SAPT
第8回知能化分科会

一色 浩
（SAPT知能化分科会長）

2018.07.21

SAPT電子会議室C

目　　次

0.　自己符号化の実験

1. 深層学習ニューロのソフト開発について

2. 瀧 雅人著「**これならわかる深層学習**」

3. 飲み会の代わり…自由討論

自己符号化器の実験

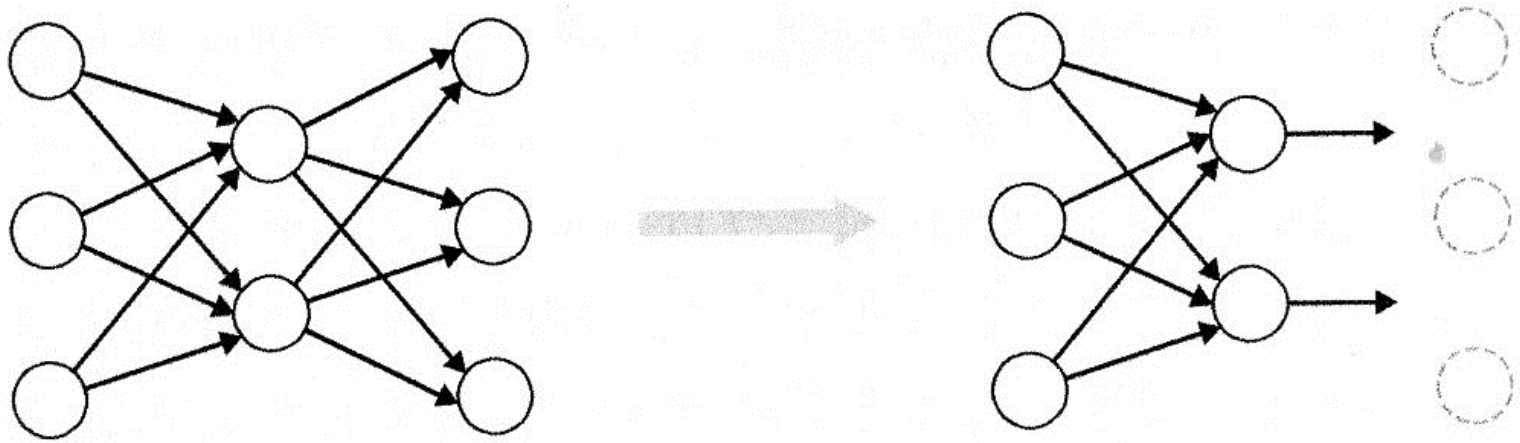


図7.2　砂時計型ニューラルネットによる自己符号化器の作成

出力層には入力データと同じ教師データを与える.

この砂時計型NNの学習は，入力$\mathbf{x}_n$が出力$\hat{\mathbf{x}}(\mathbf{x}_n)$ができる限り$\mathbf{x}_n$に近づくように行われる.

前半の変換操作$\mathbf{x} \to \mathbf{y}$を符号化と呼び，前半2層を符号化器と呼ぶ．符号化器の出力$\mathbf{y}$は符号と呼ばれる.

（1） 擬似1次元入力データ

見かけは3次元データだが，3次元空間に埋め込まれた1次元データ

$$\mathbf{e}_1 = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right)^T$$

$$\mathbf{e}_1^{\,2} = 1$$

$$\mathbf{x} = \xi_1 \mathbf{e}_1 \qquad \begin{cases} x_1 = \xi_1 / \sqrt{3} \\ x_2 = \xi_1 / \sqrt{3} \\ x_3 = \xi_1 / \sqrt{3} \end{cases}$$

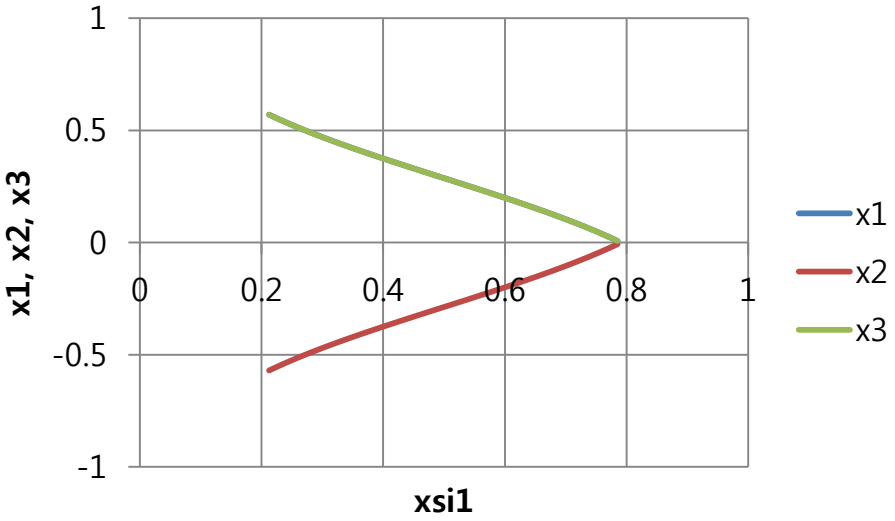
砂時計型ニューラルネットで得られた結果

OUの活性化関数は恒等関数．ただし重みは上下対称でない．

計算条件

項目	値
Lyr	3
IU	3
HU	1
OU	3
NT	40

項目	値
u0	0.5
Intwgt	0.3
Intoff	0.2
alpha	0.005
beta	0.00375
moment	0.1
dmomnt	0.002
erlimit	0.005
times	3500



