

みんなのAI講座

ゼロからPythonで学ぶ人工知能と機械学習

#Live人工知能

Section5の概要



講座の内容

Section 1. 人工知能の概要と開発環境

Section 2. Pythonの基礎

Section 3. 必要な数学の学習

Section 4. ニューラルネットワーク

 **Section 5. 機械学習**

Section 6. 機械学習ライブラリの活用

Section 7. さらに学ぶために

今回の内容

1. Section5の概要
2. 学習の仕組み
3. 出力層の学習
4. 中間層の学習
5. 演習
6. 質疑応答

教材の紹介

- **Section5の教材:**

train_output.ipynb

train_middle.ipynb

- **Section5の演習:**

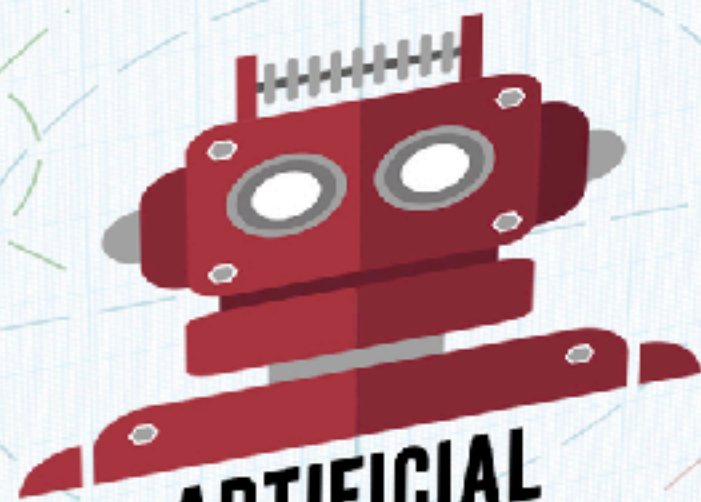
exercise.ipynb

ハッシュタグ

#Live人工知能

演習の解答 Section4

https://github.com/yukinaga/minnano_ai/blob/master/section_4/exercise.ipynb



