

Live! 人工知能

ARTIFICIAL INTELLIGENCE

#1

PyTorch + Deep Learning



ARTIFICIAL INTELLIGENCE

講座の概要

講師紹介



我妻 幸長
Yukinaga Azuma
@yuky_az

SAI-Lab株式会社 代表取締役

AI関連の教育、研究開発に従事

理学博士（物理学）

Udemyで4万人近くを指導

著書に「はじめてのディープラーニング」など



講座のゴール

PyTorchを使って、**ディープラーニング**が
実装できる

講座の内容

第1講. イントロダクション

第2講. PyTorchで実装する簡単なディープラーニング

第3講. PyTorchの様々な機能

第4講. 畳み込みニューラルネットワーク (CNN)

第5講. 再帰型ニューラルネットワーク (RNN)

第6講. AIアプリのデプロイ

今回の内容

1. 講座の概要
2. ディープラーニングとは
3. PyTorchとは
4. Google Colaboratoryの使い方
5. Tensorについて
6. 演習
7. 質疑応答

教材の紹介

- Pythonの基礎
- 第1講の教材: **tensor.ipynb**
- 第1講の演習: **exercise.ipynb**



ARTIFICIAL INTELLIGENCE

ディープラーニングとは

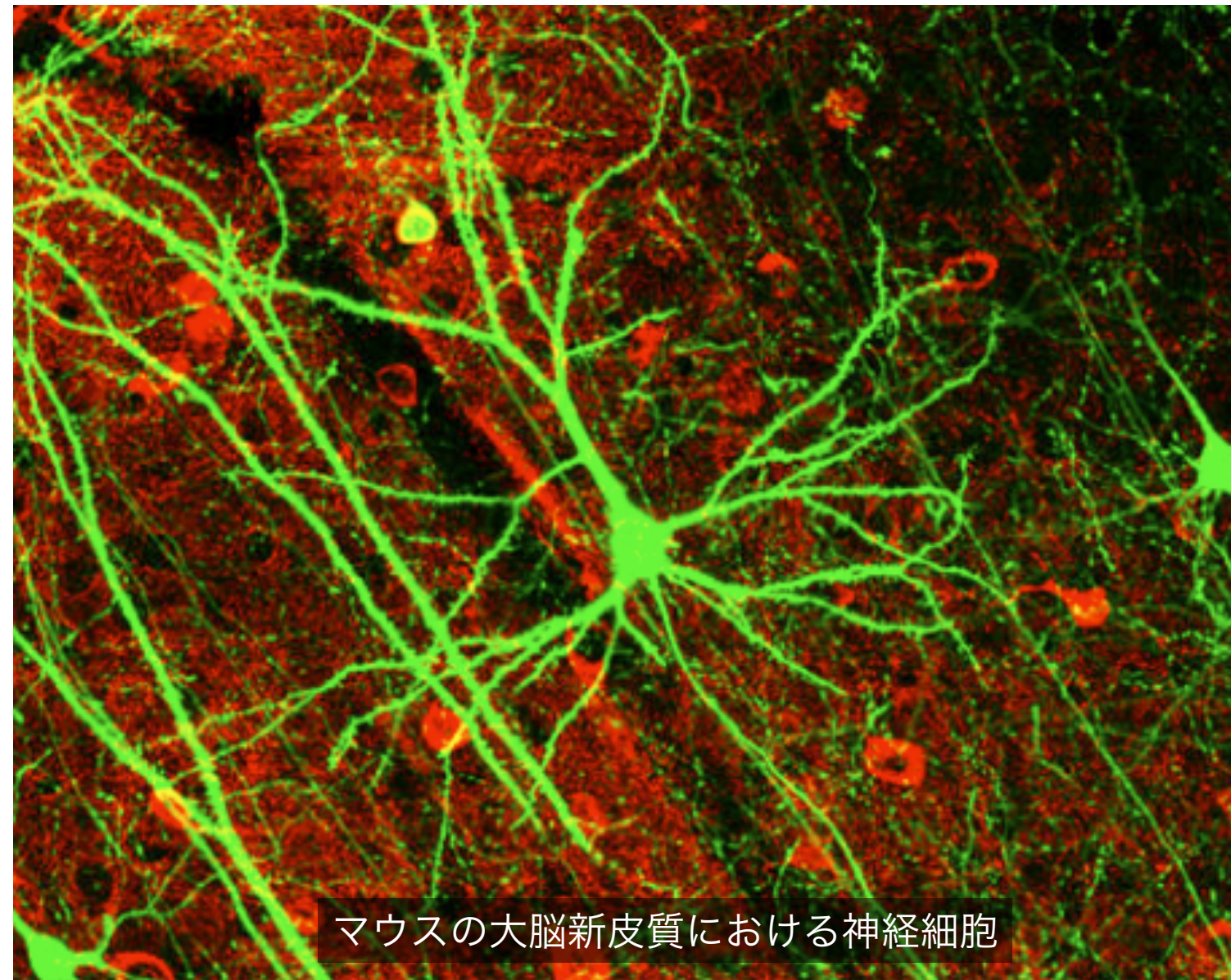
概念の整理

人工知能 (AI)

