

しおん研-ゼロつくゼミ

～ゼロからつくるニューラルネットワーク～

第1回

本ゼミのテーマ

深層学習を構成するモデル「ニューラルネットワーク」の数理的構造を知る

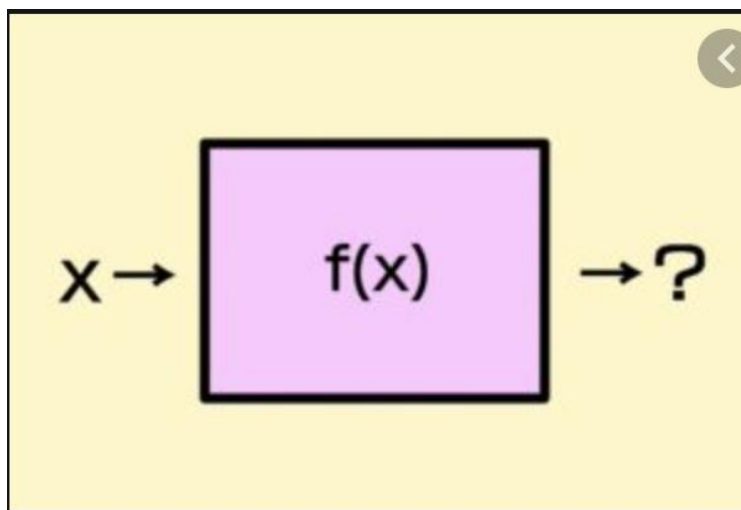
Pythonを用いて、実際にニューラルネットワークを実装する

今回やること

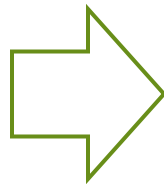
- ▶ ・ 深層学習をふわっと理解する
- ▶ ・ Pythonの基本文法をざっくりやる

深層学習ってなに？

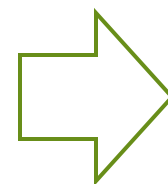
- ▶ 深層学習 : .とても賢い関数を作るシステム (by今泉doc)
- ▶ 関数 : なんか入力したら出力を返してくれるもの



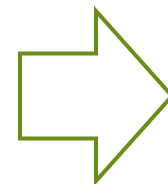
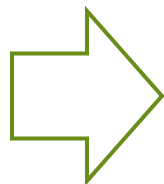
例：猫の画像から、猫であるかどうかを
判定する深層学習



