深層学習day1確認テスト
確認テスト1-1
入力データをもとに識別を行う。
その過程で重みとバイアスを学習させモデルを生成する。
3.重み、4.バイアス

確認テスト1-2
「確認テスト1-2.png」

確認テスト1-3
「確認テスト1-3.png」

確認テスト1-4
u = np.dot(x, W) + b

確認テスト1-5
z1 = functions.relu(u1)
z2 = functions.relu(u2)

↑
※模範解答では中間層の計算部分も含めている

確認テスト2-1
「確認テスト2-1.png」

確認テスト2-2
z1 = functions.relu(u1)
z2 = functions.relu(u2)

確認テスト3-1
・なぜ、引き算でなく二乗するか述べよ
→誤差を足し合わせる際に、二乗することで符号をプラスに揃えるため。

・下式の1/2じゃどういう意味を持つか述べよ
→後々誤差関数を微分する際に効率良く計算を行うため。

確認テスト3-2
① def softmax(x):
ソフトマックス関数に相当するもの。
② np.exp(x)
i番目の出力層の値の指数関数。確率の大きさを表す。
③ np.sum(np.exp(x))
全ての出力層の値の指数関数を足し合わせたもの。
分母に置くことで確率の総和を1にする。

確認テスト3-3
①def cross_entropy_error(d, y):
交差エントロピー誤差関数に相当するもの。
d:正解ラベル、y:出力結果となる。

②-np.sum(np.log(y[np.arange(batch_size), d] + 1e-7)) / batch_size
交差エントロピー誤差の計算部分。
出力結果の対数と正解ラベルを掛け合わせ、関数の最小を求めるためのマイナスをつけている。
↑
※1e-7
対数関数で0の時に-∞に発散しないようにするため

確認テスト4-1
勾配降下法に該当するソースコード
network[key] -= learning_rate * grad[key]

確認テスト4-2
オンライン学習とは何か
一度に全てのデータを学習させるのではなく、
モデルに都度データを学習させていく学習方法。
Cf.) バッチ学習

確認テスト4-3
「確認テスト4-3.png」

確認テスト5-1
誤差逆伝播法では不要な再帰的処理を避けることができる。
既に行った計算結果を保持しているソースコードを抽出せよ。
grad = backward(x, d, z1, y)
↑
(正解)
delta2 = functions.d_mean_squared_error(d, y)
delta1 = np.dot(delta2, W2.T) * functions.d_sigmoid(z1)

確認テスト5-2
2つの空欄に該当するソースコードを探せ。
1. delta1 = np.dot(delta2, W2.T) * functions.d_sigmoid(z1)
2. grad['W1'] = np.dot(x.T, delta1)