ae-7. 画像認識のための畳み込みニューラルネットワーク（CNN）入門 ：基礎概念から実装まで

（AI 演習）（全１５回）

https://www.kkaneko.jp/ai/ae/index.html

金子邦彦

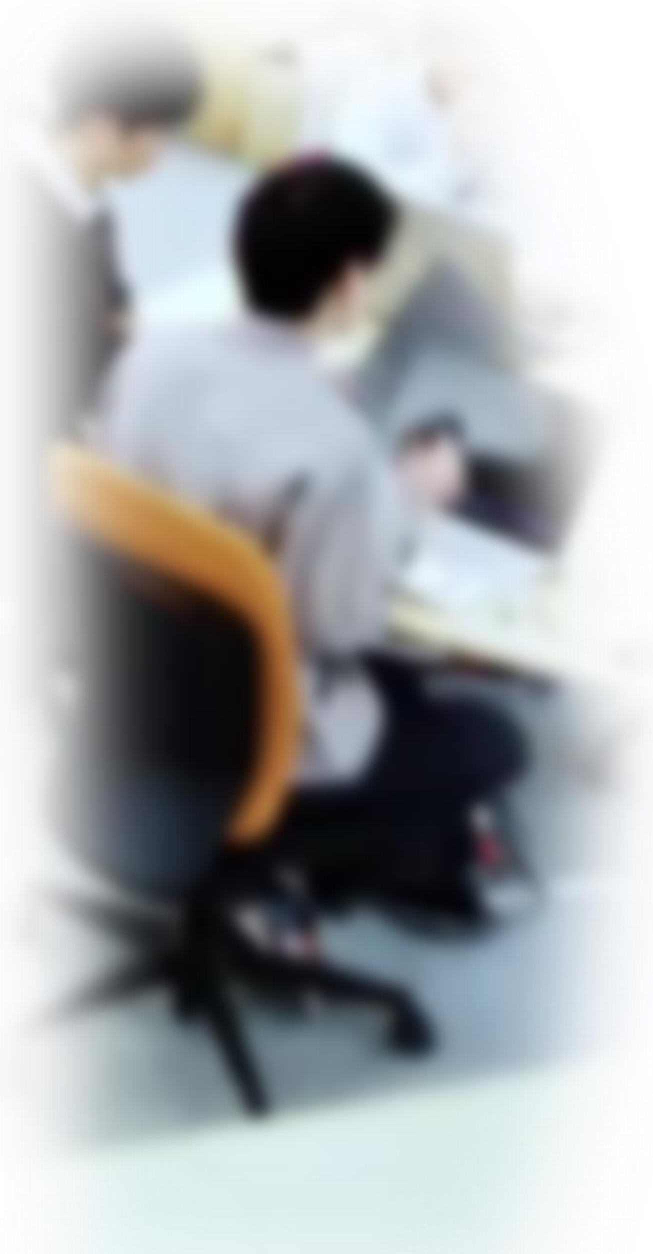
cc BY NC SA

人工知能（AI）の学習のための指針

1. **実践重視**： AIツールを実際に使用し、機能に慣れる

2. **エラーを恐れない**： 実行においては、エラーの発生の可能性がある．エラーを恐れず，むしろ学習の一部として捉えるポジティブさが大切.

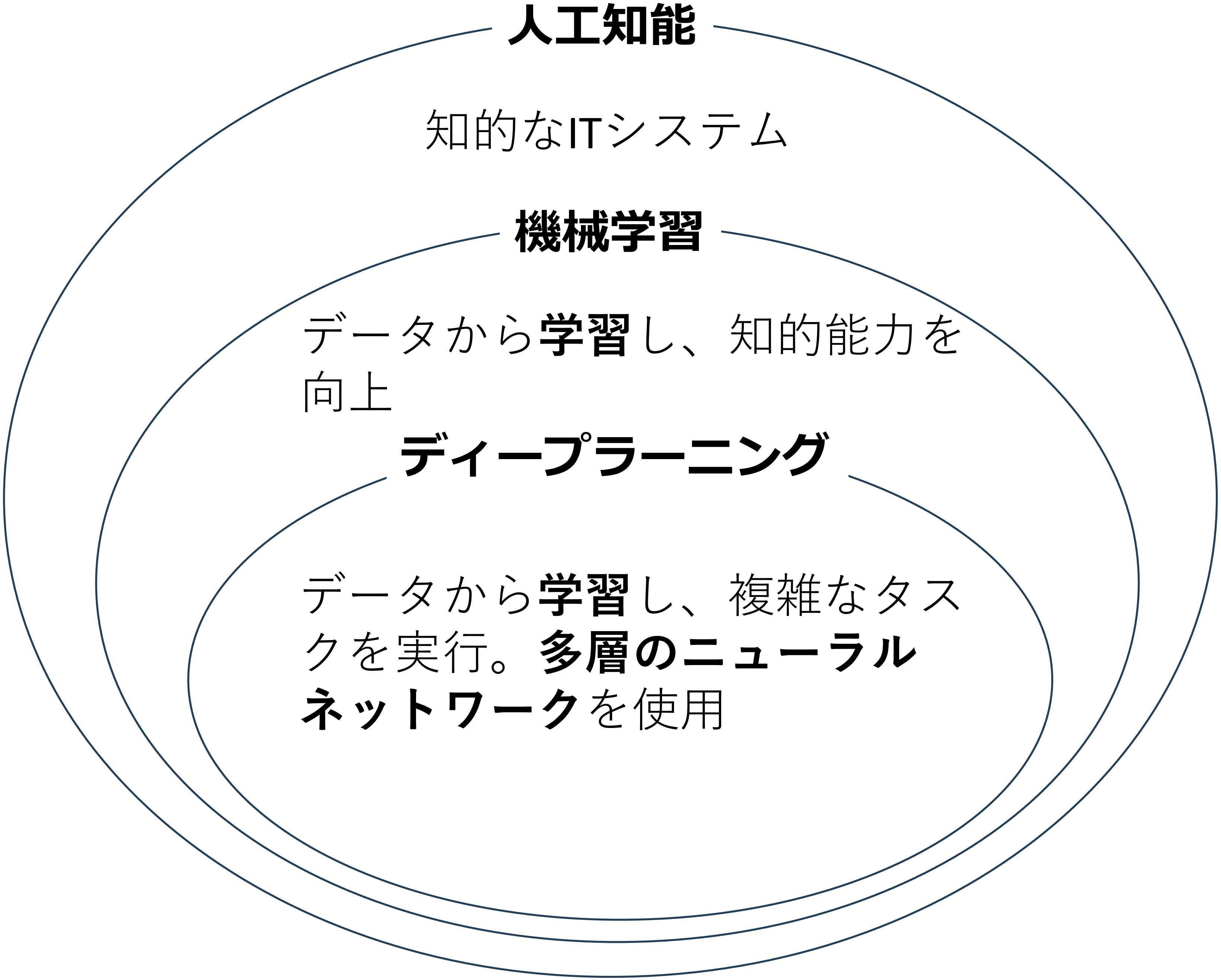
3. **段階的学習**：基礎から応用へと段階的に学習を進め，AIの可能性を前向きに捉える



アウトライン

1. イントロダクション
2. 画像データの扱い
3. Google Colaboratory の基礎
4. 畳み込みニューラルネットワーク(CNN)の基礎
5. 全結合層
6. 畳み込み層
7. プーリング層
8. CNN Explainer のデモ

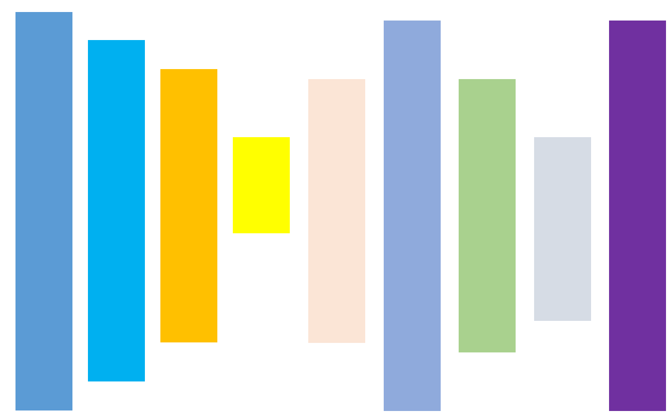
7-1. イントロダクション


人工知能
知的なITシステム
機械学習
データから**学習**し、知的能力を向上
ディープラーニング
データから**学習**し、複雑なタスクを実行。**多層のニューラルネットワーク**を使用

