

研究室ローテーション 第6回

担当：和田山 正・中井彩乃

本講義の内容

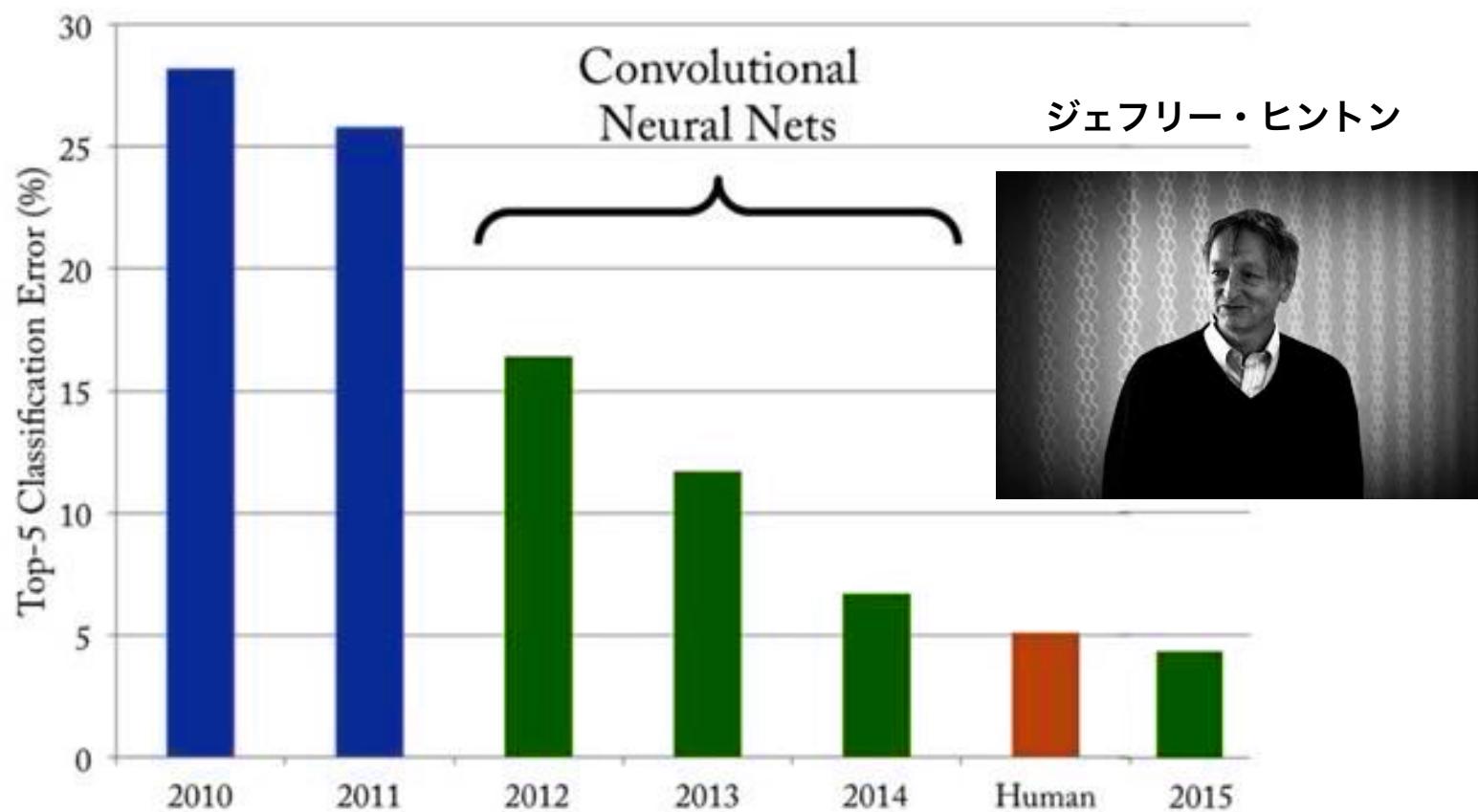
- 置み込みニューラルネットワーク

ImageNetデータ例とTOP5出力

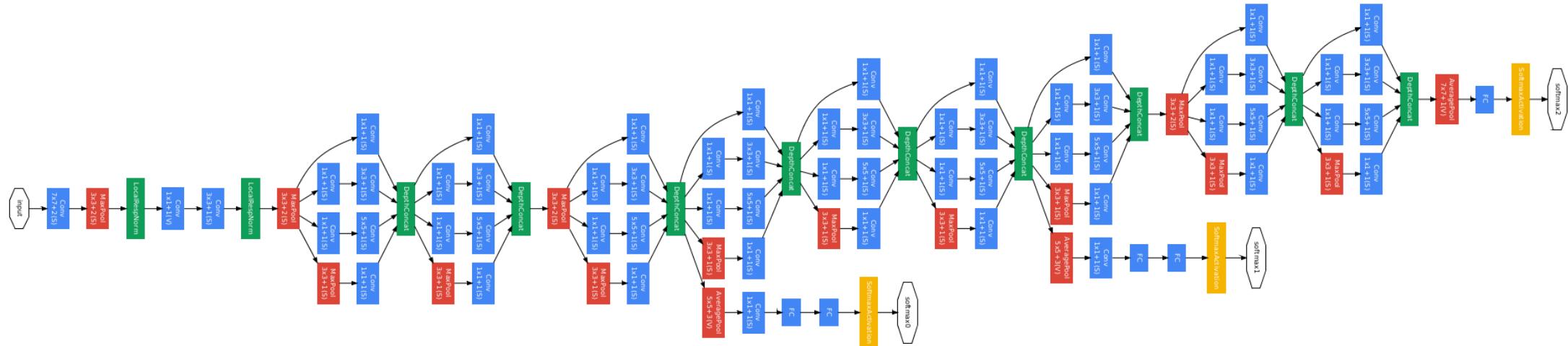


ImageNet認識の誤り率

ImageNet Classification (2010 – 2015)



GoogleNet