関数

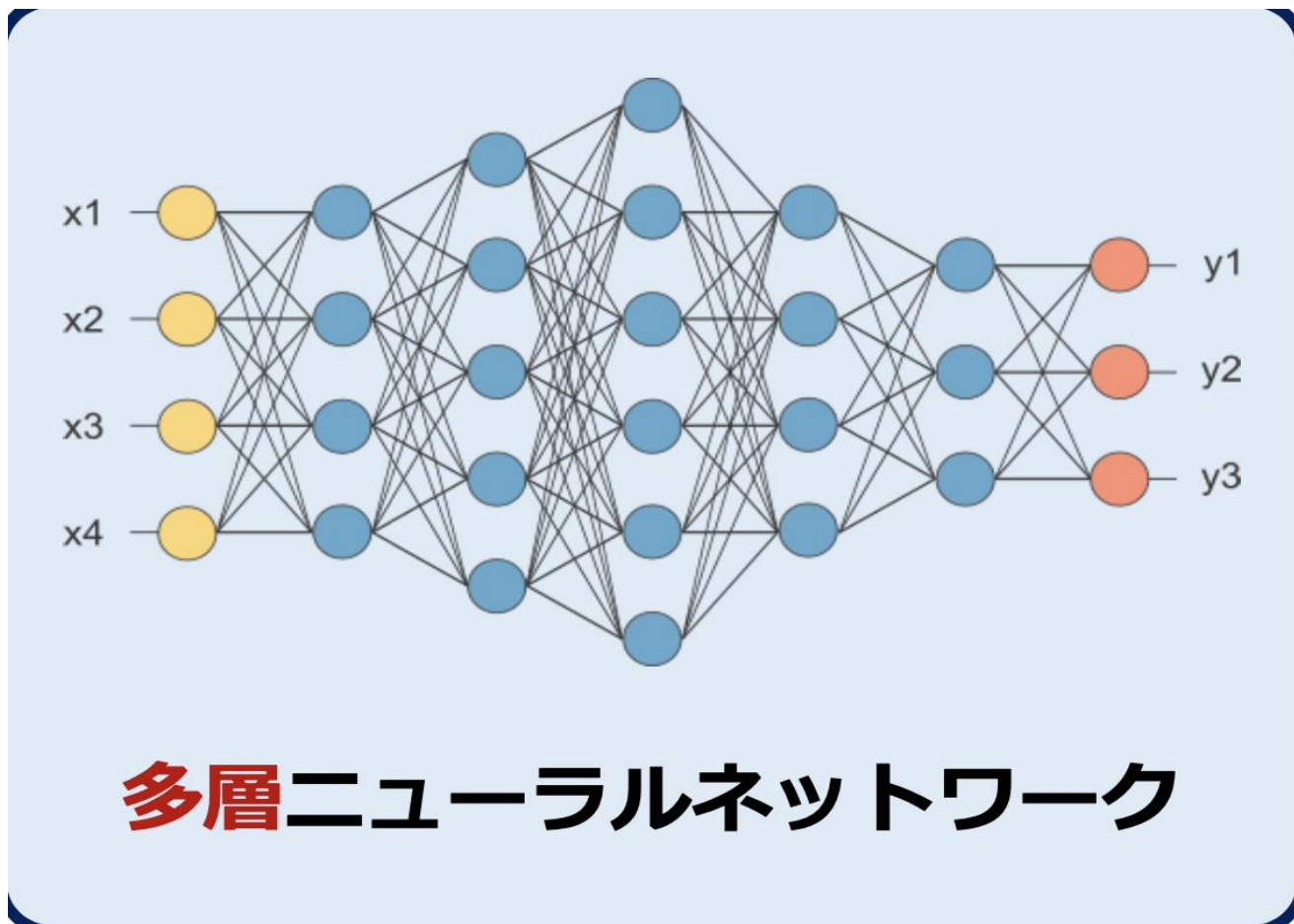


猫です



猫じゃない
です

(多層) ニューラルネットワーク：深層学習を構成する関数モデル！



深層学習でできること

活用例

- ▶ ・ 分類
- ▶ ・ 生成
- ▶ ・ 変換
- ▶ ・ 敵対的学習

活用分野

- ・ 自然言語処理
- ・ 画像認識
- ・ 音声認識
- ・ レコメンデーション

ニューラルネットワークの実装：Python によるプログラミング

- ▶ ・ Python ：データサイエンスや深層学習などに適したプログラミング言語。ライブラリがいっぱいあるのがいいところ。
- ▶ ・ Google Colaboratory（Googleが提供している、クラウドで実行可能なプログラミング開発環境）で掴みと基本文法を勉強します。
- ▶ Google ColaboratoryはJupyter Notebookのクラウド版。
- ▶ ・ Python入門の教科書：
<https://utokyo-ipp.github.io/index.html>

深層学習でよく使うライブラリ

- ▶ Numpy
- ▶ Matplotlib
- ▶ Keras

<https://udemy.benesse.co.jp/data-science/ai/keras.html>

機械学習を効率よく行うためのライブラリ。

- ▶ TensorFlow

<https://www.sejuku.net/blog/38134>

Keras, TensorFlowを使えばニューラルネットワークを簡単に実装できますが、今回はその中身を知ることが目的なので、これらのライブラリは使わずに実装します。