ディープニューラルネットワーク

ディープラーニングは**多層のニューラルネットワーク**を用いた機械学習

層の数が少ない（浅い）



層の数が多い（深い）

ディープニューラルネットワーク

- 層が浅い（**層の数が少ない**）**ニューラルネットワーク**を**組み合わせる**こともある

3層のニューラルネットワーク

3層のニューラルネットワーク

2層のニューラルネットワーク

合計で８層

（さまざまな組み合わせがありえる）

ディープラーニングが広く利用されている理由

- **多様なデータへの対応**

例：画像、テキスト、音声、動画など

- **広範な応用分野**

例：画像認識、自然言語処理、音声認識など

- **優れたパターン抽出能力**

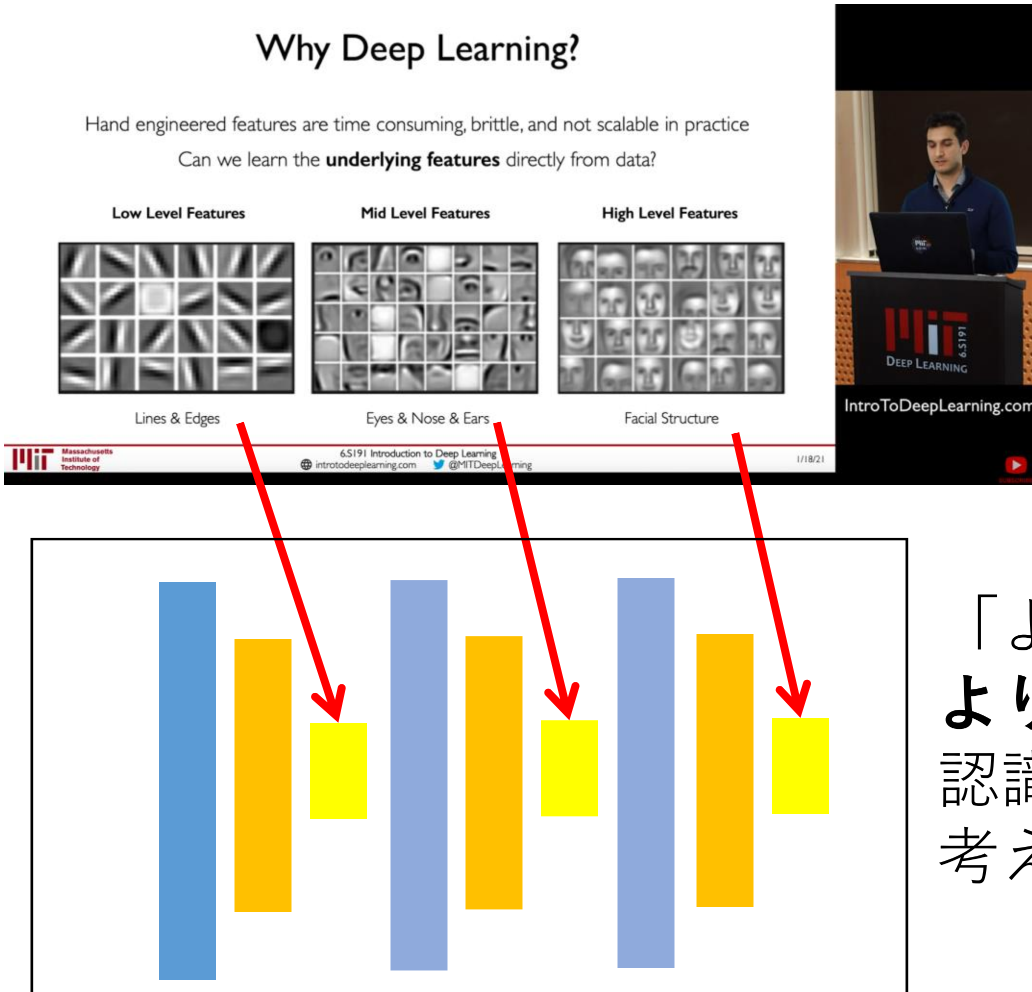
複雑なパターンを抽出する能力に優れる。

- **タスクの自動実行能力**

パターン抽出ならびにタスク実行は自動で行われる。

ディープラーニングへの期待

さまざまなレベルのパターンを**抽出・認識**できるようになる」という考える場合も



「より出力に近い層では，**より高次のパターン**を認識したい」という考え方も

ディープラーニングの応用と特徴

- **ディープラーニング**は**機械学習**の一種であり、人工**ニューラルネットワーク**を使用して**データから学習**し、複雑な**タスクを実行**する技術

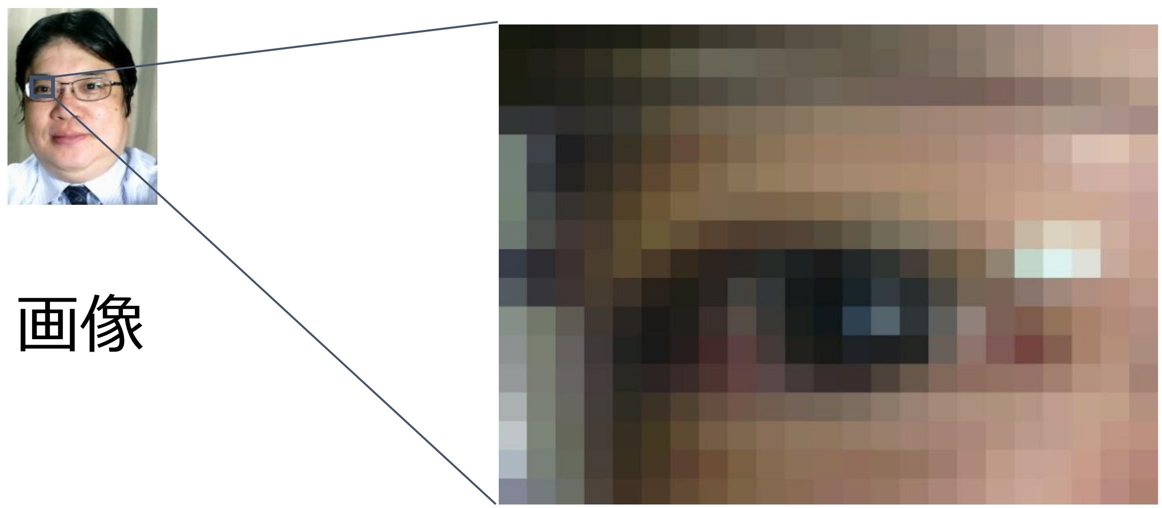
- 「ディープ」の名前は、**多層のニューラルネットワーク**を使用することに由来

- ディープラーニングが広く利用される理由は、**多様なデータに適用**でき、**さまざまなタスク**で高性能を発揮するため。例えば、**画像認識**、**自然言語処理**、**音声認識**など。

7-2. 画像データの扱い

画像と画素

- それぞれの格子が画素となる.
- 画像は画素の集合である.
- **画素**は**画像**を構成する**最小単位**である.



画像

それぞれの格子が画素

画像の種類

カラー画像

輝度と**色**の情報

濃淡画像

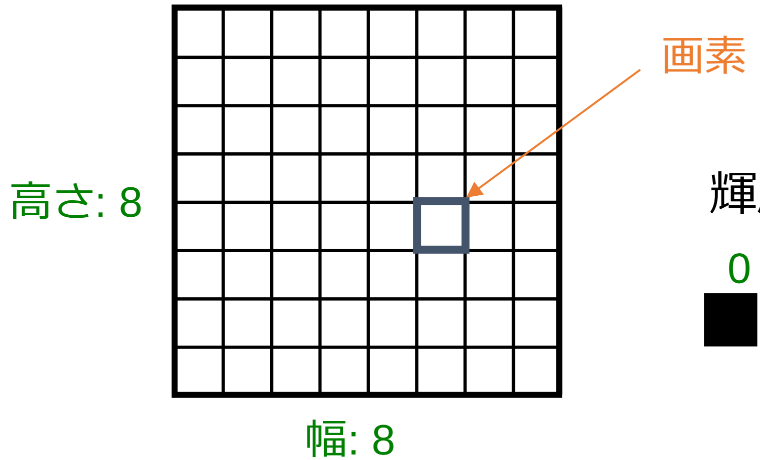
輝度のみの情報

濃淡画像でのコード化

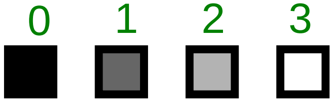
画像の輝度の情報

例えば：　黒　＝　0，
　　　　　暗い灰色　＝　1，
　　　　　明るい灰色　＝　2，
　　　　　白　＝　3

のように**コード化**



輝度が４段階の場合：



R（赤）成分，G（緑），B（青）成分によるカラー画像表現

R（赤）成分

G（緑）成分

B（青）成分

画素ごとに
１つの数値

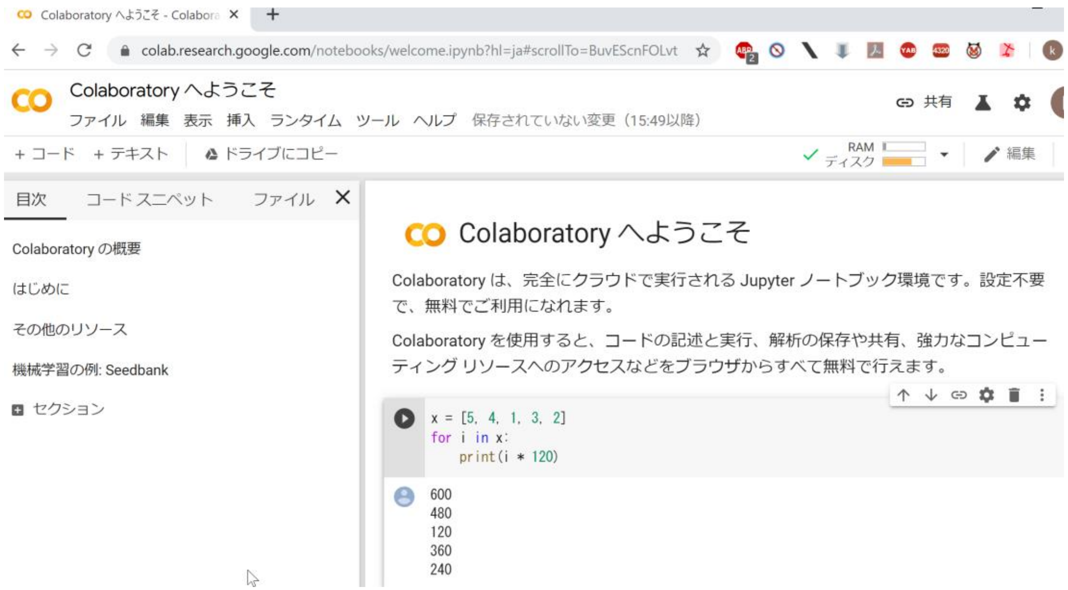
画素ごとに
１つの数値

画素ごとに
１つの数値

すべてあわせて，画素ごとに３つの数値

7-3. Google Colaboratory の基礎

Google Colaboratory (Colab)

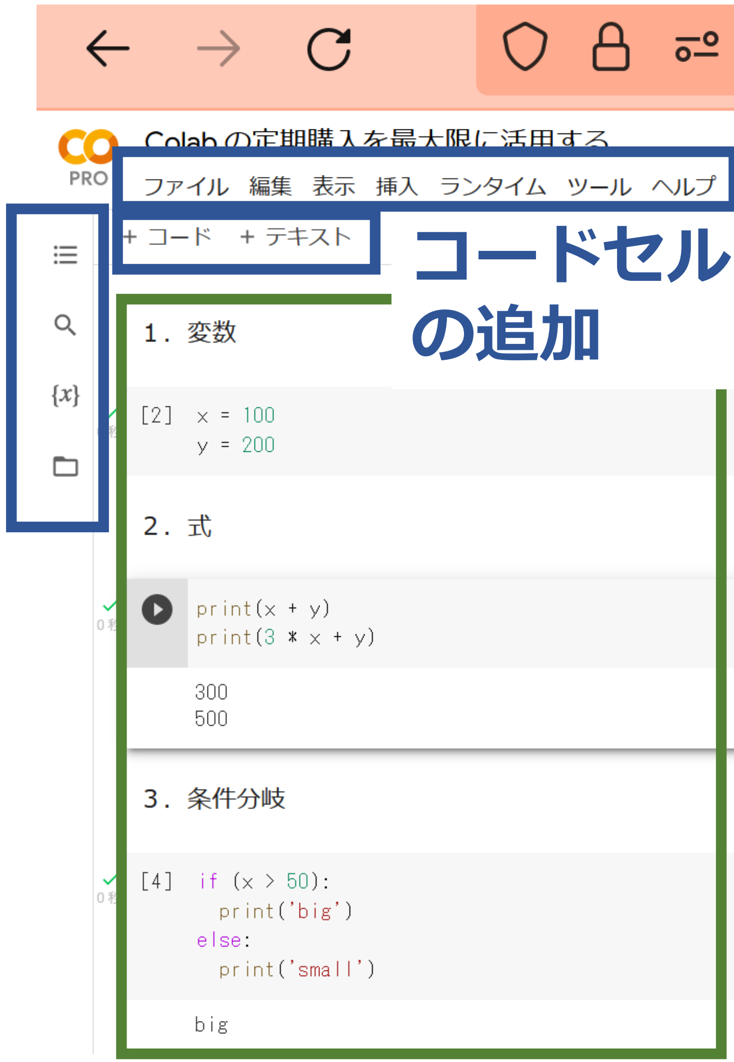


- **Google が提供**
- **クラウドベースの Python ノートブック**
- Pythonプログラミング（特に、人工知能、データサイエンス）の開発に適する
- **本格的な利用**には，**Google アカウントが必要**

