

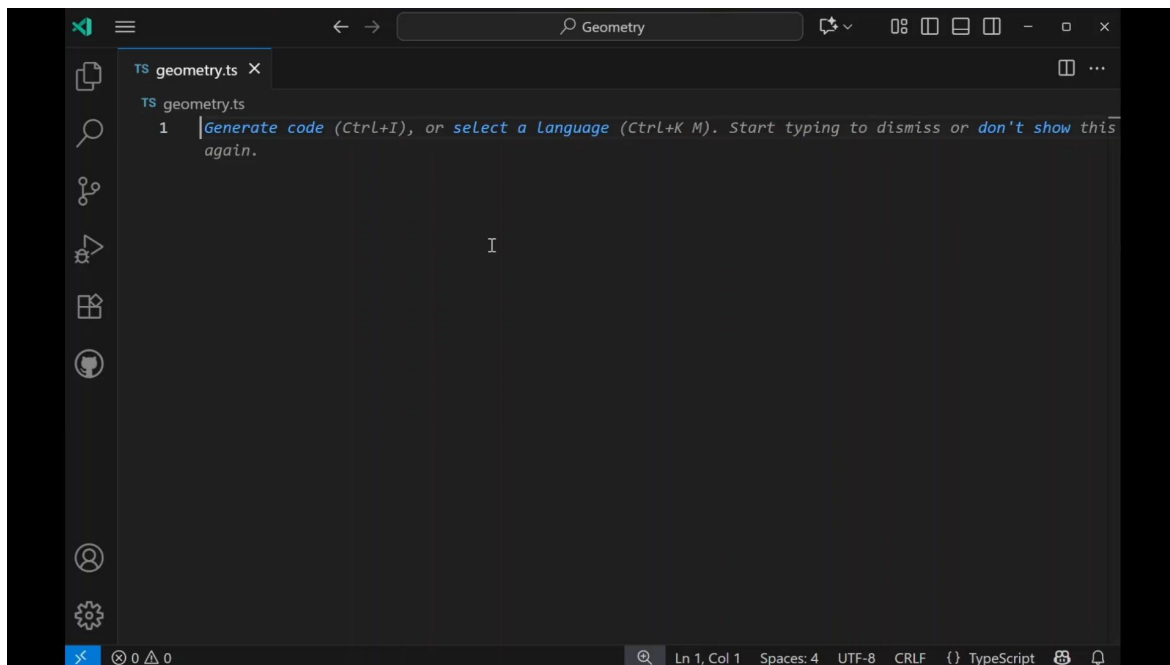


慶應義塾大学理工学部  
**機械学習基礎**

第2回 LLMによる問題解決支援

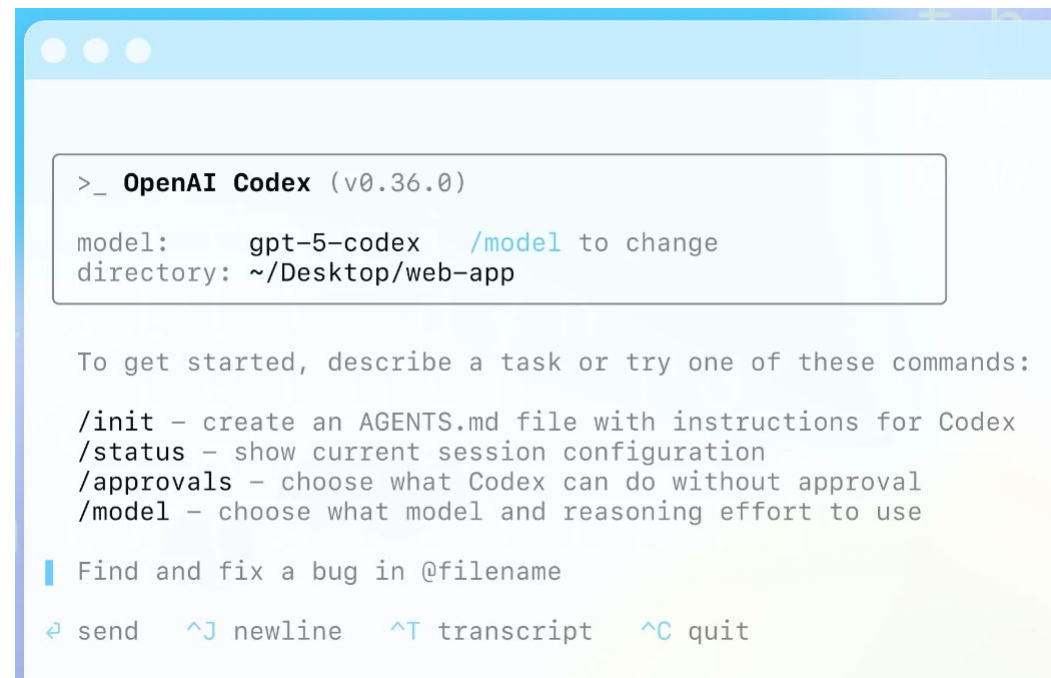
情報工学科 教授 杉浦孔明  
komei.sugiura@keio.jp

# コーディングにおけるLLM活用例： 実行・修正・テストコード作成・説明等、急拡大



## ■ GitHub Copilot in VS Code

<https://code.visualstudio.com/docs/copilot/ai-powered-suggestions>



## ■ Codex CLI

<https://developers.openai.com/codex/cli/>

# 本講義の到達目標と今回の授業の狙い



## 本講義の到達目標

- DNNの基礎理論と実装の関係を理解する
- 種々のDNNをコーディングできる

## 今回の授業の狙い

- ChatGPTを用いたコーディング習得方法を学ぶ

- 出席確認： K-LMS上の機械学習基礎のMainページへアクセス