機械学習

ディープラーニング

神経細胞の写真



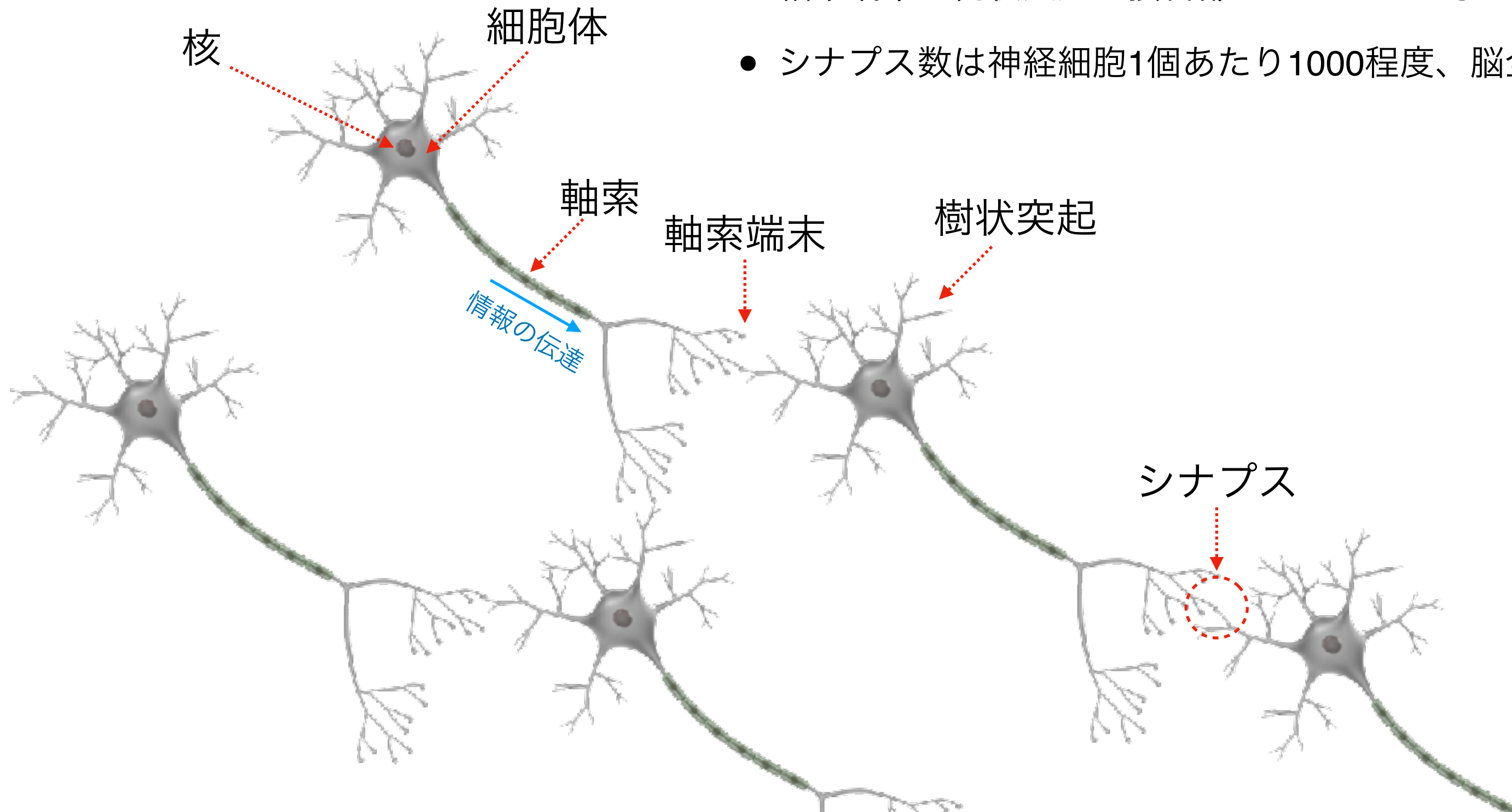
マウス的大脑新皮質における神経細胞

<https://en.wikipedia.org/wiki/Neuron> より引用

- 神経細胞には、錐体細胞、星状細胞、顆粒細胞など様々な種類が存在する

神経細胞のネットワーク

