画像を扱うニューラルネットワーク

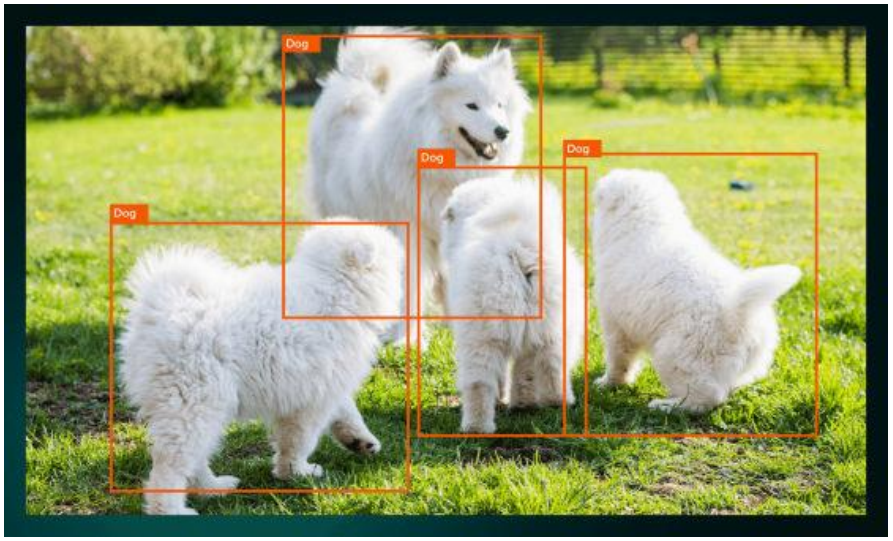
URL: https://www.kkaneko.jp/cc/ni/index.html