ARTIFICIAL
INTELLIGENCE

CLICK

Click here for more information

学習の仕組み

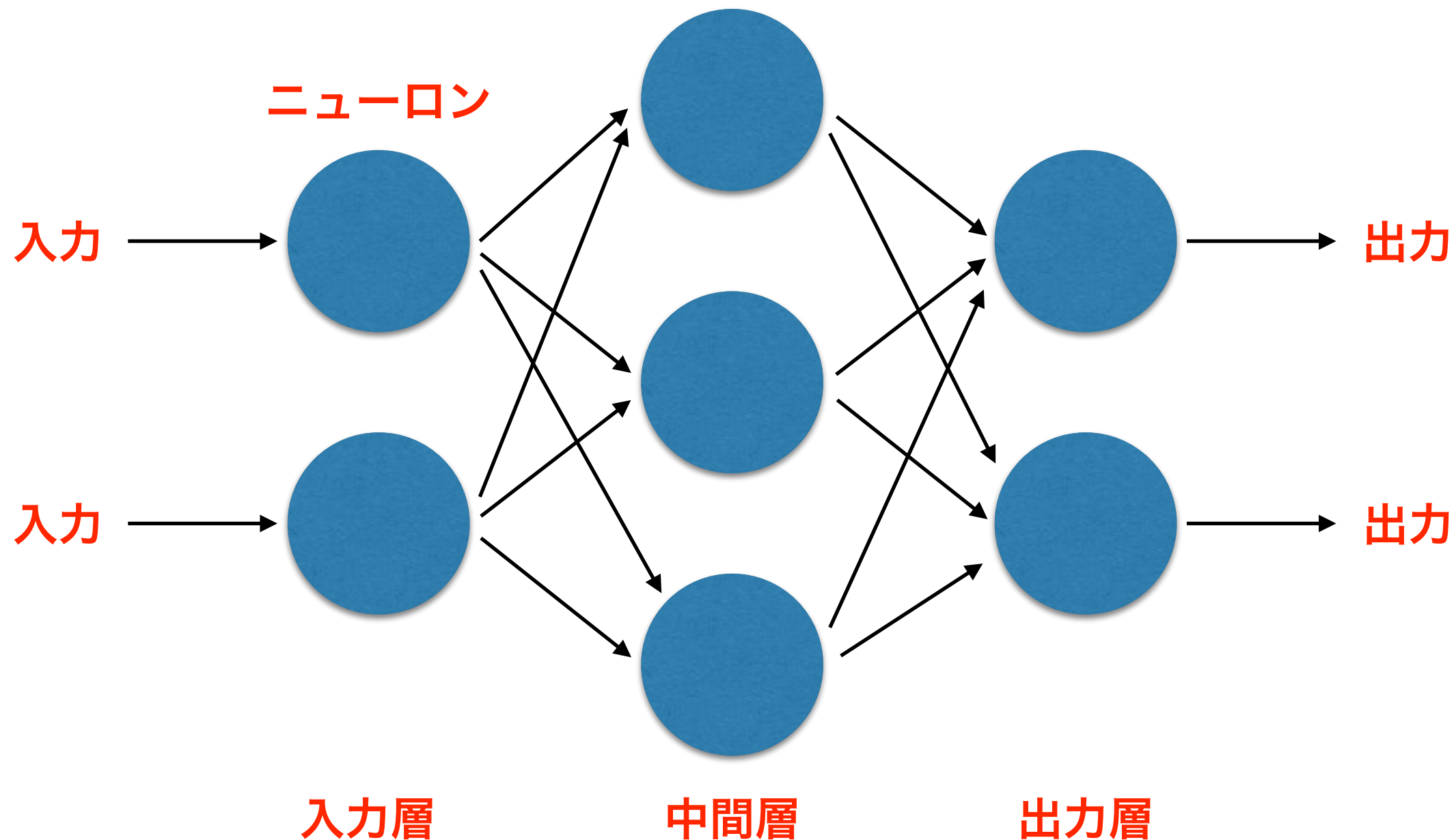
```
<br> {code*  
<html>  
<body>
```

SUNDAY	
MONDAY	
TUESDAY	
WEDNESDAY	
THURSDAY	
FRIDAY	
SATURDAY	

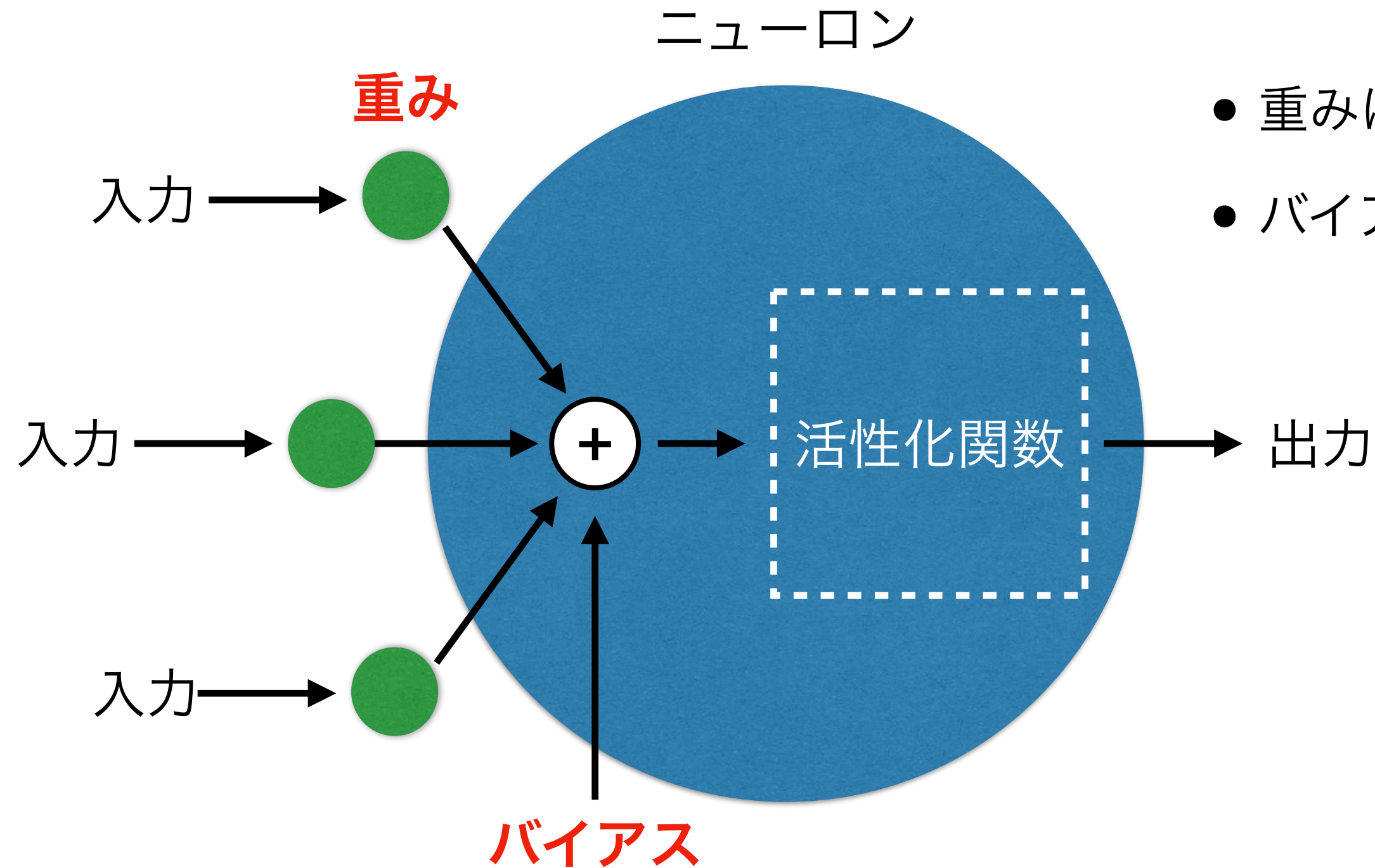