- 軸索末端と樹状突起の接合部はシナプスと呼ばれる
- シナプス数は神経細胞1個あたり1000程度、脳全体で100兆程度

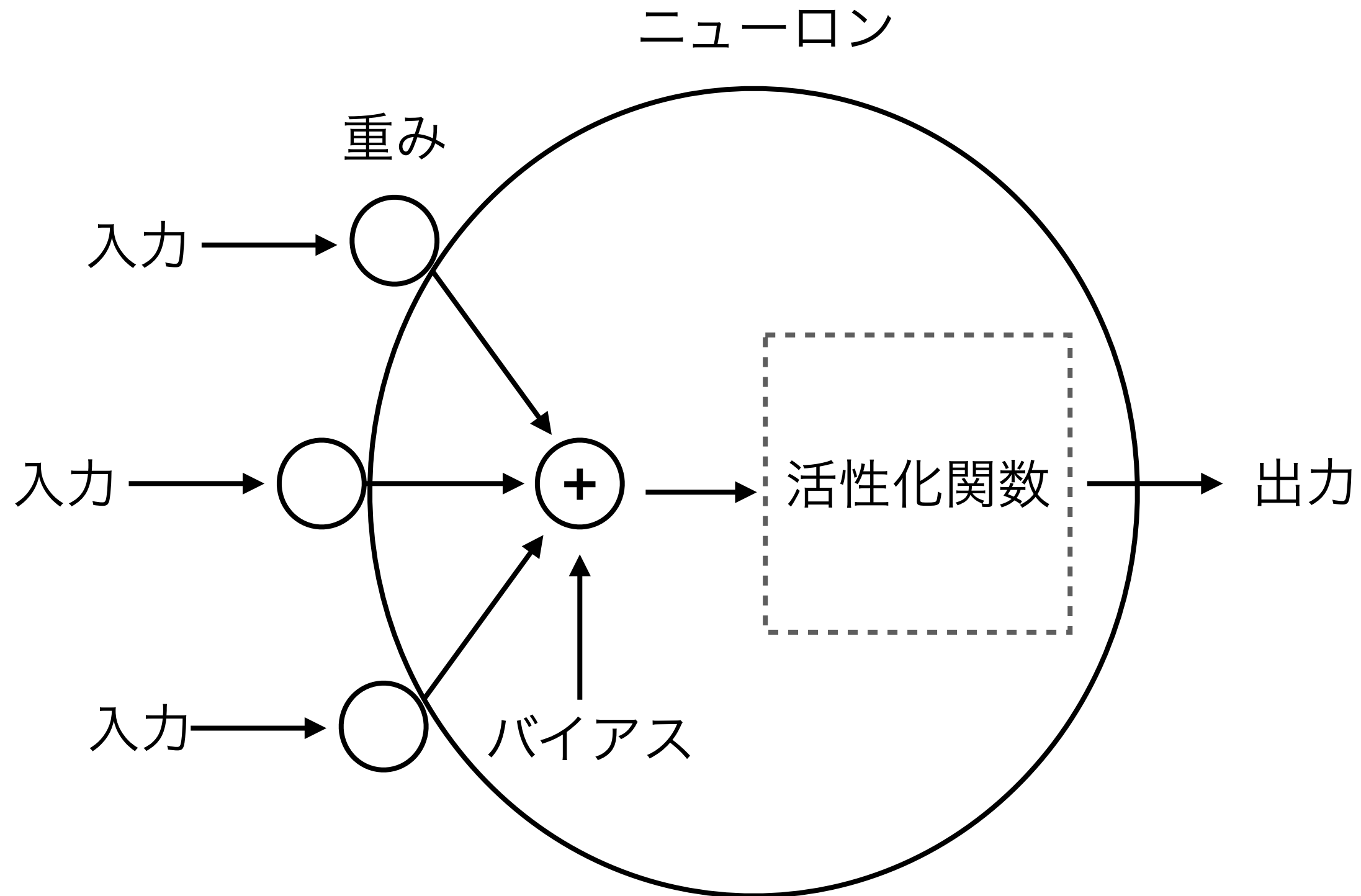


コンピュータ上における神経細胞のモデル化

- 人工ニューロン (Artificial neuron)
- 人工ニューラルネットワーク (Artificial neural network)

