

# dn-2. ニューラルネットワーク の学習不足と過学習

(ディープラーニング入門演習)

URL: <https://www.kkaneko.jp/ai/dn/index.html>

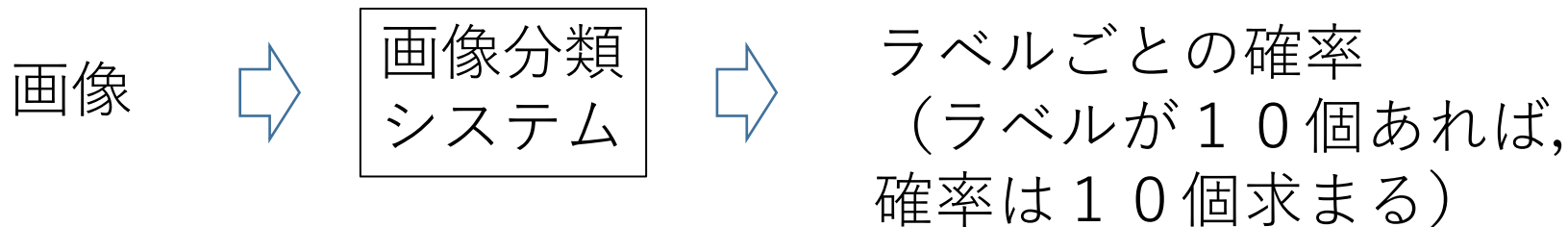
金子邦彦



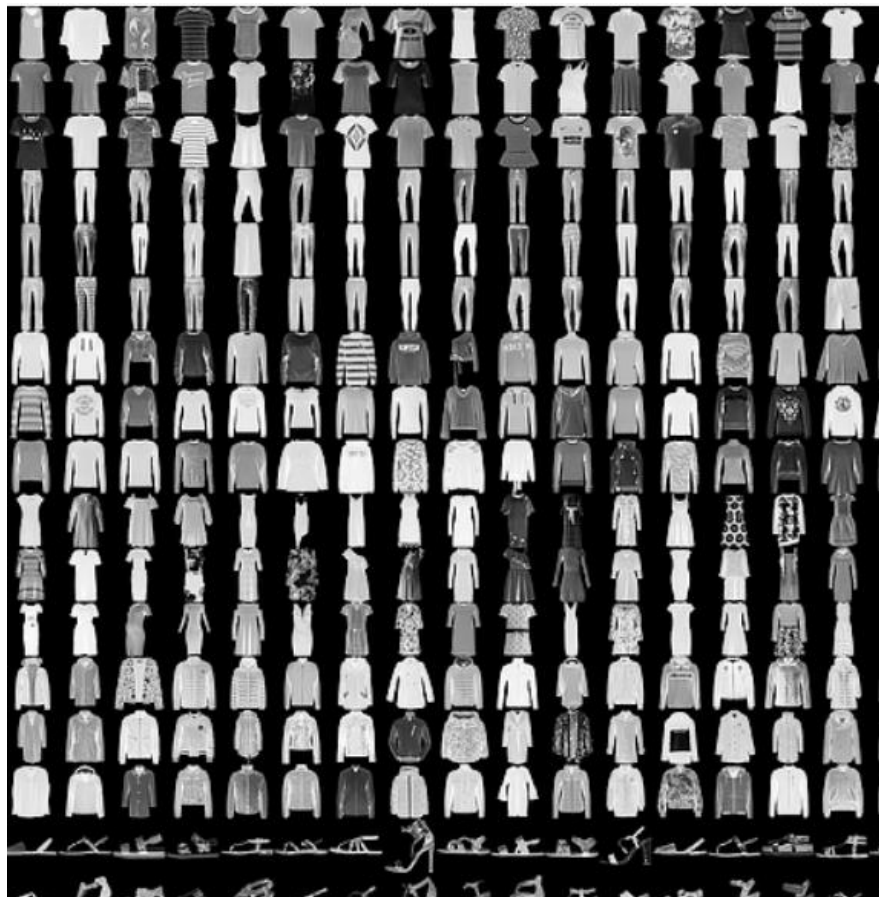
# 人工知能による画像の分類

- 画像分類は、**与えられた画像**に対して、次を得ること

## ラベルごとの確率



# 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。

# 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

ユーザー名

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

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



The screenshot shows the TensorFlow website interface. The top navigation bar includes the TensorFlow logo, a search icon, and dropdown menus for '学ぶ' (Learn), 'API', and 'もっと見る' (See more). Below the navigation bar, there are tabs for '概要' (Overview), 'チュートリアル' (Tutorials), 'ガイド' (Guides), and 'TF 1'. The left sidebar menu is expanded to show 'Keras による ML の基本' (Getting started with Keras), which is highlighted with a red box and an upward-pointing arrow. Underneath, '基本的な画像分類' (Basic image classification) is also highlighted with a red box. On the right side of the page, the 'Run in Google Colab' button is highlighted with a red box. A blue note box at the bottom right contains the following text: '★ Note: これらのドキュメントは ニティによる 翻訳はベストエドキュメントの 最新の状態を 翻訳の品質を向上させるための プルリクエストをお送りく'.