(復習) ニューラルネットワーク

- 神経細胞のネットワークを抽象化し、コンピュータ上で再現

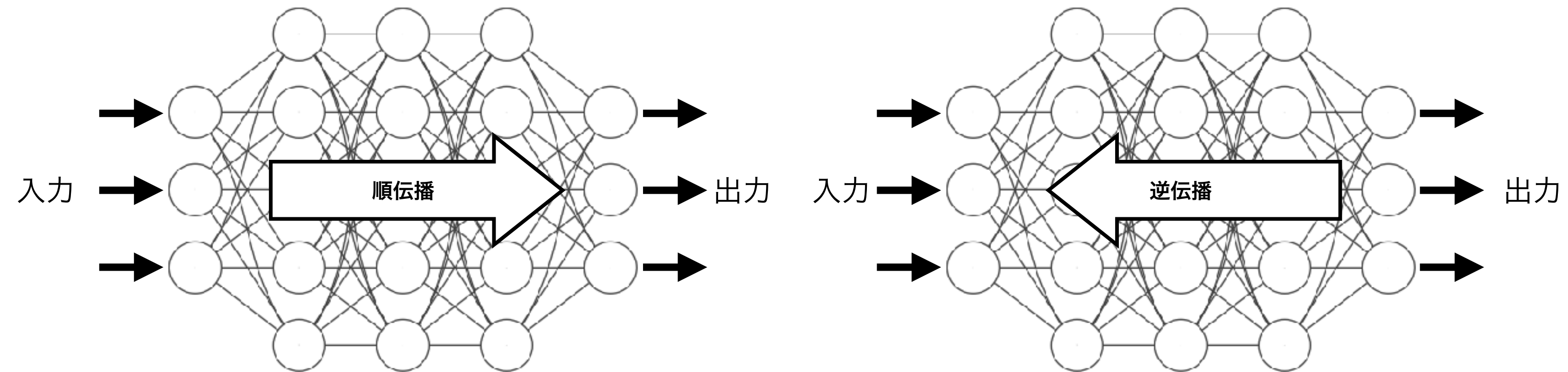


(復習) ニューロン



- 重みはシナプスの伝達効率
- バイアスはニューロンの感度

順伝播と逆伝播

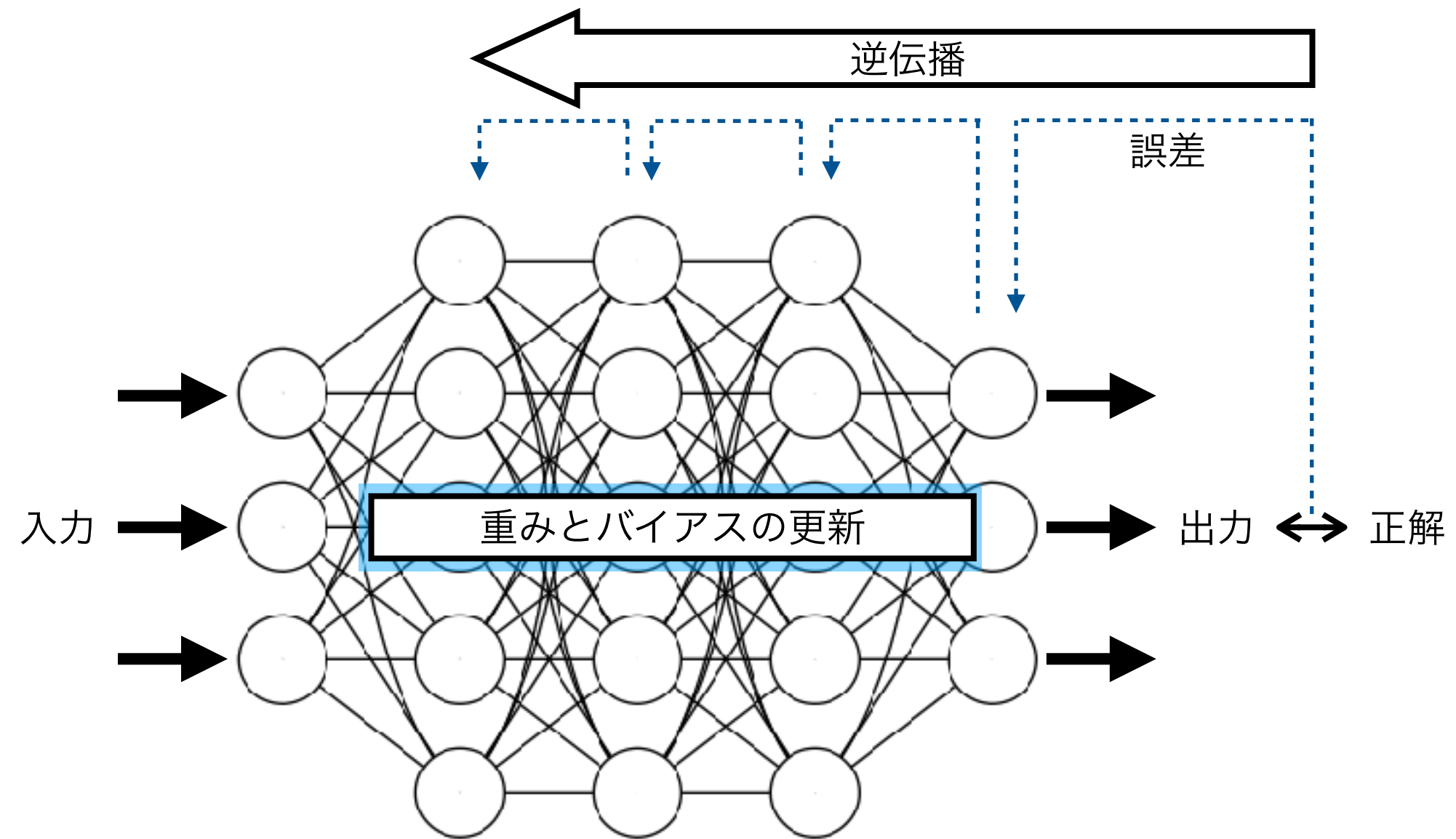


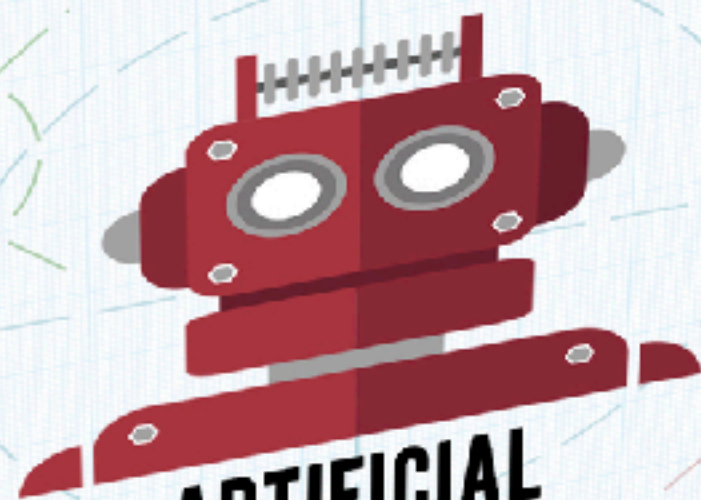
- 分類などに用いる

- 学習に用いる