- Googleが開発した深層ニューラルネットワークの一種
- 置み込みニューラルネットワーク
- ILSVRC2014 トップ

MNIST数字認識

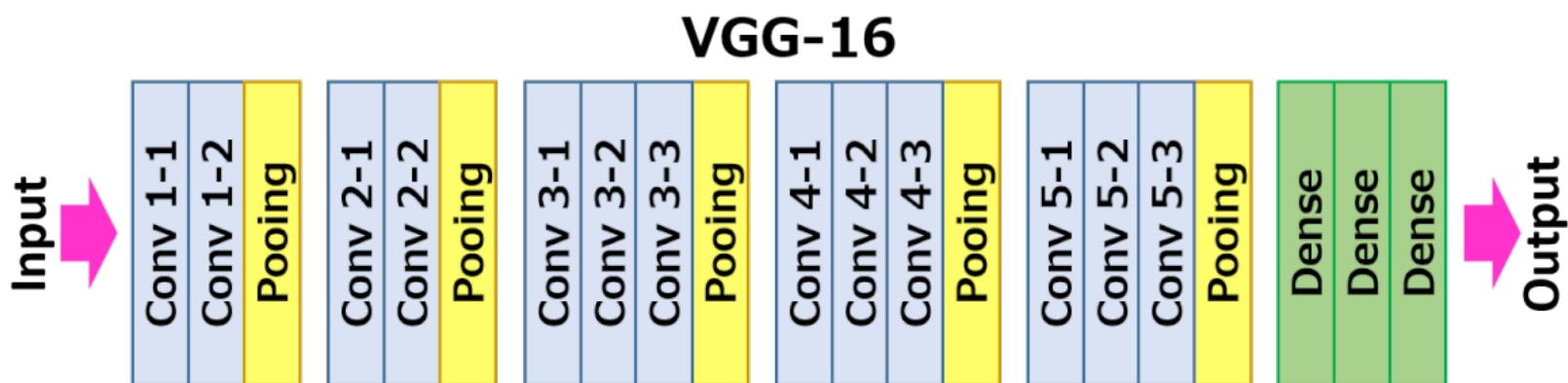
MNISTデータセット



- 0-9の手書き数字のデータセット
- 28 × 28ピクセル (深さ8bit, モノクロ)
- 訓練データ6万枚/テストデータ1万枚
- パターン認識アルゴリズムのテストにしばしば利用される (パターン認識の"Hello World" と呼ばれている)

