

ニューラルネットワーク構成図

【事前に用意する情報】

説明変数（入力値）： $\mathbf{x}_n = (x_{n1}, x_{n2}, x_{n3})$

x_{n1} → epal length (cm) : がく片の長さ

x_{n2} → sepal width (cm) : がく片の幅

x_{n3} → petal length (cm) : 花卉の長さ

目的変数（出力値との誤差算出に利用）： $\mathbf{d}_n = (d_{n1})$

d_{n1} → petal width (cm) : 花卉の幅

【多層ネットワークのパラメータ】

$$\mathbf{w}^{(l)} \begin{cases} \text{重み} : \mathbf{W}^{(l)} = \begin{pmatrix} w_{11}^{(l)} & \cdots & w_{J_l 1}^{(l)} \\ \vdots & \ddots & \vdots \\ w_{1 J_{l-1}}^{(l)} & \cdots & w_{J_l J_{l-1}}^{(l)} \end{pmatrix} \\ \text{バイアス} : \mathbf{b}^{(l)} = (b_1^{(l)} \quad \cdots \quad b_{J_l}^{(l)}) \end{cases}$$

活性化関数： $f^{(l)}(\mathbf{u}_n^{(l)}) = (f^{(l)}(u_{n1}^{(l)}) \quad \cdots \quad f^{(l)}(u_{nJ_l}^{(l)}))$

総入力： $\mathbf{u}_n^{(l)} = \mathbf{z}_n^{(l-1)} \mathbf{W}^{(l)} + \mathbf{b}^{(l)}$

中間層出力： $\mathbf{z}_n^{(l)} = (z_{n1}^{(l)} \quad \cdots \quad z_{nJ_l}^{(l)}) = f^{(l)}(\mathbf{u}_n^{(l)})$

出力層出力： $\mathbf{y}_n = (y_{n1}) = \mathbf{z}_n^{(L)}$

誤差関数： $E_n(\mathbf{w})$

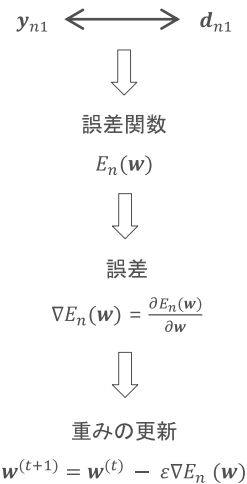
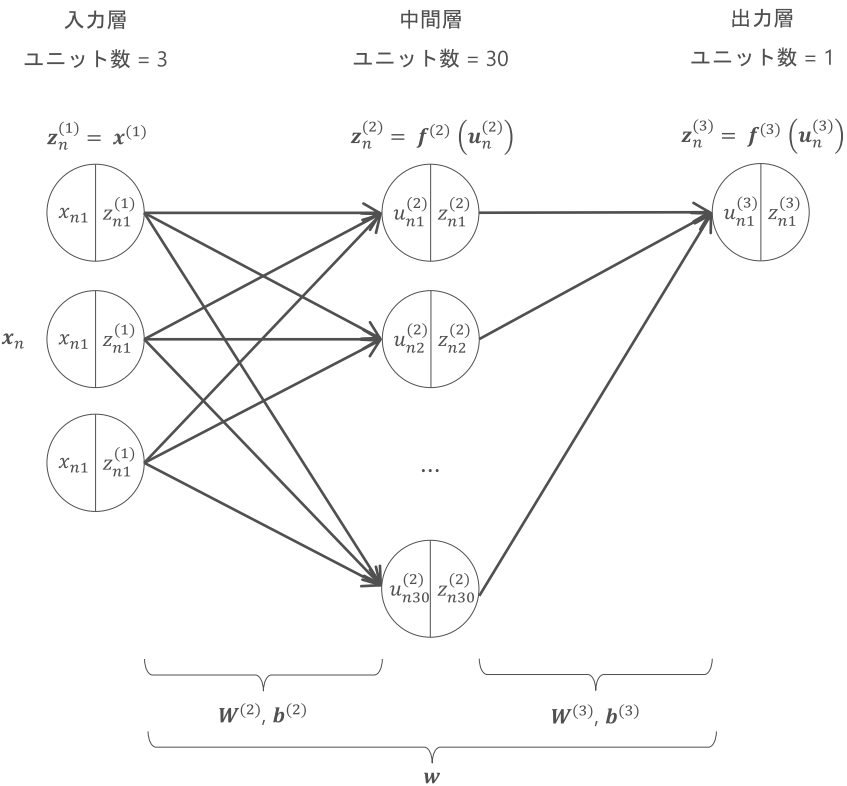
学習係数： ε

【各インデックス】

訓練データのインデックス： $n (= 1 \dots N)$

層のインデックス： $l (= 1 \dots L)$

中間層のユニットのインデックス： $j_l (= 1 \dots J_l)$



補記

- 中間層の層数は何層か？
 - 1層にしました。入力層⇒中間層⇒出力層の合計3層となります。
- 中間層の活性化関数には何を使用したか？
 - RELU を利用しました。
- 出力層の活性化関数に何を使用したか？
 - 恒等写像を利用しました。
- 誤差関数に何を使用したか？
 - 二乗誤差を利用しました。
- その他
 - クラスを使って実装したノートと、クラスを使わず実装したノートを提出します。
 - IRIS_TwoLayer_NotUsingClass.ipynb
 - IRIS_TwoLayer_UsingClass.ipynb
 - 勾配 (dW , db) を計算するタイミングで、ミニバッチのデータ数で除算しています。これは「深層学習（岡谷貴之）」の記載を参考にしました。（一方、「ゼロから作るDeep Learning（斎藤康毅）」では、出力層のデルタを求める段階で、早々に、ミニバッチのデータ数で除算しているようです）