バックプロパゲーションによる学習

- 出力と正解の誤差が小さくなるように重みとバイアスを調整することで学習が行われる
- 誤差の逆伝播による学習アルゴリズムは、バックプロパゲーション（誤差逆伝播法）と呼ばれる





ARTIFICIAL
INTELLIGENCE

CLICK

Click here for more information

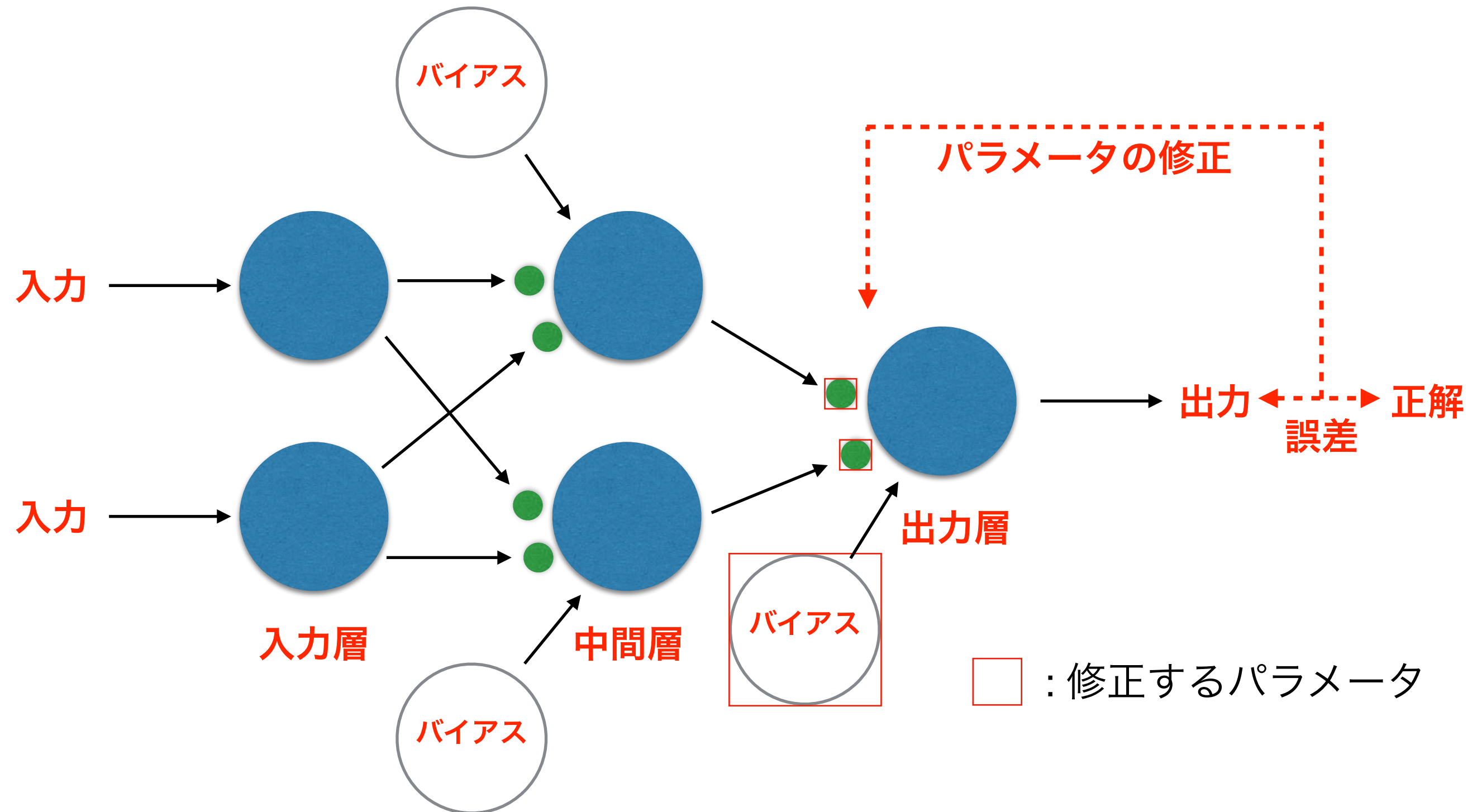
出力層の学習

```
<br> {code*  
<html>  
<body>
```