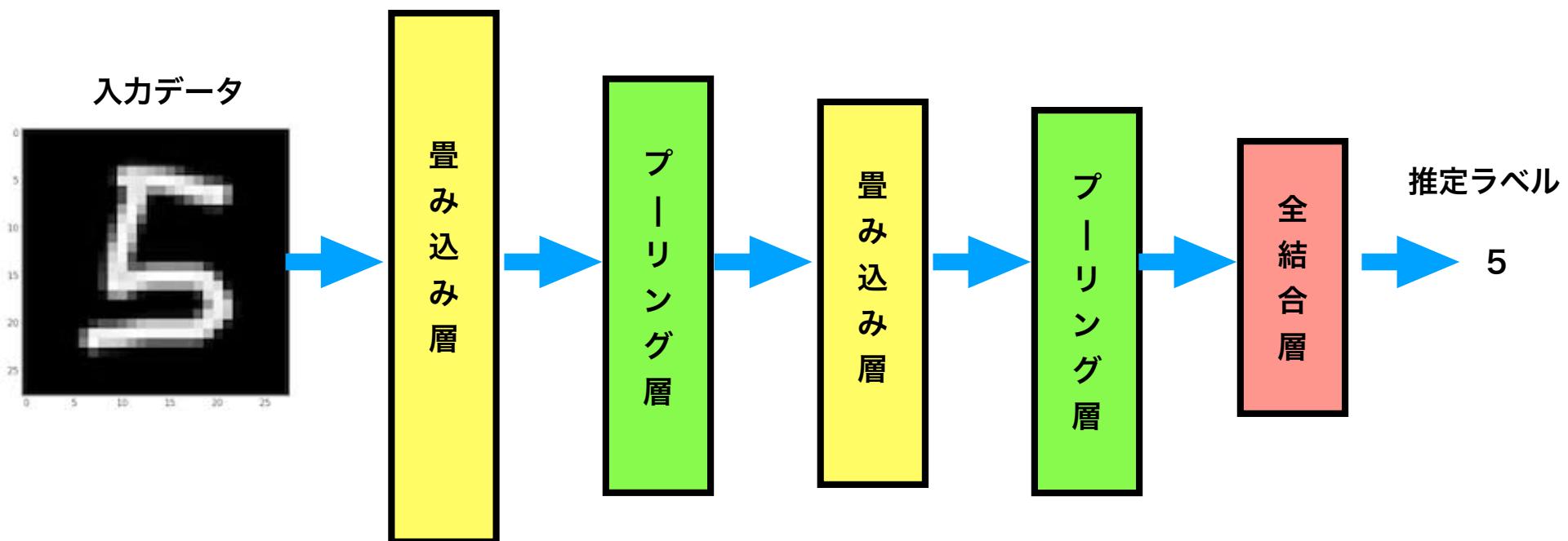
畳み込みニューラル

ネットワーク



畳み込みニューラルネットワーク

- ・2次元畳み込み演算に基づく層 = 畳み込み層(学習パラメータを含む)
- ・2次元プーリング演算に基づく層 = プーリング層
- ・全結合層



畠み込み層(1)

窓内のデータに学習可能パラメータを乗じて和を取る

x_{11}	x_{12}	x_{13}	
x_{21}	x_{22}	x_{23}	
x_{31}	x_{32}	x_{33}	

y_{11}		

窓の中の値 x_{ij} と
窓の重み w_{ij} を
かけあわせたものを
足す

$$y_{11} = \sum_{1 \leq i, j \leq 3} w_{ij} x_{ij}$$

y_{12}		

窓を1マスずつ
上下に移動して
同じ計算を行う

畳み込み層(2)

- w_{ij} は学習パラメータ(**畳み込みカーネル**と呼ばれることがある)であり、これを学習することで画像認識に適切な**2次元フィルタ**が学習される(**特徴学習・表現学習**)
- 画像の**局所的性質**を抽出している
- 上の説明は少し簡単化しており、本当は複数枚のチャネルから複数枚チャネルを生成する