中間層出力

$$x_1 \approx -(\xi_1 - 0.8),\ x_2 \approx (\xi_1 - 0.8),\ x_3 \approx -(\xi_1 - 0.8)$$

$$\mathbf{x} \approx \mathbf{e}_1'(\xi_1 - 0.8) \qquad \mathbf{e}_1' = (-1, 1, -1)^T$$

（2） 擬似2次元入力データ

見かけは3次元データだが，3次元空間に埋め込まれた2次元データ

$$\mathbf{e}_1 = \left(\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right)^T \qquad \mathbf{e}_2 = \left(\frac{-2}{\sqrt{6}}, \frac{-1}{\sqrt{6}}, \frac{1}{\sqrt{6}} \right)^T$$

$$\mathbf{e}_1{}^2 = 1 \quad \mathbf{e}_2{}^2 = 1 \quad \mathbf{e}_1 \cdot \mathbf{e}_2 = 0$$

$$\mathbf{x} = \xi_1 \mathbf{e}_1 + \xi_2 \mathbf{e}_2 \qquad \begin{cases} x_1 = \xi_1 / \sqrt{3} - 2\xi_2 / \sqrt{6} \\ x_2 = -\xi_1 / \sqrt{3} - \xi_2 / \sqrt{6} \\ x_3 = \xi_1 / \sqrt{3} + \xi_2 / \sqrt{6} \end{cases}$$

IU=3，HU=2，OU=3の砂時計NNで学習可能なので，

入力層の3次元データを中間層で2次元データにできる．

深層学習ニューロのソフト開発について

マイケル・ニールセン：ニューラルネットワークと深層学習（和訳）

http://nnadl-ja.github.io/nnadl_site_ja/index.html

Neural Network Libralies by Sony

https://nnabla.org/

みんなで情報を交換しましょう

瀧 雅人著「これならわかる深層学習」

8　畳み込みニューラルネット

この構造を開発するヒントになったのは，動物の視覚野の構造である．

8.1　一次視覚野と畳み込み

8.1.1　ヒューベルとウィーゼルの階層仮説

人間にはパターンを認識する高い能力が備わっている．

網膜で受け取られた視覚情報の電気信号は，画象としての平面構造を保ったまま，外側膝状体を経由して，1次視覚系に入力する．1次視覚系は後頭野にある．ここで，パターン認識プロセスの重要な第１段階が行われる．

1958年ヒューベルとウィーゼルは，猫の視覚野に特定の傾きをもつ線分を見せたときにだけ反応する細胞があることを発見した.

このような細胞は単純型細胞と複雑型細胞に大別され，これらの違いは図8.1の受容野の概念で説明される

図8.1(a)(b)の4つのケースそれぞれは，左の四角い領域（網膜）から視覚信号を受け取った際に，縦に並んだ4つのニューロンがどう変化するかを示す概念図である.

4つのうち上から2つ目のニューロンに注目する.

