

知能プログラミング演習 I

第 6 回: 畳み込みニューラルネットワーク I

梅津 佑太

2 号館 404A: umezu.yuta@nitech.ac.jp

前回作ったディレクトリに移動して今日の課題のダウンロードと解凍

step1: `cd ./DLL`

step2: `wget http://www-als.ics.nitech.ac.jp/~umezu/DLL19/Lec6.zip`

step3: `unzip Lec6.zip`

✓ まだ DLL のフォルダを作っていない人は, step1 の前に

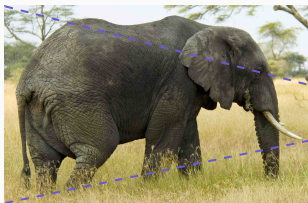
`mkdir -p DLL`

でフォルダを作成する

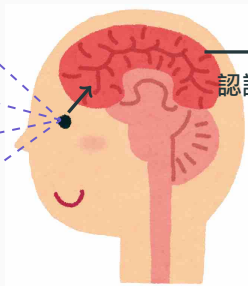
1. 畳み込みニューラルネットワークの順伝播

画像 (物体) 認識における脳活動のモデル

- 物体の視覚情報 → 特徴抽出&脳内処理 → 物体の認識



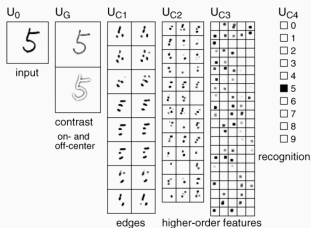
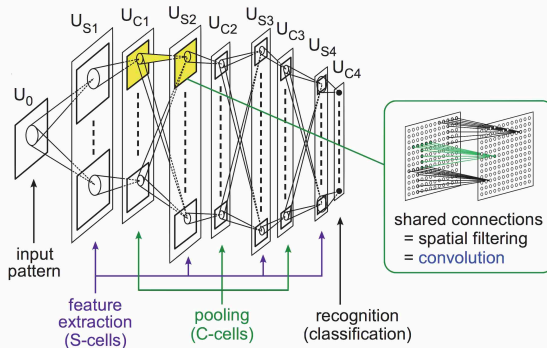
視覚情報の取得



象!