Google Colaboratory の全体画面

メニュー

（目次，検索と置換，変数，ファイル）



メニュー

コードセル，テキストセルの追加

Pythonノートブックの部分

コードセル，テキストセルの並び

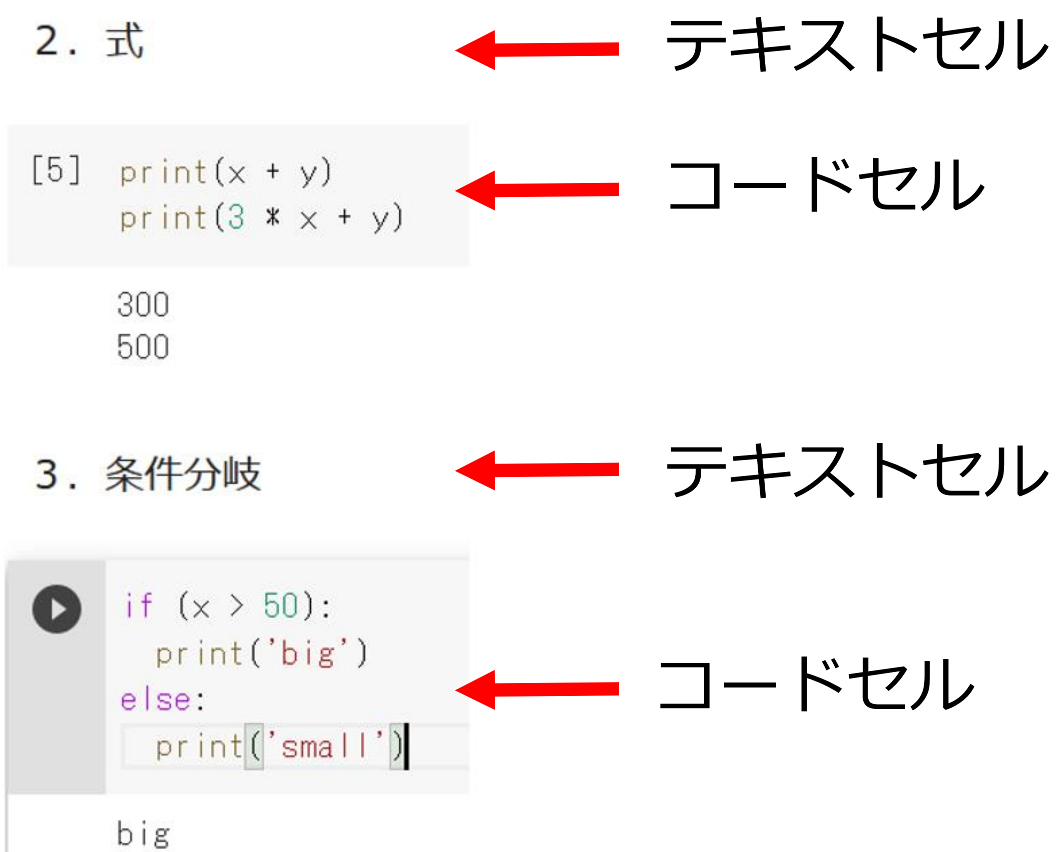
Web ブラウザの画面

Python ノートブック

コードセル，**テキストセル**の２種類

- **コードセル**： Python プログラム，コマンド，実行結果
- **テキストセル**：説明文，図

２種類のセルを組み合わせて使用



演習１

カラー画像の画素

【トピックス】

カラー画像

画素

配列

① Google Colaboratory のページを開く

https://colab.research.google.com/drive/1MMhlrh08-0Byq-U1JlTdDVwMe6dyUcaq?usp=drive_link

（このページは、演習１，２で使用する）

② **次**について、プログラムや説明や実行結果が掲載されていることを確認。各自でよく読む。

１．カラー画像の画素

１．カラー画像の画素

【プログラムの説明】

image_data は3x3ピクセルのカラー画像を模倣したもので、RGB形式で色が保存されています。

特定の位置 (x, y) の画素の色を取得するためのコードも示しています。

```
# カラー画像の画素例（RGB形式）
# 例：3x3ピクセルのカラー画像
image_data = [
    [(255, 0, 0), (0, 255, 0), (0, 0, 255)],
    [(0, 255, 255), (255, 0, 255), (255, 255, 0)],
    [(128, 128, 128), (64, 64, 64), (32, 32, 32)]
]

# 特定の画素の色を取得
x, y = 1, 1
pixel_color = image_data[y][x]
print(f"Pixel color at ({x}, {y}): {pixel_color}")
```

```
Pixel color at (1, 1): (255, 0, 255)
```

自習

目的：配列になっている**画像データ**について、**特定の位置の画素の色を取得**する方法を学び、理解を深めること。

指示：異なる (x, y) の位置の画素を取得して、期待通りであるか確認してみよう

解答例：

x, y = 2, 2

のときは次のように表示される

Pixel color at (2, 2): (32, 32, 32)

```
# 特定の画素の色を取得
x, y = 2, 2
pixel_color = image_data[y][x]
print(f"Pixel color at ({x}, {y}): {pixel_color}")
```

```
Pixel color at (2, 2): (32, 32, 32)
```

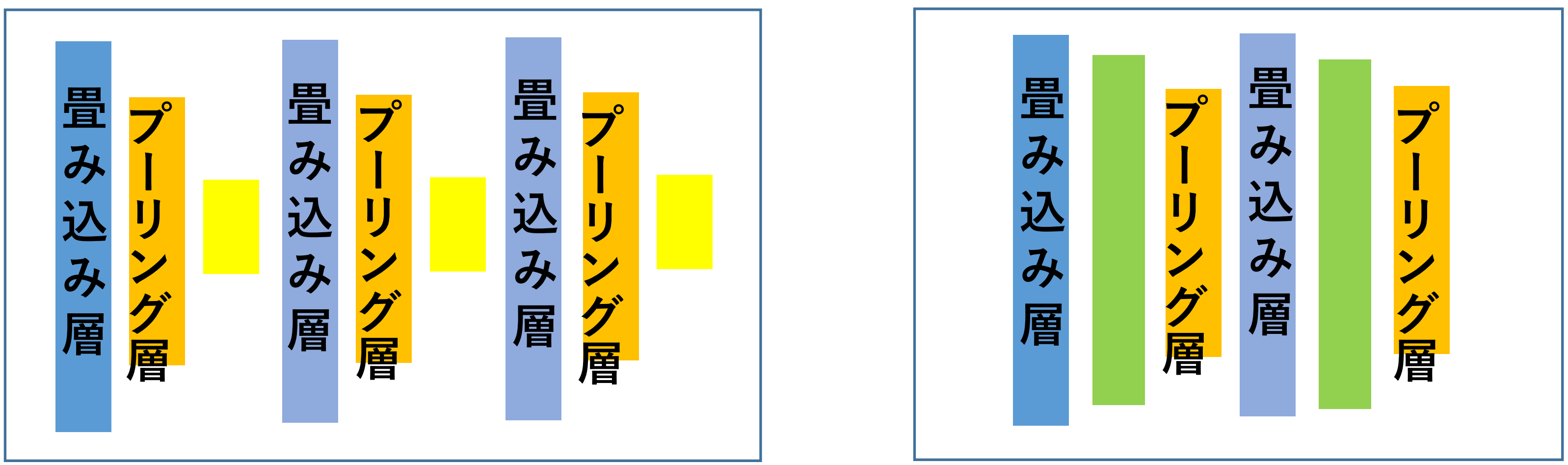
7-4. 畳み込みニューラルネットワーク(CNN)の基礎

畳み込みニューラルネットワーク（CNN）の基本構造

- **画像理解**や**画像の分析**に特化した**ディープラーニング**の一種.
- 主に**畳み込み層**、**プーリング層**、**全結合層**の３種類で構成
  - **畳み込み層**：画像の局所的な**特徴**（エッジ、テクスチャなど）をとらえる
  - **プーリング層**：**特徴マップ**の**サイズを縮小**し，**過学習を防止.**
  - **全結合層**：**特徴マップ**をもとに**画像**を処理する

畳み込みニューラルネットワーク（CNN）の例

- CNNの主要な層構成：
  - **プーリング層**
  - **畳み込み層**
  - **全結合層**
- これらの層を組み合わせることで，さまざまなバリエーションのCNNを構築できる．