金子邦彦

アウトライン

1. 画像を扱うニューラルネットワーク
2. 畳み込み
3. CNN の仕組み
4. CNN の詳細

ニューラルネットワークでの画像処理の例



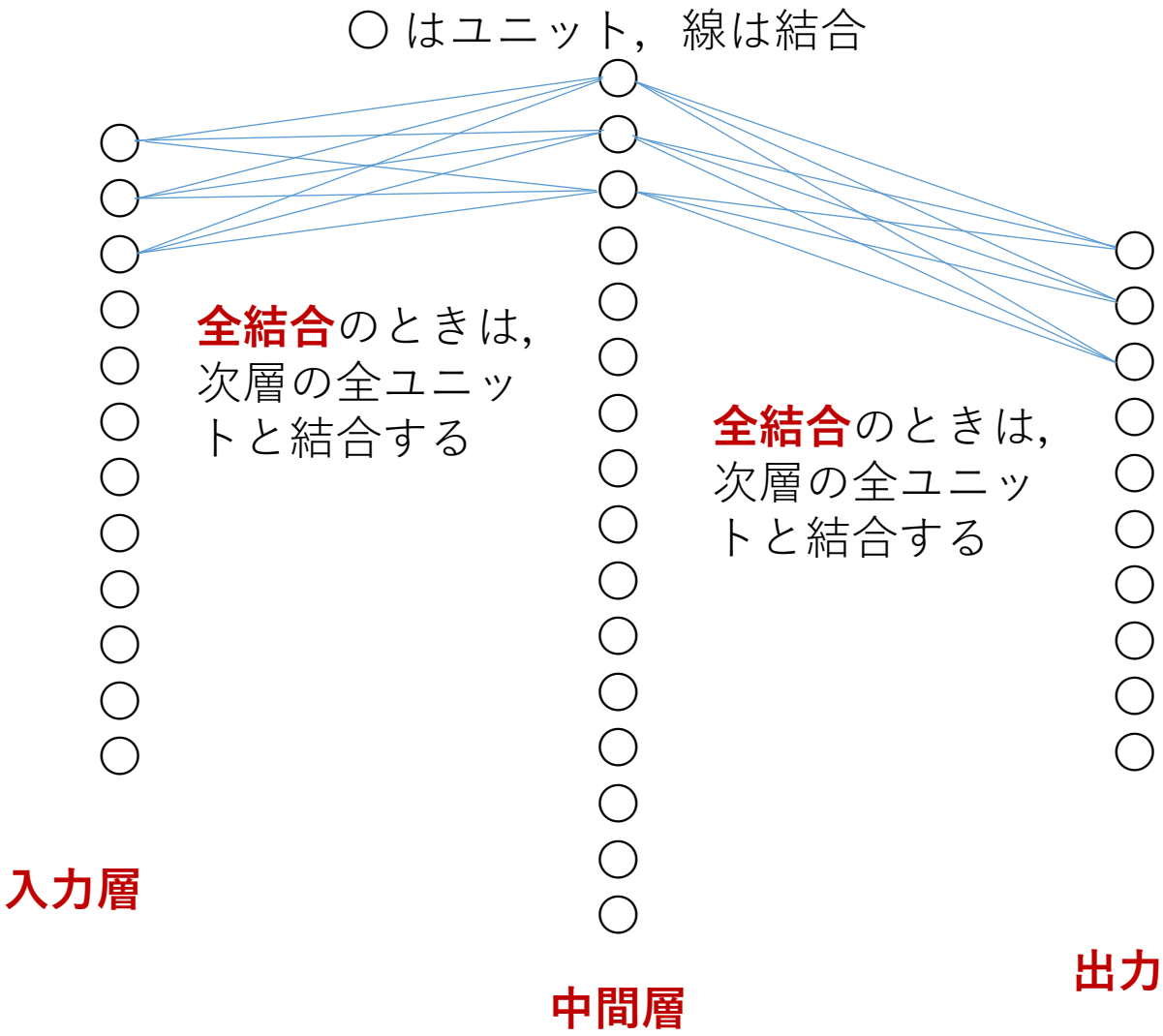
物体検出

写真の分析

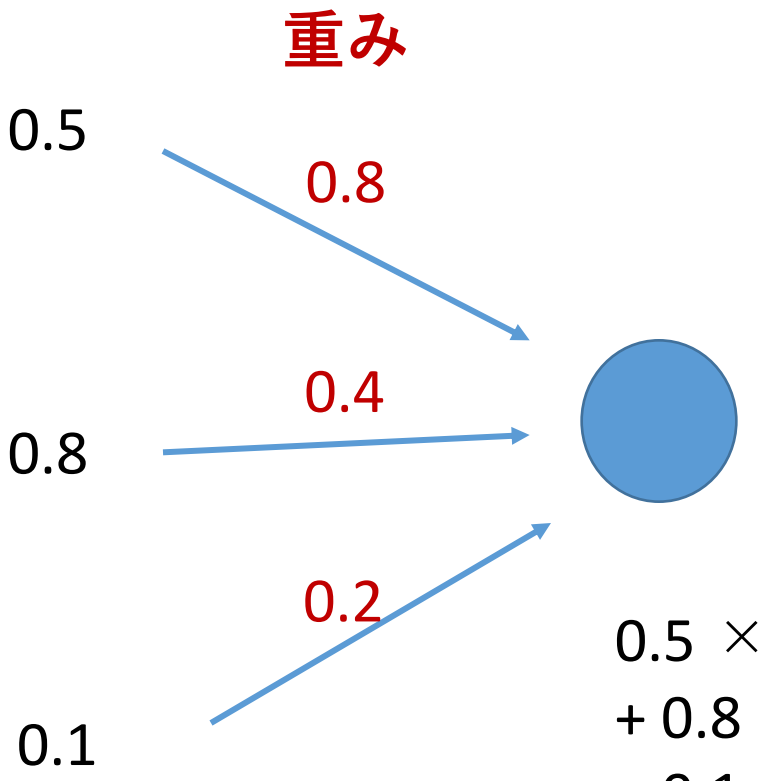
出典: https://serokell.io/blog/introduction-to-convolutional-neural-networks

ニューラルネットワークでの処理の仕組み

- 前の層から結果を受けとって，次の層へ結果を渡す
- 学習の時には，結合の重みが自動調整される



ユニットの仕組み



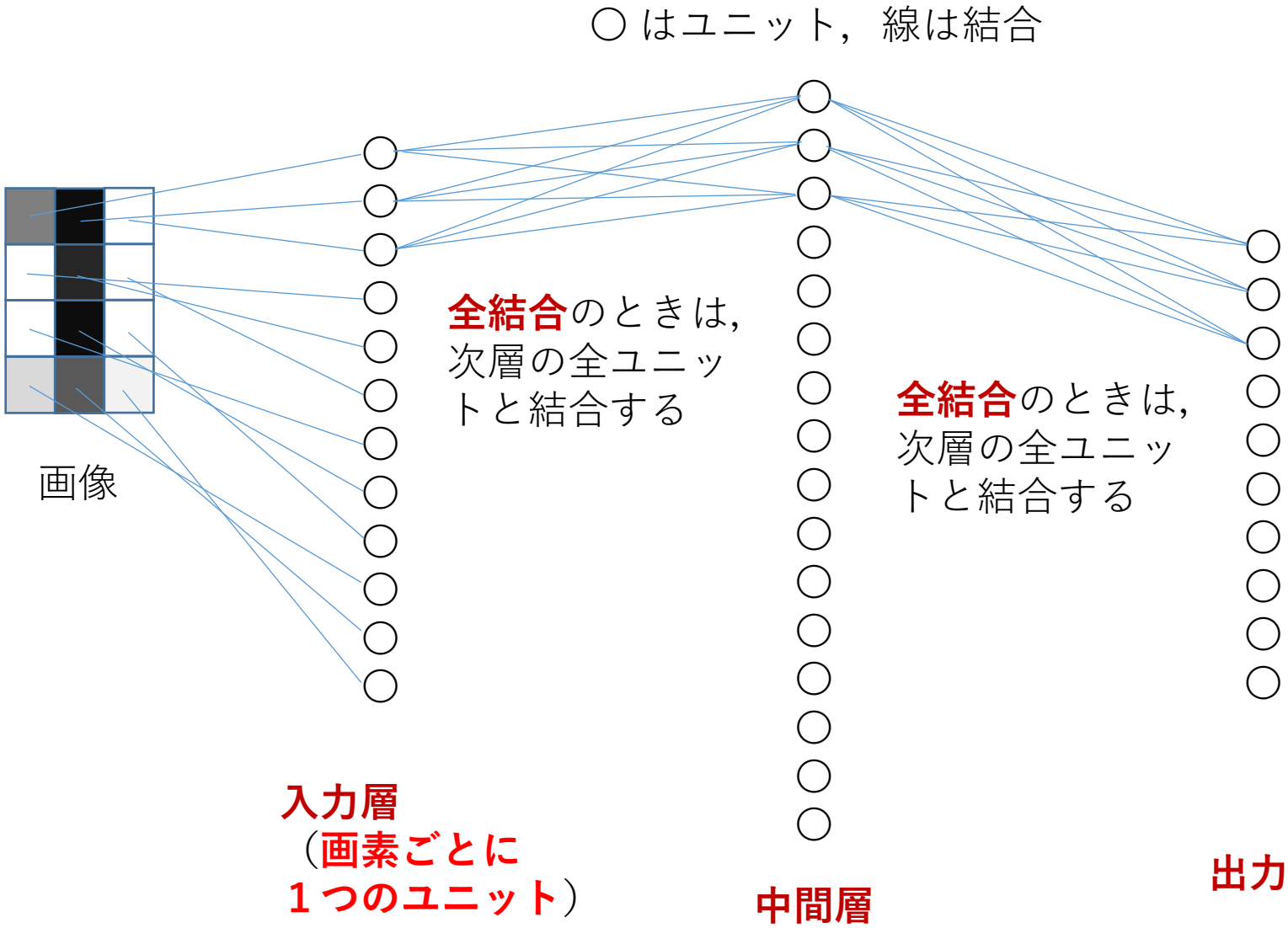
0.5×0.8
$+ 0.8 \times 0.4$
$+ 0.1 \times 0.2$

