

# Live! 人工知能

ARTIFICIAL INTELLIGENCE

## #4

# PyTorch + Deep Learning



ARTIFICIAL INTELLIGENCE

## 第4講の概要

# 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. 第4講の概要
2. CNNの概要
3. データ拡張
4. ドロップアウト
5. CNNの実装
6. 演習
7. 質疑応答

# 教材の紹介

- Pythonの基礎
- 第4講の教材: **cnn.ipynb**
- 第4講の演習: **exercise.ipynb**

# ハッシュタグ

#Live人工知能



# 演習の解答 -第3講-

[https://github.com/yukinaga/lecture\\_pytorch/blob/master/lecture3/exercise.ipynb](https://github.com/yukinaga/lecture_pytorch/blob/master/lecture3/exercise.ipynb)



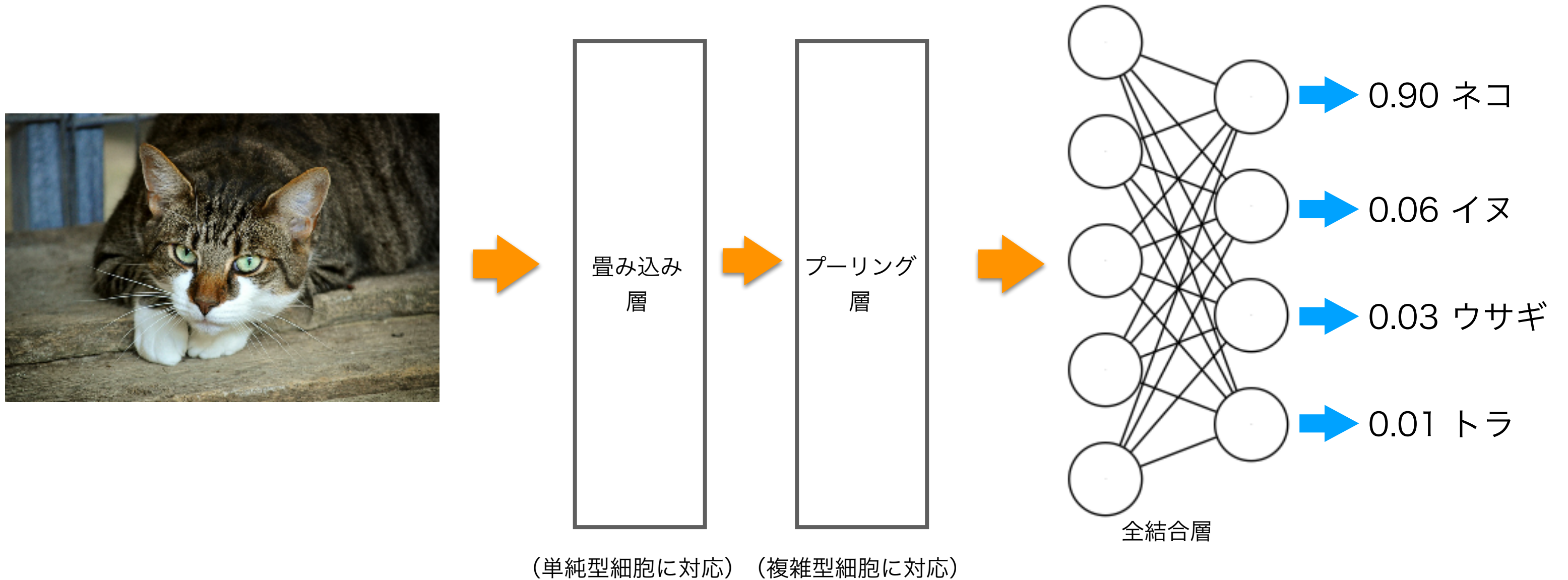
ARTIFICIAL INTELLIGENCE

# CNNの概要



# 畳み込みニューラルネットワーク (CNN) とは？

Covolutional Neural Network = CNN

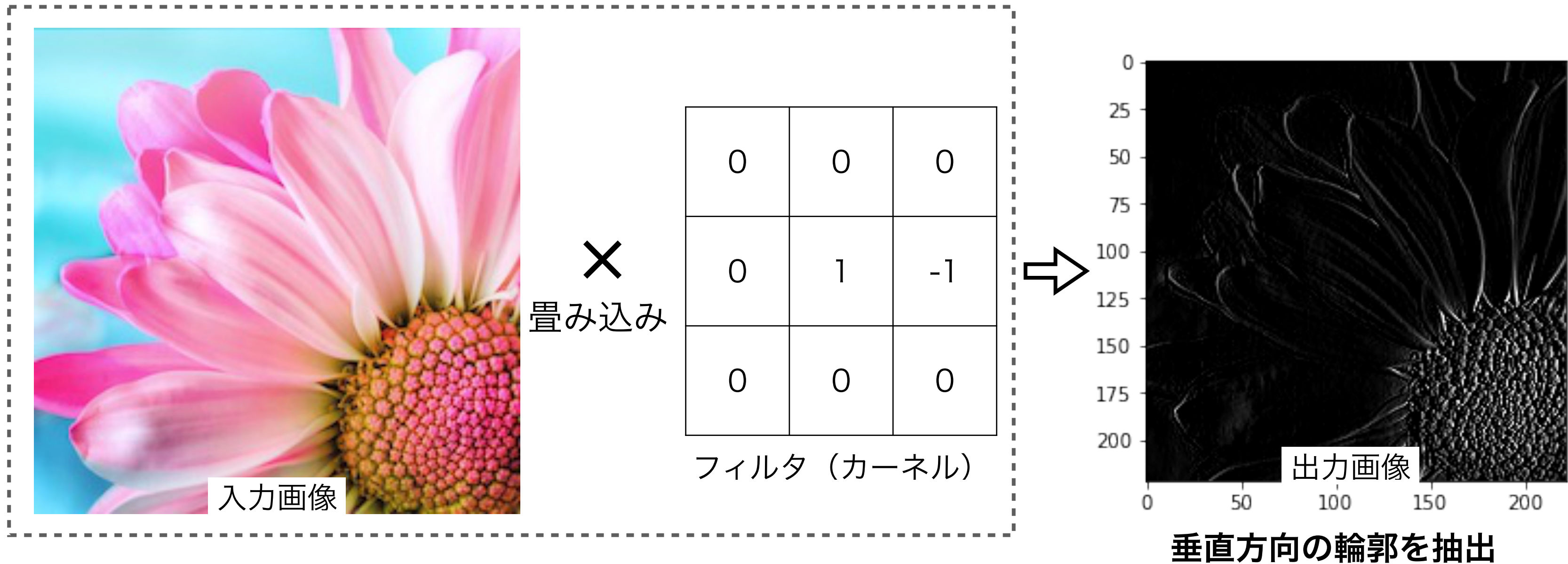


- 画像を入力とし、畳み込み層、プーリング層を経て全結合層へ
- 出力は各グループに分類される確率となる

# CNNで重要な要素

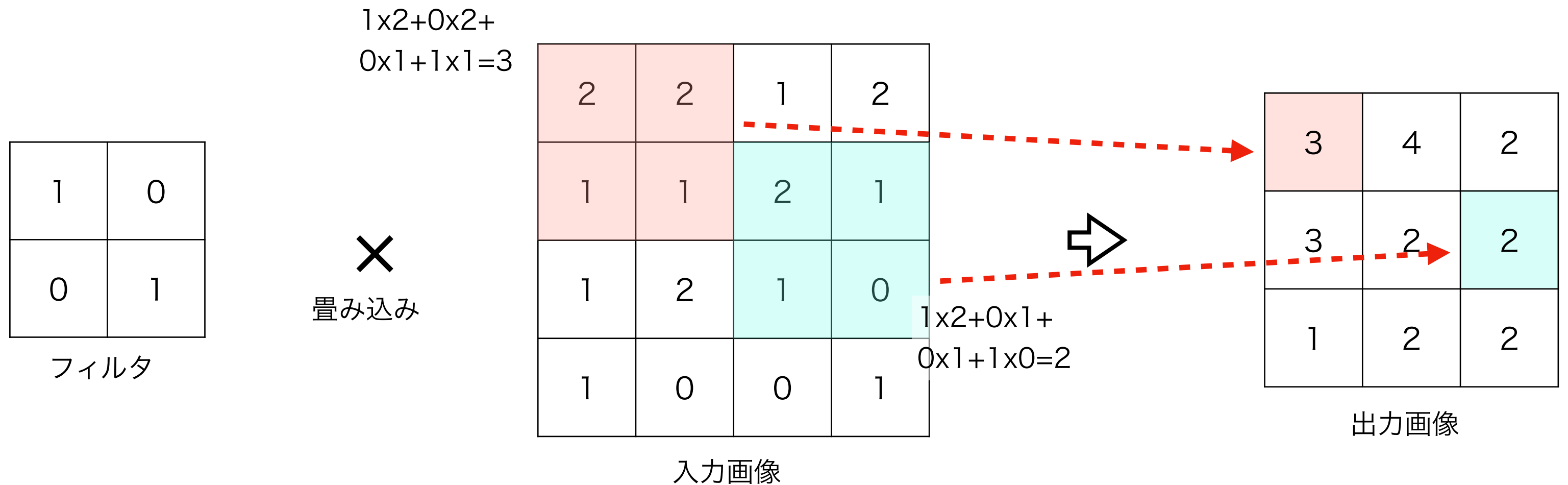
1. 畳み込み（畳み込み層）
2. プーリング（プーリング層）
3. パディング
4. ストライド