さまざまなバリエーション

畳み込みニューラルネットワークでのパターン認識

「**畳み込みニューラルネットワークの利用により，さまざまなレベルのパターン**を**抽出・認識**できるようになる」という考える場合も

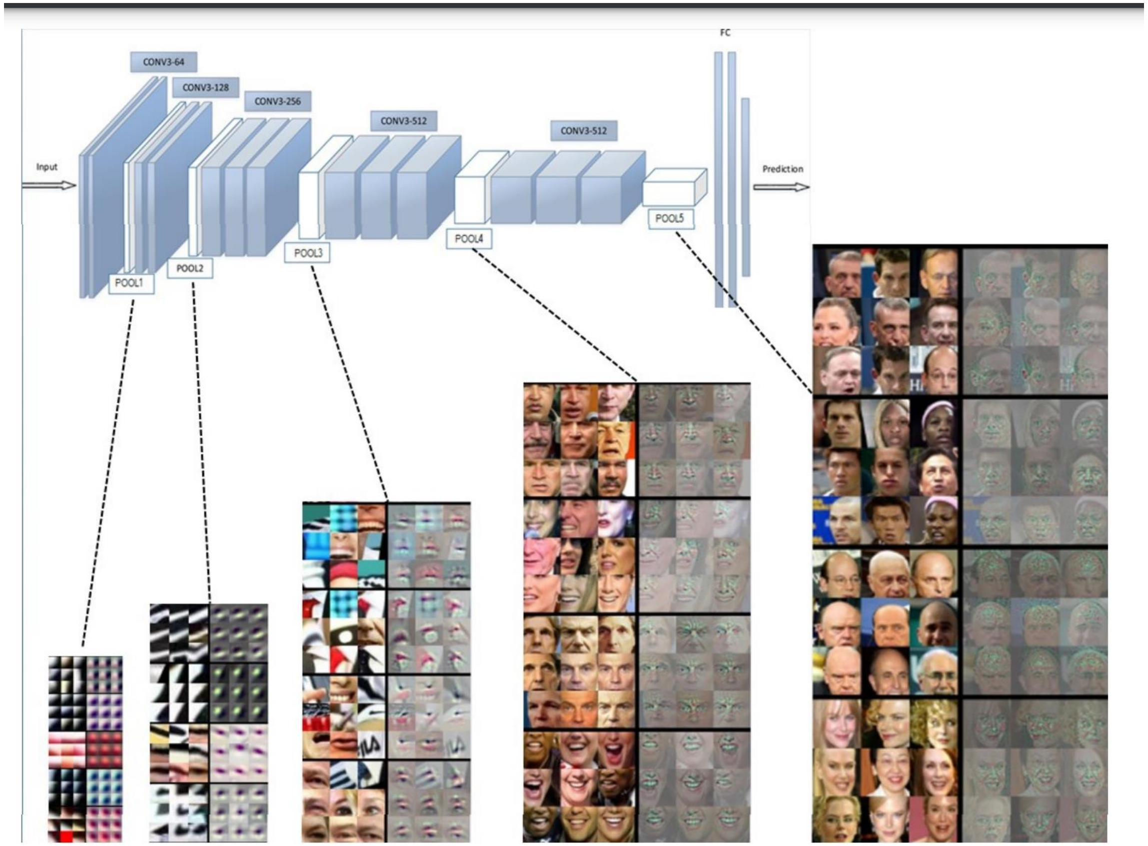


Fig. 2

Mei Wang, Weihong Deng,

Deep Face Recognition: A Survey, arXiv:1804.06655, 2018.

畳み込みニューラルネットワーク（CNN）のメリット

- 画像理解、画像の分析の**高精度**化
- **画像の特徴**を自動的に**抽出**
- 幅広い分野での**応用**

畳み込みニューラルネットワーク（CNN）の特徴

	畳み込みニューラルネットワーク	一般のニューラルネットワーク
局所的な特徴の抽出能力	画像の局所的な特徴をとらえる**畳み込み層を持つ**	畳み込み層は**ない**
仕組み	主に**畳み込み層、プーリング層、全結合層**などから構成	主に全結合層から構成
パラメータ	学習では、重み、バイアスなどのパラメータを調整。ただし、**畳み込み層**では、**カーネルを共有する**ため、学習が効率化	学習では、重み、バイアスなどのパラメータを調整

畳み込みニューラルネットワーク（CNN）の構成と特徴

畳み込みニューラルネットワーク（CNN）の構成

- **畳み込み層**: 局所的特徴抽出
- **プーリング層**: 特徴マップの生成，過学習防止
- **全結合層**: 特徴マップを用いた画像処理

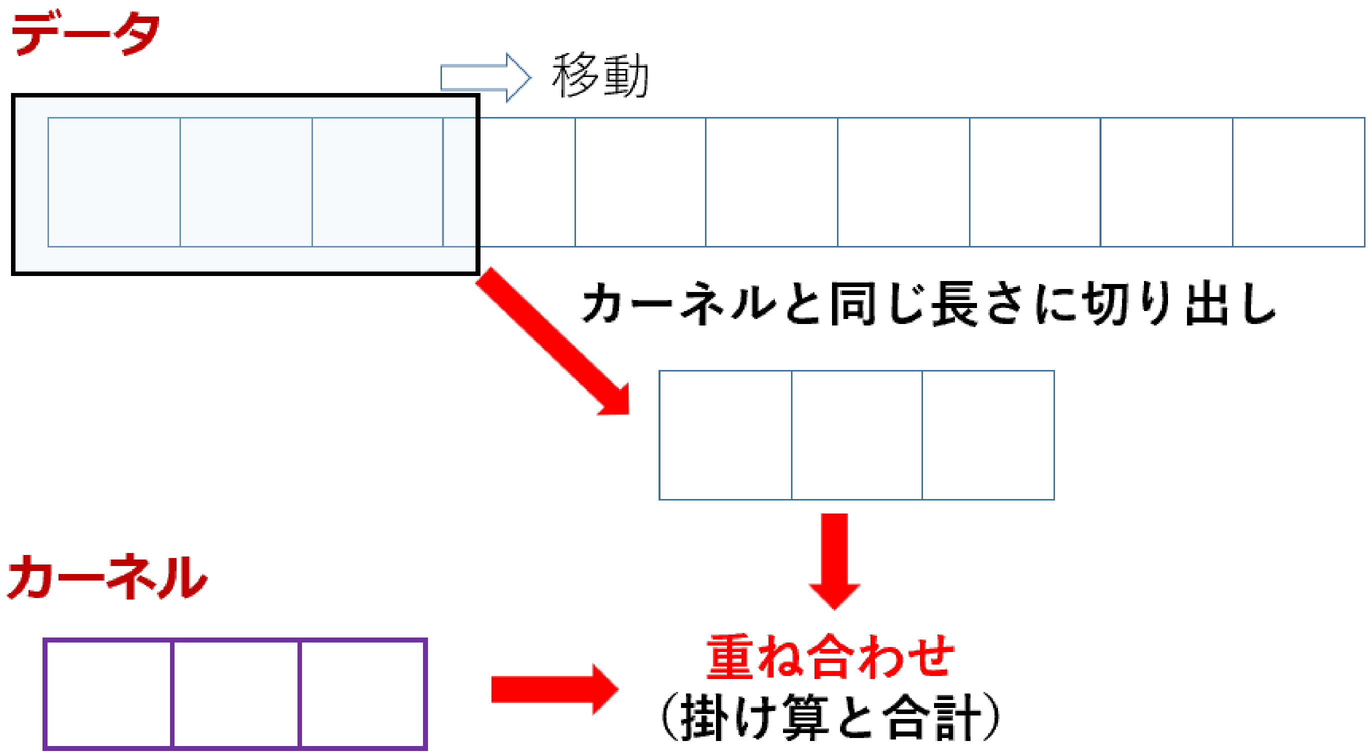
畳み込みニューラルネットワーク（CNN）の特徴

- 局所的特徴抽出
- 畳み込み層の**カーネル共有**で学習効率向上

6-5. 畳み込み

畳み込みの役割

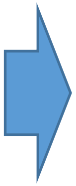
畳み込みは，**データ上を移動**しながら，**カーネル**との**重ね合わせ**を行い，**局所的特徴抽出**を行う



