

目 次

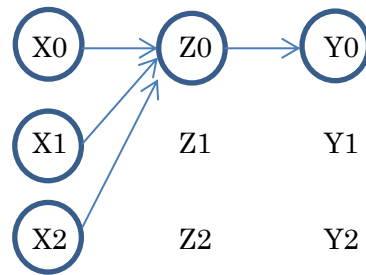
内容

1	Section1_入力層~中間層	2
2	Section2_活性化関数	3
3	Section3_出力層	6
4	Section4_勾配降下法	7
5	Section5_誤差逆伝播法	9

1 Section1_入力層~中間層

ニューラルネットワークの構造（2層）は以下のようなになる。

入力層 中間層 出力層



入力層 X_0, X_1, X_2 として、中間層へは $W_{00} \cdot X_0$ 、 $W_{01} \cdot X_1$ 、 $W_{02} \cdot X_2$ を渡す。

入力総和 $Z_0 = W_{00} \cdot X_0 + W_{01} \cdot X_1 + W_{02} \cdot X_2$

入力総和 $Z_1 = W_{10} \cdot X_0 + W_{11} \cdot X_1 + W_{12} \cdot X_2$

入力総和 $Z_2 = W_{20} \cdot X_0 + W_{21} \cdot X_1 + W_{22} \cdot X_2$

行列式より

$$\begin{bmatrix} W_{00} & W_{01} & W_{02} \\ W_{10} & W_{11} & W_{12} \\ W_{20} & W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} Z_0 \\ Z_1 \\ Z_2 \end{bmatrix}$$

2 Section2_活性化関数

ニューラルネットワークにおける活性化関数とは、あるニューロンから次のニューロンへと出力する際に、あらゆる入力値を別の数値に変換して出力する関数である。

1) ReLU

関数への入力値が0以下の場合は、出力値が常に0、入力値が0より大きい場合は、出力値が入力値と同じになる。

2) Mish

関数への入力値が0以下の場合は、出力値がほぼ0、入力値が0より大きい場合は、出力値が入力値と同じになる。

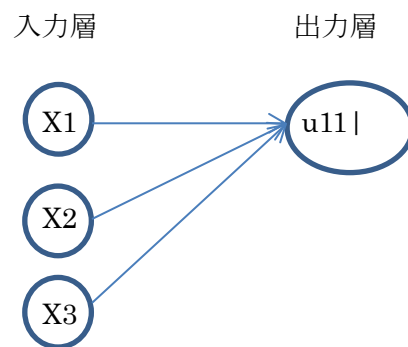
3) 活性化関数一覧

No	活性化関数	概要
1	ステップ関数	$Y = 1(x > 0)$ $0(x \leq 0)$
2	恒等関数	$Y = x$ (順伝播) $\delta y / \delta x = 1$ (逆伝播)
3	Bent Identity	$y = 1/2(\sqrt{x^2 + 1} + x)$
4	HardShrink	$Y = x(x < -\lambda \text{ or } \lambda < x)$ 0
5	SoftShrink	$Y = x + \lambda(x < -\lambda)$ (順伝播) $x - \lambda(\lambda > x)$ 0 $\delta y / \delta x = 1$ (逆伝播) 0
6	Threshold	$Y = x$ (順伝播) 0 $Y = 1$ (逆伝播) 0
7	シグモイド関数	$Y = 1/(1 + e^{-x})$ (順伝播) $\delta y / \delta x = y(1 - y)$ (逆伝播)
8	HardSigmoid	$Y = 1$ $0.2x + 0.5$ 0 $\delta y / \delta x = 0.2$ 0
9	logSigmoid	$Y = \log(1/(1 + e^{-x}))$ $\delta y / \delta x = 1/(1 + e^x)$
10	Tanh	$Y = \tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ $\delta y / \delta x = \text{sech}^2(x) = 1 / \cosh^2(x) = 4 / (e^x + e^{-x})^2$
11	tanhShrink	$Y = x \cdot \tanh(x)$ $\delta y / \delta x = \tanh^2(x)$