前の層のユニットの出力に
重みをかけたものの**総和**

この値に応じて，このユニットは**活性化**する
（値が大きいほど，**活性化**の度合いが高い）

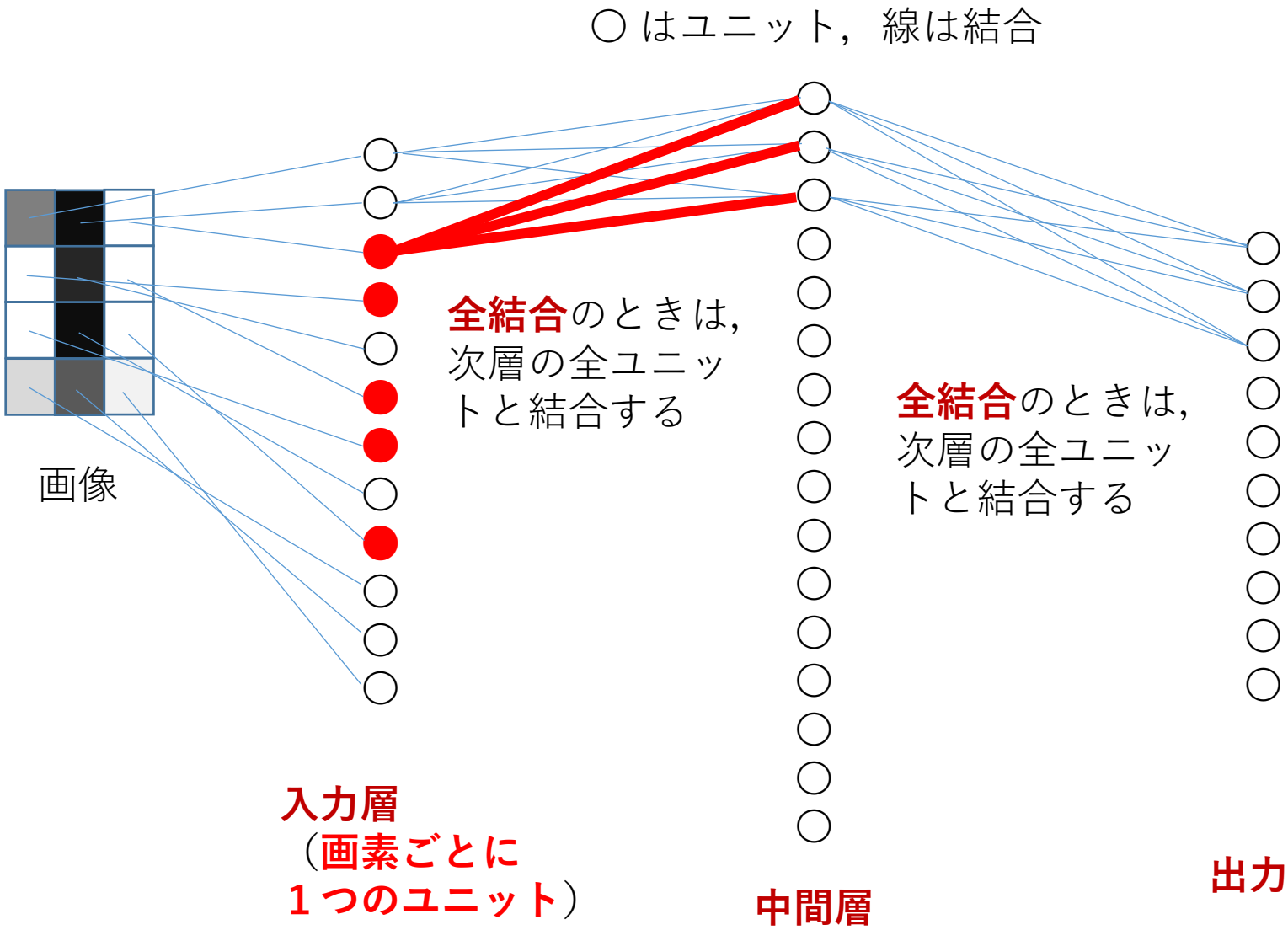
画像とニューラルネットワーク

モノクロ画像（濃淡画像）をニューラルネットワークで扱うとき，入力層では，画素ごとに１つのユニットである



伝搬

入力層では，画素が「明るい」ときに活性化．その情報は結合を通して，次の層に伝搬する



ニューラルネットワークの仕組み

- **学習**では，正しい結果が得られるように，**結合の重み**を**自動調整**する

- 入力層：　濃淡画像（モノクロ画像）を扱うときは，ユニット数が画素数に等しい（**画素1つごとにユニット１つ**）
- 中間層：　中間層の数，ユニット数などは**自由に設定できる**
- 出力層：　「**１０種類に分類したい**」というときは，**出力層のユニット数は１０**.