# ChatGPTを用いて コーディングを習得する

---



# 何でも聞ける「個人チューター」の必要性



## これまでの「機械学習基礎」受講者の意見

- 「Pythonのオブジェクト指向に関する記法に馴染みがない」
- 「調べ方がわからない」

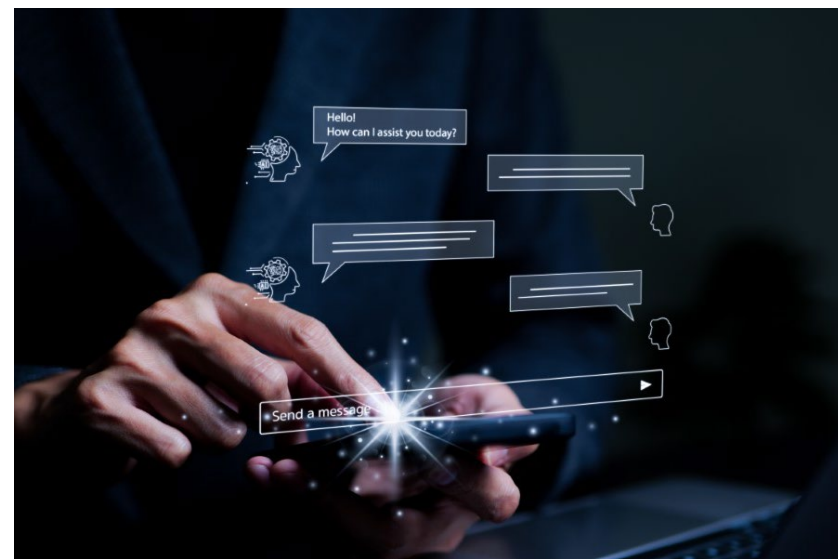
## 「個人チューター」の背景

- プログラミング入門者には個人チューターが効果的だが、受講者に対してTAが少なすぎる（40:1） 😞
- 上級者であっても、初めて使うライブラリには説明が必要

# ChatGPT [OpenAI, 2022/11/30]



- ChatGPT：大規模言語モデル（Large Language Model; LLM）を用いたチャットボット
  - 言語モデル：単語列の生成確率を推定するモデル
  - 「ChatGPT」はサービス名であるが、口語では大規模言語モデルを指すことが多い
- 以前のチャットボット
  - 数十年前から何度も盛り上がるものの、キラーアプリに欠けるため必需品とならず
- 関連技術を今後の授業で紹介



