



3-2 人工知能による画像の分類、 過学習

(情報工学応用演習 II)

金子邦彦





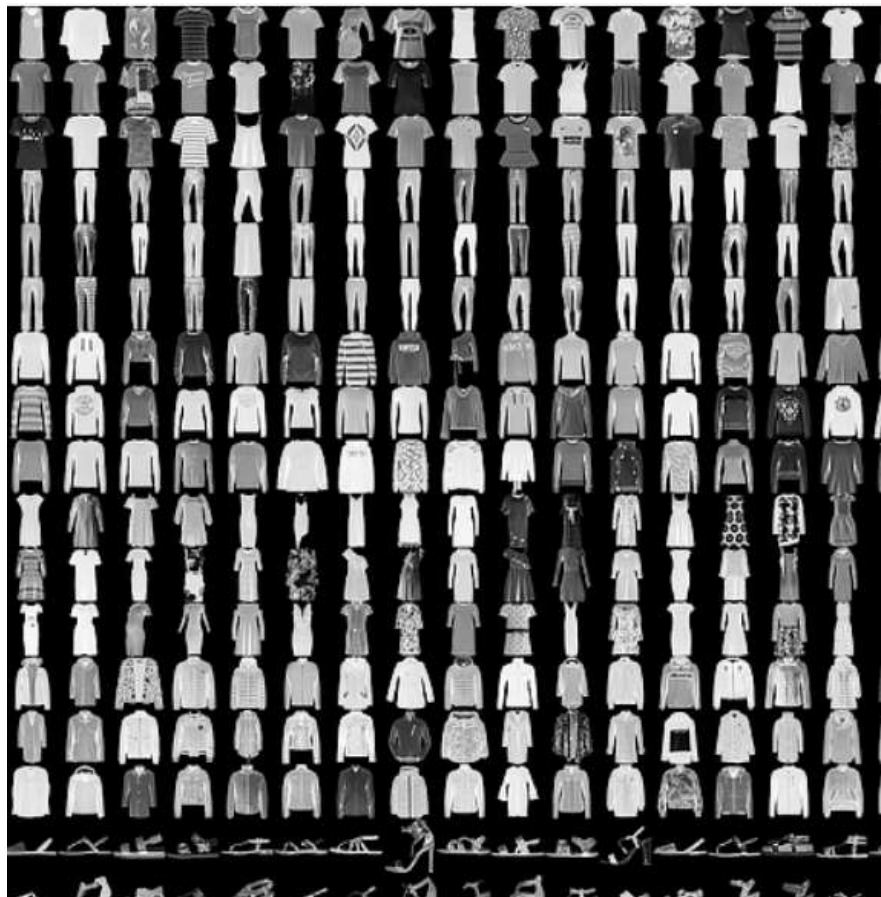
- **人工知能**の画像分類のプログラムを動作させ、理解を深める
- **学習**と**検証**のそれぞれで**精度**をみて、**過学習**が起きているかどうか判断できるようになる

過学習が起きていないことを確認しながら
人工知能を使うことが大事



人工知能による画像の分類

ここで行う画像の分類



たくさんの画像

Label	Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

- 画像を 10 種類に自動分類。
- 人工知能を使用

前準備



Google アカウントの取得が必要

- 次のページを使用

<https://accounts.google.com/SignUp>

- 次の情報を登録する

氏名

自分が希望するメールアドレス

<ユーザー名> [@gmail.com](mailto:kanekokunihiko@gmail.com)

パスワード

生年月日, 性別

Google
Google アカウントの作成

姓 名
金子 邦彦

ユーザー名
kanekokunihiko12112 @gmail.com

半角英字、数字、ピリオドを使用できます。

選択可能なユーザー名:
bangyanjinzi6 jinzibangyan6 kanekokunihiko72

代わりに現在のメールアドレスを使用

パスワード 確認
.....

半角英字、数字、記号を組み合わせて8文字以上で入力してください。

代わりにログイン 次へ

① パソコンの Web ブラウザで、次のページを開く

<https://www.tensorflow.org/tutorials>

② 左側のメニューの「Keras による ML の基本」を展開，「基本的な画像分類」をクリック，「Run in Google Colab」をクリック



③ セルを上から順に実行する.