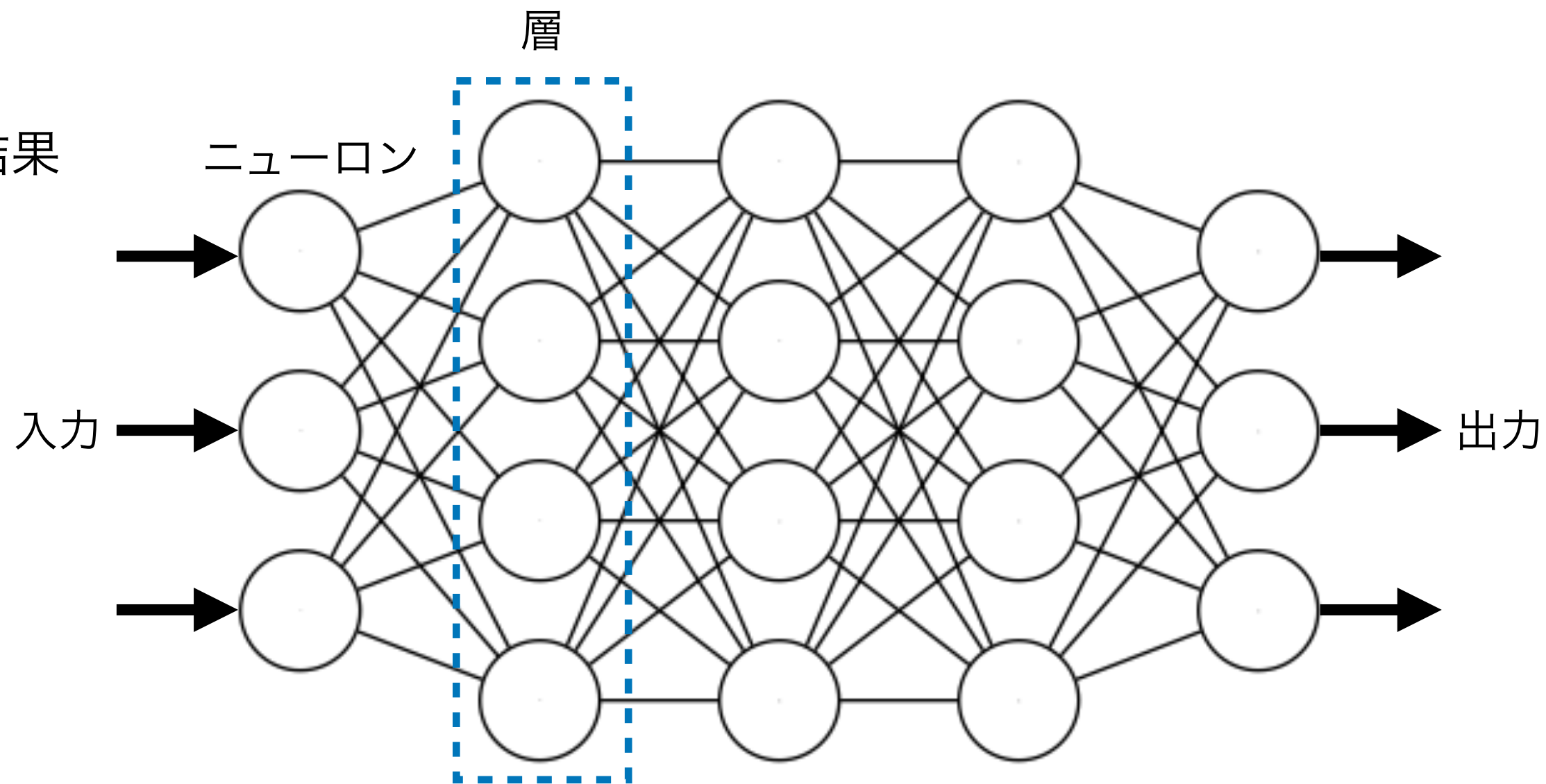
人工ニューロン

- ニューロンへの入力に**重み**をかけた値を続けて足し合わせ、**バイアス**を加える
- 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
- 上記の値を**活性化関数**で処理し出力とする
- 活性化関数は、値を信号に変換する関数