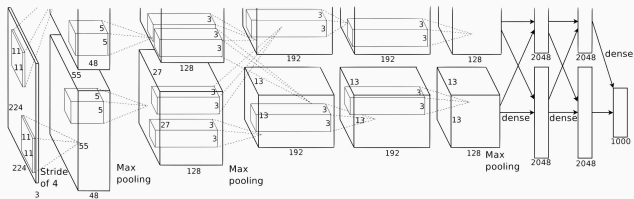
認識

ネオコグニトロン

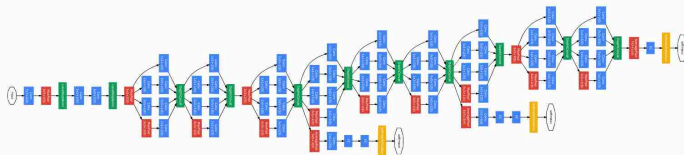


- 人間の脳モデル
- 初期の畳み込みニューラルネットワーク

畳み込みニューラルネットワーク



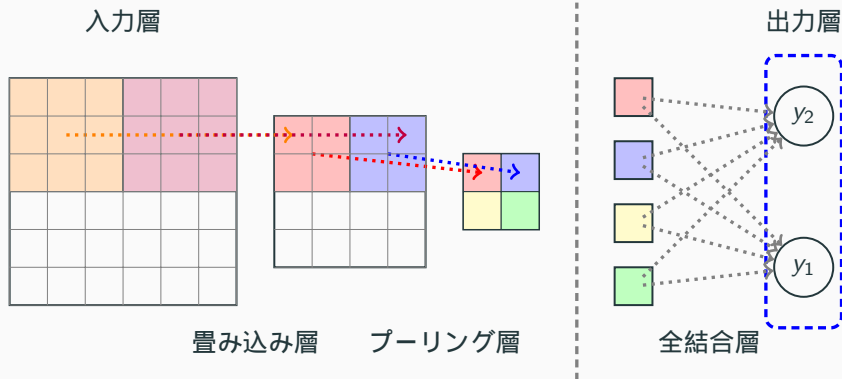
(a) AlexNet



(b) GoogLeNet

- 畳み込みとプーリングの構造を持つネットワーク
- 誤差逆伝播法でパラメータを更新

ネットワークの基本構造

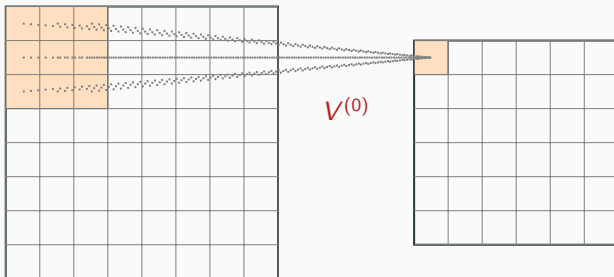


- 畳み込みを何回か繰り返した後、プーリングを行うこともある
- 全結合層は単純な (深層) ニューラルネットワーク
- 実際には、プーリング後に正規化層をはさんだり、パディングやストライドを利用することが多いが、基本は図の通り

畳み込み層のイメージ

パラメータ $V^{(0)} \in \mathbb{R}^{H \times H}$ を用いて、画像 $X \in \mathbb{R}^{W \times W}$ をのフィルタ $Z^{(1)} \in \mathbb{R}^{(W-H+1) \times (W-H+1)}$ に圧縮¹: 活性化関数を f として,

$$z_{ij}^{(1)} = f \left(\sum_{p=1}^H \sum_{q=1}^H v_{pq}^{(0)} x_{i+p-1, j+q-1} + v_0^{(0)} \right)$$

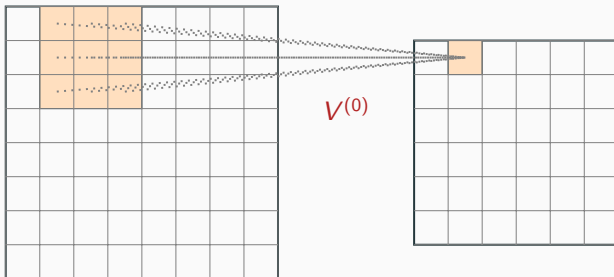


¹ 数学的な意味での畳み込みとは正確には異なるが、画像処理におけるフィルタリングに対応している

畳み込み層のイメージ

パラメータ $V^{(0)} \in \mathbb{R}^{H \times H}$ を用いて, 画像 $X \in \mathbb{R}^{W \times W}$ をのフィルタ $Z^{(1)} \in \mathbb{R}^{(W-H+1) \times (W-H+1)}$ に圧縮¹: 活性化関数を f として,

$$z_{ij}^{(1)} = f\left(\sum_{p=1}^H \sum_{q=1}^H v_{pq}^{(0)} x_{i+p-1, j+q-1} + v_0^{(0)}\right)$$

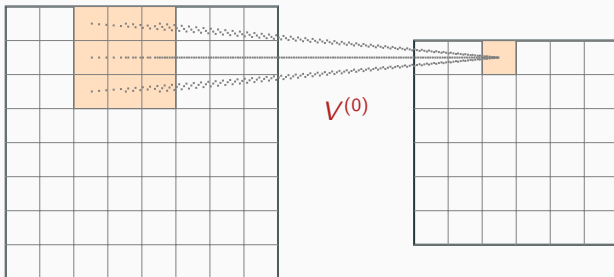


¹ 数学的な意味での畳み込みとは正確には異なるが, 画像処理におけるフィルタリングに対応している

畳み込み層のイメージ

パラメータ $V^{(0)} \in \mathbb{R}^{H \times H}$ を用いて、画像 $X \in \mathbb{R}^{W \times W}$ をのフィルタ $Z^{(1)} \in \mathbb{R}^{(W-H+1) \times (W-H+1)}$ に圧縮¹: 活性化関数を f として、

$$z_{ij}^{(1)} = f\left(\sum_{p=1}^H \sum_{q=1}^H v_{pq}^{(0)} x_{i+p-1, j+q-1} + v_0^{(0)}\right)$$