### 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

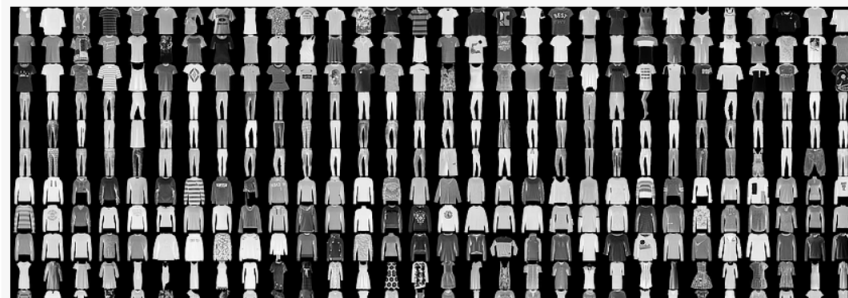
# ヘルパーライブラリのインポート
import numpy as np
import matplotlib.pyplot as plt

print(tf.__version__)
```

Google  
アカウント  
が必要

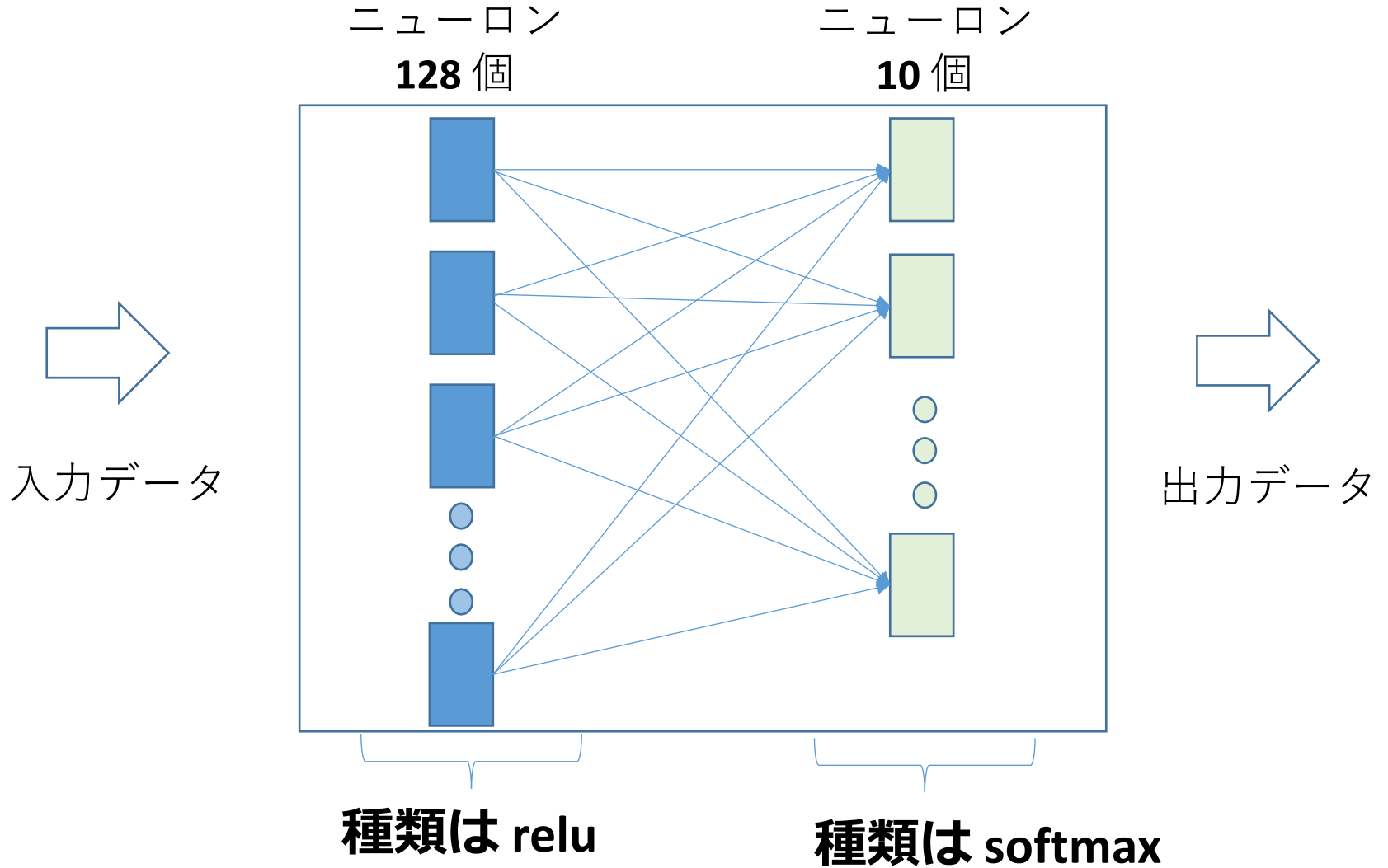