12	Hardtanh	$Y = \begin{cases} 1 & x \geq 1 \\ -1 & x \leq -1 \\ x & \text{otherwise} \end{cases}$ $\delta y / \delta x = \begin{cases} 0 & x \geq 1 \\ 1 & \text{otherwise} \end{cases}$
13	ReLU 関数	$Y = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$ $\delta y / \delta x = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$
14	ReLU6	$Y = \begin{cases} 0 & x \leq 0 \\ x & 0 < x < 6 \\ 6 & x \geq 6 \end{cases}$ $\delta y / \delta x = \begin{cases} 0 & (x \leq 0 \text{ or } x \geq 6) \\ 1 & \text{otherwise} \end{cases}$
15	Leak-ReLU	$Y = \begin{cases} x & x > 0 \\ 0.01x & x \leq 0 \end{cases}$ $\delta y / \delta x = \begin{cases} 1 & x > 0 \\ 0.01 & x \leq 0 \end{cases}$
16	ELU	$Y = \begin{cases} x & x \geq 0 \\ \alpha (e^x - 1) & x < 0 \end{cases}$ $\delta y / \delta x = \begin{cases} 1 & x \geq 0 \\ \alpha e^x & x < 0 \end{cases}$
17	SELU	$Y = \begin{cases} \lambda x & x \geq 0 \\ \lambda \alpha (e^x - 1) & x < 0 \end{cases}$ $\delta y / \delta x = \begin{cases} \lambda & x \geq 0 \\ \lambda \alpha e^x & x < 0 \end{cases}$
18	CELU	$Y = \begin{cases} x & x \geq 0 \\ \alpha (e^{x/\alpha} - 1) & x < 0 \end{cases}$ $\delta y / \delta x = \begin{cases} 1 & x \geq 0 \\ e^{x/\alpha} & x < 0 \end{cases}$
19	ソフトマックス関数	$Y_i = e^{x_i} / \sum_k e^{x_k}$ $\delta y / \delta x = y_i(1 - y_i) \sum_j y_j \delta_{ij} = \sum_j y_i (\delta_{ij} - y_j)$
20	Softmin	$Y_i = e^{-x_i} / \sum_k e^{-x_k}$ $\delta y / \delta x = -y_i(1 - y_i) + \sum_j y_j \delta_{ij} = \sum_j y_i (\delta_{ij} - y_j)$
21	Logsoftmax	$Y_i = \log(e^{x_i} / \sum_k e^{x_k})$ $\delta y / \delta x = \sum_j (\delta_{ij} - y_j)$
22	Softplus	$Y = \log(1 + e^x) = \ln(1 + e^x)$ $\delta y / \delta x = e^x / (1 + e^x) = 1 / (1 + e^{-x})$
23	Softsign	$Y = x / (1 + x)$ $\delta y / \delta x = 1 / (1 + x)^2$

24	Swish	$Y = x / (1 + e^{-\beta x})$ $\delta y / \delta x = \beta y + (1 - \beta y) / (1 + e^{-\beta x})$
25	hardSwish	$Y = 0$ $X(x+3)/6$ X $\delta y / \delta x = 0$ $(2x+3)/6$ 1
26	ACON	
27	Mish	
28	tanhExp	$Y = x \tanh(ex)$ $\delta y / \delta x = \tanh(ex) - x \exp(\tanh^2(ex) - 1)$