2 畳み込み

金子邦彦

- 画像を，**全結合のニューラルネットワーク**で扱うときの**問題**
  - **入力層のユニット数は大**

    640 × 480 のモノクロ画像：画素数は　307,200
  - **結合の数も大**
  - **学習（結合の重みの調整）のために必要となる画像も大**

- **畳み込み**を利用して，**近くの画素の情報を１つにまとめる**ことにより，ユニット数と結合数の削減を行う

畳み込み

- **カーネル（フィルタ）**を使用

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）

元画像の中から，
カーネルに**似た**
部分を取り出すために，
畳み込みを行う

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

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）

畳み込みのために，
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）

畳み込み結果

カーネルとの類似度
を表している

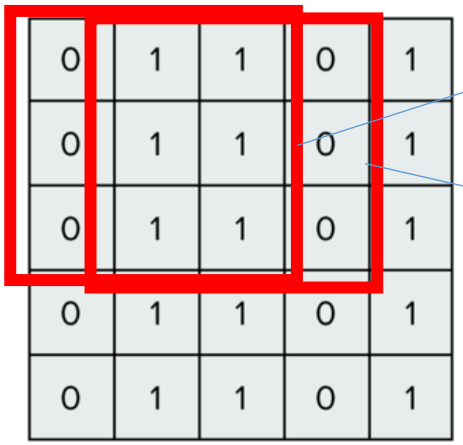
畳み込み

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）

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

切り出し領域を横にずらす

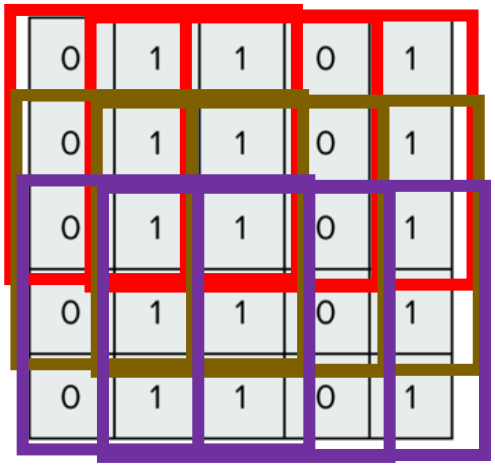
畳み込み

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**

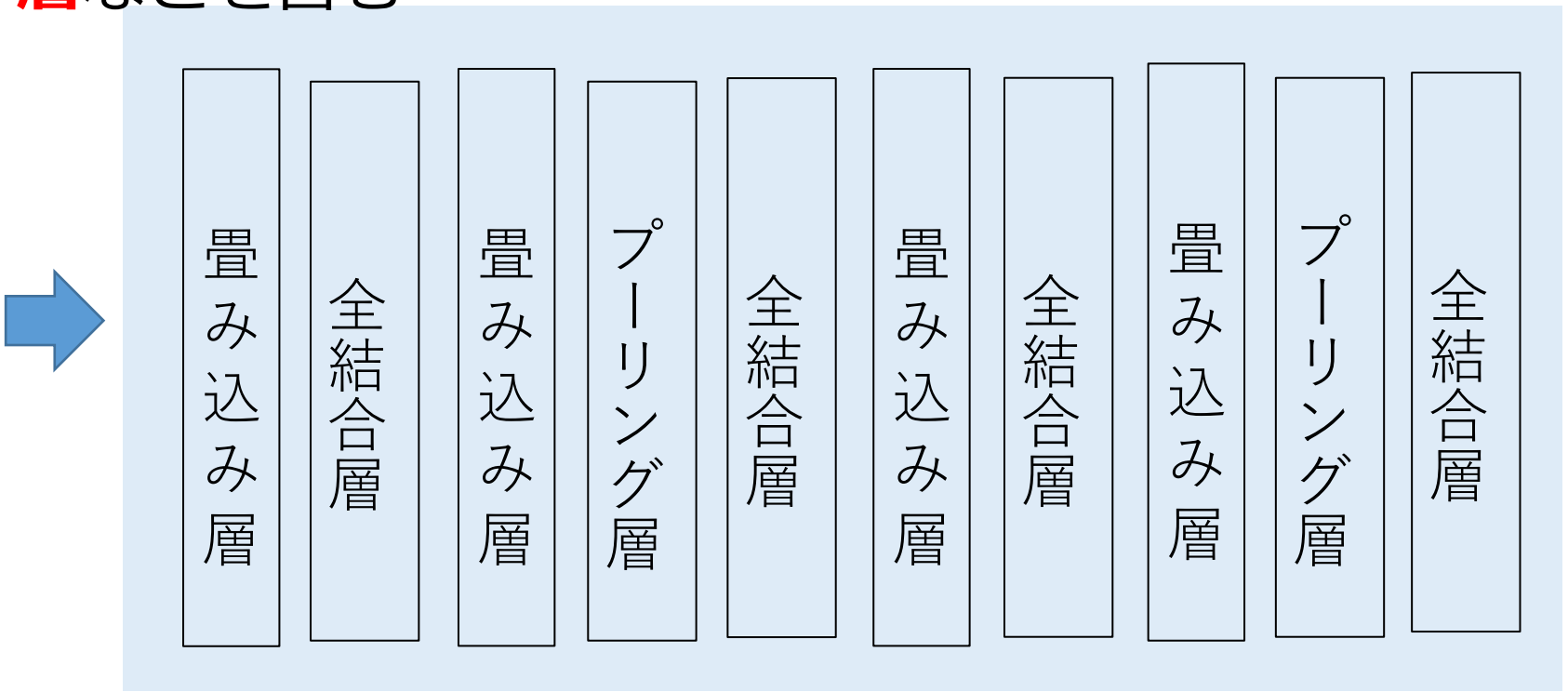
畳み込み結果

切り出し領域を縦横にずらす

3. CNN の仕組み

金子邦彦

- CNN は，畳み込みニューラルネットワーク（Convolutional Neural Network）で，**ニューラルネットワーク**の種類
- CNN は，**全結合層**の他に，**畳み込み**を行う**畳み込み層**などを含む



