第 4 回　知能システム学特論レポート

15344203　有田 裕太
15344206　緒形 裕太
15344209　株丹 亮
12104125　宮本 和

西田研究室，計算力学研究室

2015 年 6 月 29 日

進捗状況

理論研究の進捗

人工ニューラルネットワークの理論について

プログラミングの進捗

なし

ユニットの出力

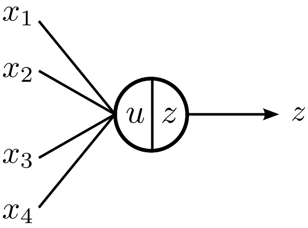

Figure: ユニット 1 つの入出力の例

- 各ユニットは複数の入力を受けて 1 つの出力を計算する．
- 各入力にそれぞれ異なる重み $w1$，$w2$，$w3$，$w4$ を入力 $x1$，$x2$，$x3$，$x4$ に掛けたものの和にバイアス値 b を加える．

ユニットの出力

総入力 u とユニットの出力 z

$$u = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \tag{1}$$

$$z = f(u) \tag{2}$$

これを一般化し
第 1 層のユニットを $i = 1, \cdots, I$，第 2 層のユニットを $j = 1, \cdots, J$ で表すと

入出力式の一般化

$$u_j = \sum_{i=1}^{I} w_{ji}x_i + b_j \tag{3}$$

$$z_j = f(u_j) \tag{4}$$

活性化 (シグモイド) 関数

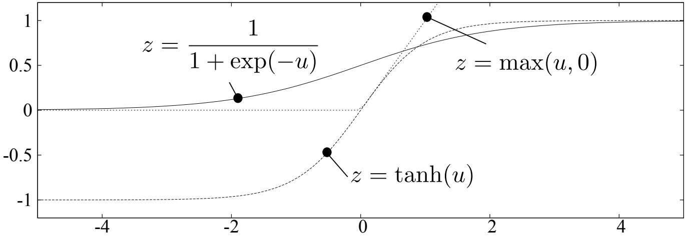

Figure: 典型的なシグモイド関数

マックスアウト

マックスアウト関数

$$u_{jk} = \sum w_{jik} z_i + b_{jk} \quad (1...K) \tag{5}$$

$$f(u_j) = max_{k=1...K}(u_{jk}) \tag{6}$$

- ユニットをまとめたような構造を持つ.
- 異なる重みとバイアスを持つそれぞれの総入力を別々に計算し最大値をユニットの出力とする.

多層ネットワーク

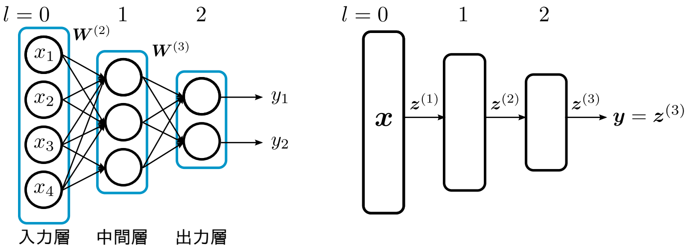

Figure: 2 層のネットワーク

- 入力 $\boldsymbol{u}^{(l)}$，出力 $\boldsymbol{z}^{(l)}$
- 各層間の結合重み $\boldsymbol{W}^{(l)}$ $(l = 2, \cdots, L)$
- ユニットのバイアス $\boldsymbol{b}^{(l)}$ $(l = 2, \cdots, L)$

多層ネットワーク

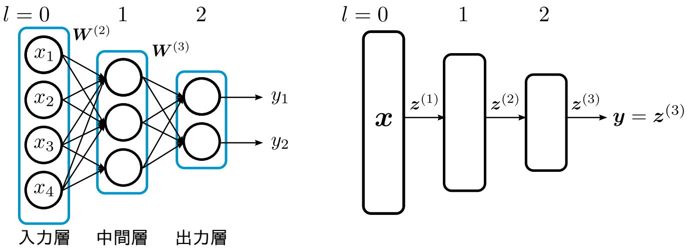

Figure: 2 層のネットワーク

中間層 ($l=2$)，出力層 ($l=3$) はそれぞれ

$$
\begin{aligned}
\boldsymbol{u}^{(2)} &= \boldsymbol{W}^{(2)}\boldsymbol{x}+\boldsymbol{b}^{(2)} \\
\boldsymbol{z}^{(2)} &= \boldsymbol{f}(\boldsymbol{u}^{(2)}) \\
\boldsymbol{u}^{(3)} &= \boldsymbol{W}^{(3)}\boldsymbol{z}^{(2)}+\boldsymbol{b}^{(3)} \\
\boldsymbol{z}^{(3)} &= \boldsymbol{f}(\boldsymbol{u}^{(3)})
\end{aligned}
$$