畳み込みの応用例

- ぼかし，エッジ抽出，テクスチャ分析などさまざまな処理

元画像

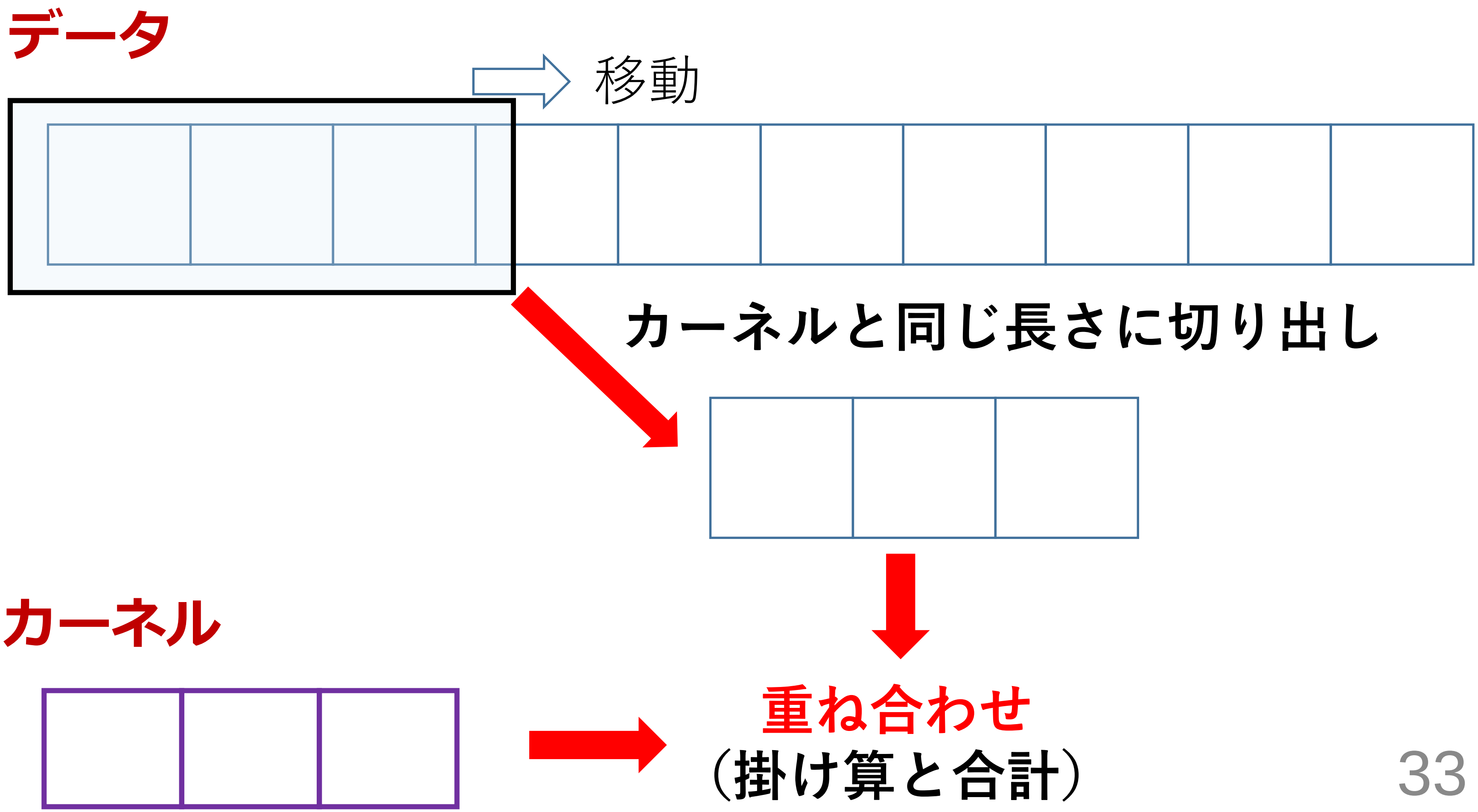


畳み込みによるぼかし

畳み込みによるエッジ抽出

畳み込みの仕組み

① **データ上を移動**しながら処理
② データをカーネルと同じ長さに切り出し
③ **カーネルとの重ね合わせ**を実行（掛け算と合計）



データの畳み込みの例

データ

0	1	0	1	0	0	1	1	1	0

この部分を切り出す

カーネル

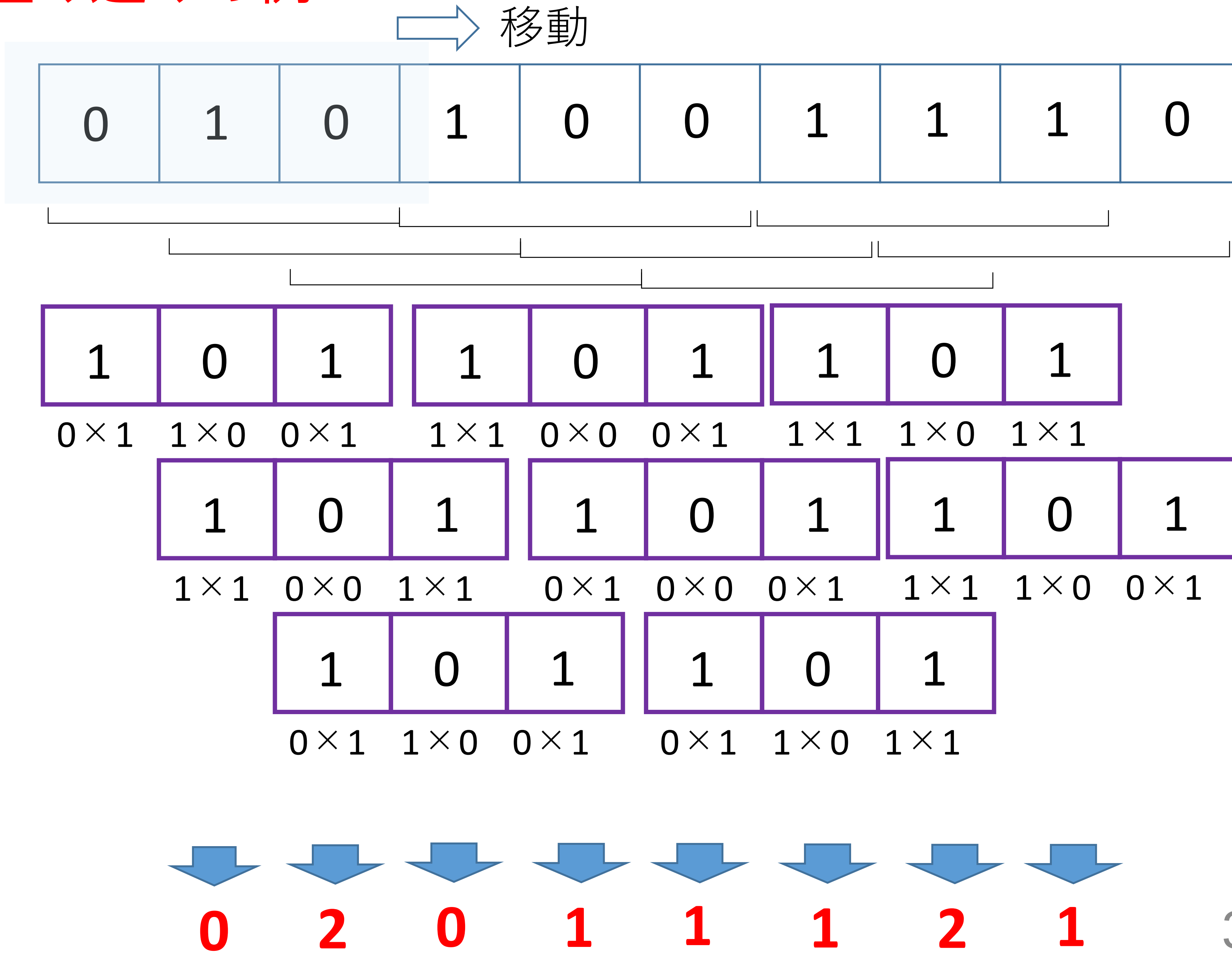
1	0	1

0×1　1×0　0×1

0

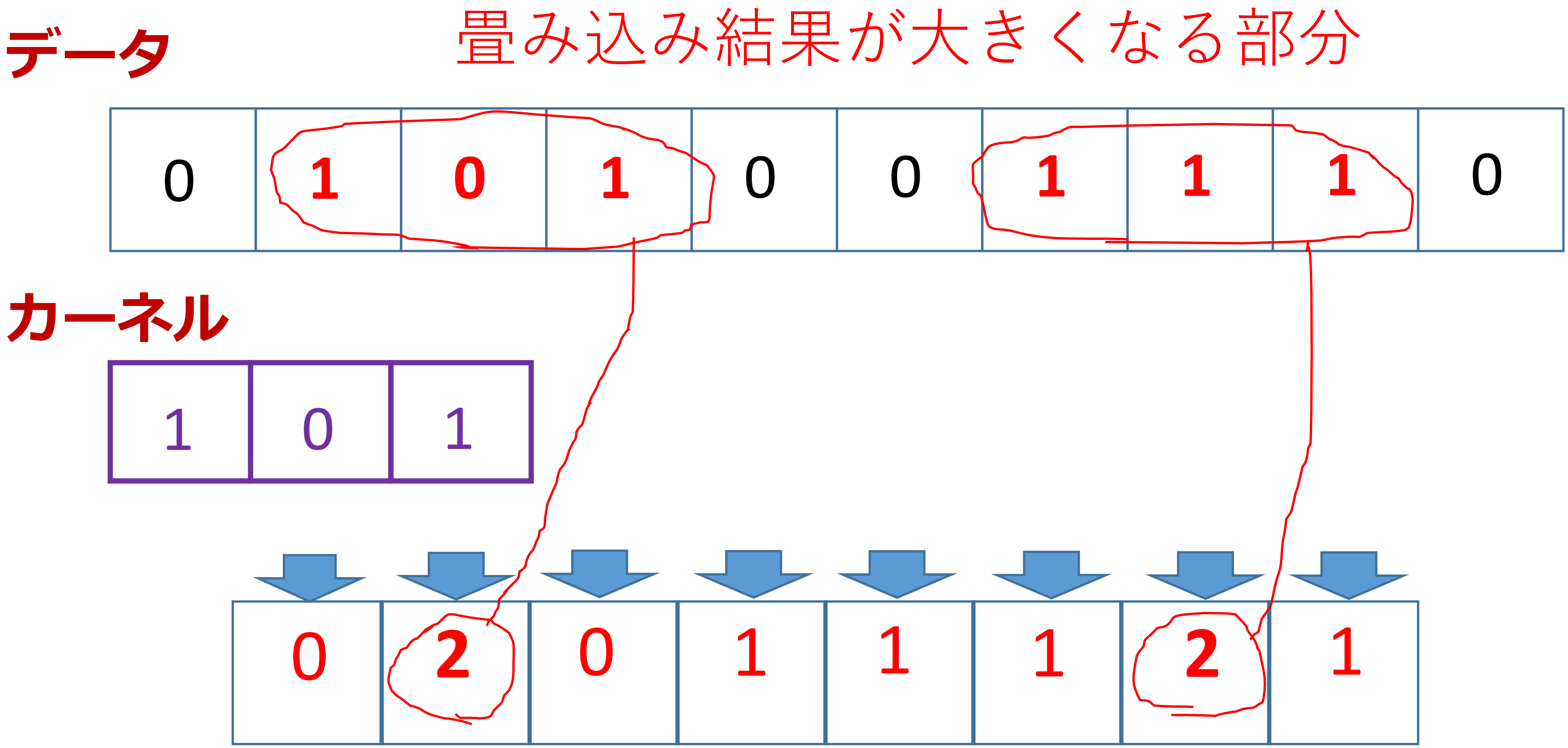
重ね合わせの結果：　0×1 + 1×0 + 0×1 = 0

畳み込みの例


移動
0 1 0 1 0 0 1 1 1 0
1 0 1 1 0 1 1 0 1
0×1 1×0 0×1 1×1 0×0 0×1 1×1 1×0 1×1
1 0 1 1 0 1 1 0 1
1×1 0×0 1×1 0×1 0×0 0×1 1×1 1×0 0×1
1 0 1 1 0 1
0×1 1×0 0×1 0×1 1×0 1×1
0 2 0 1 1 1 2 1

畳み込みによる局所的特徴抽出

畳み込みは，「**特定の局所的特徴に強く反応する**」と考えることもできる



畳み込み結果

画像の畳み込み

Input

0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1

元画像（5 × 5 マス）

Filter / Kernel

1	0	1
1	1	1
0	0	1

カーネル（3 × 3 マス）

出典: https://serokell.io/blog/introduction-to-convolutional-neural-networks

画像での畳み込み

0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1

元画像（５ × ５ マス）

1	0	1
1	1	1
0	0	1

カーネル
（３ × ３ マス）

0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1

切り出し（３ × ３ マス）

カーネルと同じサイズで切り出す

切り出した部分とカーネルの掛け算の合計

0×**1**	**1**×**0**	**1**×**1**
0×**1**	**1**×**1**	**1**×**1**
0×**0**	**1**×**0**	**1**×**1**

合計: 4 （これが畳み込み結果）

畳み込み

画像での畳み込み

0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1

元画像（5 × 5 マス）

1	0	1
1	1	1
0	0	1

カーネル
（3 × 3 マス）

0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1

切り出し（3 × 3 マス）

切り出し領域を横にずらす

0×1	1×0	1×1
0×1	1×1	1×1
0×0	1×0	1×1

合計: 4

1×1	1×0	0×1
1×1	1×1	0×1
1×0	1×0	0×1

合計: 3

4	3	

畳み込み結果

畳み込み結果

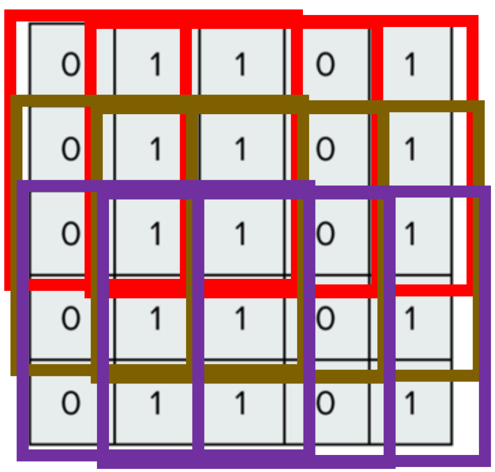
画像での畳み込み

0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1

元画像（５×５マス）

1	0	1
1	1	1
0	0	1

カーネル
（３×３マス）



切り出し（３×３マス）

画像全体について
畳み込み

4	**3**	**5**
4	**3**	**5**
4	**3**	**5**

畳み込み結果

切り出し領域を縦横にずらす

演習２

畳み込み

【トピックス】

畳み込み

二値画像（０，１の画像）の畳み込み

① Google Colaboratory のページを開く

https://colab.research.google.com/drive/1MMhlrh08-0Byq-U1JlTdDVwMe6dyUcaq?usp=drive_link