SUNDAY	
MONDAY	
TUESDAY	
WEDNESDAY	
THURSDAY	
FRIDAY	
SATURDAY	



出力層の学習



パラメータの更新（出力層）

- 最初に、修正量のベース δ_o を求める

$$\delta_o = (\text{出力} - \text{正解}) \times \text{活性化関数の微分形}$$

- 活性化関数がシグモイド関数の場合

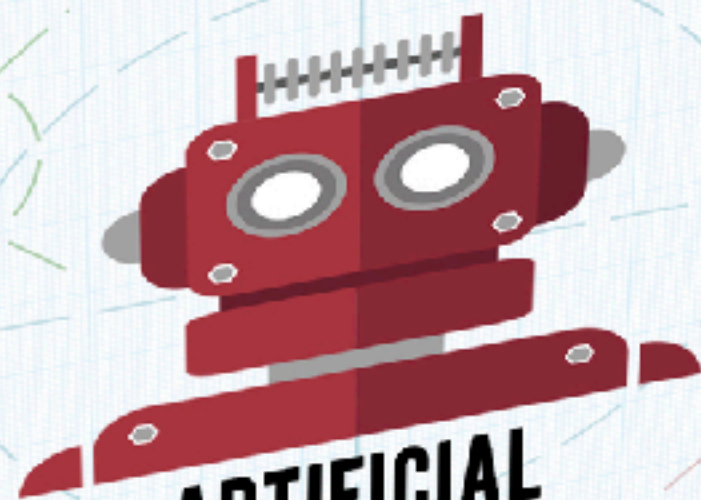
$$\delta_o = (\text{出力} - \text{正解}) \times \text{出力} \times (1 - \text{出力})$$