単純型細胞(a)では，パターンが灰色で塗りつぶされた領域にぴったり現れた時にのみニューロンが発火する．(a)の下の図では発火するが，上の図では発火しない．

複雑型細胞(b)はパターンがずれた時でも発火する．

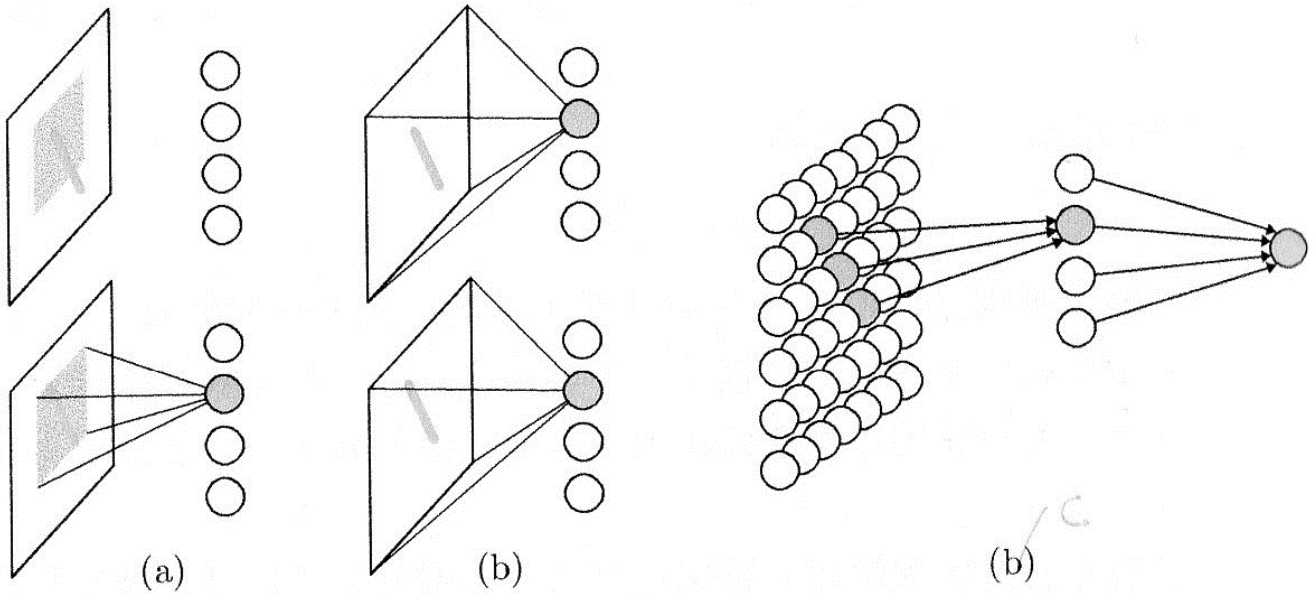


図8.1　(a) 単純型細胞、(b) 複雑型細胞．単純型細胞は、パターンの形状に対応した網膜型細胞とのみ結合を持つ．ヒューベルとウィーゼルの説では，複雑型細胞はそのような単純型細胞を束ねる形で実現している．

ヒューベルとウィーゼルの階層仮説によると，単純型細胞はパターン内の受容野とのみ結合している．刺激が受容野とぴったりあった時に最も大きく反応する．

（c）の3層目のニューロンは複雑型細胞で単純型細胞を束ねている．

したがって複雑型細胞は網膜上のより広い範囲にパターンがあれば反応する．

階層仮説が大きなヒントとなって生まれたのが畳み込みニューラルネットワーク（convolutional network CNN）である．

CNNの起源は福島邦彦のネオコグニトロン（1979）に遡る．1989年にカルカンのグループが同じネットワークに誤差逆伝播法を導入して高い性能を実現した（LeNet）．

8.1.2 ニューラルネットと畳み込み

畳み込みニューラルネットは2種類の層からなる.

1つ目の層は単純型細胞からなり図8.2(a)のように全体を覆う. (a)の右図のように次層と結合されている. どこにパターンが現れても次層のユニットが活性化する.

1層と2層の間はスパースな結合である.

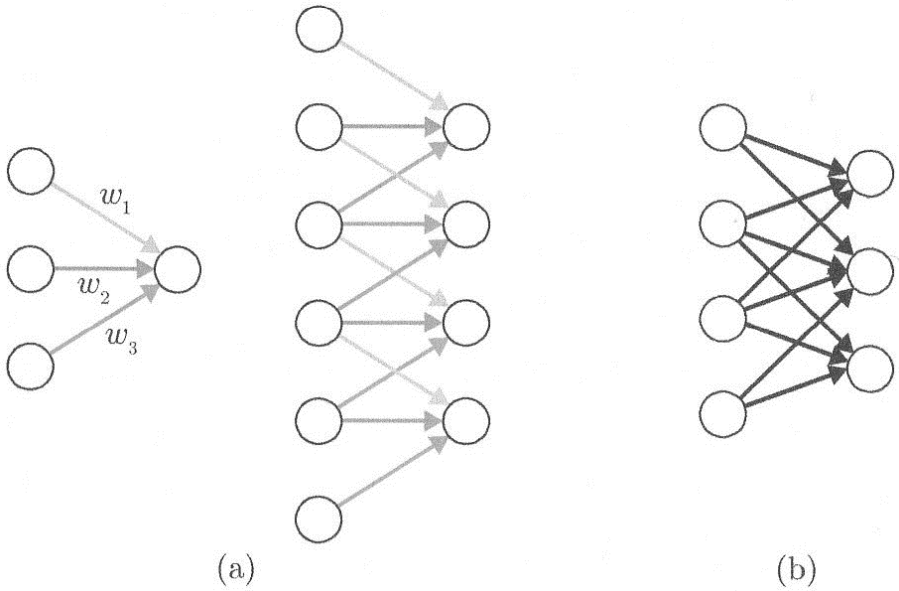


図8.2 (a) 左は3つのユニットからなる局所受容野から入力を受ける単純型細胞、(b) 単純型細胞の層から入力を受ける複雑型細胞の層.

パターンの現れる位置で認識能力が変わらないためには，結合がすべて同じ重みを持っていればよい.

結合の多さにもかかわらずパラメータ数は少ない.

畳み込みそうには重みのスパース化と共有という2つの正規化が課せられていると看做せる.

8.2 ニューラルネット畳み込み

8.2.1 画像データとチャンネル

MINISTなどの単純な手書き文字認識では，2次元の画像データを1次元状に並べて入力しても，それなりに扱える．自然画像分類などの本格的なパターン認識では，2次元構造を最大限に活用すべきである．

