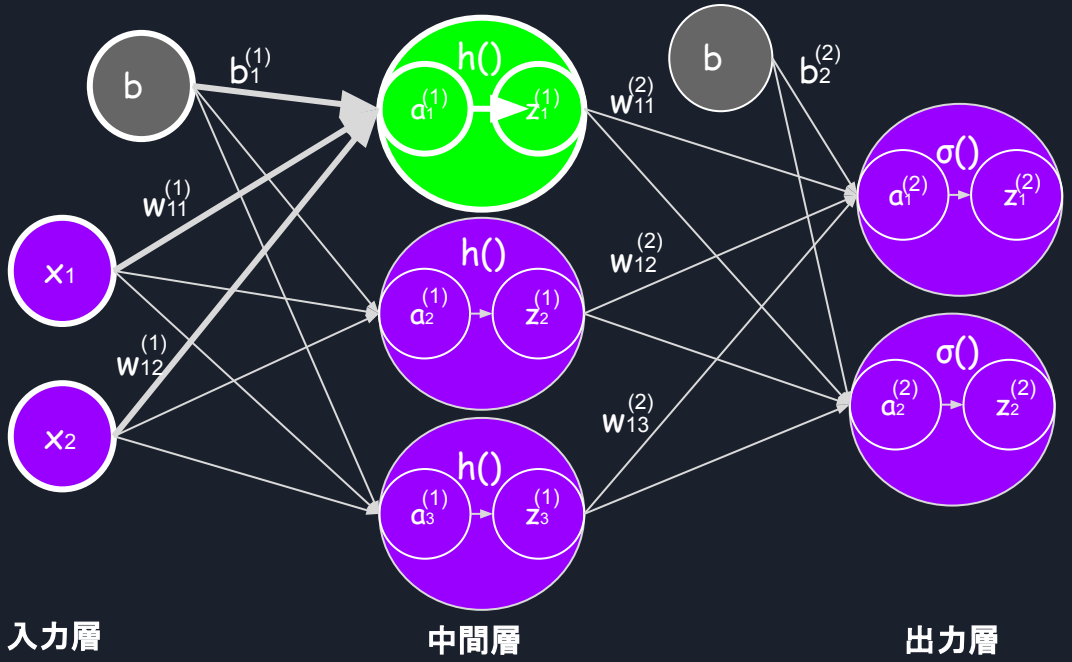
ニューラルネットワーク

2022/07/20

ニューラルネットワークの概要

ニューラルネットワークの例として、以下に3層からなるニューラルネットワークを示す。

ネットワークの左側からデータXを入力し、右側からネットワークが予測した結果Z2を出力する。一番左の層（第0層）を**入力層**、一番右の層（第2層）を**出力層**、中間の層（第1層）を**中間層**と呼ぶ。中間層は**隠れ層**とも呼ぶ。



$w_{12}^{(1)}$
- 第1層目の重み
- 前層の2番目のニューロン
- 次層の1番目のニューロン

中間層1番目のニューロン(ノード)の入出力を表す式

$$a_1^{(1)} = x_1 * w_{11}^{(1)} + x_2 * w_{12}^{(1)} + b_1^{(1)}$$

$$z_1^{(1)} = h(a_1^{(1)})$$

w：重みパラメータ、b：バイアス、$h()$：活性化関数

※重みパラメータとバイアスは訓練データによる学習により自動的に獲得するパラメータ。活性化関数は人が与える

ニューラルネットワークの式表現

第1層目の重み付き和、A1は以下の式で表すことが出来ます。また、第1層目の出力Z1は以下の式で表すことが出来ます。
ここで、A1 、X、W1、 B1は下記に示す行列です。h()は活性化関数です。

$$\boldsymbol{A1} = \boldsymbol{X} * \boldsymbol{W1} + \boldsymbol{B1}$$

$$\boldsymbol{A1} = (a_1^{(1)} \quad a_2^{(1)} \quad a_3^{(1)}), \quad \boldsymbol{X} = (x_1 \quad x_2), \quad \boldsymbol{B1} = (b_1^{(1)} \quad b_2^{(1)} \quad b_3^{(1)})$$