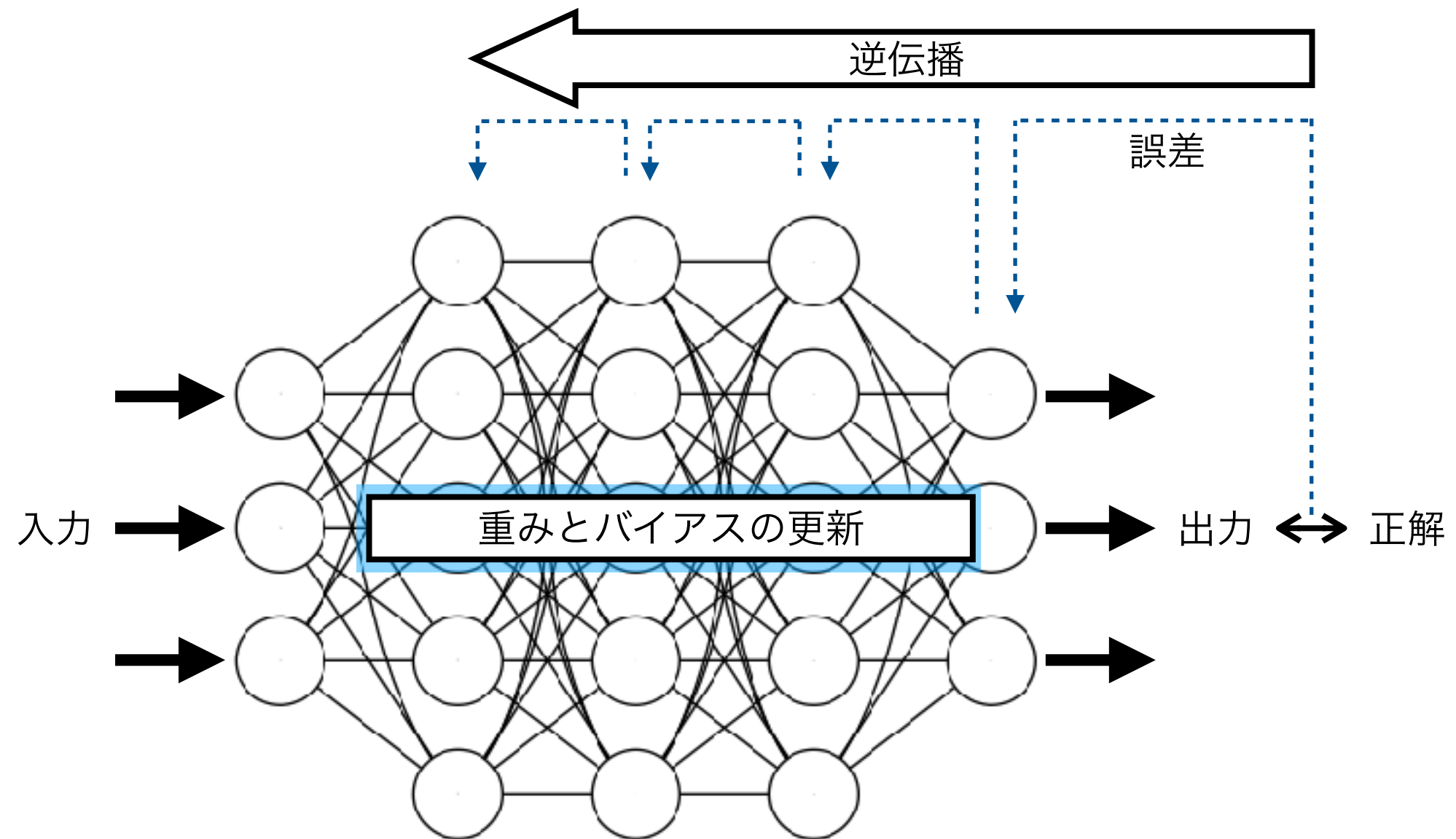
人工ニューラルネットワーク

- 人工ニューラルネットワークは、
ニューロンを層状に並べたもの
- 数値を入力し、情報を伝播させ結果
を出力する
- 出力は確率などの予測値として
解釈可能で、ネットワークにより
予測を行うことが可能
- ニューロンや層の数を
増やすことで、高い表現力
を発揮するようになる



バックプロパゲーションによる学習

- ニューラルネットワークは、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することができる
- 一層ずつ遡るように誤差を伝播させて重みとバイアスを更新するが、このアルゴリズムは、**バックプロパゲーション**（誤差逆伝播法）と呼ばれる