CNN の例

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

① Webブラウザで次の URL を開く

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

② 画面の確認

このニューラルネットワークは，画像を分類する

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

画像の分類結果.
ここでは espresso

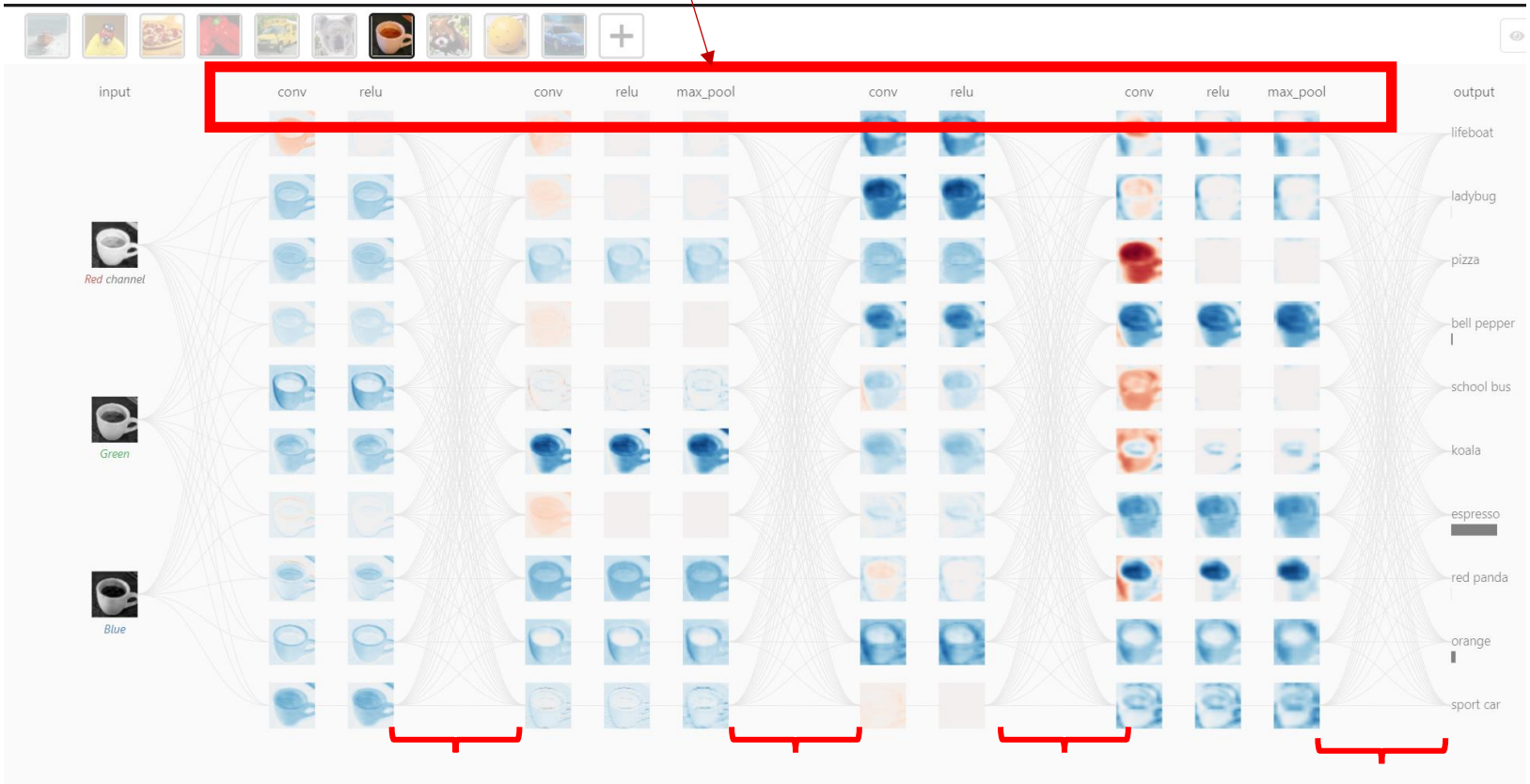
③ ニューラルネットワークの構成