セルの実行の終了を確認してから、次のセルに移ること

```
[ ] # TensorFlow と tf.keras のインポート
import tensorflow as tf
from tensorflow import keras

# ヘルパーライブラリのインポート
import numpy as np
import matplotlib.pyplot as plt

print(tf.__version__)
```

Google
アカウント
が必要

▼ ファッションMNISTデータセットのロード

このガイドでは、[Fashion MNIST](#)を使用します。Fashion MNISTには10カテゴリーの白黒画像70,000枚が含まれ、それぞれは下図のような1枚につき1種類の衣料品が写っている低解像度（28×28ピクセル）の画像です。

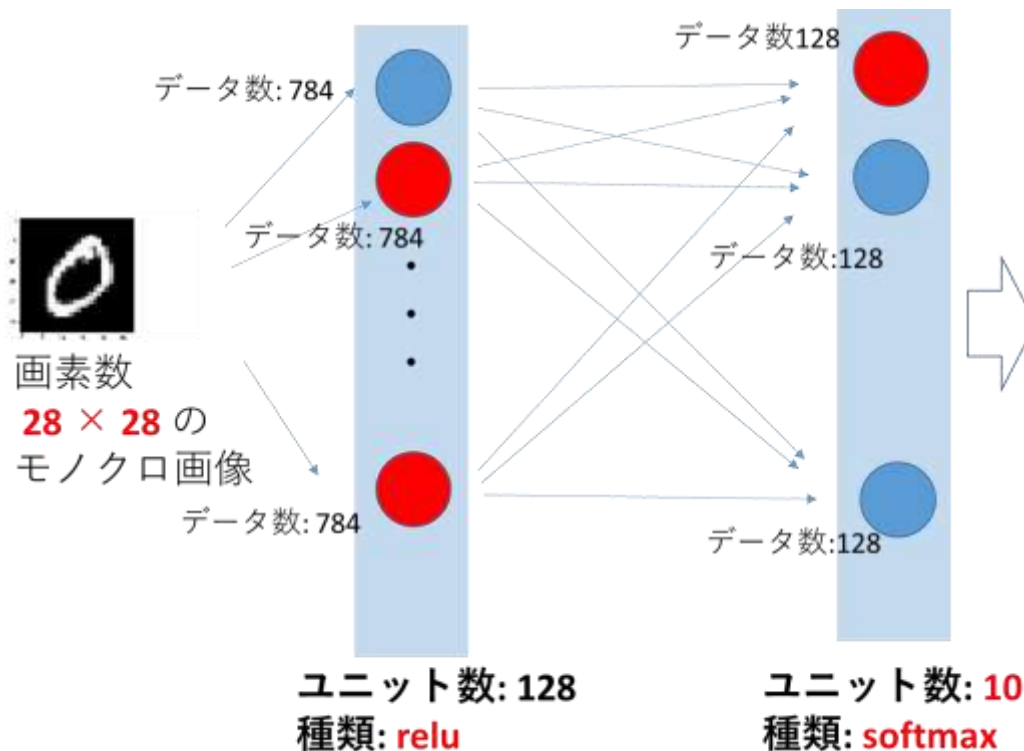


最後まで続ける

ニューラルネットワークを作成するプログラム



```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```



ニューラルネットワークの学習の様子



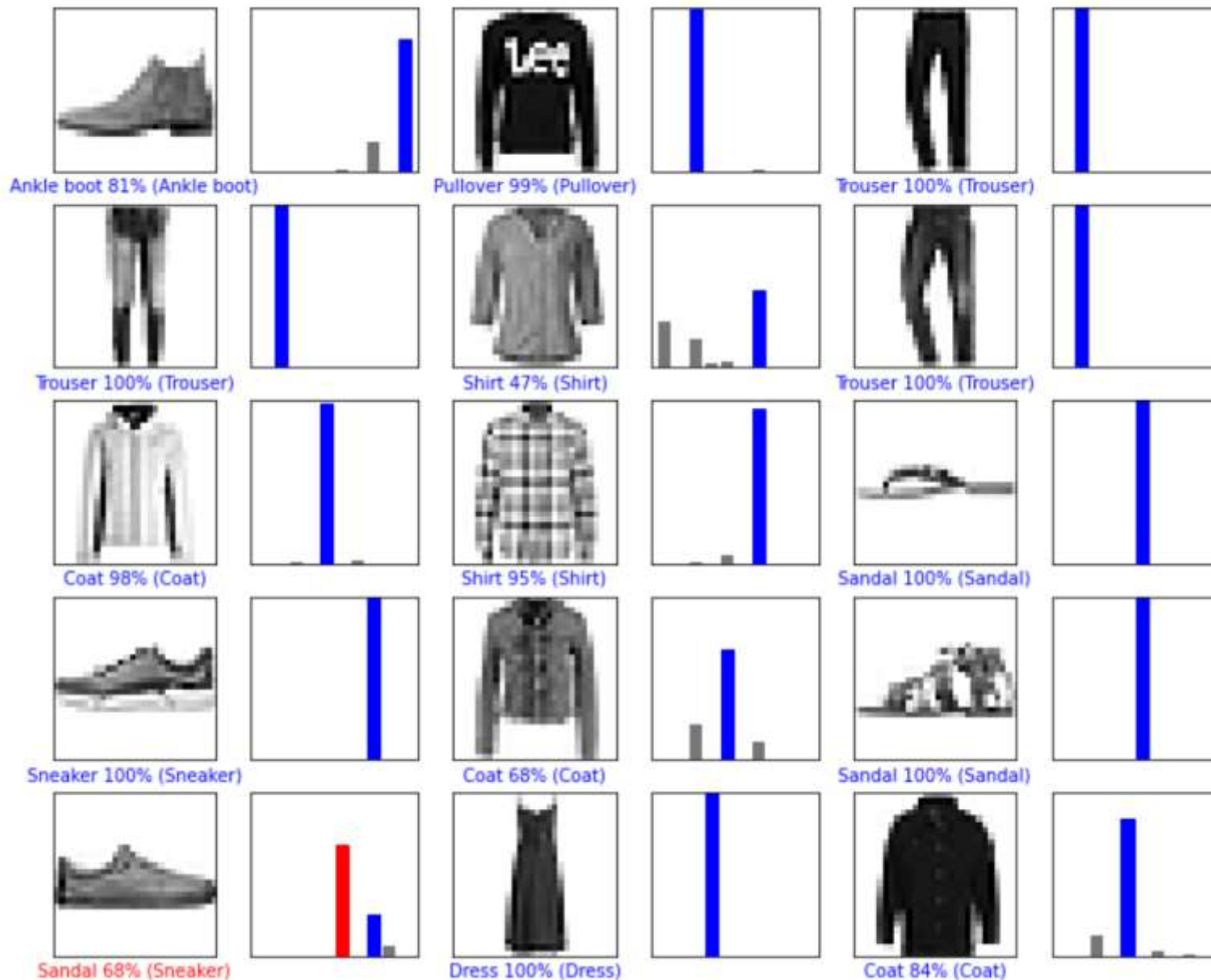
- 同じデータを用いた学習の繰り返し回数
(エポック数) : 5