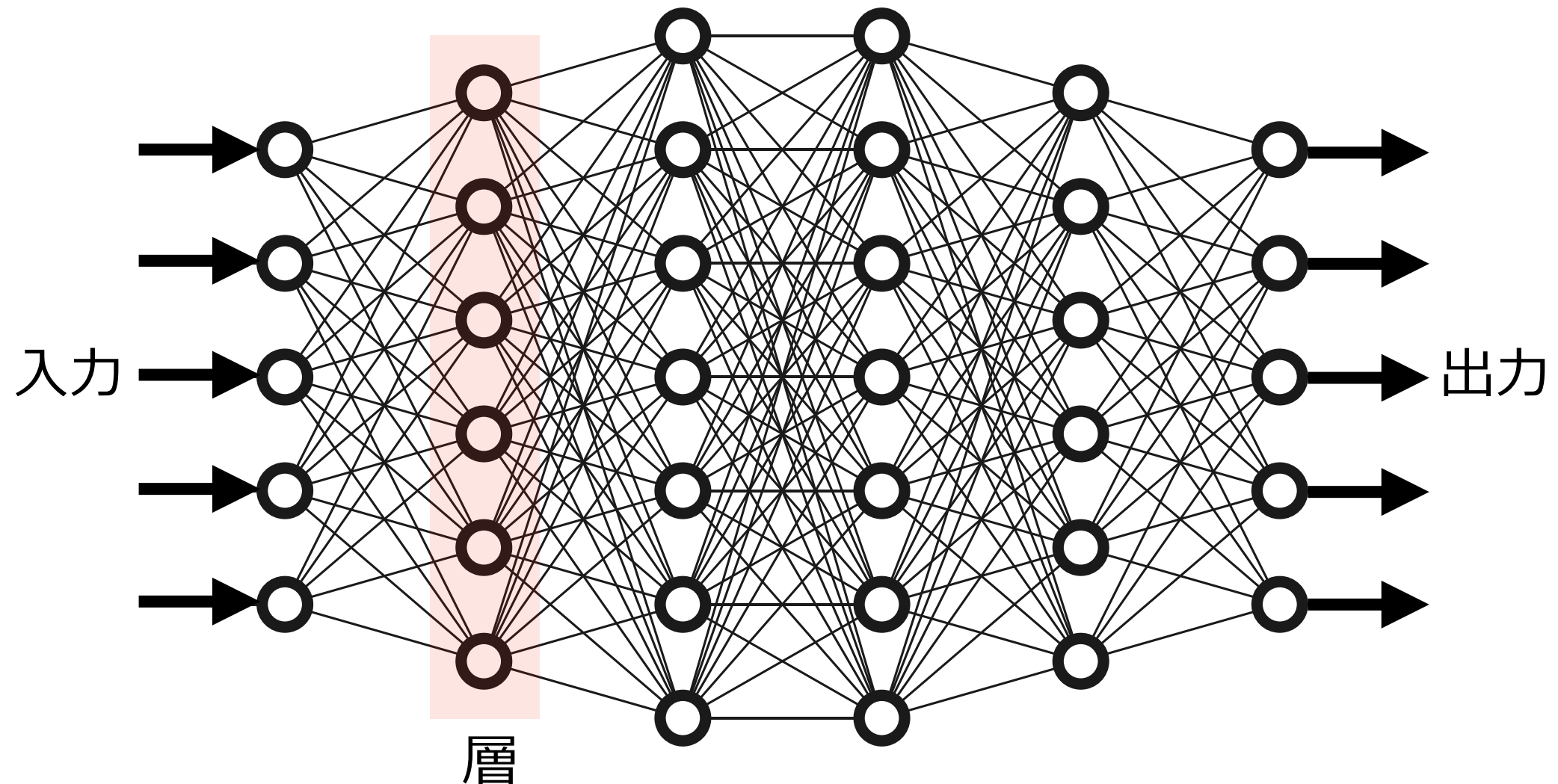


ディープラーニングとは？

- 多数の層からなるニューラルネットワークの学習のことを、**ディープラーニング**

(深層学習) と呼ぶ

- ヒトの知能に部分的に迫る
(あるいは凌駕する) 高い性能を
しばしば発揮する





ARTIFICIAL INTELLIGENCE

PyTorchとは

PyTorchの概要

- オープンソースの機械学習ライブラリ
- 海外を中心にコミュニティが活発で、ネット上の情報が豊富
- 簡潔な記述が可能なため、最新の研究成果の実装によく使われる
- Define by Run方式を採用し、データを流しながら
ネットワークの定義と演算を行う
- Numpyと類似した操作の**Tensor**クラスによりデータを扱う
- 簡潔さ、柔軟性、速度のバランス

PyTorchの人気

Googleトレンド

すべての国

<https://trends.google.co.jp/trends/explore?date=today%205-y&q=pytorch,keras,tensorflow>

アメリカ合衆国

<https://trends.google.co.jp/trends/explore?date=today%205-y&geo=US&q=pytorch,keras,tensorflow>

各モジュール

- **autograd**

→ Tensorの各要素による微分を自動で行う機能

- **optim**

→ 様々な最適化アルゴリズムを実装したモジュール

- **nn**

→ モデルを構築するためによく使われる層を集めたモジュール



ARTIFICIAL INTELLIGENCE