畳み込み層と**プーリング層**を含む

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

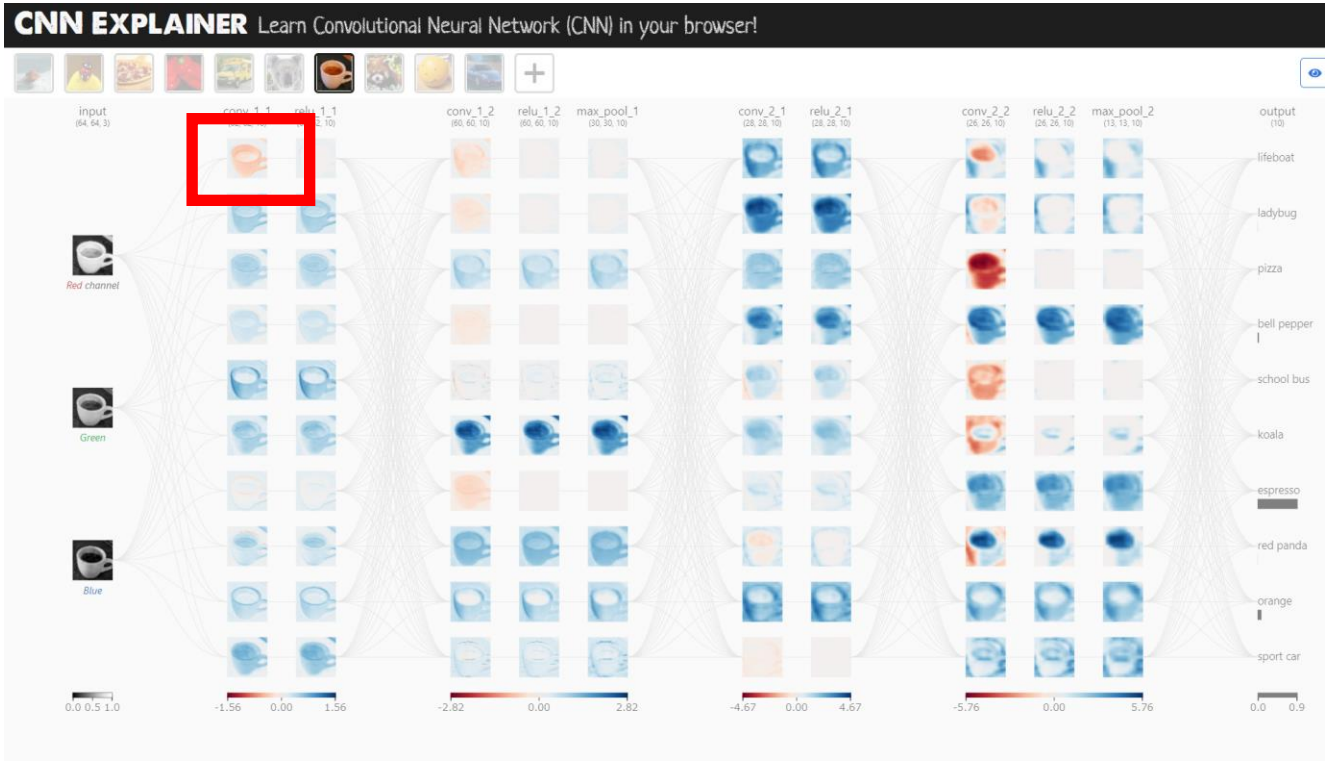
conv は畳み込み層で，max_pool はプーリング層



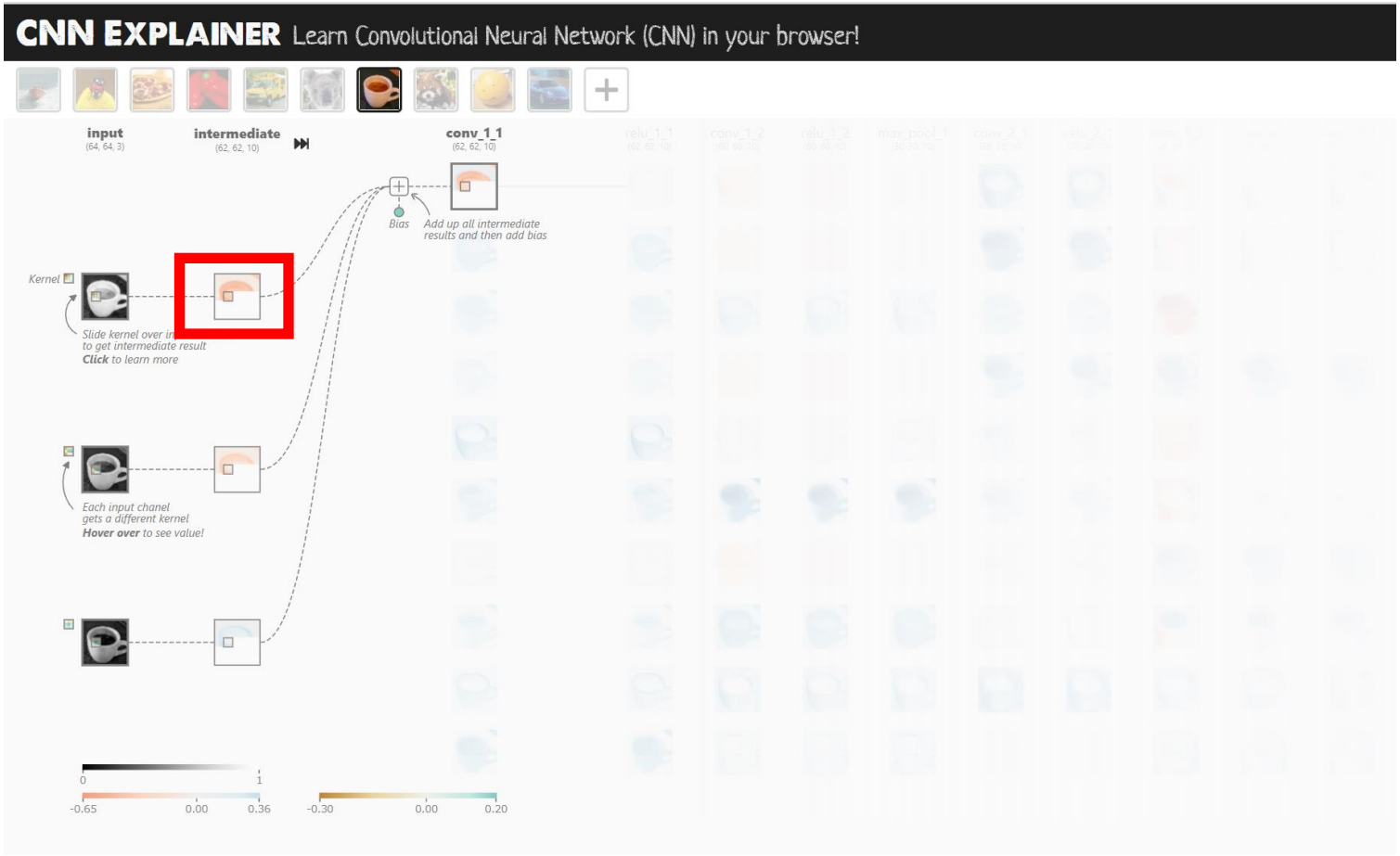
全結合層　全結合層　全結合層　全結合層

④ 左上の画像をクリック

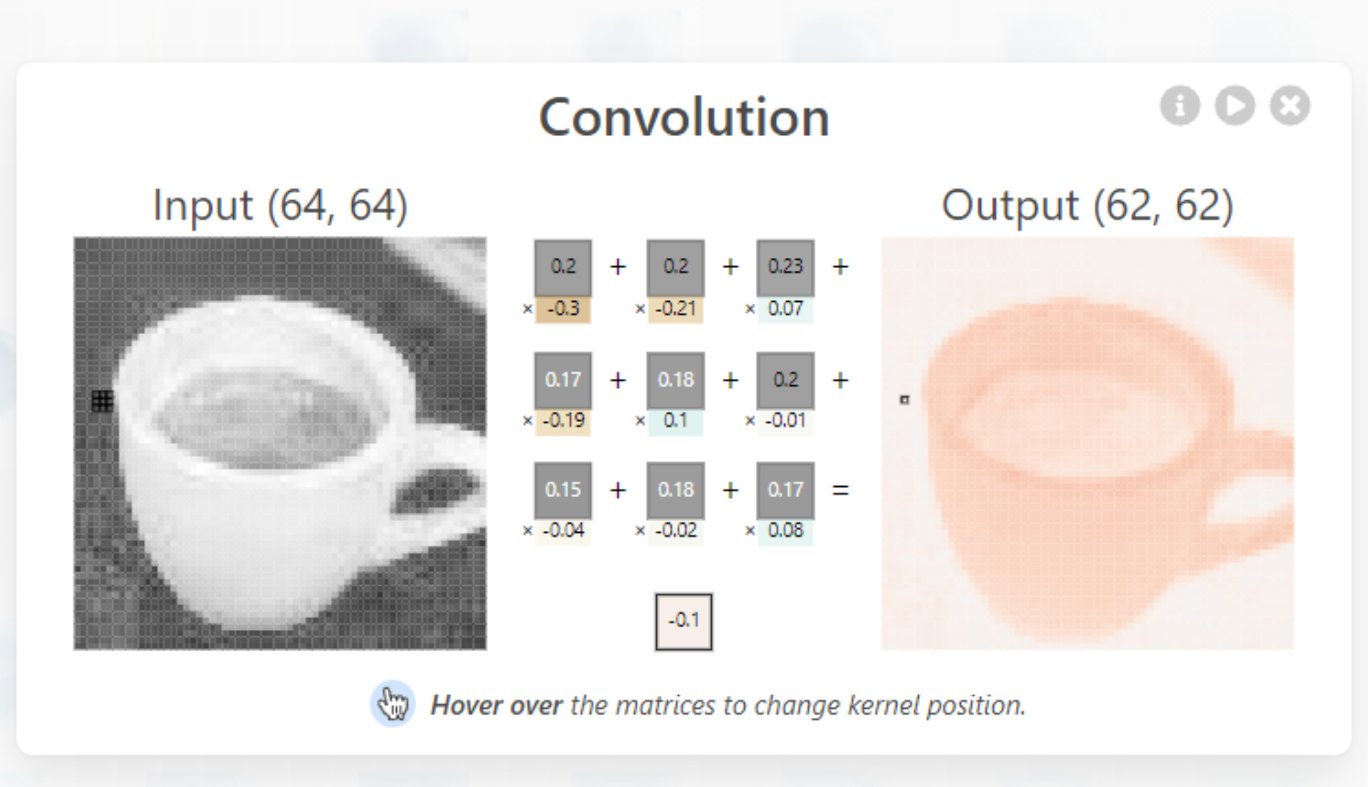
（この画像は，各層での処理結果である．画像１個がユニット１つというわけではない）