プーリング層

情報を縮約し、層のサイズを縮小させる

x_{11}	x_{12}	x_{13}	
x_{21}	x_{22}	x_{23}	
x_{31}	x_{32}	x_{33}	

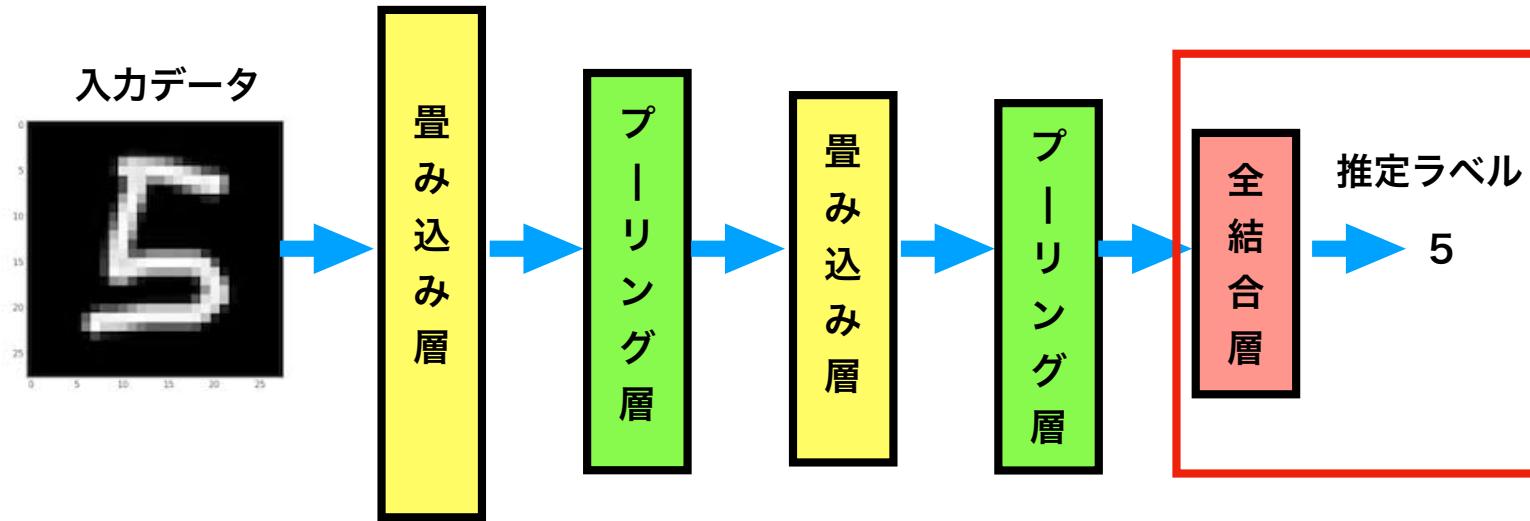
y_{11}		

窓の中の最大値を
取ってくる

$$y_{11} = \max_{1 \leq i, j \leq 3} x_{ij}$$

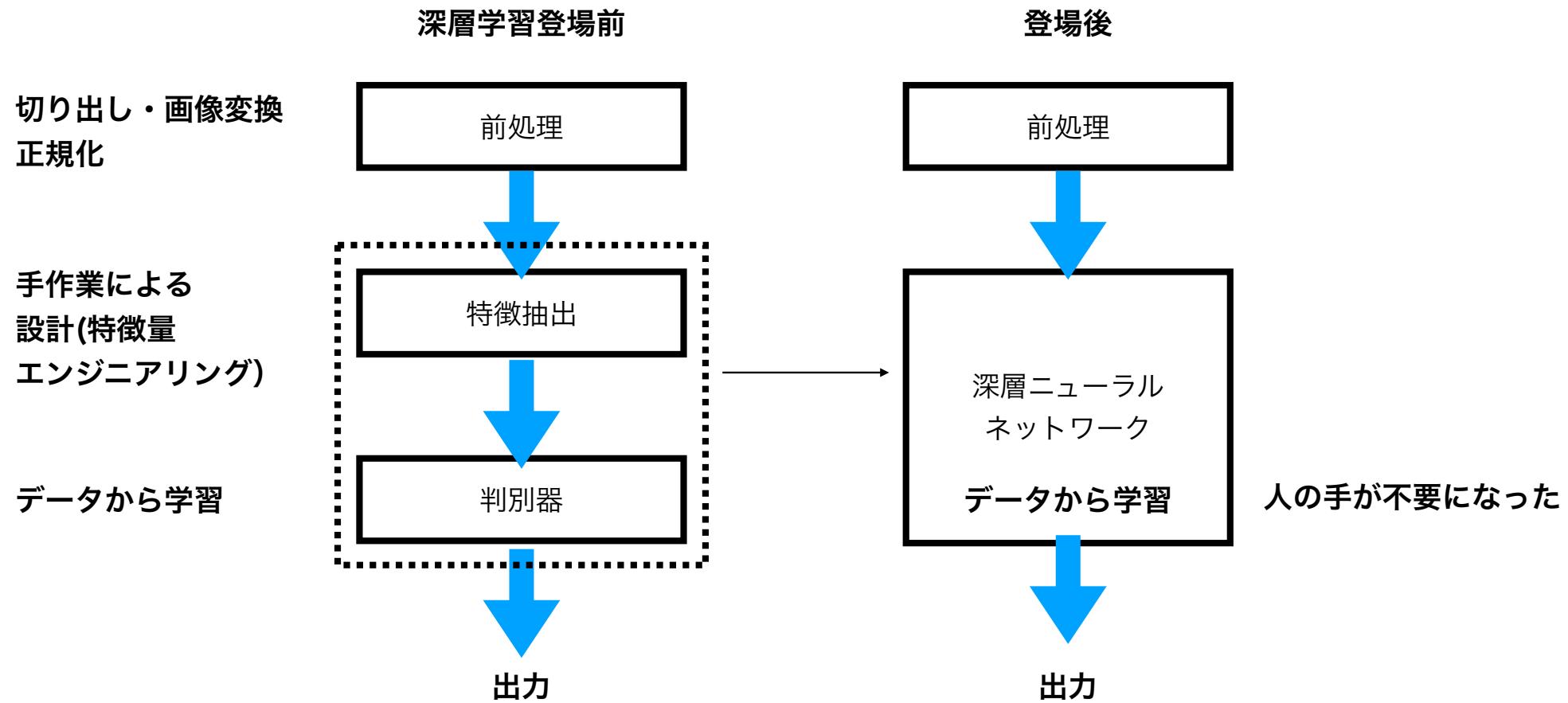
y_{12}		

畳み込みニューラルネットワークの構造



- 全結合層は $f(Wx + b)$ 型の通常のNN（前回までの構造）
- 最終層はソフトマックス層
- 現在においても優れた画像クラス判別のアルゴリズムは畳み込みNN構造

パターン認識構成のためのアプローチ



(過去)特徴量エンジニアリング → (現在) DNNによる表現学習

PyTorchで畳み込みNN

ややこしいですが、
畳み込みとプーリングで
画像サイズが
どう変化するかを
計算しながら
実装する必要があります

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 2, 3) # 28x28x1 -> 26x26x2
        self.conv2 = nn.Conv2d(2, 3, 3) # 26x26x3 -> 24x24x3
        self.pool = nn.MaxPool2d(2, 2) # 24x24x3 -> 12x12x3
        self.fc1 = nn.Linear(12 * 12 * 3, 16)
        self.fc2 = nn.Linear(16, 10)

    def forward(self, x):
        x = F.relu(self.conv1(x)) # 第1畳み込み層
        x = F.relu(self.conv2(x)) # 第2畳み込み層
        x = self.pool(x) # プーリング層
        x = x.view(-1, 12 * 12 * 3)
        x = F.relu(self.fc1(x)) # 全結合層 1
        x = self.fc2(x) # 全結合層 2
        return F.log_softmax(x, dim=1)
```

入力チャネル数 (1で固定)
出力チャネル数 (画像を複製する枚数を設定できる)
カーネル (窓) サイズ
窓を何マスずつずらすか
カーネル (窓) サイズ
全結合層 (前回までと同様)

GPUの使用方法

The screenshot shows a Jupyter Notebook interface. In the code cell, the following Python code is written:

```
[ ] isgpu = False      # True:GPUを使う, False:GPUを使わない, ランタイム->ランタイムのタイプを変更->GPU
```

A large blue arrow points upwards from the "ランタイム" menu towards the code cell, indicating the step to change the runtime type.

Trueに変更

The "ランタイム" menu is open, showing various options like "すべてのセルを実行" and "ランタイムのタイプを変更". The "ランタイムのタイプを変更" option is highlighted with a blue box and a blue arrow pointing to it from the left.

GPUを選んで保存

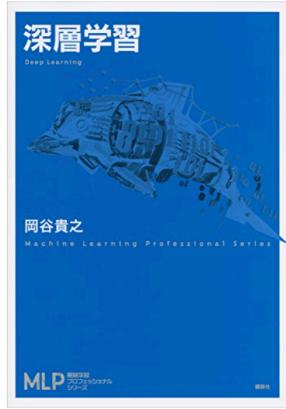
The "ランタイムのタイプ" dropdown is set to "Python 3". Below it, the "ハードウェアアクセラレータ" section shows four options: "CPU" (unchecked), "T4 GPU" (checked), "A100 GPU" (unchecked), and "V100 GPU" (unchecked). A blue arrow points downwards from the "ランタイムのタイプ" dropdown towards the "T4 GPU" button.

GPUに変更したあとは最初のセルから実行しなおす。

本講義のまとめ

- 置み込みニューラルネットワーク

深層学習についてもっと知りたい人は



深層学習 (機械学習プロフェッショナルシリーズ) Kindle版

岡谷貴之 (著) | 形式: Kindle版

★★★★☆ 103個の評価

すべての形式と版を表示

Kindle版 (電子書籍)

¥2,156

獲得ポイント: 42pt

今すぐお読みいただけます: 無料アプリ

単行本 (ソフトカバー)

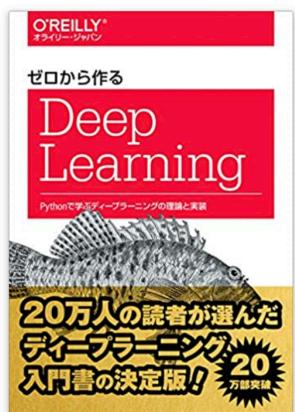
¥3,080

獲得ポイント: 93pt

※この商品はタブレットなど大きいディスプレイを備えた端末で読むことに適しています。また、文字だけを拡大することや、文字列のハイライト、検索、辞書の参照、引用などの機能が使用できません。

いま最も注目されている機械学習手法である深層学習（ディープラーニング）を、トップ研究者が解説した。基礎から、SGD、自己符号化器、CNN、RNN、ボルツマンマシンまでと、盛りだくさん。軽快な語り口なので、無理なく理解できる！

手に取りやすい厚さで
内容が多岐にわたる。



ゼロから作るDeep Learning —Pythonで学ぶディープラーニング の理論と実装 単行本 (ソフトカバー) – 2016/9/24

斎藤 康毅 (著)

★★★★☆ 629個の評価

すべての形式と版を表示

単行本 (ソフトカバー)

¥3,740

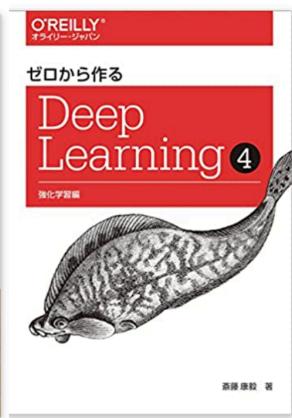
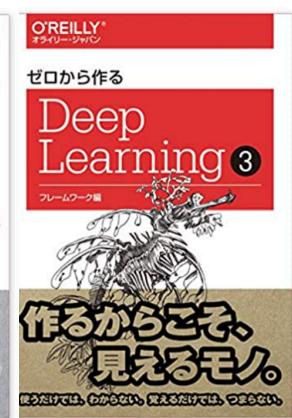
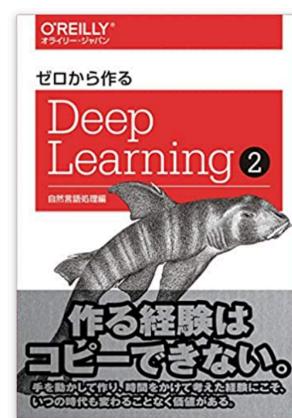
獲得ポイント: 178pt

¥2,050 より 50 中古品

¥3,740 より 18 新品

¥7,480 より 1 コレクター商品

本の購入で10



手を動かしたい人向け。
入門から応用まで
詳細に記述。