Google Colaboratoryの使い方

セッションとインスタンス

- セッション (Session)
 - ある活動を継続して行っている状態のこと
 - 接続を確立してから切断するまでの一連の通信のこと
- インスタンス (Instance)
 - 「ソフトウェアとして実装された仮想的なマシン」を起動したもの
 - オブジェクト指向におけるインスタンスとは異なる

90分ルール

- 90分ルール

→ ノートブックのセッションが切れてから90分経過すると、
インスタンスが落とされる

① 新しくノートブックを開く

② 新しいインスタンスが立ち上がる

③ ブラウザを閉じる、PCがスリープに入るなどして**セッションが切れる**

④ セッション切れから**90分**経過すると、インスタンスが落とされる

12時間ルール

- 12時間ルール

→新しいインスタンスを起動してから12時間経過すると、
インスタンスが落とされる

① 新しくノートブックを開く

② 新しいインスタンスが立ち上がる

③ その間、新しくノートブックを開いても同じインスタンスが使われる

④ **インスタンスの起動**から**12時間**経過すると、インスタンスが落とされる



ARTIFICIAL INTELLIGENCE

Tensorについて



ARTIFICIAL INTELLIGENCE

演習

次回

第1講. イントロダクション

 **第2講. PyTorchで実装する簡単なディープラーニング**

第3講. PyTorchの様々な機能

第4講. 畳み込みニューラルネットワーク (CNN)

第5講. 再帰型ニューラルネットワーク (RNN)

第6講. AIアプリのデプロイ



ARTIFICIAL INTELLIGENCE

質疑応答