⑤ 出てきた画像をクリック



⑥ 畳み込みの様子がアニメーションで表示される



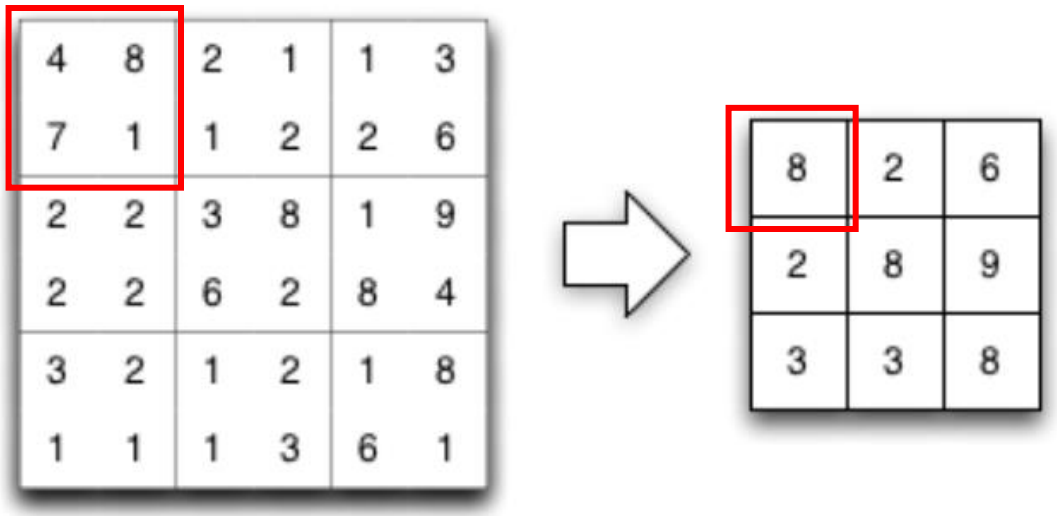
その他の層についてもビジュアルに表示できる（いろいろ試すことは，各自の自主的な自習とする）

4. CNN の詳細

金子邦彦

プーリング

- **プーリングにより，画像サイズが小さくなる**
- Max pooling （マックス・プーリング）は，**最大値を利用するプーリング**



・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**.
すると，サイズは ¼ になる

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

CNN の作成プログラム

- 第２回授業のプログラム

```
[ ]  from tensorflow.keras import backend as K
     from tensorflow.keras.models import Model, Sequential
     import tensorflow.keras

     # 畳み込みニューラルネットワークの作成
     num_classes = 10
     img_rows, img_cols = 28, 28
     if K.image_data_format() == 'channels_first':
         input_shape = (1, img_rows, img_cols)
     else:
         input_shape = (img_rows, img_cols, 1)

     m = Sequential()
     m.add(Conv2D(32, kernel_size=(3, 3),
                  activation='relu',
                  input_shape=input_shape))
     m.add(Conv2D(64, (3, 3), activation='relu'))
     m.add(MaxPooling2D(pool_size=(2, 2)))
     m.add(Dropout(0.25))
     m.add(Flatten())
     m.add(Dense(128, activation='relu'))
     m.add(Dropout(0.5))
     m.add(Dense(num_classes, activation='softmax'))
     m.compile(loss='categorical_crossentropy', optimizer='adam',
               metrics=['accuracy'])
     m.summary()
```

① １層目：畳み込み層

② ２層目：畳み込み層

③ ３層目：プーリング層

④ ４層目：全結合層

⑤ ５層目：全結合層

全体まとめ

- 画像の画素数は多い
- **全結合層**だけでは，**画像をうまく扱えない**
- **CNN（畳み込みニューラルネットワーク）**の**畳み込み層**，**プーリング層**を用いて，画像を扱う技術が考案されている
- **CNN（畳み込みニューラルネットワーク）**の**作成**は，**プログラムで簡単にできる**