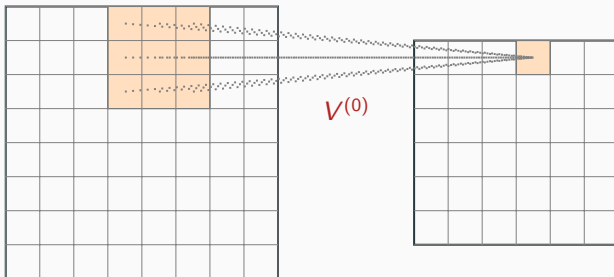


¹ 数学的な意味での畳み込みとは正確には異なるが、画像処理におけるフィルタリングに対応している

畳み込み層のイメージ

パラメータ $V^{(0)} \in \mathbb{R}^{H \times H}$ を用いて、画像 $X \in \mathbb{R}^{W \times W}$ をのフィルタ $Z^{(1)} \in \mathbb{R}^{(W-H+1) \times (W-H+1)}$ に圧縮¹: 活性化関数を f として,

$$z_{ij}^{(1)} = f \left(\sum_{p=1}^H \sum_{q=1}^H v_{pq}^{(0)} x_{i+p-1, j+q-1} + v_0^{(0)} \right)$$

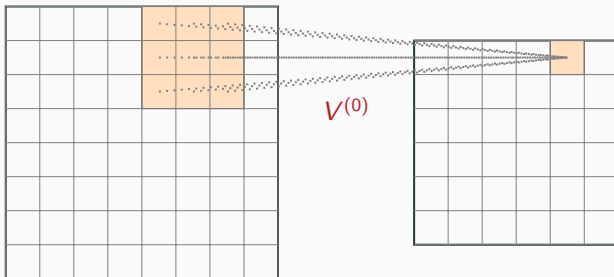


¹ 数学的な意味での畳み込みとは正確には異なるが、画像処理におけるフィルタリングに対応している

畳み込み層のイメージ

パラメータ $V^{(0)} \in \mathbb{R}^{H \times H}$ を用いて、画像 $X \in \mathbb{R}^{W \times W}$ をのフィルタ $Z^{(1)} \in \mathbb{R}^{(W-H+1) \times (W-H+1)}$ に圧縮¹: 活性化関数を f として、

$$z_{ij}^{(1)} = f \left(\sum_{p=1}^H \sum_{q=1}^H v_{pq}^{(0)} x_{i+p-1, j+q-1} + v_0^{(0)} \right)$$

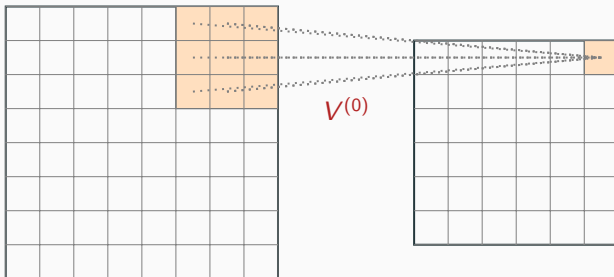


¹ 数学的な意味での畳み込みとは正確には異なるが、画像処理におけるフィルタリングに対応している

畳み込み層のイメージ

パラメータ $V^{(0)} \in \mathbb{R}^{H \times H}$ を用いて、画像 $X \in \mathbb{R}^{W \times W}$ をのフィルタ $Z^{(1)} \in \mathbb{R}^{(W-H+1) \times (W-H+1)}$ に圧縮¹: 活性化関数を f として、

$$z_{ij}^{(1)} = f\left(\sum_{p=1}^H \sum_{q=1}^H v_{pq}^{(0)} x_{i+p-1, j+q-1} + v_0^{(0)}\right)$$



