を

学習するのではなく、

1つの集合の非空部分集合

[3] Equivariance

$$f(g(x)) = g(f(x)) \iff f(x) \in g \text{ (等値関係)}$$

guard page shift 時間数を抑える

これは量や述詞関数 f ? と等価

量や形は画像のスケールが回転といった変換に対して等価でない (別のx変換が必要)

Pooling

プーリングは入力の小さな移動に対してほぼ不変(Invariant)な表現を作るのに役立つ

プーリングを利用するというのは、学習する関数に対して小さな移動に対して不変であるという強い前提を置いてみるに見る

9.4 事前分布

prior probability distribution

○ 弱い事前分布

大きな分散を持つガウス分布

エントロピーの大きい事前分布

→ テータがパラメータを
多く自由に動かす

ことを許容している

○ 強い事前分布

小さな分散を持つガウス分布

エントロピーの小さい事前分布



無限に強い事前分布: テータがどれほどあり得るかを制限するかわからず、パラメータがいくつかの値にのみ確率をゼロにして、完全に使用禁止とする事前分布

畳み込みネットワークは重みに対して無限に強い事前分岐を持つ全結合ネットワークに類似したものとみなせる
Pooling も同様

Convolution とは実装する方が効率的だけ

解析において全結合と比べた計算量が
得られる

→ 過剰適合の可能性

仮定が妥当な時にだけ有効

e.g. 月27の中心に離れた場所の情報を入札に組み込む
必要がある場合、畳み込みに基づく
事前分岐は不適切

Attendants

9.5 畳込みのバリエーション

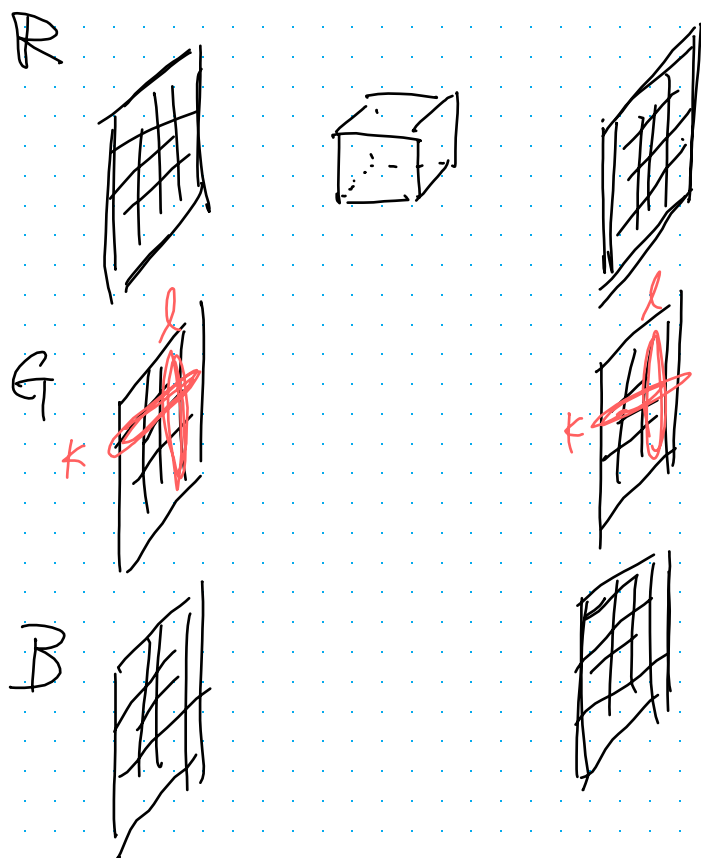
3-D Tensor $\left\{ \begin{array}{l} 1 \text{ channel (RGB!?)} \\ x \\ y \end{array} \right\}$ 空間的座標

4-D Tensor 上記 + Batch Index

4-D の Kernel Tensor $K_{i,j,k,l}$

$\rightarrow$ RGB 色 の channel
場所の index

出力 i
入力 j



入力は観測データ V
要素は $V_{i,j,k}$

$\downarrow$
出力も V と同じ形式 Z

$$Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m-1, k+n-1} K_{i-l, m, n}$$

ストライド幅のセル数が1より大きい畳み込みは
ストライド幅の畳み込みの後にダウンサンプリングを行うことと
ユニット 数学的に等価

Zero padding → カーネル幅と出力のサイズを独立に
制御できる

これがないと、ネットワークの空間的なサイズが
急速に縮小するのを防ぐため OR
小さなカーネルを利用するため

⇒ これはNNの表現力を制限する

ネットワークは特徴的なfeatureの獲得を犠牲に計算コスト削減

これは fullの Convolution + Downsampling
と等価

Locally connected layers (局所結合層)

↳ 非共有重みを持つ層

(unshared conv)

Tiled Convolution : 共有重みを持つ unshared の中継

CNN は 分類、回帰タスクに使える

segmentation などにも使われる

(出力 = リルは 各ピクセルが各クラスに属する確率)

$\hat{y}_{i,j,k}$

ピクセル j,k がクラス i