# 畳み込みとは？（畳み込み層）



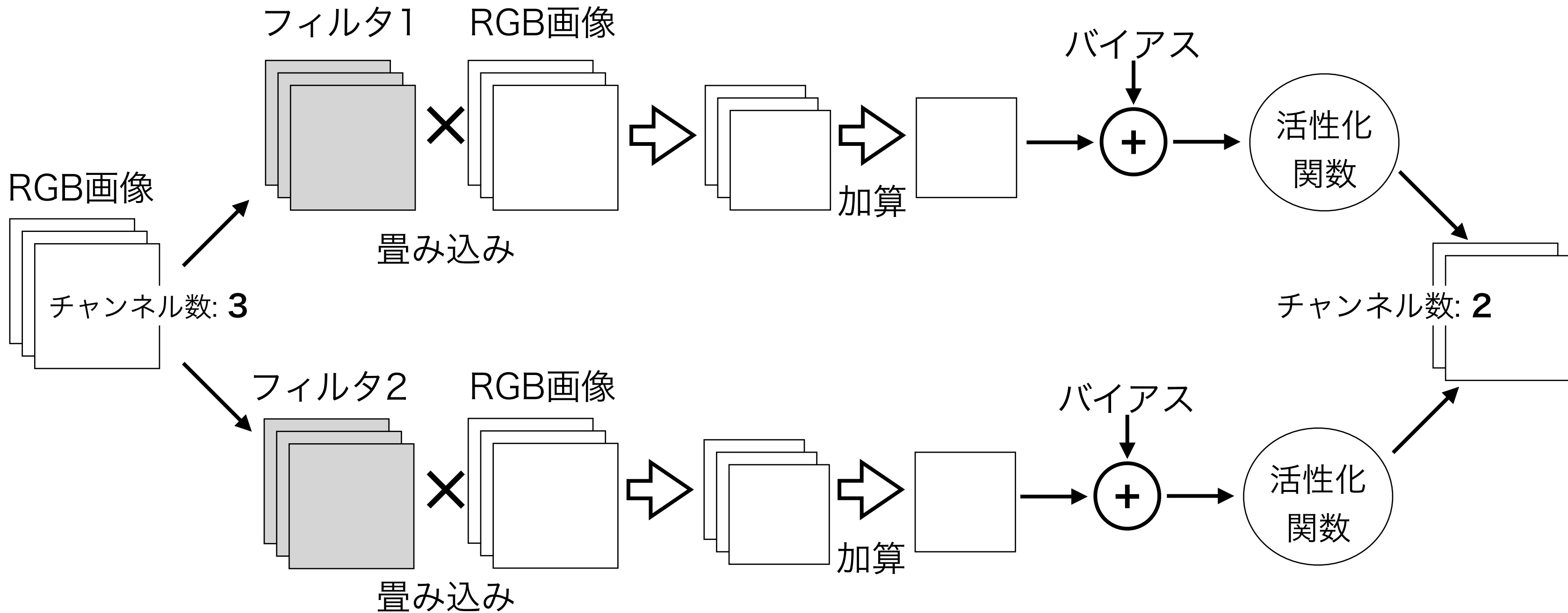
- 畳み込み層では、フィルタを用いて画像の特徴の抽出が行われる
- 上記の例では、畳み込みの結果垂直方向の輪郭が抽出される

# 畳み込みの計算



- フィルタにより、画像の局所的な特徴が抽出される
- フィルタの各値は学習により最適化される

# 畳み込みによるチャンネル数の変化



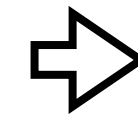
- 出力画像のチャンネル数は、フィルタの数に等しくなる



# プーリングとは？（プーリング層）

2	0	1	2	3	2
1	1	0	1	0	1
1	2	1	0	2	0
1	0	0	1	0	1
0	1	0	0	1	0
1	3	2	1	0	0

入力画像



各領域の最大値  
(MAXプーリング)