¹ 数学的な意味での畳み込みとは正確には異なるが、画像処理におけるフィルタリングに対応している

パディング

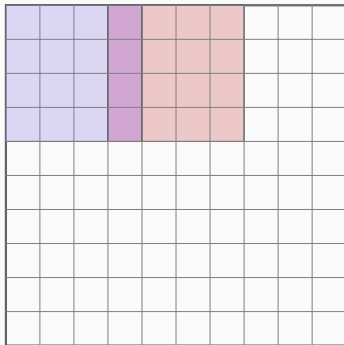
- 通常の畳み込みでは, 画像の端の情報が薄れてしまう
- 画像 X のフチにゼロを加える (ゼロパディング)
- パディング数 1 で画像を変換すると図のようになる

0	0	0	0	0	0	0	0	0	0
0									0
0									0
0									0
0									0
0									0
0									0
0									0
0									0
0	0	0	0	0	0	0	0	0	0

パディング数を α とすると, 元の画像 $X \in \mathbb{R}^{W \times W}$ は
 $(W + 2\alpha) \times (W + 2\alpha)$ の画像になる

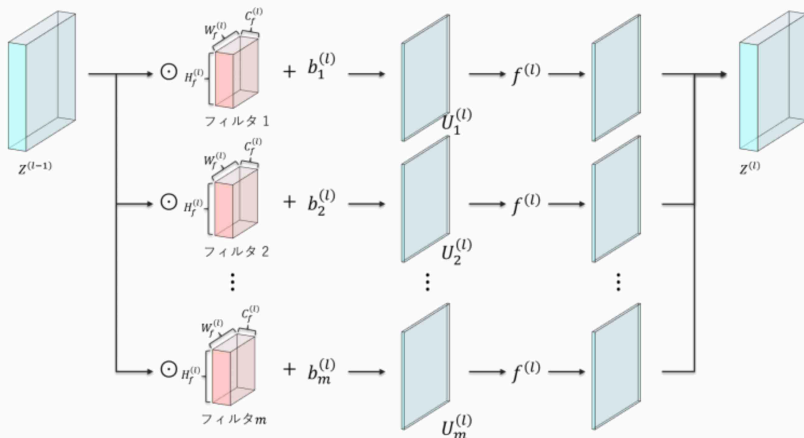
ストライド

- 1画素づつフィルタをずらさず, 数画素づつフィルタを移動させる
ことがあり, ストライドと呼ばれる
- ストライド 3 で 4×4 のフィルタを畳み込むと図のようになる



ストライド数を β とし $H \times H$ のフィルタで画像 $X \in \mathbb{R}^{W \times W}$ を畳み込むと, 畳み込み後の画像は $(W - H)/\beta \times (W - H)/\beta$ に圧縮される

畳み込み層



- 画像 $Z \in \mathbb{R}^{W \times W \times K}$ に対して, $V \in \mathbb{R}^{H \times H \times K \times m}$ 個の重みパラメータと m 個のバイアスパラメータ $b \in \mathbb{R}^m$ を用いて畳み込まれる
 ✓ $V_j \in \mathbb{R}^{H \times H \times K}$, $v_j \in \mathbb{R}$ が各チャンネルごとのパラメータとなる.

畳み込み層: チャンネルごとの出力

パディング数 α でパディング後の入力を $Z \in \mathbb{R}^{W \times W \times K}$, $V \in \mathbb{R}^{H \times H \times K}$, $v_0 \in \mathbb{R}$ をパラメータ, β_c をストライド数とする.

$$u_{ij} = \sum_{k=1}^K \sum_{p=1}^H \sum_{q=1}^H V_{pqk} Z_{\beta_c(i-1)+p, \beta_c(j-1)+q-1, k} + v_0$$

によって得られる $U = (u_{ij})$ に対して, 畳み込み層の出力は, 活性化関数 f を用いて

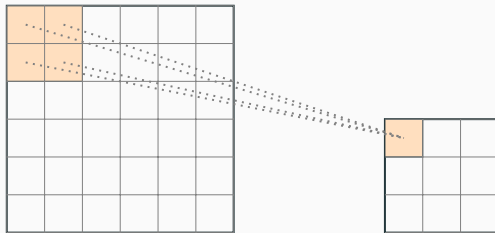
$$Z^{(1)} = f(U) = f(u_{ij})$$

で計算する².

²実際には, この計算はチャンネル数の数だけ行われる.