（このページは、演習１，２で使用する）

② **次**について、プログラムや説明や実行結果が掲載されていることを確認。各自でよく読む。

２．畳み込み

３．二値画像（画素値が０，１のみ）の畳み込み

```
x = [0, 1, 0, 1, 0, 0, 1, 1, 1, 0]
k = [1, 0, 1]
y = [0, 0, 0, 0, 0, 0, 0, 0]

y[0] = x[0] * k[0] + x[1] * k[1] + x[2] * k[2]
y[1] = x[1] * k[0] + x[2] * k[1] + x[3] * k[2]
y[2] = x[2] * k[0] + x[3] * k[1] + x[4] * k[2]
y[3] = x[3] * k[0] + x[4] * k[1] + x[5] * k[2]
y[4] = x[4] * k[0] + x[5] * k[1] + x[6] * k[2]
y[5] = x[5] * k[0] + x[6] * k[1] + x[7] * k[2]
y[6] = x[6] * k[0] + x[7] * k[1] + x[8] * k[2]
y[7] = x[7] * k[0] + x[8] * k[1] + x[9] * k[2]

print(y)
```

```
[0, 2, 0, 1, 1, 1, 2, 1]
```

```
x = [[0, 1, 1, 0, 1],
     [0, 1, 1, 0, 1],
     [0, 1, 1, 0, 1],
     [0, 1, 1, 0, 1],
     [0, 1, 1, 0, 1]]
k = [[1, 0, 1],
     [1, 1, 1],
     [0, 0, 1]]
y = [[0, 0, 0],
     [0, 0, 0],
     [0, 0, 0]]

def conv(i, j):
    return x[i + 0][j + 0] * k[0][0] + x[i + 0][j + 1] * k[0][1] + x[i + 0][j + 2] * k[0][2] ¥
    + x[i + 1][j + 0] * k[1][0] + x[i + 1][j + 1] * k[1][1] + x[i + 1][j + 2] * k[1][2] ¥
    + x[i + 2][j + 0] * k[2][0] + x[i + 2][j + 1] * k[2][1] + x[i + 2][j + 2] * k[2][2]

y[0][0] = conv(0, 0)
y[0][1] = conv(0, 1)
y[0][2] = conv(0, 2)
y[1][0] = conv(1, 0)
y[1][1] = conv(1, 1)
y[1][2] = conv(1, 2)
y[2][0] = conv(2, 0)
y[2][1] = conv(2, 1)
y[2][2] = conv(2, 2)

print(y)
```

```
[[4, 3, 5], [4, 3, 5], [4, 3, 5]]
```

発展演習（余裕のある人向け）①

目的: 元データの変化が、畳み込みの結果がどのように影響するか確認

指示: 演習１のプログラムについて、x を [0, 1, 2, 1, 0, 0, 1, 1, 1, 0] に変更して結果を確認してください

解答例: [2, 2, 2, 1, 1, 1, 2, 1]

プログラムの変更と再実行には，各自の Google アカウントでのログインが必要.

発展演習（余裕のある人向け）②

目的: 元データとカーネルの変化が、畳み込みの結果がどのように影響するか確認

指示: 演習１のプログラムについて、x を [0, 1, 2, 1, 0, 0, 1, 1, 1, 0], kを[1,2,1]にして結果を確認してください

解答例: [4, 6, 4, 1, 1, 3, 4, 3]

プログラムの変更と再実行には，各自の Google アカウントでのログインが必要.

畳み込みのまとめ

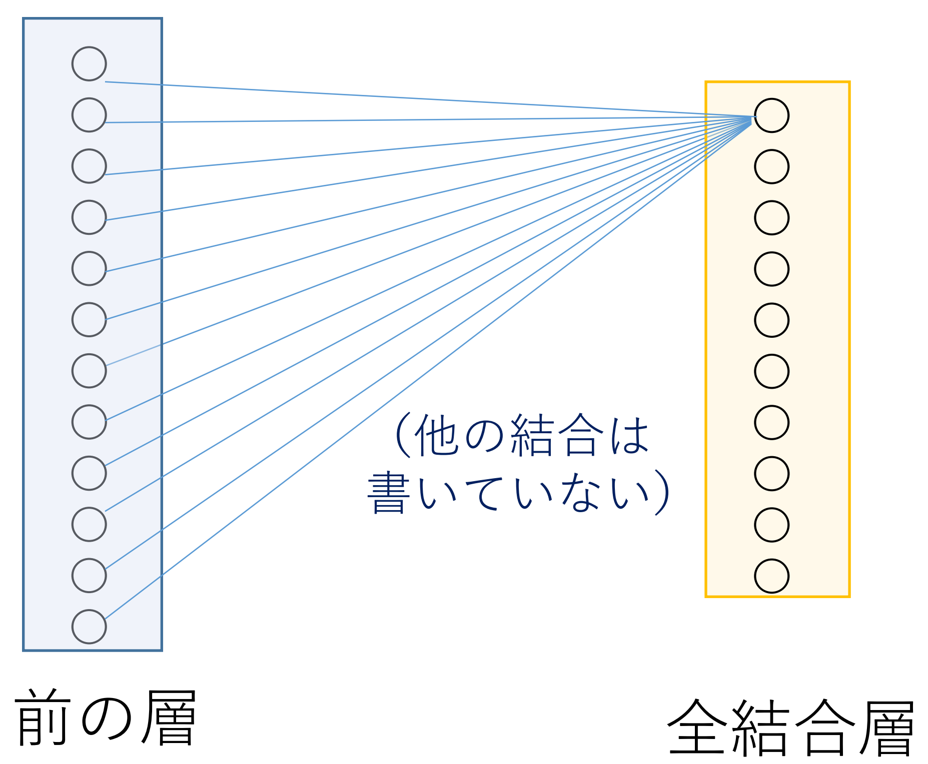
- **畳み込みニューラルネットワーク（CNN）**は画像理解や画像の分析に特化.
- CNNの主層：**畳み込み層**、**プーリング層**、**全結合層**。
- **畳み込み**は、**データ上を移動**しながら、**カーネル**との**重ね合わせ**を行い、**局所的特徴抽出**を行う.

    例：エッジ、テクスチャなど

7-5. 全結合層

全結合層の特徴

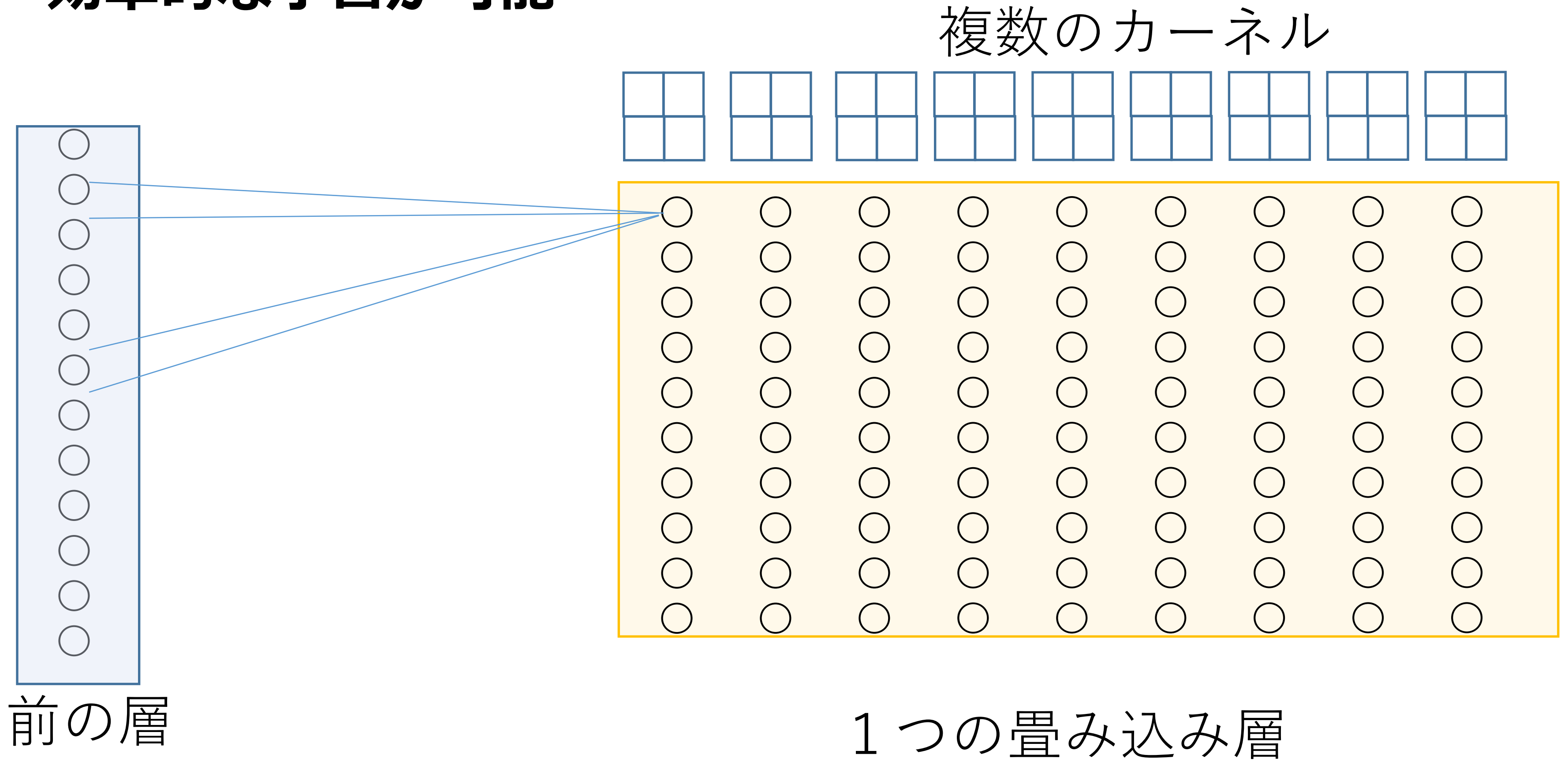
- **全結合層**の**ニューロン**は，**前の層**の**すべてのニューロンと結合**している
- **畳み込みニューラルネットワーク**だけでなく、**通常のニューラルネットワークにも**用いられる



7-6. 畳み込み層

畳み込み層の特徴

- １ニューロンで，１つの畳み込み（掛け算と合計）を行う
- **同じ層の全ニューロン**はすべて、**同一のカーネルを共有し、パラメータの量**は**少ない**
- **効率的な学習が可能**