2	2	3
2	1	2
3	2	1

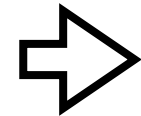
出力画像

- 画像の各領域を代表をする値（最大値など）を取り出して、出力画像にする
- 出力は、位置の微小なずれが吸収された、本質的な特徴を表す画像となる

# パディングとは？

2	2	1	2
1	1	2	1
1	2	1	0
1	0	0	1

入力画像

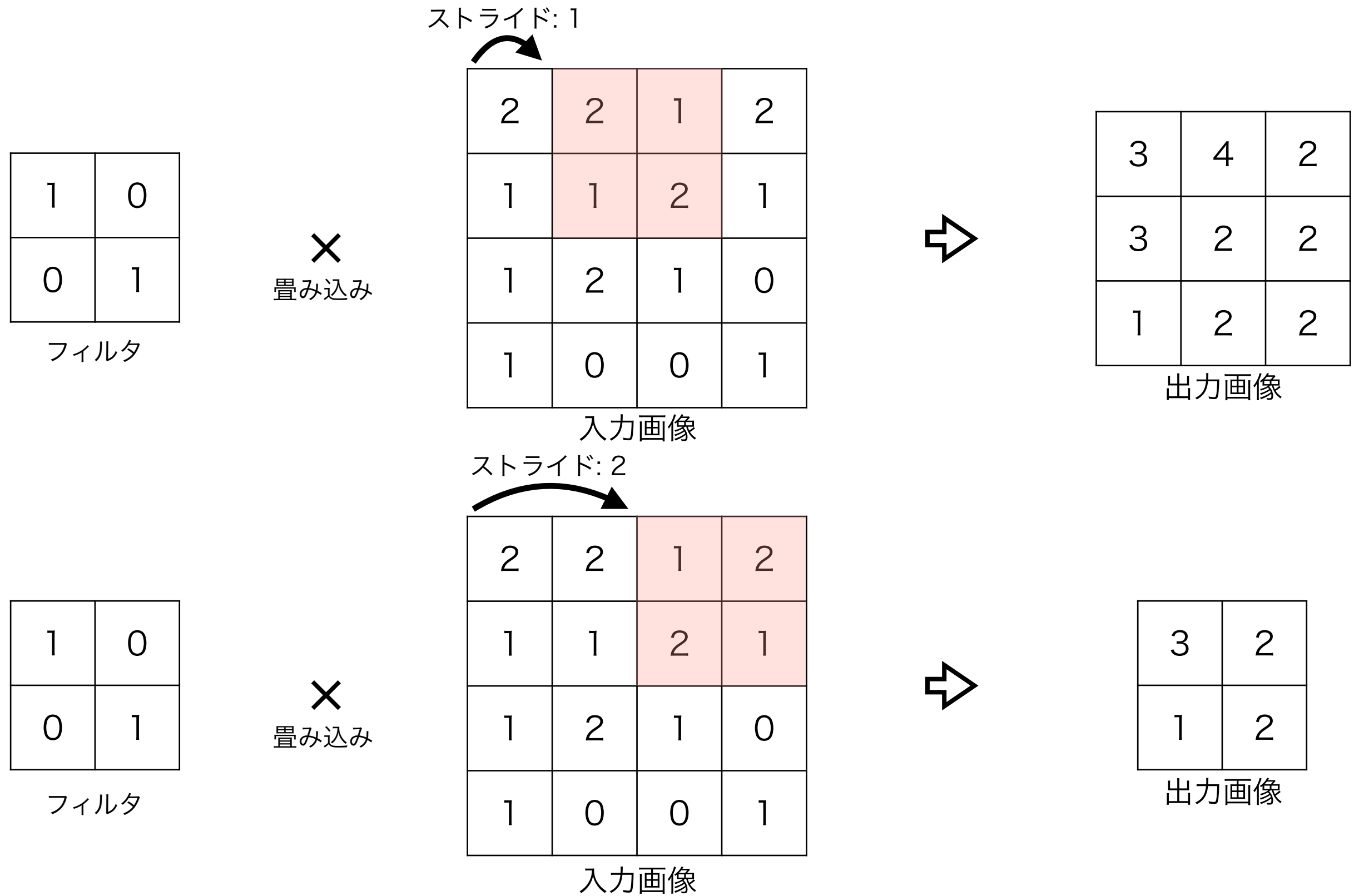


ゼロパディング

0	0	0	0	0	0
0	2	2	1	2	0
0	1	1	2	1	0
0	1	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