Scikit-Learn

Pandas

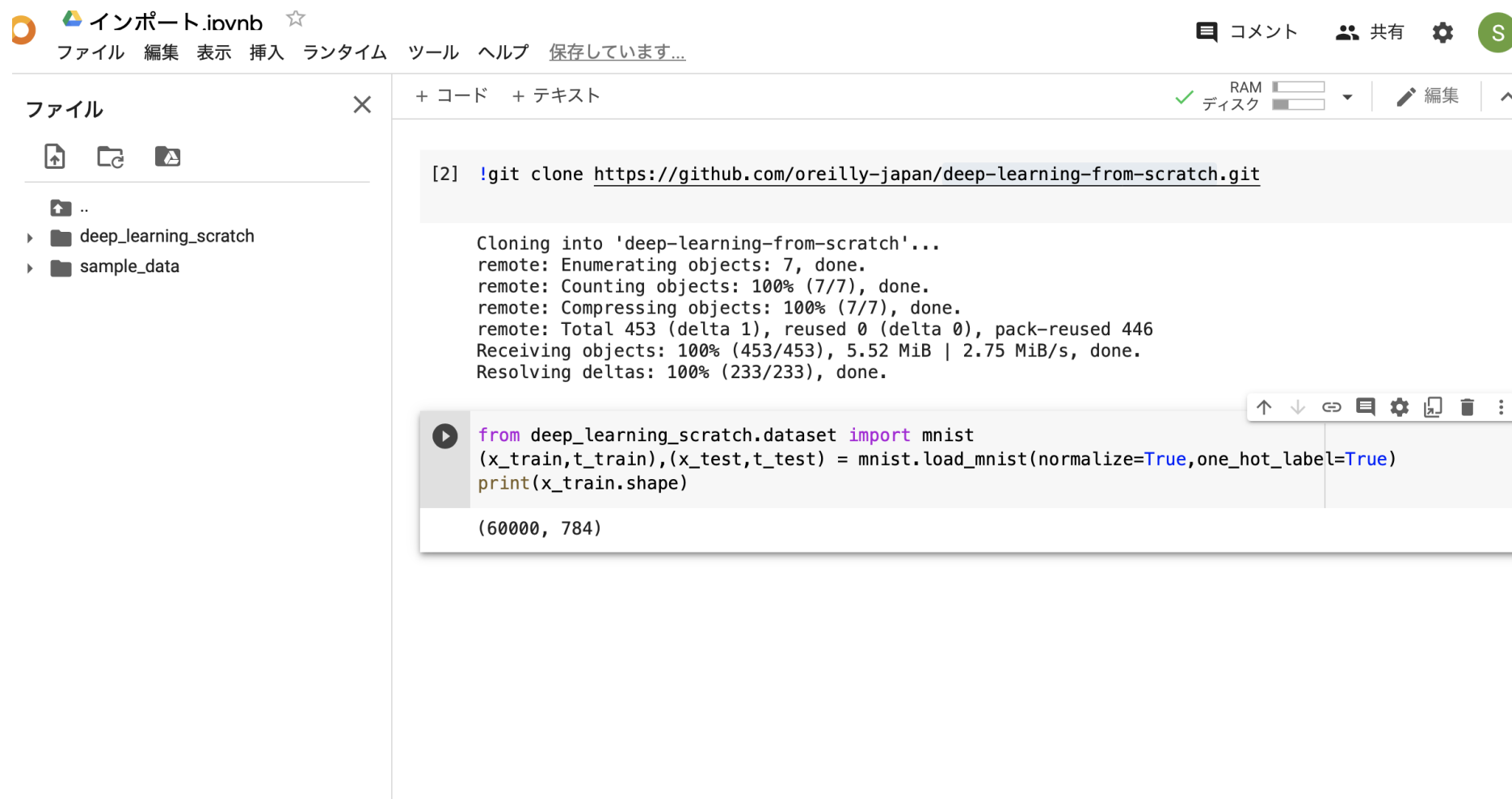
Scipy.sparse

応用問題

https://drive.google.com/file/d/1ug8N_o8Meh8LuKl6Qy1DkHWdmRE0nCkP/view?usp=sharing

課題 1 を解いてスレッドに上げてください

今回使う参考書の添付コードをGithubからクローンしてください。（次回やります）



The screenshot shows the JupyterLab interface. At the top, there's a header with the JupyterLab logo, the text "インポート.ipynb", a star icon, and a menu bar with "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", "ヘルプ", and "保存しています...". On the right side of the header, there are icons for "コメント", "共有", "設定", and a user profile icon labeled "S".

Below the header, there's a sidebar on the left labeled "ファイル" with a close button "X". It shows a file explorer with a home icon, a refresh icon, and a trash icon. Below these, there's a folder icon and a list of files: "..", "deep_learning_scratch", and "sample_data".

The main area is divided into two panes. The top pane is a terminal window with the title "+ コード + テキスト". It shows the command `[2] !git clone https://github.com/oreilly-japan/deep-learning-from-scratch.git` and its output:

```
Cloning into 'deep-learning-from-scratch'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 453 (delta 1), reused 0 (delta 0), pack-reused 446
Receiving objects: 100% (453/453), 5.52 MiB | 2.75 MiB/s, done.
Resolving deltas: 100% (233/233), done.
```

The bottom pane is a code editor with the title "+ コード + テキスト". It shows the following Python code:

```
from deep_learning_scratch.dataset import mnist
(x_train, t_train), (x_test, t_test) = mnist.load_mnist(normalize=True, one_hot_label=True)
print(x_train.shape)
```

Below the code, the output is displayed: `(60000, 784)`. The code editor has a toolbar with icons for undo, redo, copy, paste, search, and other editing functions.