#### ▼ ファッションMNISTデータセットのロード

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の例





# 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**教師データ**を繰り返し使って学習を行う

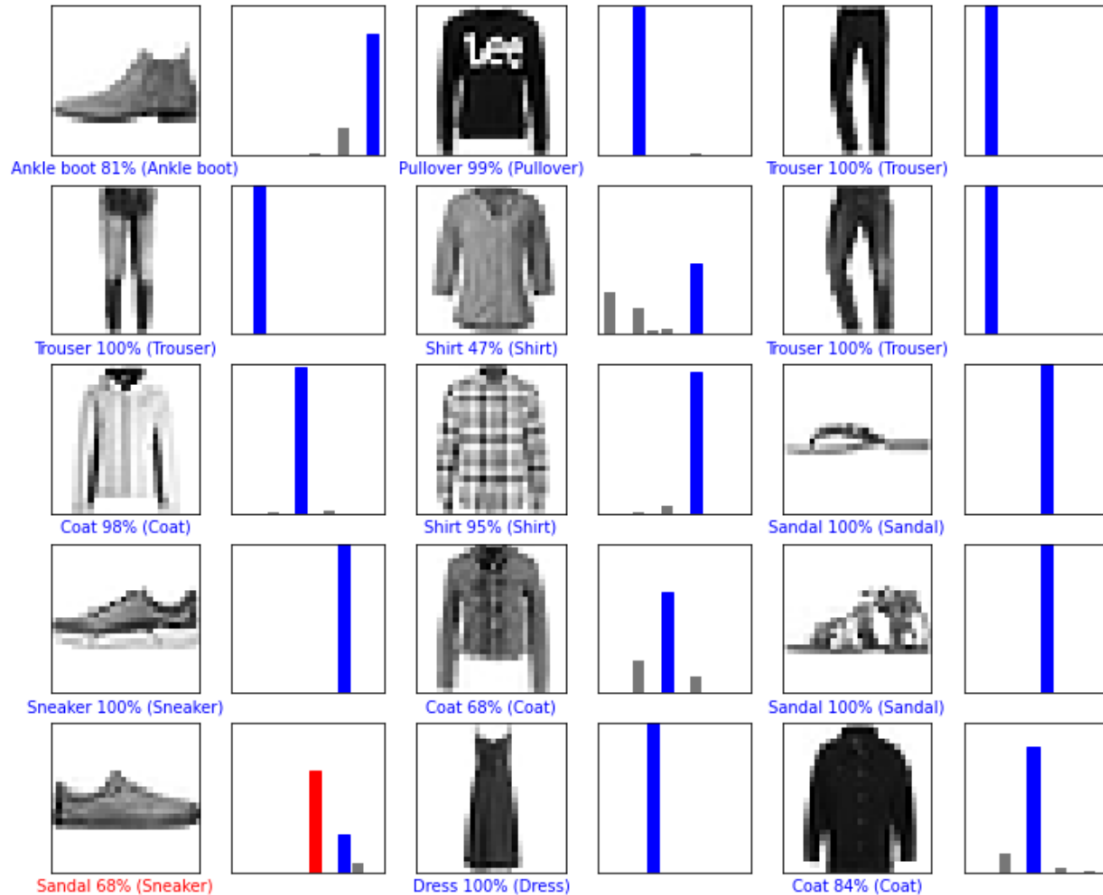
```
[14] model.fit(train_images, train_labels, epochs=5)
```