- 入力画像の周囲を0などの値で囲む

# ストライドとは？



# 出力画像のサイズ

$I_h$ : 入力画像高さ  $I_w$ : 入力画像幅

$F_h$ : フィルタ高さ  $F_w$ : フィルタ幅

$O_h$ : 出力画像高さ  $O_w$ : 出力画像幅

$D$ : パディング幅  $S$ : ストライド幅

$$O_h = \frac{I_h - F_h + 2D}{S} + 1$$

$$O_w = \frac{I_w - F_w + 2D}{S} + 1$$

# 畳み込み、プーリングの例

- 畳み込み層

- 入力画像: 32x32、RGB (3チャンネル)
- フィルタ: 5x5、フィルタ数は6
- パディングなし、ストライド1
- 出力画像:  $(32-5+2\times 0)/1+1 = 28$  なので 28x28、6チャンネル

- プーリング層

- 入力画像: 28x28、6チャンネル
- 領域サイズ: 2x2
- 出力画像:  $28/2 = 14$  なので 14x14、6チャンネル





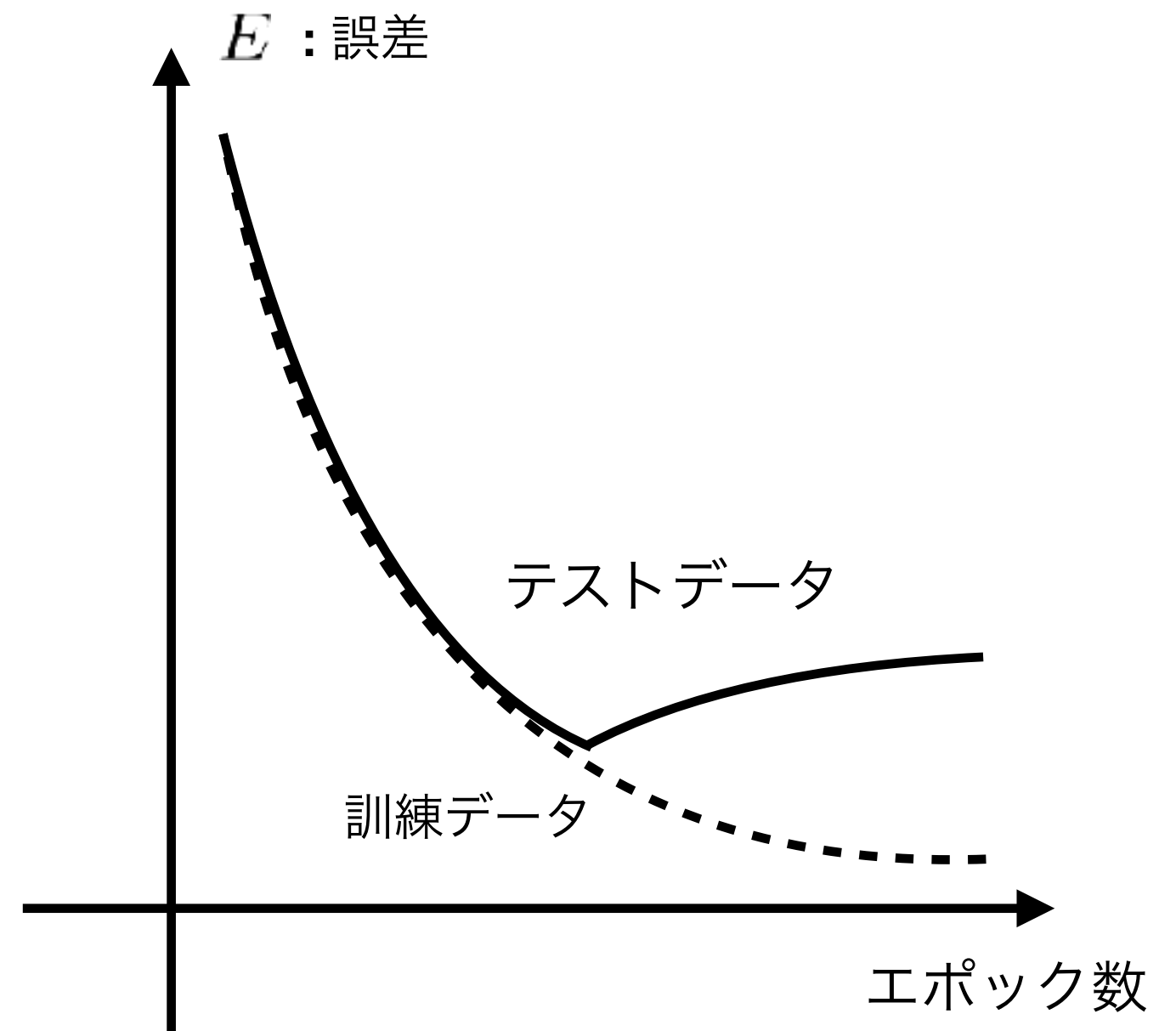
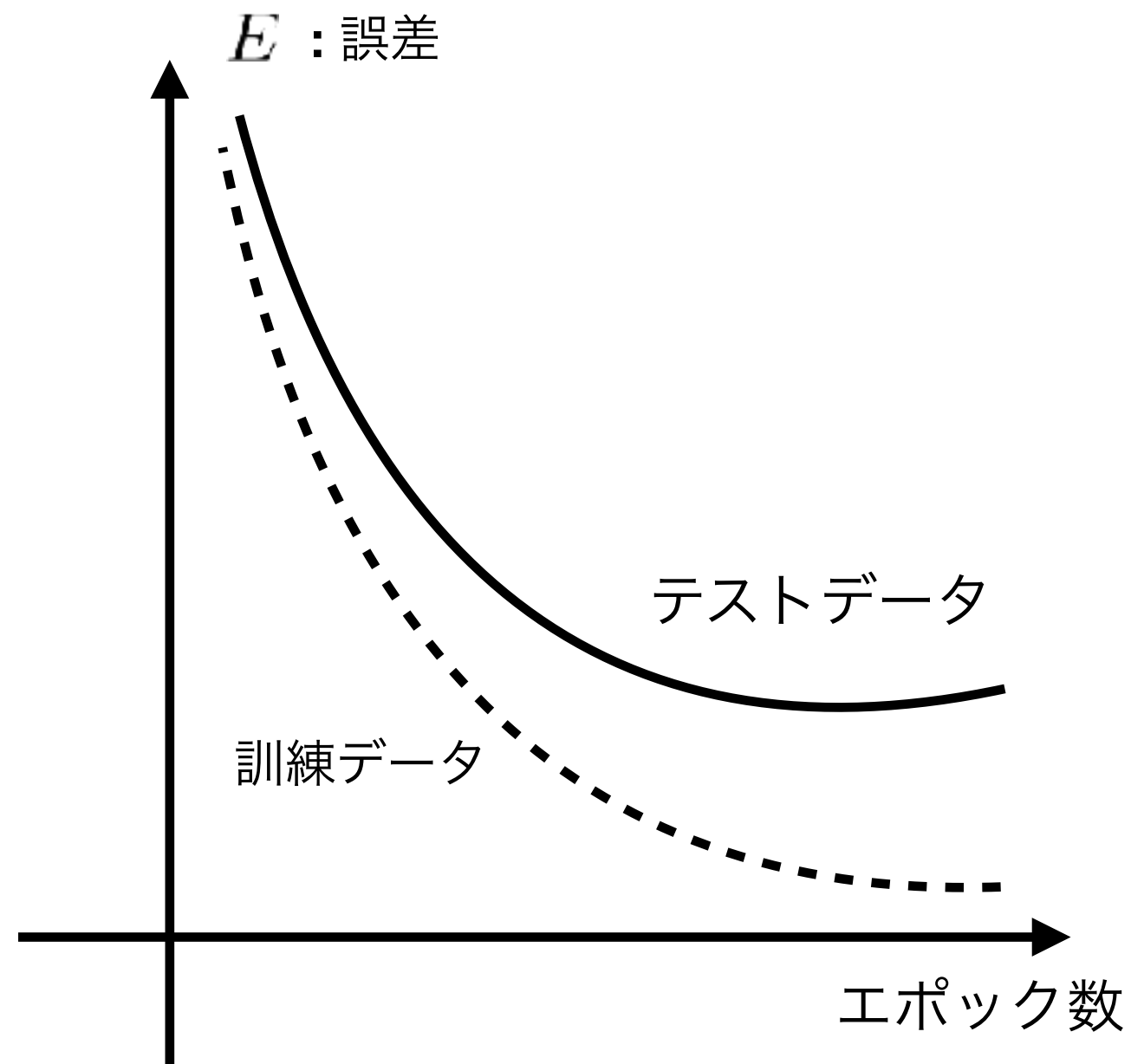
ARTIFICIAL INTELLIGENCE