3 Section3_出力層



出力層では、活性化関数を使用して非線形変換を行う。

出力層の設計

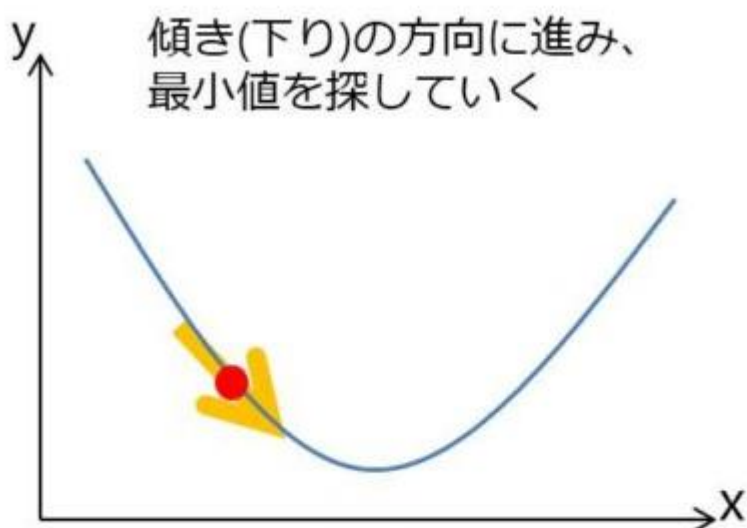
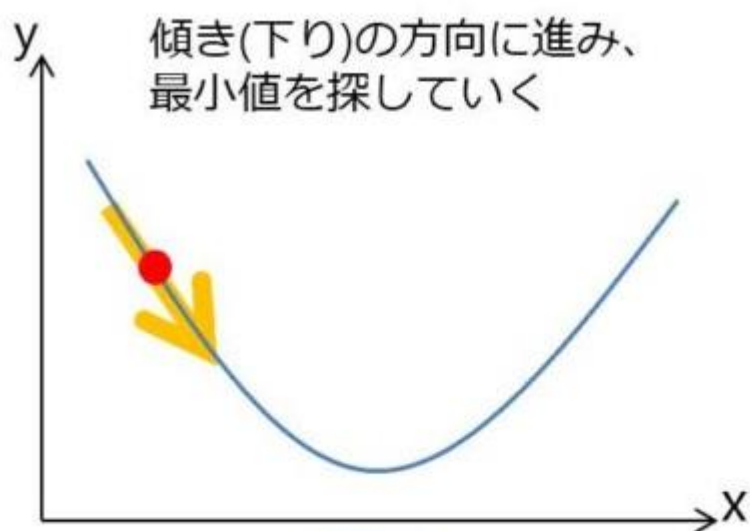
- 1) ニューラルネットワークは分類、回帰の両方に用いられる。
- 2) どちらの問題を解決するかで、活性化関数を変更する必要がある。
- 3) 回帰問題では恒等関数を、分類問題ではソフトマックス関数を用いる。
- 4) クラス分類では、出力層のニューロンの数はクラス数

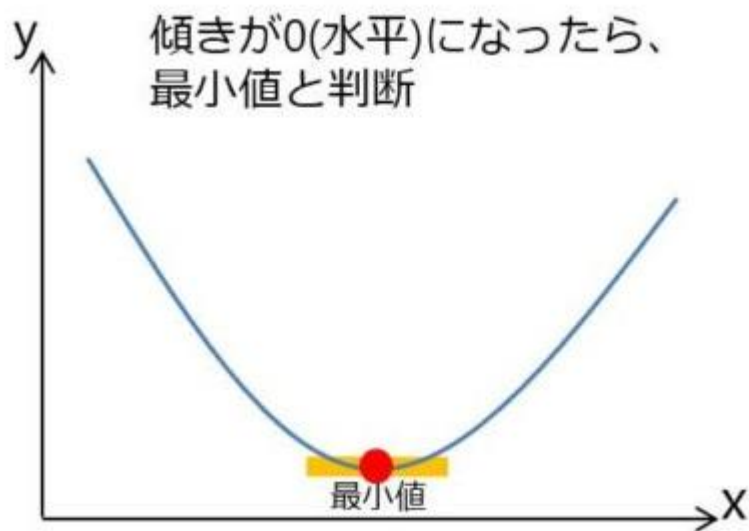
4 Section4_勾配降下法

モデルに対してコストが最小になるようにパラメータを少しずつ変化させて、トレーニングに適合したパラメータを算出するアルゴリズムである。

$$x = x - \eta \left(\frac{d f(x)}{d x} \right)$$

1) 連鎖律





手法	利用データ	計算時間	メリット	デメリット
バッチ勾配 降下法	全てのデータ	大	<ul style="list-style-type: none"> ・ 解への到達が早い ・ 結果が安定 	<ul style="list-style-type: none"> ・ メモリの使用量が多い ・ 局所解にはまりやすい
SGD	1つのデータ	小	<ul style="list-style-type: none"> ・ メモリの使用量が少ない ・ オンライン学習が可能 ・ 局所解を回避する可能性がある 	<ul style="list-style-type: none"> ・ 解への到達が遅いことがある ・ はづれ値の影響を大きく受ける
ミニバッチ勾配 降下法	一部のデータ	中	<ul style="list-style-type: none"> ・ バッチ勾配降下法と SGD のそれぞれのメリットがある 	それぞれのデメリットがある