$$\boldsymbol{W1} = \begin{pmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{pmatrix}$$

$$\boldsymbol{Z1} = \boldsymbol{h}(\boldsymbol{A1})$$

第2層目の重み付き和、A2は以下の式で表すことが出来ます。また、第2層目の出力Z2は以下の式で表すことが出来ます。
ここで、A2、Z1、W2、 B2は下記に示す行列です。σ()は出力層で使用する活性化関数です。

$$\boldsymbol{A2} = \boldsymbol{Z1} * \boldsymbol{W2} + \boldsymbol{B2}$$

$$\boldsymbol{A2} = (a_1^{(2)} \quad a_2^{(2)} \quad a_3^{(2)}), \quad \boldsymbol{Z1} = (h(a_1^{(1)}) \quad h(a_2^{(1)}) \quad h(a_3^{(1)})), \quad \boldsymbol{B2} = (b_1^{(2)} \quad b_2^{(2)} \quad b_3^{(2)})$$

$$\boldsymbol{W2} = \begin{pmatrix} w_{11}^{(2)} & w_{21}^{(2)} & w_{31}^{(2)} \\ w_{12}^{(2)} & w_{22}^{(1)} & w_{32}^{(1)} \end{pmatrix}$$

$$\boldsymbol{Z2} = \boldsymbol{\sigma}(\boldsymbol{A2})$$

ニューラルネットワークの実装

行列式はpythonを使うと簡単に実装できます。
以下に、P.3のニューラルネットワークの式をpythonで実装した例を示す。

```
X = np.array([1.0, 0.5])                  # 1x2の行列
W1 = np.array([[0.1, 0.3, 0.5], [0.2, 0.4, 0.6]]) # 2x3の行列
B1 = np.array([0.1, 0.2, 0.3])            # 1x3の行列


#第1層の実装
A1 = np.dot(X, W1) + B1      #[0.3, 0.7, 1.1]の1x3の行列
Z1 = h(A1)                   #h()は中間層の活性化関数、Z1は1x3の行列

#第2層の実装
A2 = np.dot(Z1, W2) + B2     #A2は1x3の行列
Z2 = σ(A2)                   #σ()は出力層の活性化関数、Z2は1x3の行列
```

ニューラルネットワークの実装例1

“train_neuralnet.ipynb”は、3層からなるニューラルネットワークtwo_layer_net.ipynb”を使って、MNISTデータを学習するコードである。

事前準備:deep-learning-from-scratch.zipの取得
github（HidekazuYamasaki/autoaichallenge）から、deep-learning-from-scratch.zipをダウンロードし、Googleドライブのマイドライブフォルダ以下に展開します。

<<train_neuralnet.ipynb>>
以下の順で訓練データに対して学習を行うコード。

ステップ1:MNISTデータ取得と入力データ前処理(データ正規化と一次元変換)

ステップ2:モデルの作成(入力層:784、隠れ層:50、出力層:10)
　※入力層は入力画像サイズの28x28=784、出力層は数値0～9の各ラベルに対応する10

ステップ3:ミニバッチを抽出
　※訓練データ(6,0000)から無作為に100個のデータを抽出

ステップ4:勾配の算出

ステップ5:パラメータの更新

ステップ6:ステップ3～ステップ5を繰り返し

ニューラルネットワークの実装例2

<<two_layer_net.ipynb>>
以下の順で訓練データに対して学習を行うコード。

ステップ1：MNISTデータ取得と入力データ前処理(データ正規化と一次元変換)

ステップ2：モデルの作成(入力層：784、隠れ層：50、出力層：10)
　※入力層は入力画像サイズの28x28=784、出力層は数値0～9の各ラベルに対応する10

ステップ3：ミニバッチを抽出
　※訓練データ(6,0000)から無作為に100個のデータを抽出

ステップ4：勾配の算出

ステップ5：パラメータの更新

ステップ6：ステップ3～ステップ5を繰り返し