データ拡張

# データ拡張とは？

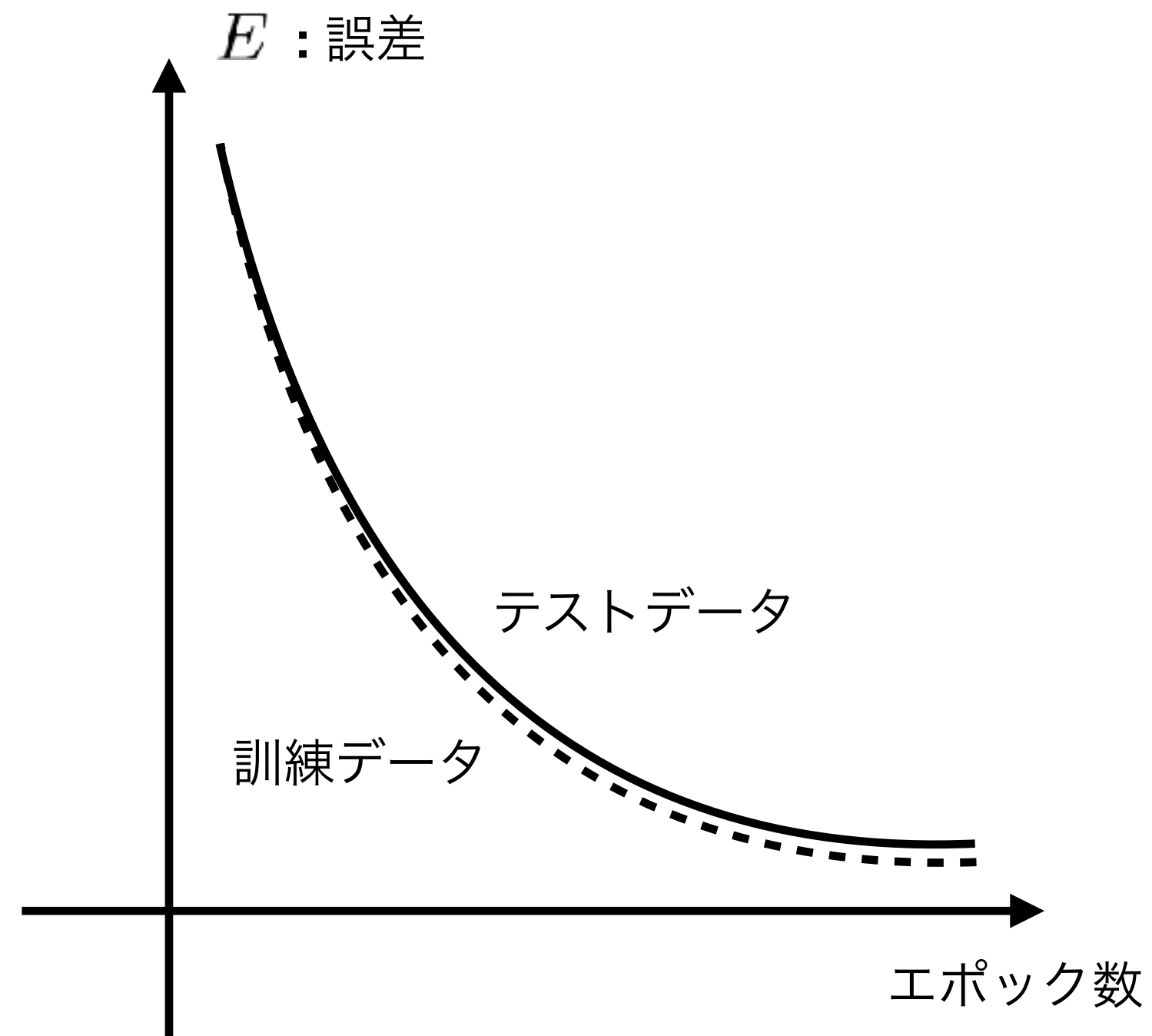
- データ拡張 (data augmentation)
  - 「過学習」は狭い範囲のデータにモデルが最適化されてしまい、モデルが汎用性を失ってしまうこと
  - 訓練データ内のサンプル数が少ないと、過学習が起きやすい
  - **データ拡張**ではデータに何らかの加工を加えることで、データを「水増し」することにより過学習を抑制する

# 過学習の例



- 過学習が発生すると、モデルが未知のデータに対して性能を発揮できない

# 理想的な学習



- モデルが汎用性を伴うため、未知のデータに対しても高い性能を発揮できる

# データ拡張の例

- 画像の回転
- 画像の拡大/縮小
- 画像を上下左右にシフト
- 画像を上下左右に反転
- 画像の一部を隠す
- etc...