# ChatGPTを 個人チューターとして使おう

- おそらく本科目が日本で初めて  
ChatGPTを用いたコーディング習得  
を大学科目で正式に開始  
(2023/1/6)



杉浦孔明研究室@慶大 @keio\_smilab · 1月7日

#情報工学科 B2の講義で、[#ChatGPT](#) を使ってPyTorchを学んでもらいました。

受講者全員に「理解度に応じて」「コードを説明してくれる」「個人チューター」を用意できる点は、革新的な教育方法ですね。

案の定「レポート作成に使いたい」という感想はありましたが、もっと大きい変革だと思います。

- 将来的に当然のスキルになるはず

## ChatGPT



### Examples

"Explain quantum  
computing in simple terms"  
→

"Got any creative ideas for a  
10 year old's birthday?" →

"How do I make an HTTP  
request in Javascript?" →



### Capabilities

Remembers what user said  
earlier in the conversation

Allows user to provide  
follow-up corrections

Trained to decline  
inappropriate requests



### Limitations

May occasionally generate  
incorrect information

May occasionally produce  
harmful instructions or  
biased content

Limited knowledge of world  
and events after 2021

ChatGPTと会話することでPyTorchを学んでみたいと思います。私はPython初心者です。オブジェクト指向プログラミングもわかりません。まずはPyTorchの初心者用プログラムについて教えてください。

# ChatGPTとの対話例



1. ChatGPTと会話することでPyTorchを学んでみたいと思います。私はPython初心者です。オブジェクト指向プログラミングもわかりません。まずはPyTorchの初心者用プログラムについて教えてください。
2. PyTorchをインストールしました。MNIST画像データセットをダウンロードするにはどうすれば良いですか？
3. ダウンロードできました。データを加工して、モデルに入力する形にするにはどうすれば良いですか？
4. 変換できました。次に多層パーセプトロンで画像分類を行うにはどうすれば良いですか？
5. 最初の行にあるピリオドはどのような意味ですか？



# 杉浦が監修した「ニュートン別冊 が出版されました（2024/2/1）」

## ChatGPTの未来」



# コーディング習得でのChatGPTのメリット・デメリット



- ビギナーほど説明が必要 → ChatGPTは初級コードが得意😊
  - 高度知識に比べ初級知識に関する学習サンプルが多いためと予想
- 公式ドキュメント（含教科書）は正式な知識源だが、あらゆるレベルに合わせたドキュメントを書くことは不可能
  - ⇒ ChatGPTの回答は正しいとは限らないが、**ChatGPTでは個々のレベルに合わせた説明が可能**
- TAに聞くのは効率的だが気を使う
  - ⇒ ChatGPTはいつでも・どこでも・気を使うことなく質問可能

# ハルシネーションとコーディングへの影響



## ■ ハルシネーション（錯誤）

- いかにも正しそうだが実際には誤りである内容の生成
- ハルシネーションの影響を軽減するには？
  - 調べ物： ハルシネーションかどうかわからないこともある😞
  - コーディング： **コンパイラのエラーメッセージ**をもとに間違いを修正しやすい😊
- 教科書で調べる、検索する、人に聞く、をバランスよく組み合わせるのが良い

# コーディング用プロンプトのコツ



## ①「○○にわかるように」

初心者・大学生・高校生・小学○  
年生等の単語を○○に入れる

## ②条件を細かく指定

Pythonのコード、コメントを多く、  
○○ライブラリを使う/使わず、等

## ③提案されたコードでうまくいかない場合→エラーメッセージそのものをプロンプトに入力

文脈に合わせて解決策を出力する



はじめまして！ PyTorchを使ってニューラルネットワークを構築する方法を簡単に説明します。

まず、PyTorchをインポートする必要があります。次に、ニューラルネットワークを構築するために、torch.nn モジュールをインポートする必要があります。次に、ニューラルネットワークを構築するために必要な畳み込み層や全結合層を定義します。最後に、ニューラルネットワークを訓練するために、最適化アルゴリズムを定義します。

以下に、PyTorchを使ってニューラルネットワークを構築するコード例を示します。この例では、MNISTデータセットを使用して手書き数字を分類するニューラルネットワークを構築しています。