多層ネットワーク

任意の階層 L のネットワークに一般化すると

$$\boldsymbol{u}^{(l+1)} = \boldsymbol{W}^{(l+1)}\boldsymbol{z}^{(l)} + \boldsymbol{b}^{(l+1)} \tag{7}$$

$$\boldsymbol{z}^{(l+1)} = \boldsymbol{f}(\boldsymbol{u}^{(l+1)}) \tag{8}$$

- $l = 1,\ 2,\ 3, \cdots, L-1$ の順に繰り返していくと最終的な出力 $\boldsymbol{y}$ を決定することができる.
- 各層間の結合重み $\boldsymbol{W}^{(l)}$ とユニットのバイアス $\boldsymbol{b}^{(l)}$ を成分に持つベクトル $\boldsymbol{w}$ を定義する.
- これを $\boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w})$ と表現する.

出力層の設計と誤差関数

順伝播型ネットワークが表現する関数 $\boldsymbol{y}(\boldsymbol{x};\boldsymbol{w})$ をネットワークのパラメータ $\boldsymbol{w}$ を変えることで変化させ，望みの関数を与える．

$$\{(\boldsymbol{x}_1, \mathrm{d}_1), (\boldsymbol{x}_1, \mathrm{d}_1), ..., (\boldsymbol{x}_N, \mathrm{d}_N)\} \tag{9}$$

これらのペア $(\boldsymbol{x}, \mathrm{d})$ 1 つ 1 つを訓練サンプル (training samples) といい，その集合を訓練データ (training data) という．ネットワーク $\boldsymbol{w}$ を調整することで訓練データの入出力ペアをできるだけ再現すること学習という．
この場合，ネットワークが表す関数と訓練データとの近さ $(\boldsymbol{y}(\boldsymbol{x}_n;\boldsymbol{w}))$ を誤差関数 (error function) で定義する．誤差関数は問題の種別や活性化関数によって異なる．

Table: 問題の種別ごとの活性化関数と誤差関数

問題の種別	出力層の活性化関数	誤差関数
回帰	正接双曲線関数や恒等写像	二乗誤差 式 (10)
二値分類	ロジスティック関数	式 (11)
多クラス分類	ソフトマックス関数	交差エントロピー 式 (13)

回帰

回帰

出力連続値をとる関数を対象に訓練データを良く再現する関数を求めることをいう．回帰では活性化関数に正接双曲線関数や恒等写像を用いる

評価関数

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N} ||\boldsymbol{d}_n - \boldsymbol{y}(\boldsymbol{x}_n; \boldsymbol{w})||^2 \tag{10}$$

二値分類

二値分類

入力 $\boldsymbol{x}$ に応じて 2 種類に区別する問題を考える．すなわち，$d \in \{0, 1\}$ とする．このとき，活性化関数はロジスティック関数 $y = 1/(1 + \exp(-u))$ とする．

評価関数

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N} [d_n \log y(\boldsymbol{x}_n; \boldsymbol{w} + (1 - d_n) \log\{1 - y(\boldsymbol{x}_n; \boldsymbol{w})\})] \tag{11}$$

多クラス分類

多クラス分類

入力 $\boldsymbol{x}$ に応じて有限個のクラスに分類する問題である．活性化関数にはソフトマックス関数 (softmax function) が良く用いられる．

活性化関数と評価関数

$$y_k \equiv z_k^{(L)} = \frac{\exp(u_k^{(L)})}{\sum_{j=1}^{K} exp(u_j^{(L)})} \tag{12}$$

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{N} d_{nk} \log y_k(\boldsymbol{x}_n; \boldsymbol{w}) \tag{13}$$

今後の課題

理論研究

DNN，CNN，caffe について理解を深める

プログラミング

中間層の出力，可視化