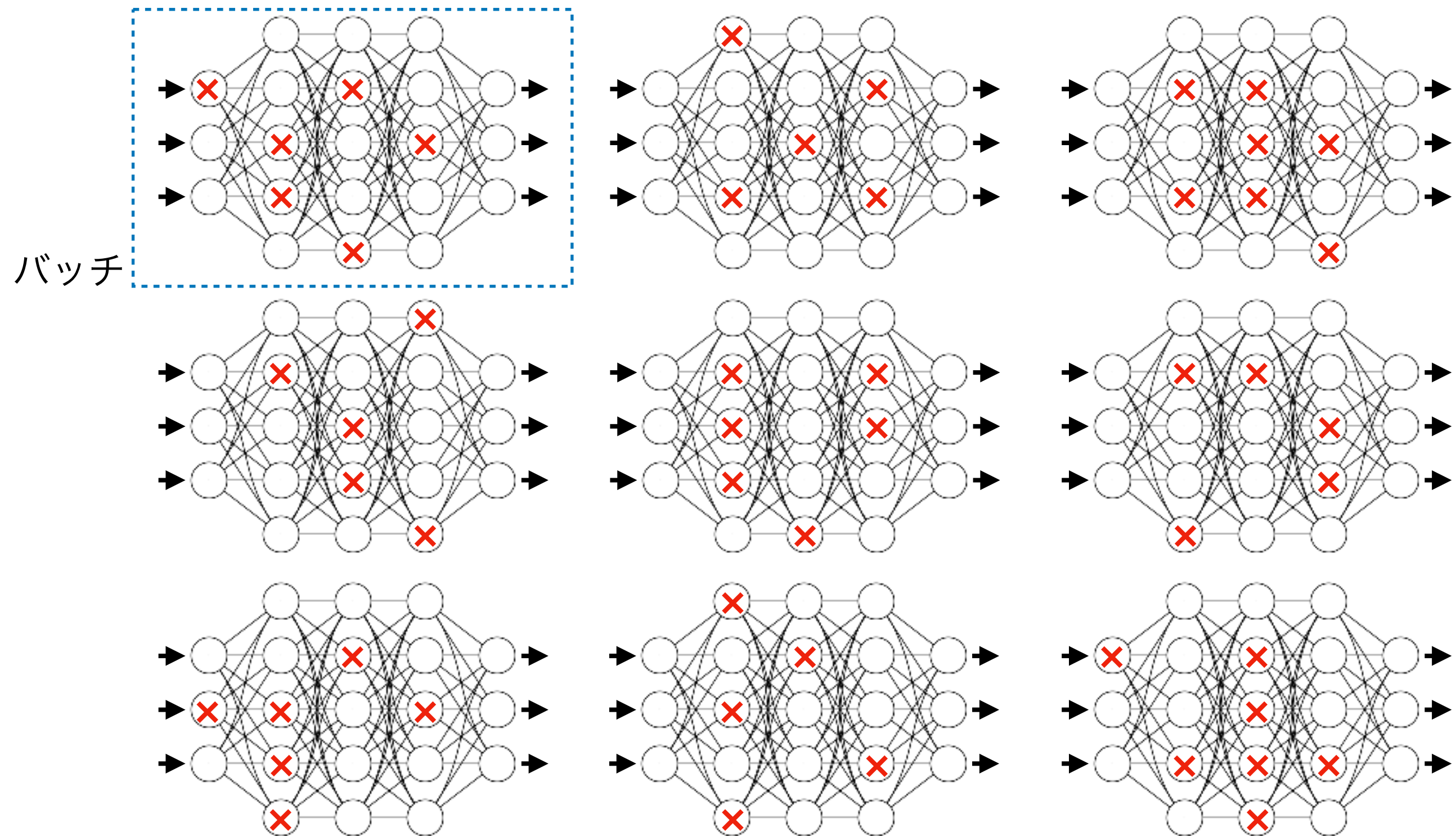
ARTIFICIAL INTELLIGENCE

ドロップアウト

# ドロップアウトとは？

- **ドロップアウト (dropout)**
  - 出力層以外のニューロンを一定の確率でランダムに消去するテクニック
  - 実装が容易でありながら、過学習を大きく抑制できる
  - 消去されるニューロンは、バッチごとに入れ替わる

# ドロップアウトのイメージ



- 実質的に、バッチごとに異なるニューラルネットワークを使用していることになる



ARTIFICIAL INTELLIGENCE

# CNNの実装





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