Copy code

```
import torch
import torch.nn as nn # PyTorchのニューラルネットワーク用のパッケージをインポート

# MNISTデータをロードするためのtorchvision.datasetsパッケージをインポート
import torchvision.datasets as datasets

# データをロードするためのtorchvision.transformsパッケージをインポート
import torchvision.transforms as transforms

# ニューラルネットワークを定義するためのクラスを定義
class NeuralNetwork(nn.Module):
    def __init__(self):
        super(NeuralNetwork, self).__init__()
        self.conv1 = nn.Conv2d(in_channels=1, out_channels=32,
                                kernel_size=3, stride=1, padding=1)
```

# プロンプトエンジニアリング： 望ましい出力を得るために指示を設計・改良する作業



- 背景
  - プロンプトで出力を制御できるAIの台頭
- 今後
  - ChatGPT等が日常で使われるようになり、プロンプトエンジニアリングのスキルが重要になる



# プロンプトエンジニアリングスキルを身につけるには？



## プロンプト集を作る

- 有益だったプロンプトをメモ
- 使う頻度が高いほど上に来るようにランキング

## 有益なプロンプトを調べる

プロンプトエンジニアリング講座、  
検索、周囲と情報交換

### 前提

- ChatGPTは、今後当たり前のスキルになる
  - 今後、他大学でもChatGPTを装備した学生が多数発生する。
    - ちょっとしたメールを書く時間についても、装備した人で差が生まれる。装備せずに戦うのはナンセンス。
  - J科B2には、全員装備できるように授業で練習させた
  - B2でも使うぐらいなのだから、孔明研ではもっとディープに
- 【締切1/15】 ChatGPTを**自分で使用**し、有益だった例を書いてください
  - その際のChatGPTへの指示文をコピー
  - 名前の下に書いてください
- アカウント作成方法
  - <https://drive.google.com/drive/u/1/folders/10Xyt-SY71cZdOPLKHG>

### 1. テンプレ

- a. PyTorchの勉強
  - i. ChatGPTと会話することでPyTorchを学んでみたいとPython初心者です。オブジェクト指向プログラミング。まずはPyTorchの初心者用プログラムについて教
- b. データセット（造語で良い）の名称とか、モジュールの名称
  - i. 日本語（母語）で造語を作るのも大変なのに、英語でが大きかった
  - ii. 今後はChatGPTと対話して候補を5個ぐらい作って選
  - iii. I want you to create a new acronym based on mu

# コーディング習得以外でも大規模言語モデルを学ぶことは有益



2023年6月のサム・アルトマン氏の講演（於三田キャンパス）

「みなさんは革新期に居合わせた幸運な世代であり、このような世代は少ない。ライフプランや技術についてまだイメージできなくても、大規模言語モデル技術を学び上達することは将来きっと価値がある」

革新期を捉えた成功の例：  
電話、半導体、Internet等





# ChatGPTを使う際の 注意点

---





# 大規模言語モデルによるコード生成で起こった問題の例 (2022/10 GitHub Copilot)



- Texas A&M Univ.のTim Davis教授：「自分が著作権を有するコードを、attributionやLGPL表示なしでCopilotが生成した」
- プロンプト：「sparse matrix transpose, cs\_」
- 感想：確かに多くは類似
  - コメントまでコピーされている点は類似とみなされてもやむを得ない



Tim Davis  
@DocSparse

Curious about the "cs\_" in the prompt, I tried no function name, just "/\* sparse matrix transpose in the style of Tim Davis \*/". And got my code again, slight tweaked. So the @github AI "knows" this is my code.