2次元状に配列した実数値画素値をx_{ij} として，サイズを$W \times W$とする．添字範囲を以下のように取る：

$$i = 0,1,\ldots,W-1, \qquad j = 0,1,\ldots,W-1 \tag{8.1}$$

実際の画像データは1つの位置(i, j) に対して，色成分などの多くの情報が付与されている．

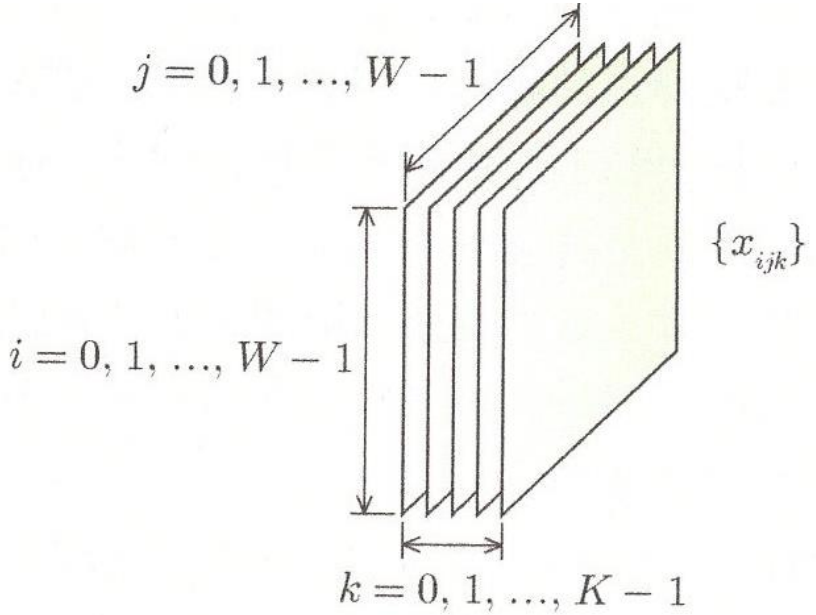


図8.3　*K*チャンネルからなる*W*×*W*画像.

そのような自由度をチャンネルという. RGBカラーを用いた画像では1つの位置(i, j)に，赤，緑，青の成分量に応じて3種類の数値が付随する.

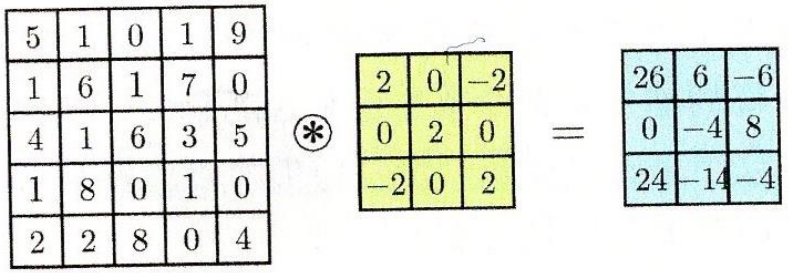
8.2.2　畳み込み層

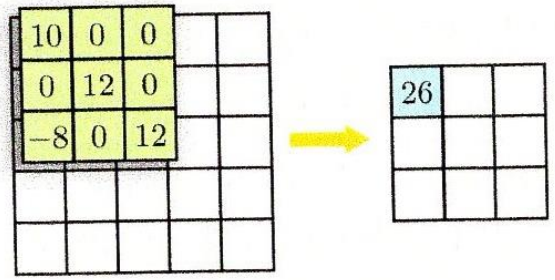
1チャンネルの簡単な場合から議論する. フィルタは入力よりも小さなサイズの*H*×*H*画像とする. その画素値をh_{pq}とする.

$(z_{ij}^{(l-1)})$ に対するフィルタ (h_{pq}) の畳み込み $\otimes$ とは

$$u_{ij}^{(l)} = \left[(z^{(l-1)}) \otimes (h) \right]_{ij} = \sum_{p,q=0}^{H-1} z_{i+p,j+q}^{(l-1)} h_{pq} \qquad (8.3)$$

畳み込み後画像のサイズは（W-H+1）×（ W-H+1 ）.





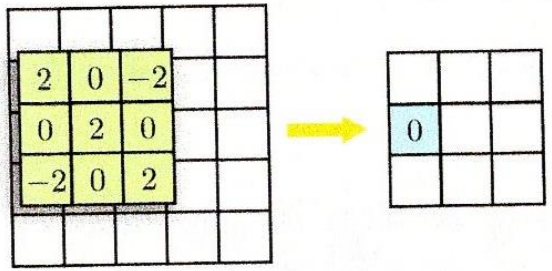


図8.4　5×5画像への3×3フィルタの畳みI込み. 例えば畳み込み後の画像値26は,
5×2＋1×0+0×(-2)＋
1×0＋6×2＋1×0＋
4×(-2)＋1×0＋6×2
＝26と
いう計算にできます.

重み共有までは必要でない場合は

$$u_{ij}^{(l)} = \sum_{p,q=0}^{H-1} z_{i+p,j+q}^{(l-1)} h_{ijpq} \qquad (8.4)$$

フィルタは図8.4のように作用し，画像からあるパターンを抽出する.

図8.4の緑のフィルタでは対角線に3つの大きな数値が並んでいる.
これを画像に畳み込むことで，フィルタ同様に斜めに大きな数が3つ並んでいる領域を抽出できる.
青の画像には26と24という飛び抜けて大きな画素がある.

26の例では元画像に5,6,6という大きな数値の並びがある.

畳み込みとプーリングをそれぞれ2回行うCNNを学習させて得られたフィルタが図8.5に例示してある.

きれいなパターンは認めがたいが，斜めの線やスポット構造などを取り出しているように見える．意味不明？

図8.5　1層目の5×5フィルタをいくつか画像下したもの.