- δ_o を使って、重みとバイアスの修正量を求める

$$\text{重みの修正量} = - \text{学習係数} \times \delta_o \times \text{入力}$$

$$\text{バイアスの修正量} = - \text{学習係数} \times \delta_o$$

- δ_o は、中間層のパラメータ更新にも利用する



ARTIFICIAL
INTELLIGENCE

CLICK

Click here for more information

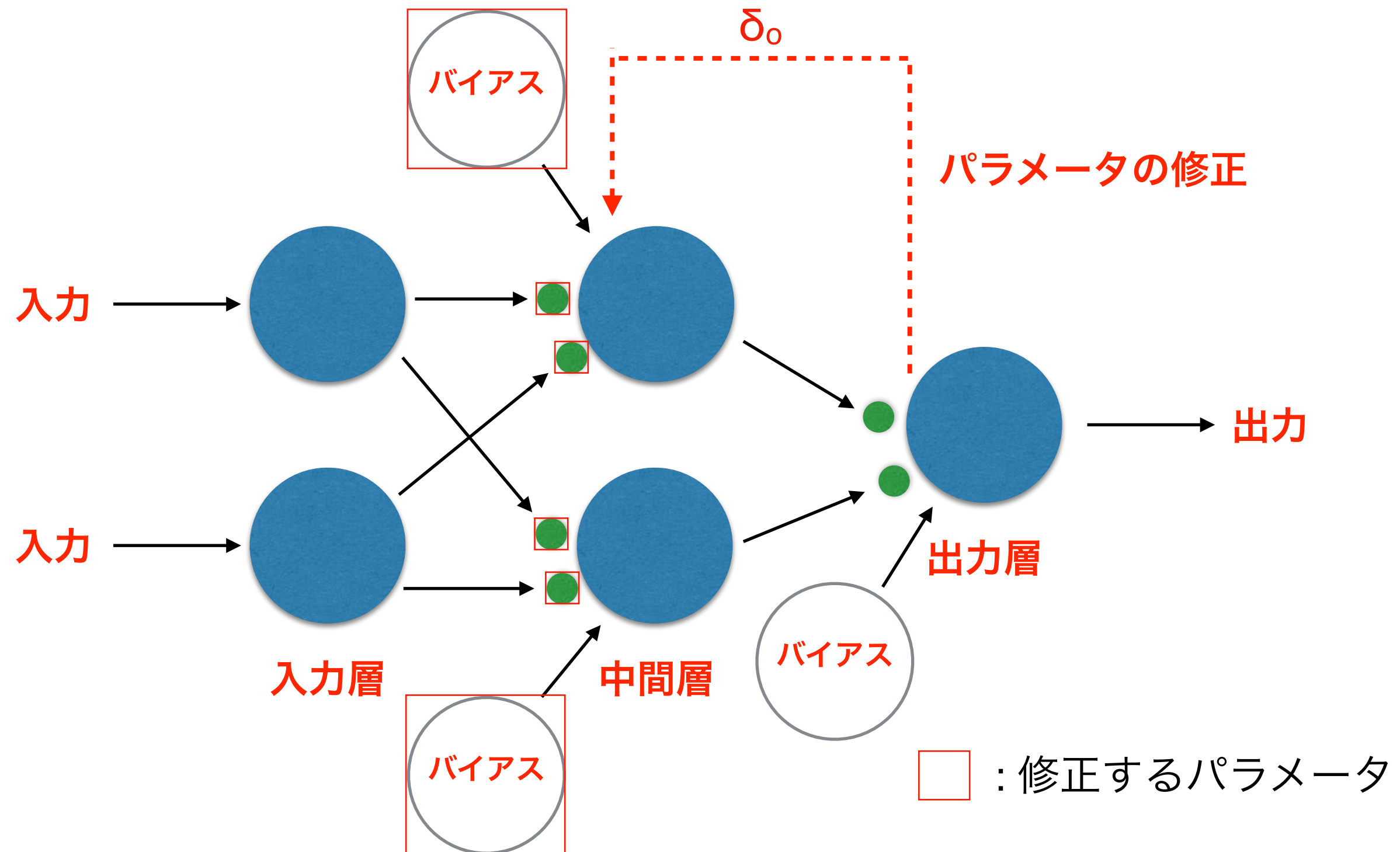
中間層の学習

```
<br> {code*  
<html>  
<body>
```