```
[14] model.fit(train_images, train_labels, epochs=5)
```



```
Epoch 1/5  
1875/1875 [=====] - 3s 2ms/step - loss: 0.5026 - accuracy: 0.8225  
Epoch 2/5  
1875/1875 [=====] - 3s 2ms/step - loss: 0.3774 - accuracy: 0.8638  
Epoch 3/5  
1875/1875 [=====] - 3s 2ms/step - loss: 0.3413 - accuracy: 0.8754  
Epoch 4/5  
1875/1875 [=====] - 4s 2ms/step - loss: 0.3162 - accuracy: 0.8845  
Epoch 5/5  
1875/1875 [=====] - 3s 2ms/step - loss: 0.2960 - accuracy: 0.8907  
<tensorflow.python.keras.callbacks.History at 0x7f43d53c1e48>
```

ニューラルネットワークによる予測の様子





過學習

過学習なし



精度

良くない

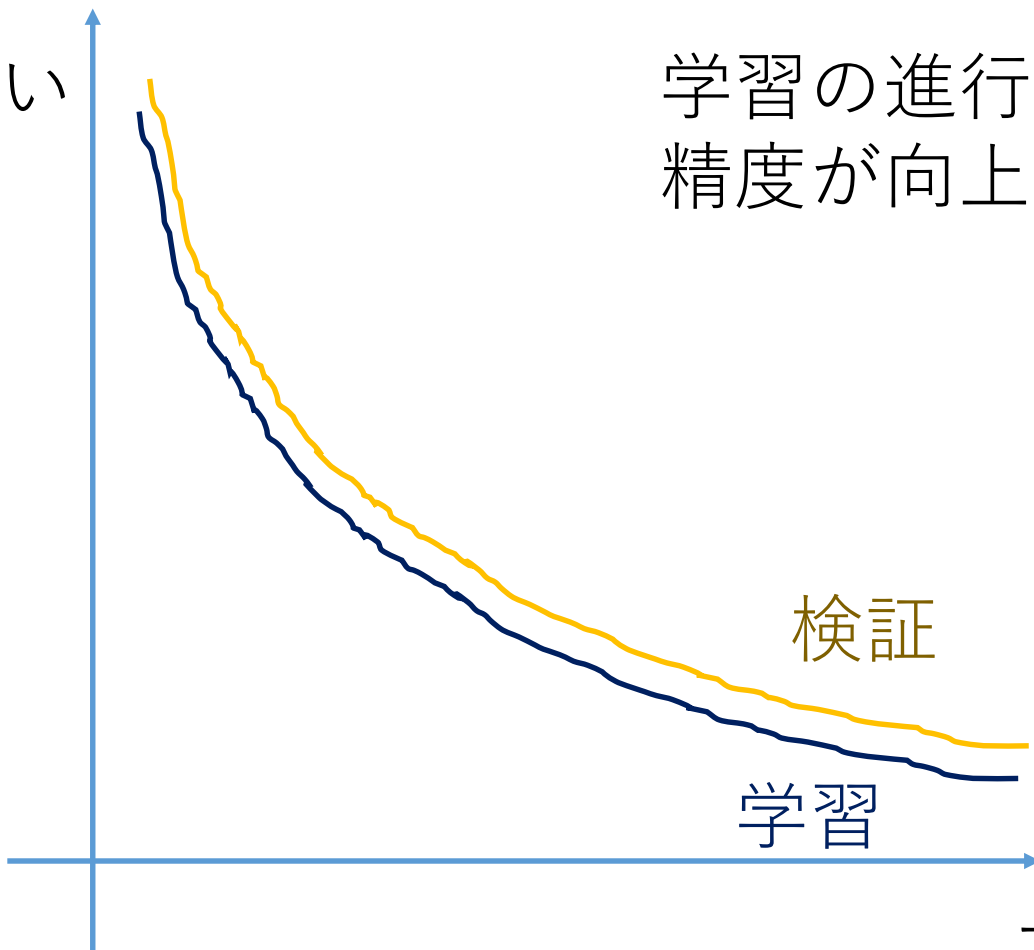
学習の進行とともに
精度が向上

検証

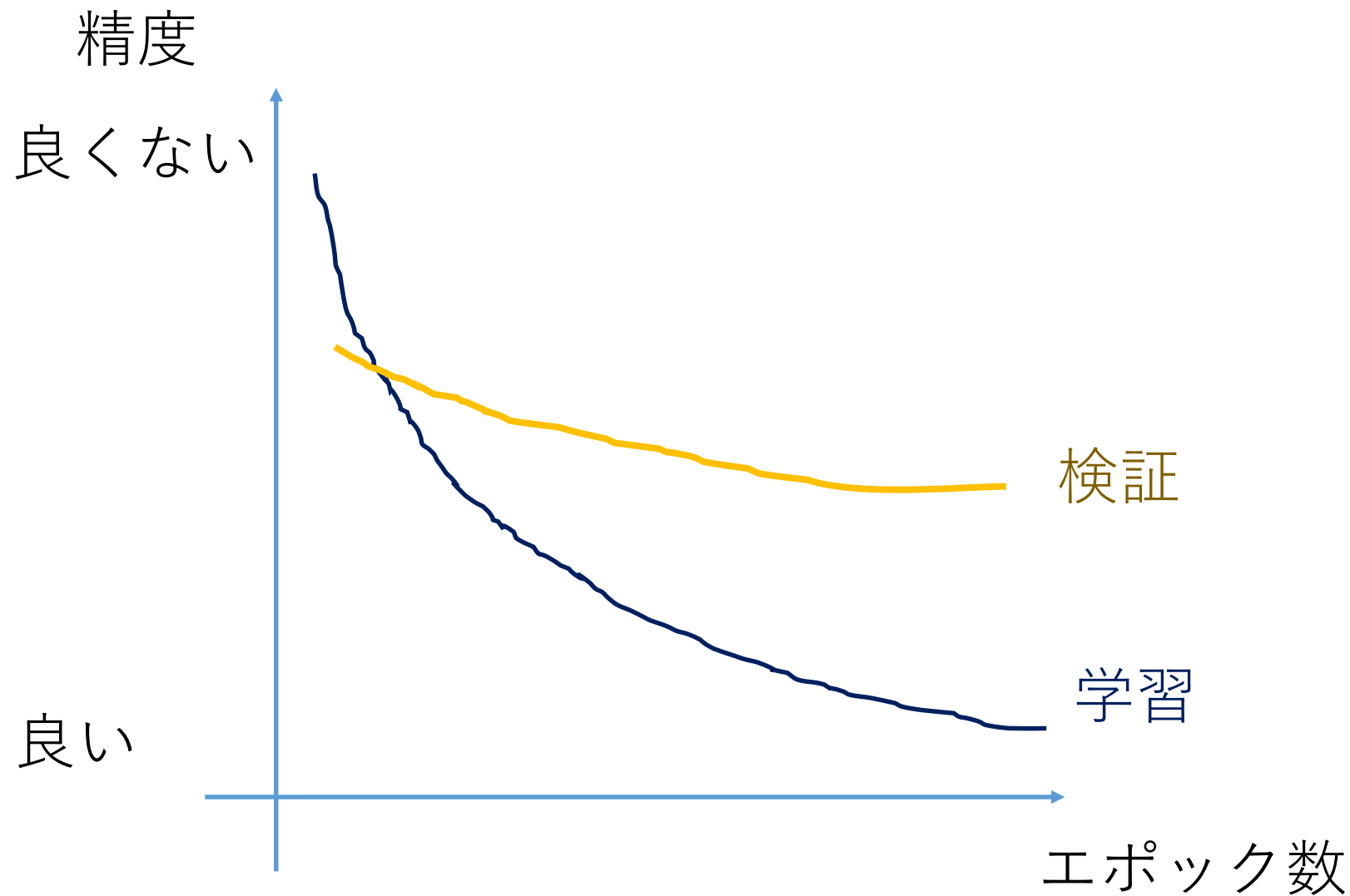
学習

良い

エポック数



過学習あり



手順



① パソコンの Web ブラウザで、次のページを開く

<https://www.tensorflow.org/tutorials>

② 左側のメニューの「Keras による ML の基本」を展開，「オーバーフィットとアンダーフィット」をクリック，「Run in Google Colab」をクリック

The screenshot shows a web browser window with the URL `tensorflow.org/tutorials/keras/overfit_and_underfit`. The page header includes the TensorFlow logo and navigation links like 