フィルタを畳み込むことで，画像から特定のパターンを抽出できる．フィルタは我々が与えるのではなく，学習で得られる.

フィルタh_{pq}が重みパラメータに他ならない.

つぎにKチャンネル画像について考える.

Kチャンネル画像は $z_{ijk}^{(l-1)}$ のように3つの添字でラベルされるので，$W \times W \times K$画像と看做せる.

そこで$H \times H \times K$ フィルタh_{pqk} を用意する.

各チャンネルごとに畳み込みを行い，同じ位置の画素値の総和を取る. 畳み込み後の画像の画素数は（W-H+1）×（W-H+1）である.

$$u_{ij}^{(l)} = \sum_{k=0}^{K-1} \sum_{p,q=0}^{H-1} z_{i+p,\, j+q,k}^{(l-1)} h_{pqk} + b_{ij} \qquad (8.5)$$

この式にはバイアス b_{ij} も含まれている.

畳み込み後も多チャンネルが欲しいならば. M種類のフィルタh_{pqkm}（m=0,1,…, M）を用意して

$$u_{ijm}^{(l)} = \sum_{k=0}^{K-1} \sum_{p,q=0}^{H-1} z_{i+p,j+q,k}^{(l-1)} h_{pqkm} + b_{ijm} \qquad (8.6)$$

畳み込み層の最終的な出力は，この画像に活性化関数を作用させて

$$z_{ijm}^{(l)} = f(u_{ijm}^{(l)}) \qquad (8.7)$$

畳み込み層からの出力を特徴マップともいう．活性化関数としては，最近ではReLU関数がよく用いられる．

8.2.3 1×1畳み込み

最近のCNNでは，サイズ1×1のフィルタによる畳み込みが用いられる．チャンネル方向に画素値を加算するので無意味ではない．チャンネル数の削減を行っている．

8.2.4 因子化した畳み込み

畳み込みにまつわる計算量削減法を紹介する.

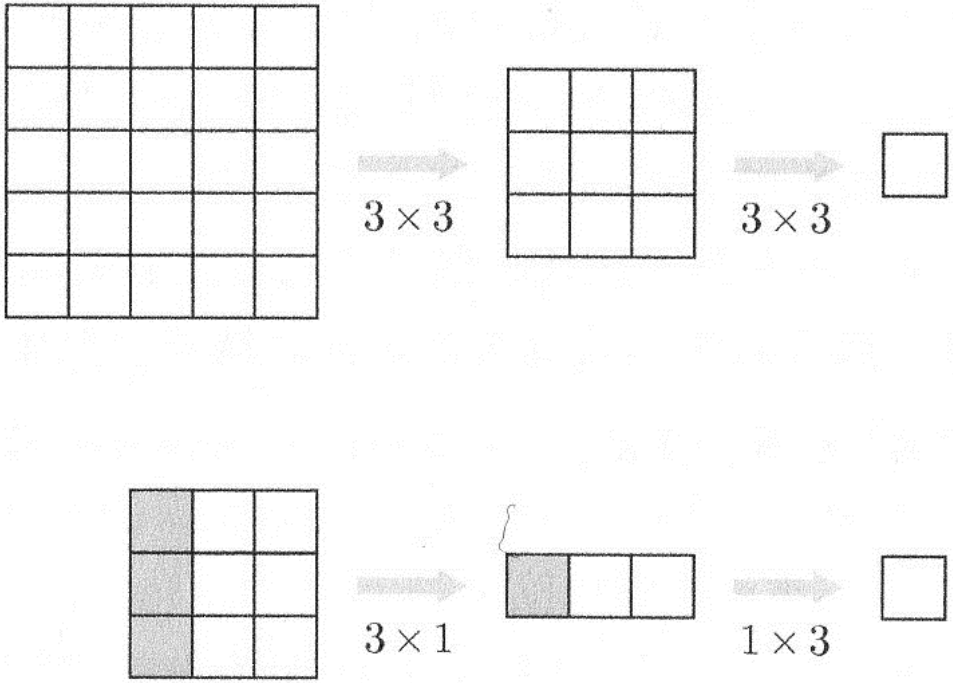


図8.6 畳み込みの因子化. 上図では, 5×5の領域が3×3フィルタの畳み込み2回で1×1へ縮小している. したがってこれは5×5畳み込み1回とみなせる. 下側では3×1の後に1×3を行い, 3×3フィルタの畳み込み1回と同様の作用を実現. はじめの3つのピクセルが, 3×1の作用によって1個の赤いピクセルに変換している.

図8.6上は5×5畳み込みは3×3畳み込み2回で代用できる.

5×5畳み込み演算を2回の3×3畳み込み演算で置き換えると，フィルタのパラメータ数が（3×3）×2／（5×5）＝0.72まで減る.

このようなものを因子化された畳み込みと呼ぶ.

3×3畳み込み演算を2回の2×2畳み込み演算で置き換えると，フィルタのパラメータ数が（2×2）×2／（3×3）＝0.89まで減る.

図8.6下を見ると，非対称な畳み込み3×1と1×3を用いている．フィルタのパラメータ数が（3×1）×2／（3×3）＝0.67まで減る.