SUNDAY	
MONDAY	
TUESDAY	
WEDNESDAY	
THURSDAY	
FRIDAY	
SATURDAY	



中間層の学習



パラメータの更新（中間層）

- δ_o を使って、中間層における修正量のベース δ_m を求める

$$\delta_m = \delta_o \times \text{出力にかける重み} \times \text{活性化関数の微分形}$$

- 活性化関数がシグモイド関数の場合

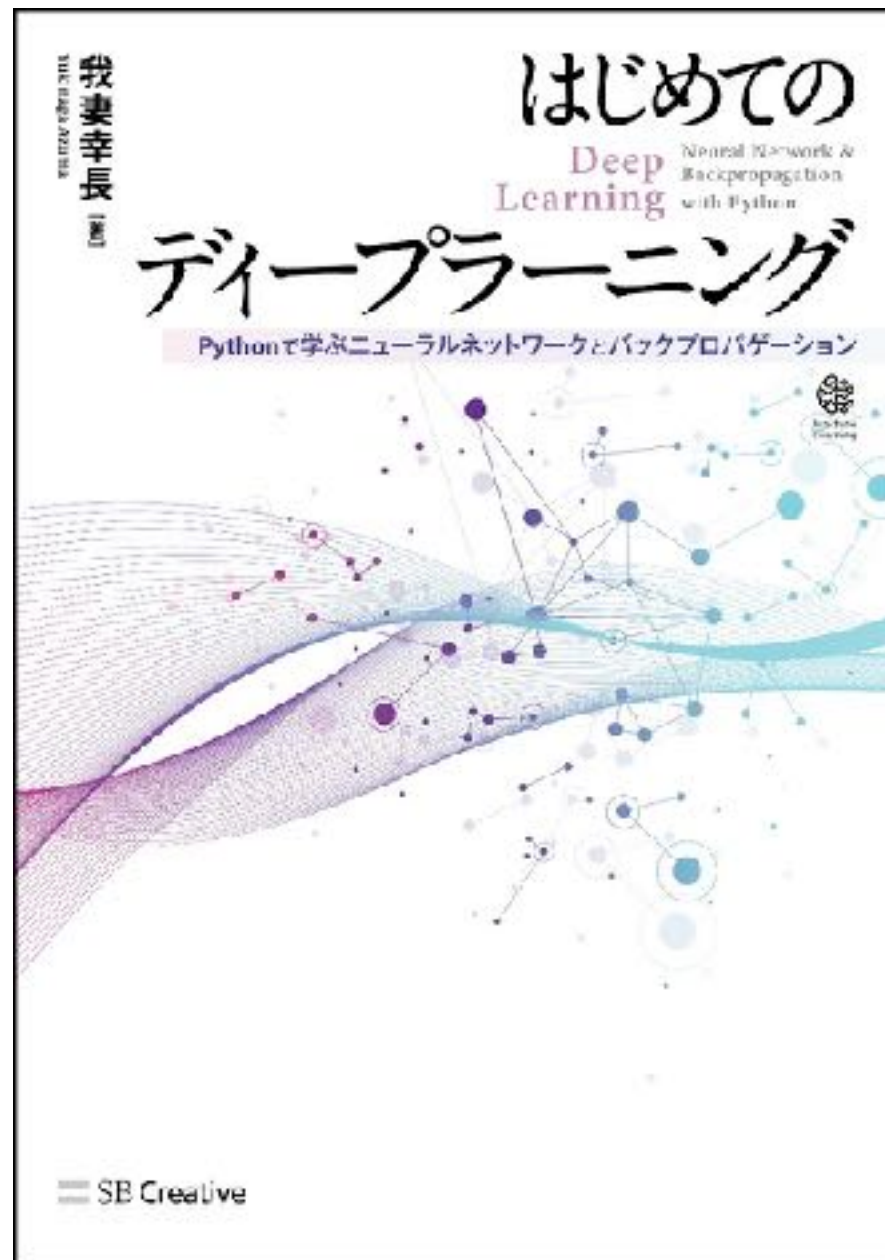
$$\delta_m = \delta_o \times \text{出力にかける重み} \times \text{出力} \times (1 - \text{出力})$$

- δ_m を使って、重みとバイアスの修正量を求める

$$\text{重みの修正量} = - \text{学習係数} \times \delta_m \times \text{入力}$$

$$\text{バイアスの修正量} = - \text{学習係数} \times \delta_m$$

より詳しく学びたい方へ...



Udemy講座

ディープラーニング : Pythonで
ゼロから構築し学ぶ人工知能
(AI) と深層学習の原理



演習

```
<br>
<html> {code*
<body>
```

SUNDAY	
MONDAY	
TUESDAY	
WEDNESDAY	
THURSDAY	
FRIDAY	
SATURDAY	

次回

Section 1. 人工知能の概要と開発環境

Section 2. Pythonの基礎

Section 3. 必要な数学の学習

Section 4. ニューラルネットワーク

Section 5. 機械学習

 **Section 6. 機械学習ライブラリの活用**

Section 7. さらに学ぶために



質疑応答

```
<br>
<html> {code*
<body>
```

SUNDAY	
MONDAY	
TUESDAY	
WEDNESDAY	
THURSDAY	
FRIDAY	
SATURDAY	