```
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>
```

繰り返し回数：**5回**

繰り返しのたびに**誤差が減少**（loss の右横の数値）

# ニューラルネットワークによる予測の様子



10種類のどれに分類されたかを棒グラフで表示  
(ラベルが10個あるので、**確率は10個求まる**)

青：正解、 赤や黒：不正解

# 学習不足と過学習

# ニューラルネットワークの学習で気を付けること



- **学習**には大量のデータが必要  
学習の成功のため
- 同じ教師データを使って**学習**を繰り返す  
学習不足の解消
- **学習**の**検証**が必要  
過学習が無いことの確認

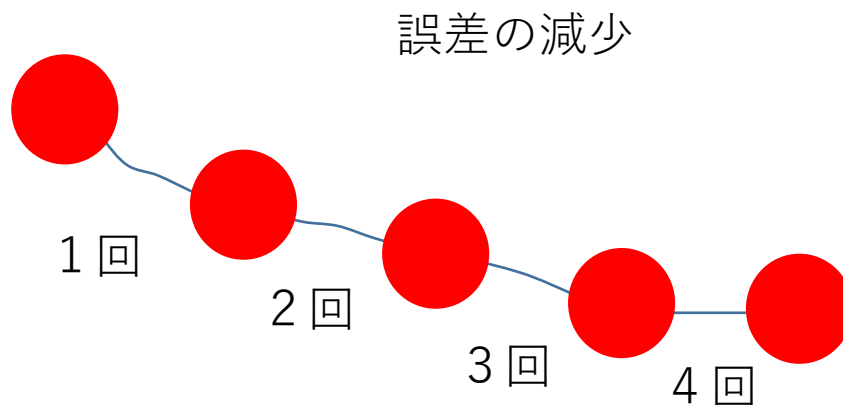
# ニューラルネットワークの学習

- **教師データ**（学習のためのデータ）を使用
- **学習**は**自動**で行われる