Try it. Compare with [github.com/DrTimothyAlden...](https://github.com/DrTimothyAlden...)  
ツイートを翻訳

```
1 /* sparse matrix transpose in the style of Tim Davis */
2
3 #include "cs.h"
4
5 cs_t* cs_transpose(const cs *A, int values)
6 {
7     int p, q, j, *Cp, *Cl, n, m, *w;
8     double *Cx, *Cy;
9     cs_t C;
10
11     if (!CSC(A)) return (NULL); /* check inputs */
12     m = A->m; n = A->n; Cp = A->Cp; Cl = A->Cl; Cx = A->Cx;
13     C = cs_spalloc(m, n, Cp, n), values && Cx != NULL, 0); /* alloc result */
14     w = cs_malloc(n, sizeof(int)); /* get workspace */
15     if (!C) return (cs_done(C, w, NULL, 0));
16     if (!C) return (cs_done(C, w, NULL, 0));
17     Cy = C->Cx; Cl = C->Cl; Cp = C->Cp;
18     for (p = 0; p < Cp[n]; p++) w[Cl[p]]++; /* row counts */
19     cs_csum(Cp, w, n); /* row pointers */
20     for (j = 0; j < n; j++)
21     {
22         for (p = Cp[j]; p < Cp[j+1]; p++)
23         {
24             Cl[q = w[Cl[p]]++] = j; /* place A(i,j) as entry C(j,i) */
25             if (Cx) Cy[q] = Cx[p];
26         }
27     }
28     return (cs_done(C, w, NULL, 1)); /* success; free w and return C */
29 }
30
31 =====
32
33 #include "cs.h"
34 cs_t* cs_transpose(const cs *A, int values)
```

午後10:06 · 2022年10月16日

<https://twitter.com/DocSparse/status/1581632706693079042/photo/1>

# 注意：リテラシーや法的問題



- プロンプトに含めるべきではない情報に注意
  - 個人情報、パスワード、APIキー、機密情報等
  - 「公に書けない情報」を入力してはならない
- 他者の著作権を侵害してはならない
  - 例：「生成されたコードを公開したが、後になって他者の著作物と著しく似ていることがわかった」という状況は許されない
- 利用規約に反する行為
  - 利用規約が書かれたページがサービスごとに存在するので読むこと

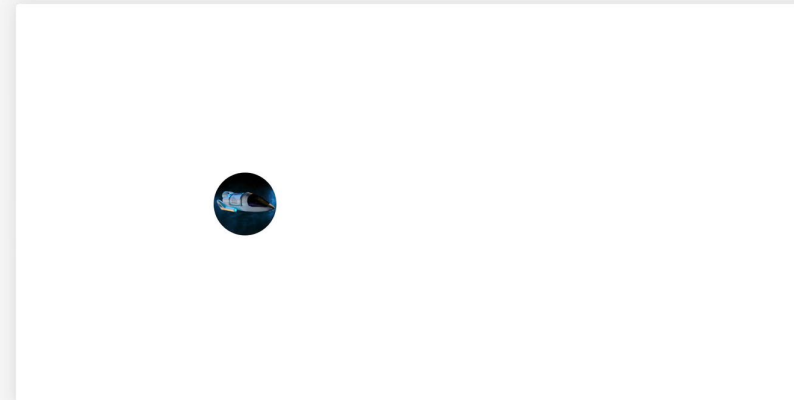


# ChatGPTをコーディング習得補助に使うのであって、 コードをすべて生成させるわけではない



- 大規模言語モデルのみで商品レベルのプログラムを簡単に作ることはできない
- 例えばゲームを作るとして、「シューティングゲームなのか、2次元なのか、ボス敵はどんな形か、...（仕様）」が明確でなければ、良いゲームは作れない

## ■ OpenAI CodeX



Now set background to the color of space.



```
/* Animate the rocketship horizontally, bouncing off the left/right walls. */  
var x = 0;  
var xDirection = 1;  
var xSpeed = 10;  
function animate() {  
  x += xDirection * xSpeed;  
  if (x > window.innerWidth - 100) {  
    xDirection = -1;  
  }  
  if (x < 0) {  
    xDirection = 1;  
  }  
  rocketship.style.left = x + 'px';  
  
  window.requestAnimationFrame(animate);  
  animate();  
  
  /* Go half speed. */  
  xSpeed = 5;  
  
  /* Disable scrollbars. */  
  document.body.style.overflow = 'hidden';
```