'インストール', '学ぶ', and 'もっと見る'. A secondary navigation bar contains '概要', 'チュートリアル', 'ガイド', and 'TF 1'. The left sidebar menu is expanded to show 'Keras による ML の基本' (highlighted with a red box), which includes sub-items like '基本的な画像分類', '基本的なテキスト分類', and 'TF Hub によるテキスト分類'. Below these, 'オーバーフィットとアンダーフィット' is also highlighted with a red box. The main content area shows a 'Run in Google Colab' button (highlighted with a red box) and a 'View' button. A blue note box contains text about the quality of the translation and a GitHub repository link.



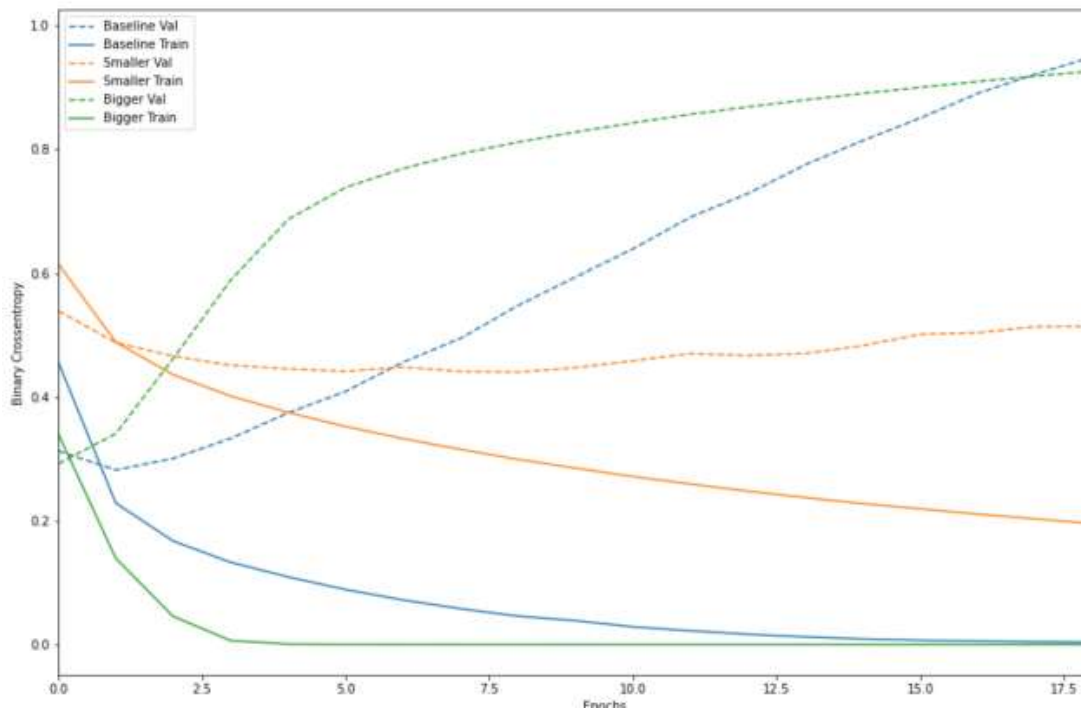
- 3つのニューラルネットワークの学習曲線
ユニット数

Baseline: 1層目は16個, 2層目は16個

Smaller: 1層目は4個, 2層目は4個

Bigger: 1層目は512個, 2層目は512個

ユニット数が多いと過学習が起きやすい



点線は検証

実践は学習



考察の例

次のグラフから次を読み取りなさい

- エポック数はいくつがよさそうか？
- 検証の結果、精度はどれくらいか？
- 過学習は発生しているか、していないか？

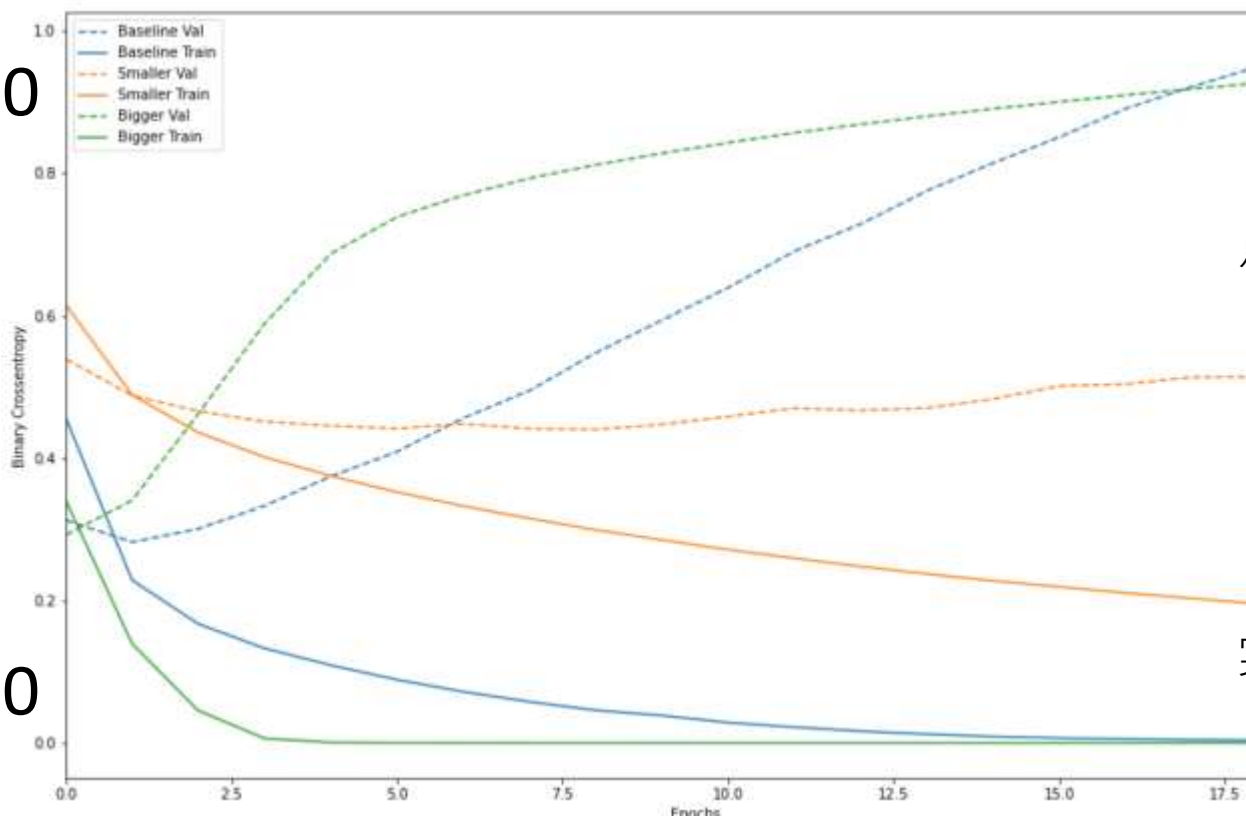
良くない

1.0

精度

0.0

良い



点線は検証

実線は学習

過学習の防止に役立つ技術

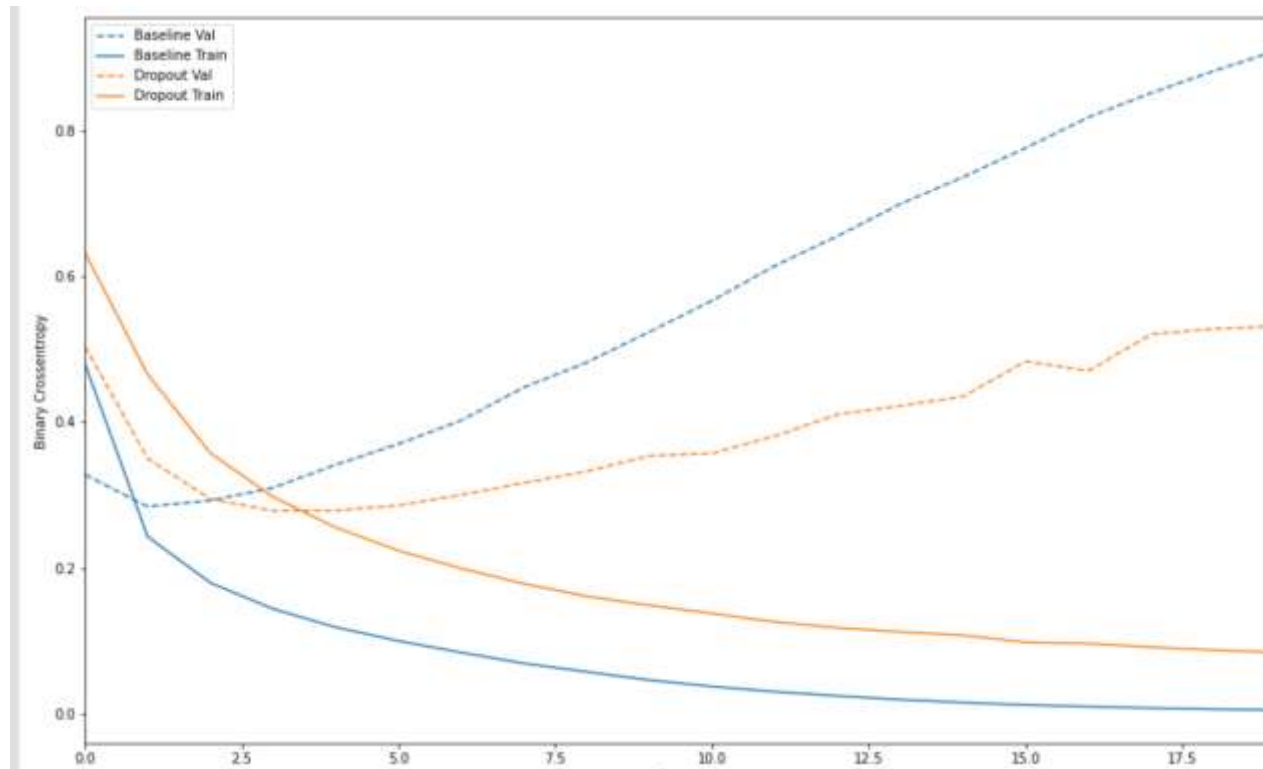


- データの拡張
学習データを増量する

- ドロップアウト
学習の途中で、結合をランダムに無効化する



- ドロップアウト等の技術により，過学習を緩和



青：ドロップアウト等なし オレンジ：ドロップアウト等あり