  - ① **教師データ**により、**ニューラルネット**を動かし、誤差を得る
  - ② **ニューロン間の結合の重みの上げ下げ**により、**誤差**を減らす（最終層の結果が、手前の層の結合の重みに伝搬することから、フィードバックともいわれる）
- ニューロンの数が増えたり減ったりなどではない
- **誤差が減らなくなったら、最適**になったとみなす

# 学習不足



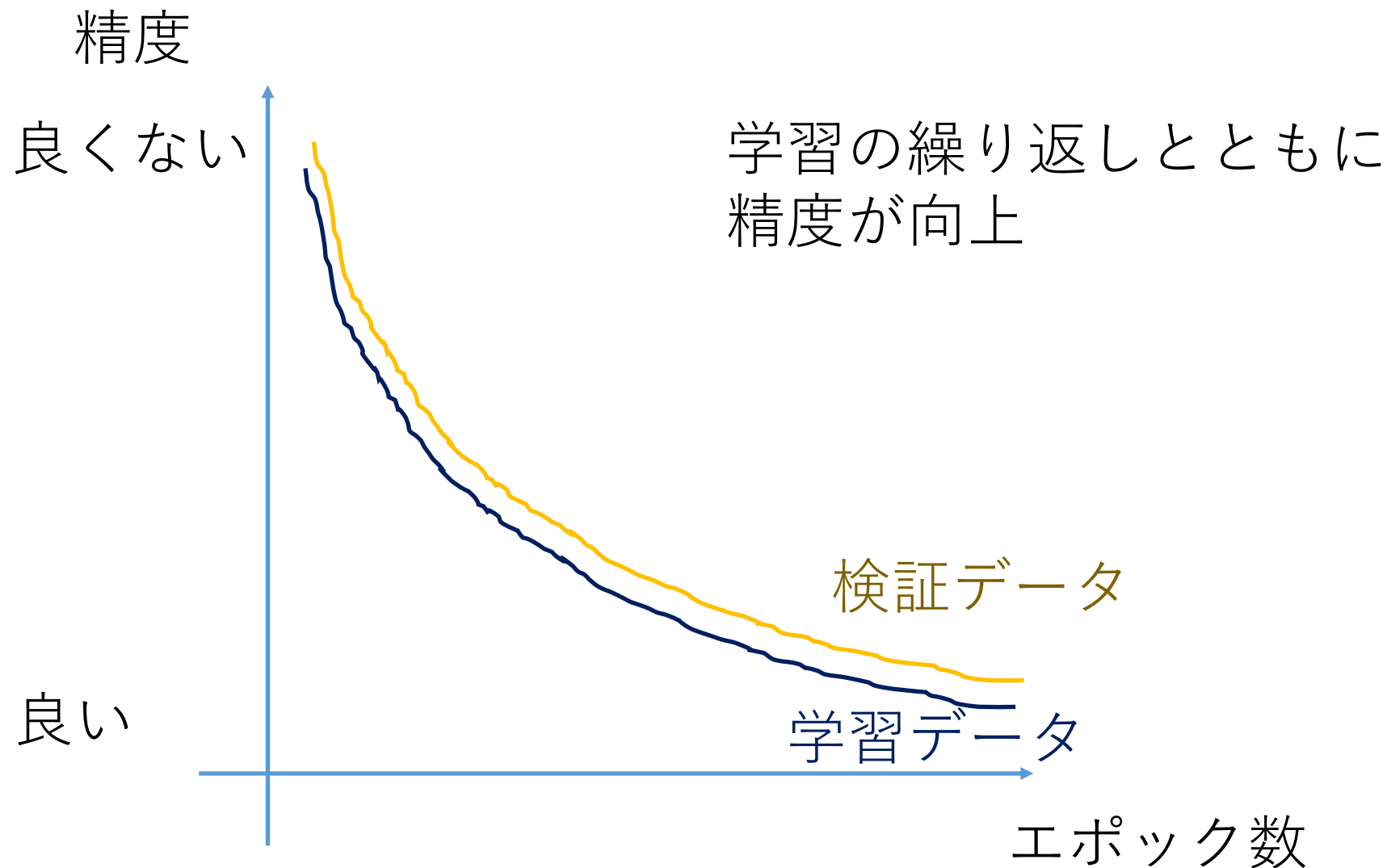
- ニューラルネットワークの**学習**では、学習のためのデータ（**教師データ**）を使う
- **教師データ**を1回使っただけでは、**学習不足**の場合がある
- 同じ**教師データ**を繰り返し使って学習を行う。  
繰り返しながら、誤差の減少を確認



- **教師データ**での学習を終了したとき
- 検証データで**検証**すると、  
学習がうまくいっていないことが分かる場合がある



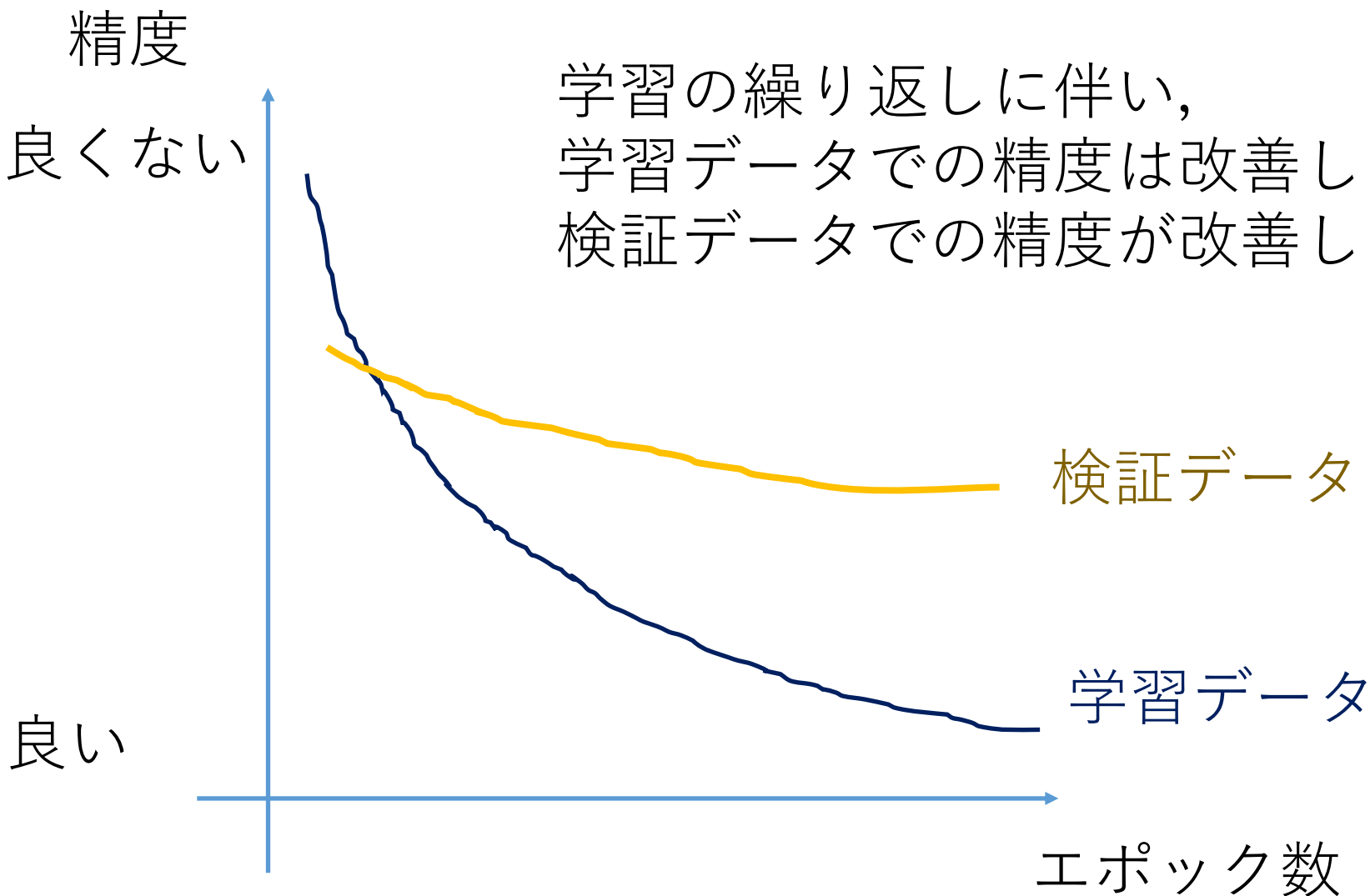