フィルタサイズ$H \times H$がおおきくなると$H \times 1$と$1 \times H$へ因子化させると，コスト削減効果が大きくなる.

因子化の方法は多くのモデルで採用されている.

8.2.5 ストライド

これまでフィルタを1メモリずつ動かしてきたが，大雑把に画像の構造を捉えたい場合には，数メモリー度に動かすストライドという方法が取られる.

フィルタがS画素動くごとに1回の畳み込みを行うストライドSでは

$$u_{ijm}^{(l)} = \sum_{k=0}^{K-1} \sum_{p,q=0}^{H-1} z_{Si+p,Sj+q,k}^{(l-1)} h_{pqkm} + b_{ijm} \qquad (8.8)$$

畳み込み後の画像サイズは

$$\left(\left[\frac{W-H}{S} \right] + 1 \right) \times \left(\left[\frac{W-H}{S} \right] + 1 \right) \qquad (8.9)$$

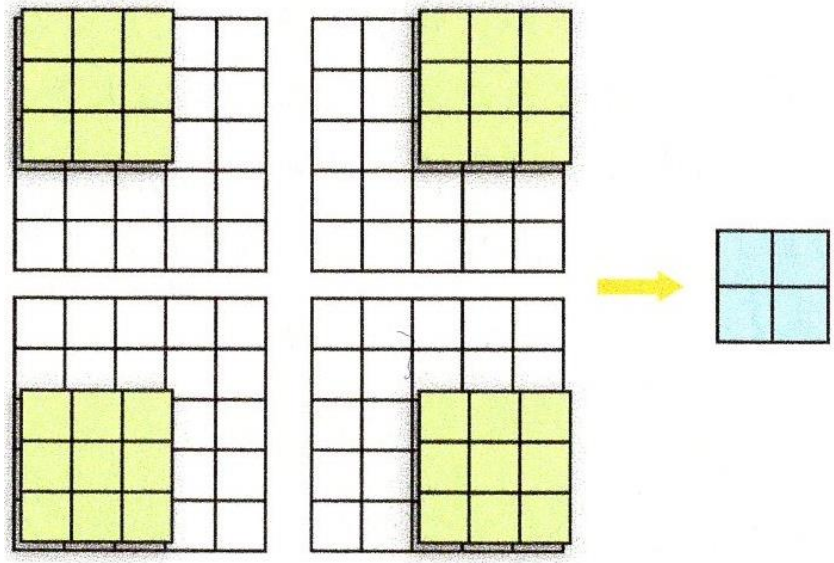


図8.7　5×5画像へのストライド2での3×3フィルタの畳み込み.

8.2.6　パディング

畳み込みをすると画像はサイズが小さくなる. ストライドが大きいほど顕著である.

畳み込み後のサイズをある程度大きくできるパディングという手法がある.

パディングPとは元の画像の周りに幅Pの溝を加える操作である．畳み込み後の画像サイズは

$$\left(\left[\frac{W-H+2P}{S}\right]+1\right)\times\left(\left[\frac{W-H+2P}{S}\right]+1\right) \quad (8.9)$$

パディングで加えた画像にどのような値をどのように入れるかには複数の考え方があります．

ゼロパディングでは，0を入れます．

ストライドが1のゼロパディングに3つが多用される．

validパディング：一切パディングせず．画像は縮小．

sameパディング：画像サイズが畳み込みの前後で不変

fullパディング：幅がW+H-1となるよう最大限パディング

パディング域に0以外の値をとっても良い.

図8.8では，画像が左右上下に周期的に続いていると考えて元画像と同じ画素値をパディングに用いている.

境界での画素値をそのまま外部に拡張する方法もある.