畳み込み層で得られた $Z^{(1)}$ をさらに圧縮³: フィルタサイズを s , ストライド数を β_p として,

$$z_{ij}^{(2)} = g(z_{\beta_p(i-1)+p, \beta_p(j-1)+q}^{(1)}; p, q = 1, \dots, s)$$



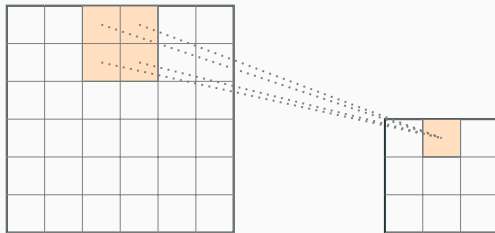
- 畳み込み層と同様に 1 ピクセルずつ動かすこともできるが、画像領域が重複しないようにプーリングする事が多い

³画像処理における縮小処理に対応している

プーリング層

畳み込み層で得られた $Z^{(1)}$ をさらに圧縮³: フィルタサイズを s , ストライド数を β_p として,

$$z_{ij}^{(2)} = g(z_{\beta_p(i-1)+p, \beta_p(j-1)+q}^{(1)}; p, q = 1, \dots, s)$$



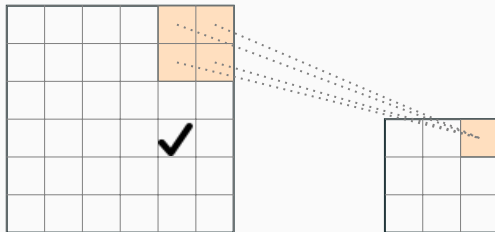
- 畳み込み層と同様に 1 ピクセルずつ動かすこともできるが、画像領域が重複しないようにプーリングする事が多い

³画像処理における縮小処理に対応している

プーリング層

畳み込み層で得られた $Z^{(1)}$ をさらに圧縮³: フィルタサイズを s , ストライド数を β_p として,

$$z_{ij}^{(2)} = g(z_{\beta_p(i-1)+p, \beta_p(j-1)+q}^{(1)}; p, q = 1, \dots, s)$$



- 畳み込み層と同様に 1 ピクセルずつ動かすこともできるが、画像領域が重複しないようにプーリングする事が多い

³画像処理における縮小処理に対応している

いろいろなプーリング

$P_{ij} = \{(s(i-1) + p, s(j-1) + q) \mid p, q = 1, 2, \dots, s\}$ としたとき,

- max pooling

$$z_{ij}^{(2)} = \max_{(k,l) \in P_{ij}} z_{kl}^{(1)}$$

- average pooling

$$z_{ij}^{(2)} = \frac{1}{s^2} \sum_{(k,l) \in P_{ij}} z_{kl}^{(1)}$$

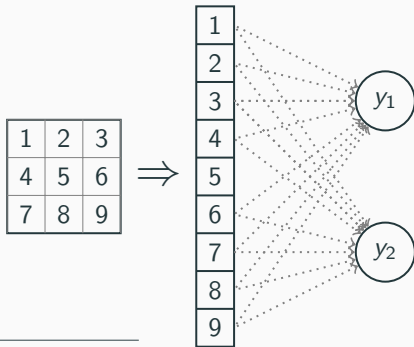
- L_p -pooling

$$z_{ij}^{(2)} = \left(\frac{1}{s^2} \sum_{(k,l) \in P_{ij}} \left(z_{kl}^{(1)} \right)^p \right)^{1/p}$$

全結合層

プーリング後のデータを $Z^{(2)} \in \mathbb{R}^{s' \times s'}$ として, (深層) ニューラルネットワークを適用: 出力層の数を m とすれば, 活性化関数を h として⁴

$$y_j = h\left(\sum_{p=1}^{s'} \sum_{q=1}^{s'} v_{pqj}^{(2)} z_{pq}^{(2)} + v_0^{(2)}\right) \Leftrightarrow \mathbf{y} = h\left(\sum_{p=1}^{s'} \sum_{q=1}^{s'} \mathbf{v}_{pq}^{(2)} z_{pq}^{(2)} + v_0^{(2)} \mathbf{1}_m\right)$$



⁴分類の場合は通常 softmax 関数を用いる