# 過学習なし



# 過学習あり



学習の繰り返しに伴い、  
学習データでの精度は改善しても、  
検証データでの精度が改善しない

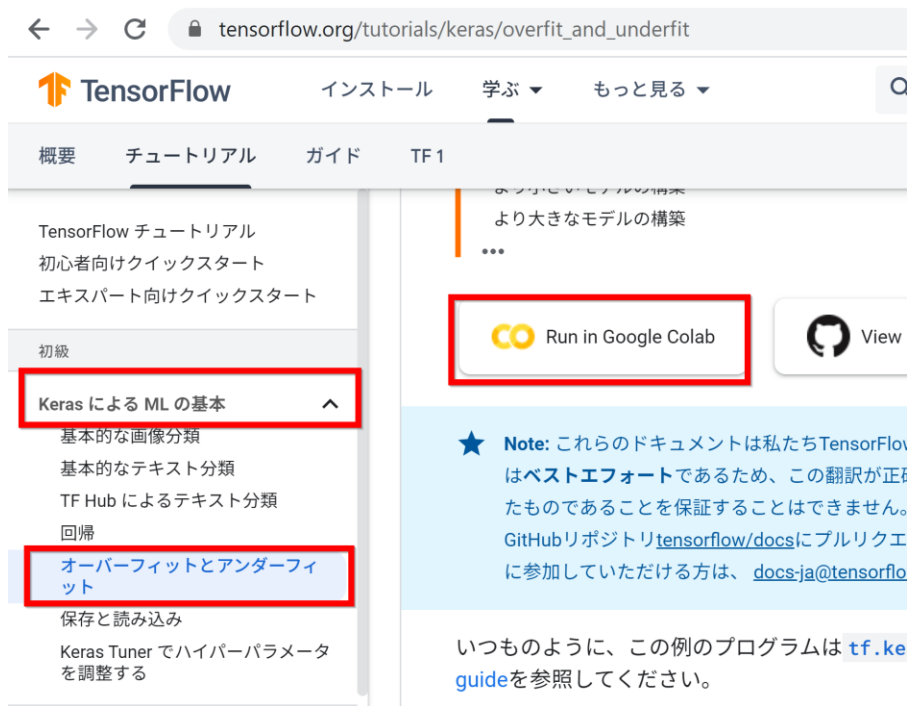


ニューラルネットワークの学習では、  
過学習が発生していないことを確認すること

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 the TensorFlow website at the URL [tensorflow.org/tutorials/keras/overfit\\_and\\_underfit](https://www.tensorflow.org/tutorials/keras/overfit_and_underfit). The left sidebar menu is expanded to show the 'Keras による ML の基本' (Basic ML with Keras) section, which