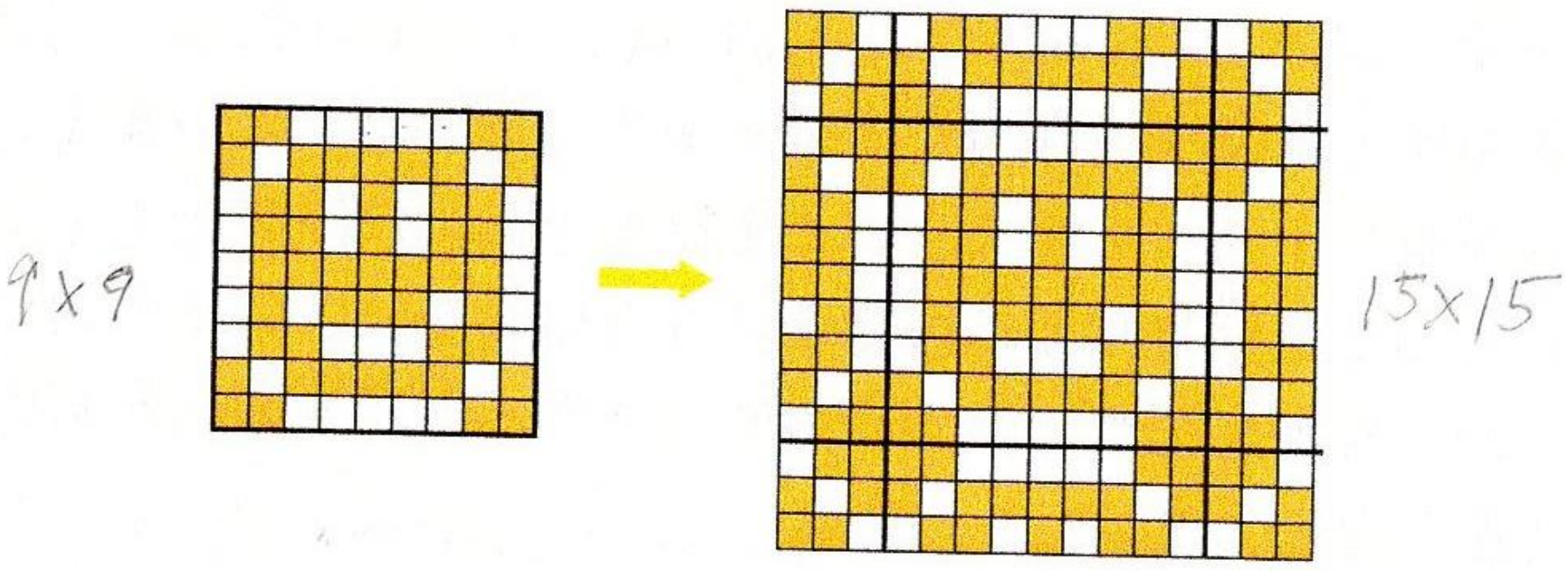


図8.8　9×9画像へのパディング3.

8.2.7 プーリング層

複雑型細胞に対応するプーリング層（pooling layer）の説明をする.

プーリング層の役割は単純型細胞の出力を合算して入力位置の平行移動に対してロバストにすること.

$W \times W \times K$入力画像 $z_{i,j,k}^{(l-1)}$ の各チャンネルに対し，各画素位置(i,j)を中心とした$H \times H$の領域 P_{ij} を考える.

ただし画像はパディングしておき，どの点に対しても P_{ij} の領域を取れるようにしておく.

P_{ij} 中の画素から，その領域での代表的な値 $u_{i,j}^{(l)}$ を決定する.

最大プーリングでは，領域中の最大の画素値を代表値とする.

$$u_{ijk}^{(l)} = \max_{(p,q)\in P_{ij}} z_{pqk}^{(l-1)} \qquad (8.11)$$

最大プーリング意外にも，平均プーリング：

$$u_{ijk}^{(l)} = \frac{1}{H^2} \sum_{(p,q)\in P_{ij}} z_{pqk}^{(l-1)} \qquad (8.12)$$

この2つのプーリングを拡張した方法として，L^P プーリング：

$$u_{ijk}^{(l)} = \left(\frac{1}{H^2} \sum_{(p,q)\in P_{ij}} \left(z_{pqk}^{(l-1)} \right)^P \right)^{\frac{1}{P}} \qquad (8.13)$$

P=1の場合は明らかに平均プーリングになるが，対極の$P\rightarrow\infty$では，最大プーリングになる．（証明省略）

確率的プーリングというのもある．

プーリング層の役割は畳み込み層の出力を束ねるだけだから重み学習はしない．学習中もプーリングの式は変更されない．

8.2.8　局所コントラスト正規化層

最近はあまり使われないとのことであるので，省略．

8.2.9　局所応答正規化層

実際に計算を実行する段階でないと理解不能．省略．

8.2.10　ネットワーク構造

典型的なCNNの例を図8.9に示す（オックスフォード）．

8.3 CNNの誤差逆伝播層

CNNはネットワーク構造が複雑なため，学習などが難しく見えるが，本質的な違いはなく，誤差逆伝播法を用いる学習が可能である.

8.3.1 畳み込み層

畳み込み層(8.6)，(8.7)の構造は，重み共有された順伝播層に過ぎない. したがって，l-1層とl層の間の重みの勾配は

$$\frac{\partial E}{\partial h_{pqkm}} = \sum_{i,j} \frac{\partial E}{\partial u_{ijm}^{(l)}} \frac{\partial u_{ijm}^{(l)}}{\partial h_{pqkm}} = \sum_{i,j} \delta_{ijm}^{(l)} z_{Si+p,Sj+q,k} \qquad (8.15)$$

最後の行で(8.8)を用いた. デルタの逆伝播式は

$$\delta_{abk}^{(l-1)} \equiv \frac{\partial E}{\partial u_{abk}^{(l-1)}} = \sum_{i,j,m} \frac{\partial E}{\partial u_{ijm}^{(l)}} \frac{\partial u_{ijm}^{(l)}}{\partial z_{abk}^{(l-1)}} f'(u_{abk}^{(l-1)}) \qquad (8.16)$$

と得られる．式(8.8)を使うと

$$\frac{\partial u_{ijm}^{(l)}}{\partial z_{abk}^{(l-1)}} = \sum_{p=a-Si} \sum_{q=b-Sj} h_{pqkm}$$

であるので

$$\delta_{abk}^{(l-1)} = \sum_{m} \sum_{i,j} \sum_{\substack{p \\ p=a-Si}} \sum_{\substack{q \\ q=b-Sj}} \delta_{ijm}^{(l)} h_{pqkm} f'\left(u_{abk}^{(l-1)}\right) \qquad (8.17)$$

となる．

ベクトル表記による書き直しは，議論の本質とは関係ないので省略する．

8.3.2 プーリング層

プーリング層も通常の順伝播層に書き換えることで逆伝播できる.