演習３

畳み込み層

【トピックス】

畳み込み

畳み込み層

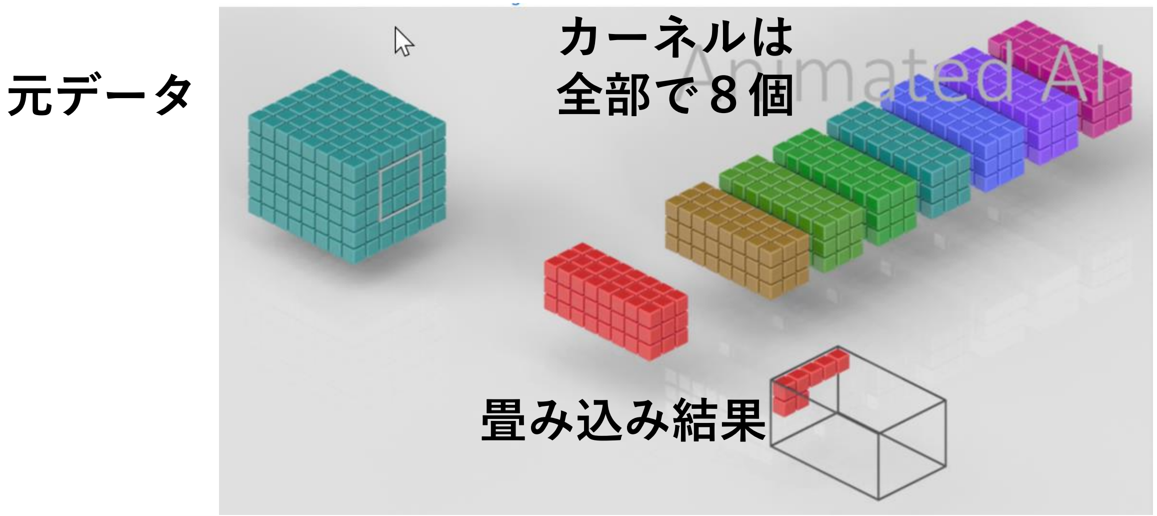
①「Animated AI」のサイトにアクセスする

https://animatedai.github.io/

②このサイトでは、さまざまな**ニューラルネットワーク**の**アルゴリズム**が解説されている

③トップの「**The Basic Algorithm**」は、**畳み込み**をアニメーションで示している。

- 元データとカーネルは３次元である
- **畳み込み**の仕組み上，畳み込み層の**同じ層のニューロン**はすべて，**同一のカーネルを共有**



学習の仕組み

<u>パラメータを調整して誤差を最小化する</u>

【全結合層のパラメータ】

- **パラメータ：**ニューロンの**重み**や**バイアス**
- **パラメータは大量**

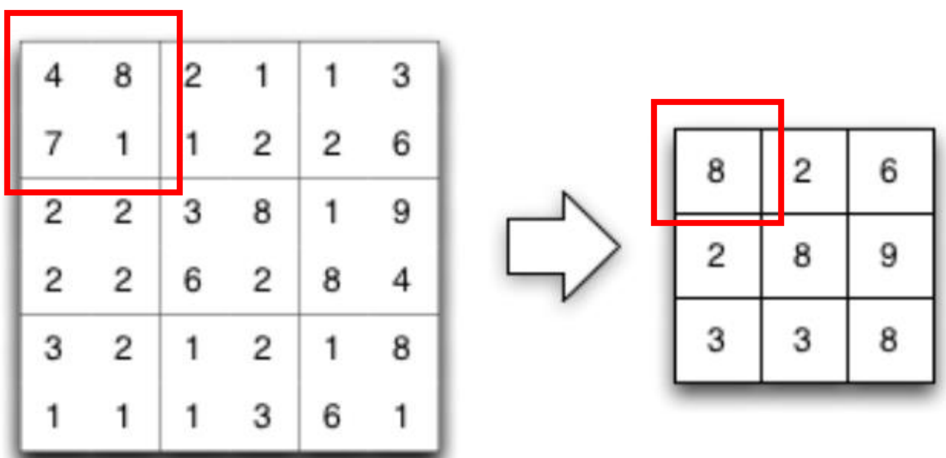
【畳み込み層のパラメータ】

- パラメータ：**カーネル**
- **同じ層の全ニューロン**はすべて、**同一のカーネルを共有**
- **パラメータは少量**
- カーネルの共有によって，過学習を避けつつ高速な学習が実現される.

7-7. プーリング層

プーリング層の機能

- ２次元のデータの**縮小**
  （例）サイズ 100 × 100 ⇒ 50 × 50 のように
- 一定領域内の**結果**を，**１つにまとめる**.
- まとめるときに，最大値でまとめる Max Pooling がよく利用される（特徴抽出の結果を保持）.
- 学習の対象では**ない**



· 4, 8, 7, 1 の最大値は 4
・「**4, 8, 7, 1**」**の 4マス**
から，**最大値の 8 を選ぶ.**

出典: https://github.com/jeffheaton/t81_558_deep_learning/blob/084023876b6cf09c931b452584dbd44c56314a03/t81_558_class_06_2_cnn.ipynb

7-8. CNN Explainer のデモ

畳み込みニューラルネットワーク（CNN）の特徴

- 画像の局所的な特徴をとらえる**畳み込み層を持つ**
- 主に**畳み込み層、プーリング層、全結合層**などから構成

•学習では、重み、バイアスなどのパラメータを調整。ただし、**畳み込み層**では、**カーネルを共有する**ため、学習が効率化

演習４

畳み込みニューラルネットワーク

CNN Explainer

- **CNN Explainer**　ジョージア工科大学 Polo Club
- 畳み込み層などの仕組みをビジュアルに学ぶことができるサイト

【アクセス方法】

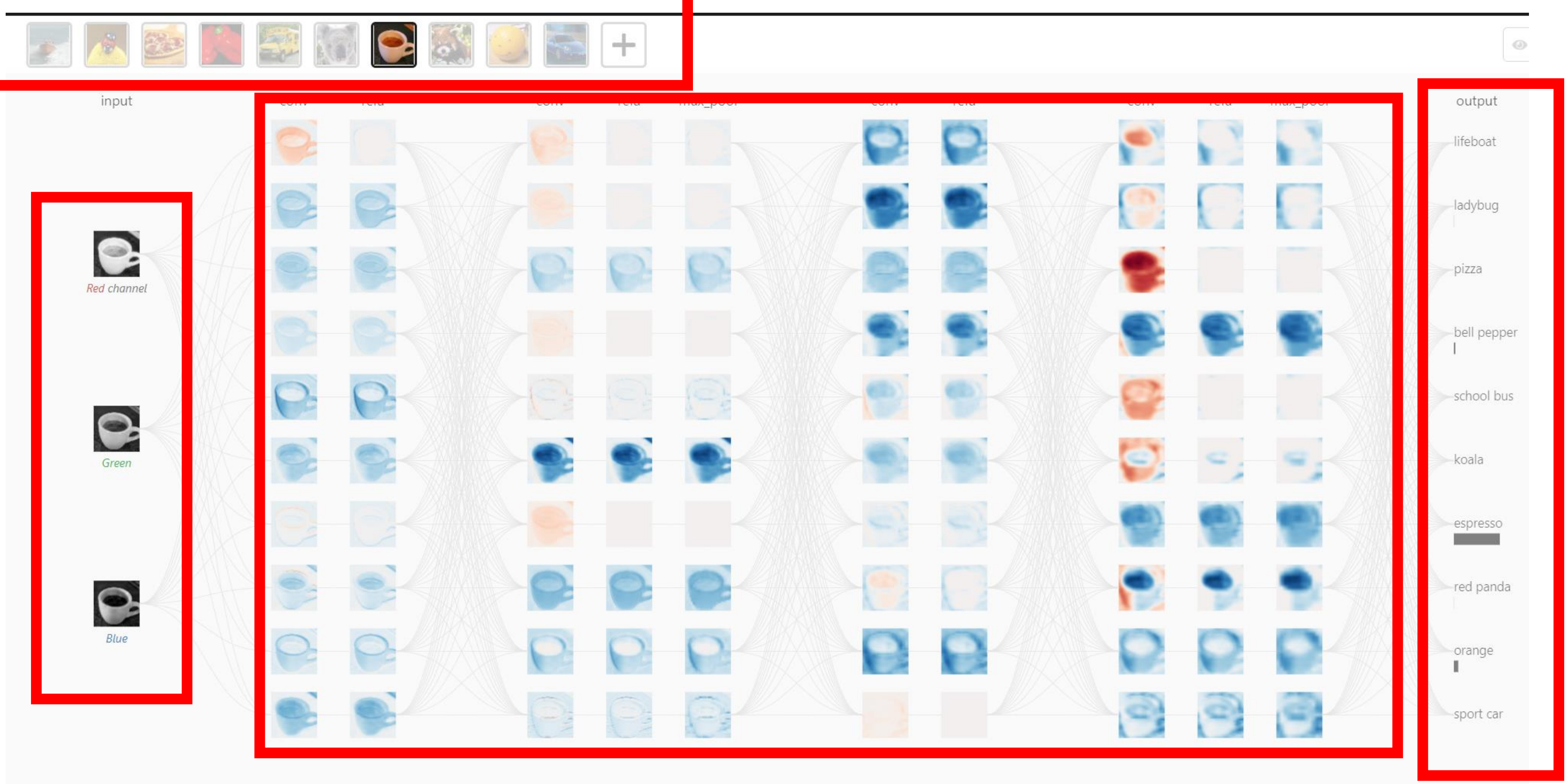
Webブラウザで次の URL を開く

https://poloclub.github.io/cnn-explainer/

CNN Explainerの基本操作

画像分類を行うニューラルネットワーク

画像を選ぶ



元画像の
赤青緑の成分

ニューラルネットワークの可視化

画像の分類結果.
ここでは espresso

ネットワーク構成の確認

全結合層，畳み込み層，プーリング層から構成される

conv relu conv relu max_pool conv relu conv relu max_pool
畳み込み層 畳み込み層 プーリング層 畳み込み層 畳み込み層 プーリング層