is highlighted with a red box. Under this section, the 'オーバーフィットとアンダーフィット' (Overfitting and Underfitting) item is also highlighted with a red box. In the main content area, the 'Run in Google Colab' button is highlighted with a red box. A blue note box is visible below the button, and a footer note at the bottom of the page states: 'いつものように、この例のプログラムは [tf.keras](#) を参照してください。'

- 3つのニューラルネットワークの学習曲線

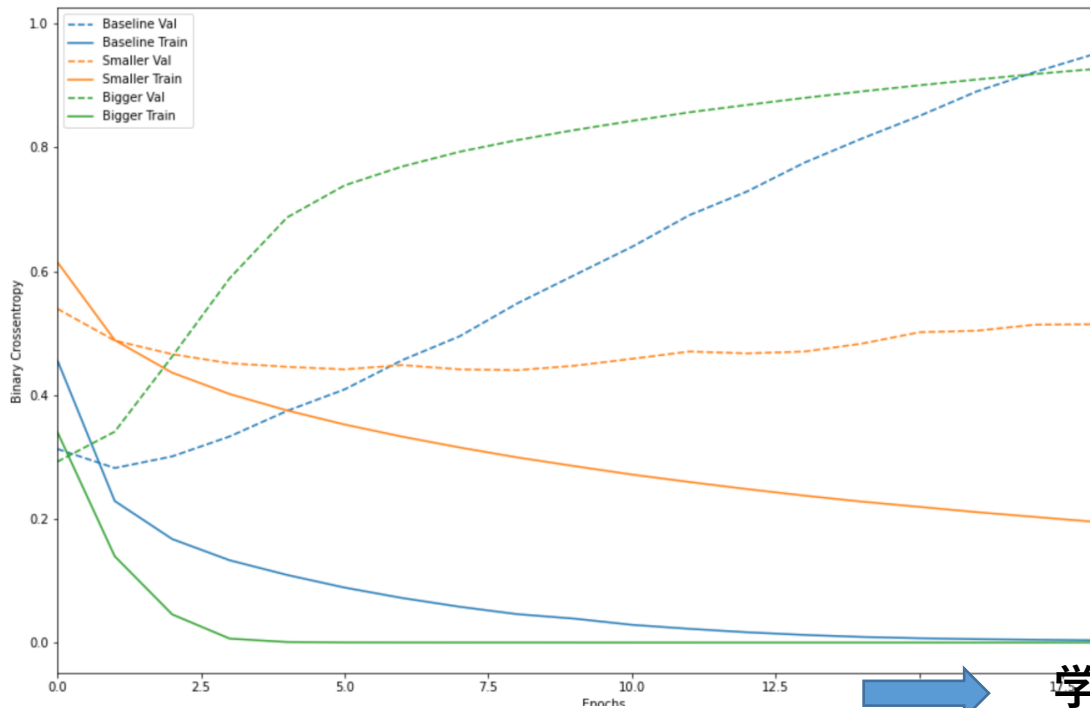
ニューロン数