平均プーリング層は

$$w_{JI} = \begin{cases} \frac{1}{H^2} & I \in R_J \\ 0 & \text{otherwise} \end{cases} \qquad (8.21)$$

という重みの順伝播層. 重みは学習中も変わらない.

最大プーリング層では, 順伝播時の最大画素位置を用いて, 逆伝播時には

$$w_{JI} = \begin{cases} 1 & I = \text{argmax}_{I'} z_{I'}^{(l-1)} \\ 0 & \text{otherwise} \end{cases} \qquad (8.22)$$

という重みの順伝播層とみなせば, 通常の逆伝播可.

8.4 学習済みモデルと転移学習

ImageNetなどの大きなデータで学習したモデルは，他の機械学習とは比較にならないほどの高い画像分類性能を達成できる.

しかし多層のCNNを用いた自然画像の学習には多くのコストと計算時間を要する.

もしも，学習済のモデルがImageNetデータの分類以外には全く役立たないならば，コストが高すぎて，魅力が半減する.

しかし，ニューラルネットには高い汎化性能があり，深いネットワーク構造では，中間層にさまざまな特徴量を獲得していると期待される.

そこで，学習済のモデルの出力側の層を取り除いて，中間層からの出力を入力画像の表現として用いることが考えられる.

出力に近い側は出力層に設定した学習タスクに強く影響を受けていて，他のタスクへの応用に向かない.

出力層だけを置き換えて他のタスクのために学習し直す方法も用いられる.

学習済みパラメータが初期値となって，学習がスムースに進行する.

この方法を転移学習という.

8.5 CNNはどのようにパターンを捉えているのか

CNNの高い学習能力の詳細は多く分かっていない．しかし，CNNの中を覗き見る技術はさまざま開発されている．

図8.10の(1)，(2)，(3)はそれぞれVGG16のある特定の画像で，(1)から(3)の順に出力寄りから入力よりの順に並んでいる．

これを見ると，低層から高次の層に行くに連れ，各フィルタがより複雑なパターンを捉えている．ただ，高次の層でのパターンを直感的に理解できるものでない．

図8.10の(a)から(f)は，出力層から選んだ6個のユニットに対応した画像である.

(a)三葉虫，(b)シベリアンハスキー，（c）シマウマ，（d）タクシー，(e)図書館，（f）エスプレッソである.

ランダムノイズを初期値として作成した入力画像である．99.9%という高い確率で，各カテゴリに分類される.

人間がこの画像を見ても，全く認識できない.

ノイズに対するこの謎めいた挙動はadversarial examplesという現象と類似のもので，現在でも活発に研究されている.

8.6 脱畳み込みネットワーク

初学者には不要と思われるので，省略.

8.7 インセプションモジュール

初学者には不要と思われるので，省略.

飲み会の代わり…自由討論

分科会運営方法に関するコメント

最近の世相，時代の潮流

年寄りは無条件にモノづくりは良いもの、楽しいものと信じている？若者の考え方は？

本田宗一郎や松下幸之助の生き方に共感？

夢：遠隔地音楽アンサンブル

時間遅れをどう克服するか？

音楽合奏に関する鍋島さんの経験

ユニソンからアンサンブルへ

ノイズキャンセラーが悪さをする？

A, Bの2信号があると、一方が信号，他方がノイズになり、抑えられる。

夢：遠隔地音楽アンサンブル

ヤマハNETDUETTO（宮竹さんからの情報）

現行の通信回線でよい.

各パーツの時間遅れを極力小さくする.

ダウンロードの仕方
NETDUETTOラボ http://netduetto.net/manual/

７月４日（水）に前田，澤井，鍋島の諸氏と一色で WiFi ではなくて，有線接続で実験しました.

音の強弱，テンポが安定して，ようやく合奏出来る可能性が出てきました. 遅れ時間80ms.

新井先生から，オーディオ･インターフェースの仕組みのご報告をいただけるのを期待しています.

７月１１日（水)にもNETDUETTOによる合奏の練習をしました．その結果，**オーディオ･インターフェース**が時間遅れを解消するためにはどうしても必要だという気がしてきました．

その理由は，飛び込みの人の遅延時間が↑が35ms，↓が75ms程度だったということです．我々の遅延時間は上下ともに80ms程度です．

オーディオ･インターフェースに関する鍋島さんの解釈

一種のADコンバータで，パソコンのCPUを使わずに，アナログのオーディオ信号をデジタル信号にするので，CPUは、通信にかかりっきりになれる．

安いオーディオ･インターフェースでは改善しない

Web上の記事を見ると1万円クラスのものでないとダメ.

鍋島さんの実験（2台のPCのNet環境が同じ）

1階のデスクトップPCと2階のノートPCでNETDUETTOを立ち上げたところ，遅れが両者に僅かな差があるが，いままでの結果とほぼ同じ．タスクマネージャーでCPUの負荷をチェックしたが、負荷は小さい.

PCのAD，DAが常時働いていると仮定すると，オーディオ･インターフェースの早いAD、DAに置き換わると遅延が小さくなり得る.