conv は畳み込み層で，max_pool はプーリング層

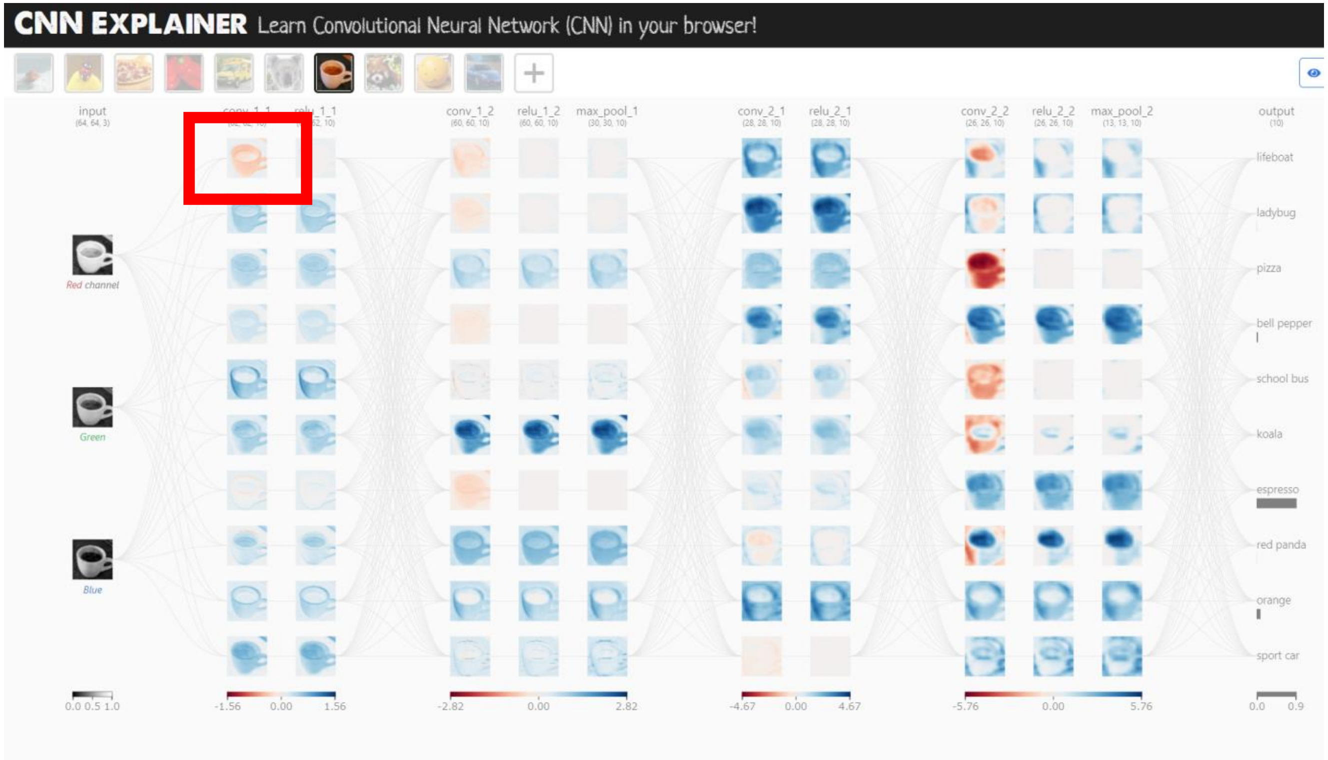
全結合層 全結合層 全結合層 全結合層

畳み込み処理の可視化

左上の画像をクリック

→ **畳み込みの様子をアニメーションで確認できる**

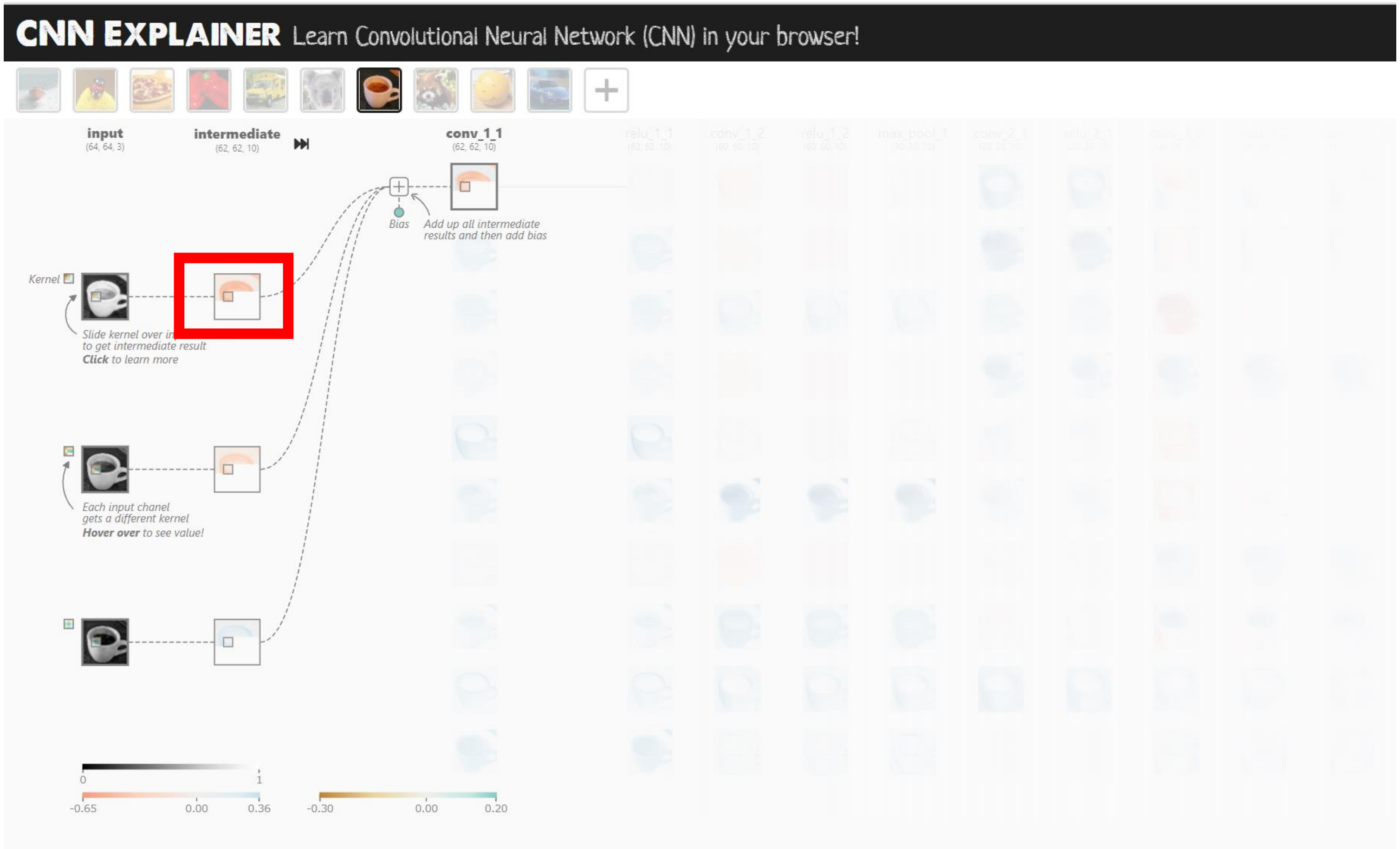
（この画像は，**各層での処理結果**である）



詳細な畳み込み過程

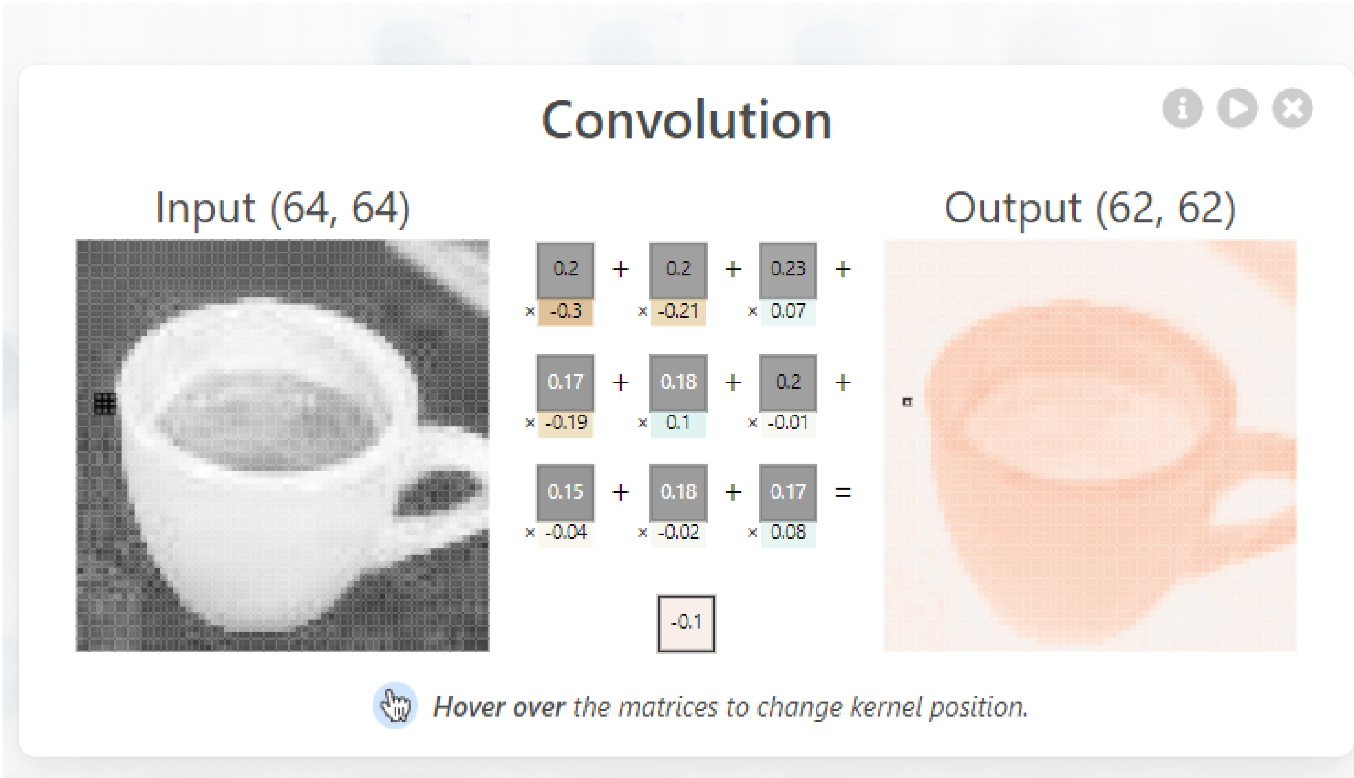
出てきた画像をクリック

→　畳み込みの詳細をアニメーションで確認できる



アニメーションによる学習

畳み込みの様子がアニメーションで表示される



その他の層についてもビジュアルに表示できる（いろいろ試すことは，各自の自主的な自習とする）

全体まとめ

畳み込みニューラルネットワーク（CNN）

- **畳み込みニューラルネットワーク（CNN）**は**画像理解**や**画像の分析**に特化した**ディープラーニング**の一種。
- CNNは主に**畳み込み層**、**プーリング層**、**全結合層**の３種類
- **畳み込み層**：画像の**局所的な特徴をとらえる**ための層。**特徴**は、画像内の**顕著なパターンや属性**（例：エッジ、テクスチャ）
- **プーリング層**：**特徴マップ**の**サイズを縮小**するための層。**過学習を防止。計算効率を向上**
- **全結合層**：全ニューロンが前の層のすべてのニューロンと接続された層。畳み込み層とプーリング層を通過した後の特徴を基に、**画像**の**分類**や**回帰**を行う

畳み込み層のパラメータの量は少ない。これにより、過学習を回避