**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

ニューロン数が多いと過学習が起きやすい



点線は検証データ

実線は学習データ

学習の繰り返し

# 考察の例

次のグラフから次を読み取る

- **学習の繰り返し回数**はいくつがよさそうか？
- **過学習**は発生しているか、していないか？

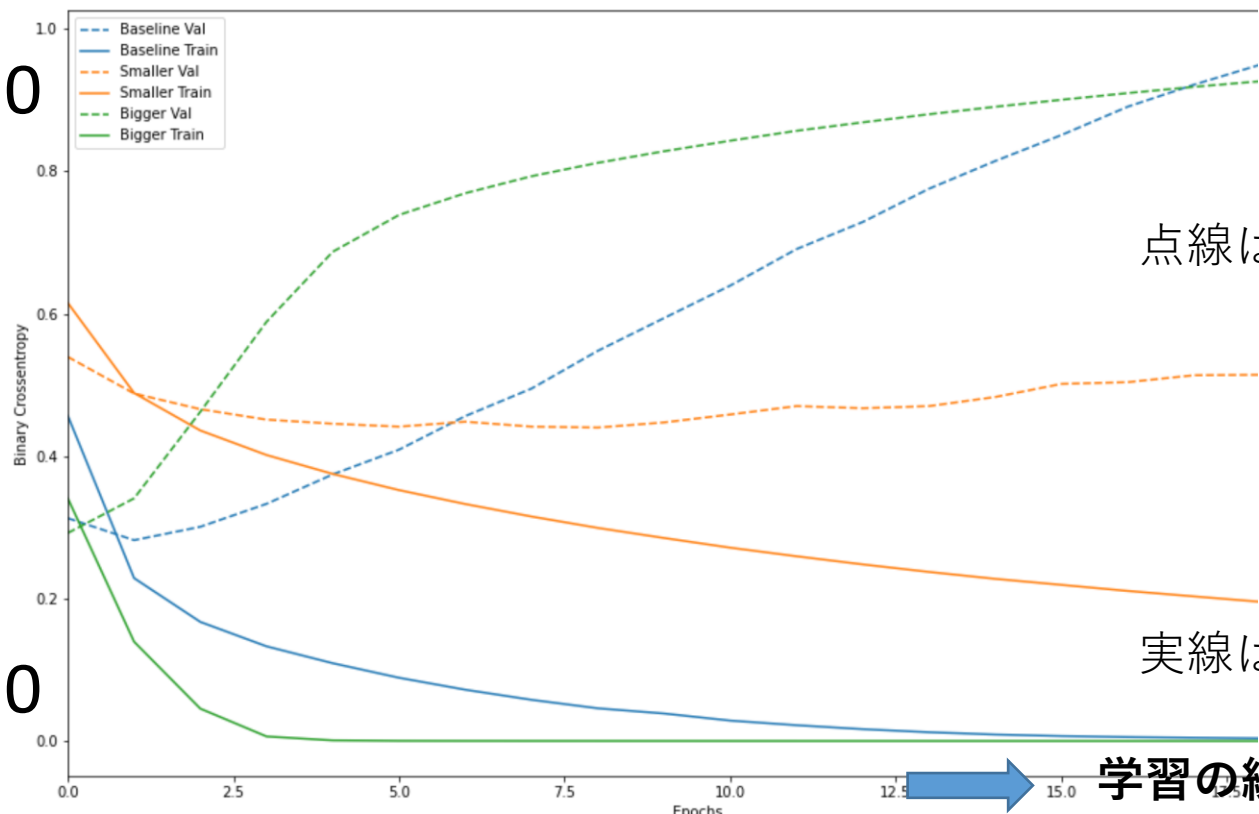
良くない

1.0

精度

0.0

良い



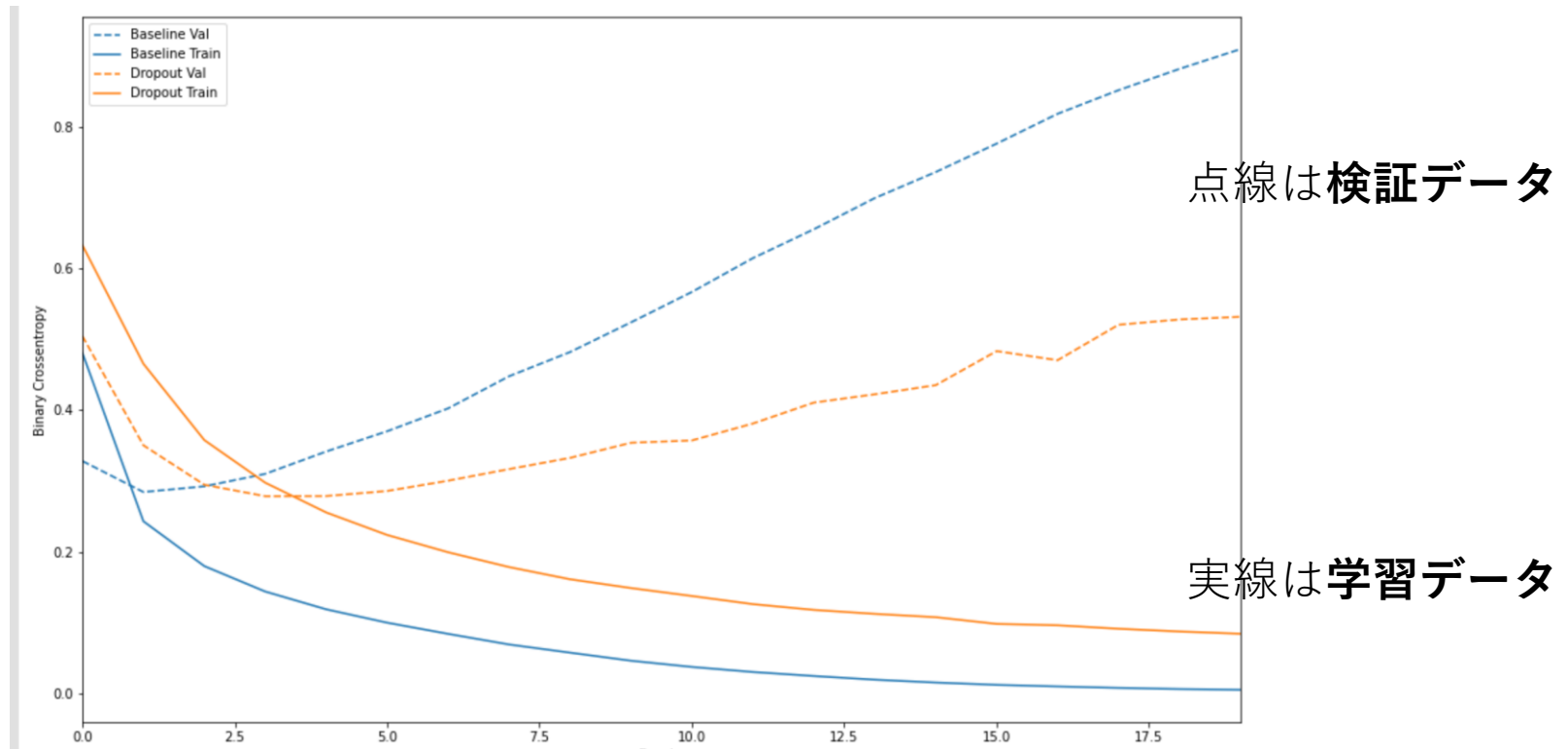
点線は検証データ

実線は学習データ

学習の繰り返し

- データの拡張
  - 教師データの拡張（増量）と再学習による解決**
- ドロップアウト
  - 学習の途中で、**ニューロン間の結合をランダムに無効化すること**で解決
- その他（正則化など）

# • 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