<https://www.youtube.com/watch?v=Zm9B-DvwOgw>

# 「大規模言語モデルがコーディングを全部行う」は誤り



- ネット上に「大規模言語モデルがあれば人間がコーディングしなくて良くなる」という発言があるが、**これは誤り**
  - 実際に、情報系学生から将来に対する不安の声があった
  - 入門レベルのコードしか書かないケースに対する発言では。
- 入門者： 大規模言語モデルが自分より良いコードを書くように見えるので、惑わされないように注意
  - 上級者になれば惑わされない（例：就職した卒業生談「**変な情報に騙されすぎですね**」）

# 長所・短所を理解し学びに活用することが重要



## 誤りの例

- 「9.9と9.11の大小比較」 → 😞 「9.11の方が大きいです」 (2024/7月)
- (CNNを出力させたうえで) 「そのモデルのパラメータ数はいくつですか？」 → 「18762です」 (😞実際には約9000)

## ポイント

- 初級知識を質問することにとどめ、**検証を忘れない**ように
- ChatGPTは英訳サービス・予測変換・文法校正ツールと本質的に同じ
- **LLMはツールである** (鉛筆や辞書同様、使い方を学べばよい)



# 実習

---



# 初歩的な内容を質問する（3分）



## 課題：以下をChatGPTに入力せよ

1. NumPyについて教えて
  - 上記回答中のわからない内容についてさらに問い合わせよ
  - 例：「ブロードキャストとは何ですか」
2. 現在、google colaboratoryのノートブックの画面を見えています。このノートブックを実行する手順を教えてください
3. 以下のコードを説明してください。  
`np.loadtxt(fname, delimiter=",")`

# エラーメッセージと修正方法の解説



## ■ 課題：以下をChatGPTに入力せよ

以下のエラーについて説明し、どのように修正すればよいか教えてください。

ZeroDivisionError

Traceback (most recent call last)

```
<ipython-input-6-b457b0d96231> in <module>
```

```
1 ans = 0
```

```
2 for i in range(10):
```

```
----> 3     ans += A[i] // B[i]
```

```
4 print(ans)
```

ZeroDivisionError: integer division or modulo by zero



# ChatGPT補助付きNumPy入門（10分）



課題：以下のページの確認問題を解け。 **その際にChatGPTに説明させよ。**

【要keio.jp】

[https://colab.research.google.com/drive/1r\\_JfRv3eL4-eyJ3-G2xcBain1CTS41y8?usp=sharing](https://colab.research.google.com/drive/1r_JfRv3eL4-eyJ3-G2xcBain1CTS41y8?usp=sharing)



## ■ 課題：以下をChatGPTに入力せよ

以下の関数のdocstringを生成してください。出力はdocstringのみとしてください：

```
def zeller(year, month, day):  
    if month <= 2:  
        year -= 1  
        month += 10  
    else:  
        month -= 2  
    w = day + int((13 * month - 1) / 5) + year + int(year / 4) - int(year / 100) + int(year / 400)  
    x = w % 7  
    return x
```

# コンソールから入力を受け取る

```
year = input("誕生年を入力してください(Enterで決定)：")
```

```
month = input("誕生月を入力してください(Enterで決定)：")
```

```
day = input("誕生日を入力してください(Enterで決定)：")
```

# 曜日の計算を0-6の間で計算する

```
x = zeller(year, month, day)
```

# 出力

```
ws = ["日", "月", "火", "水", "木", "金", "土"]
```

```
print(f"{year}年{month}月{day}日の曜日は{ws[x]}曜日です")
```

# 問題点の指摘とコーディング規約に準拠した修正案の生成



## ■ 課題：以下をChatGPTに入力せよ

以下の問題点を指摘し、正しいコードを出力させてください。コードはPEP8に準拠させてください。：

```
for i in range(1, 1001):
    sum_divisor = 0
    for j in range(1, i//2):
        if i % j == 0:
            sum_divisor += j

    if i == sum_divisor:
        print(i)
```



## ■ 課題：以下をChatGPTに入力せよ

以下のコードを、内包表記を使って高速になるように書き換えてください：

```
for i in range(1, 1001):  
    sum_divisor = 0  
    for j in range(1, i//2 + 1):  
        if i % j == 0:  
            sum_divisor += j  
  
    if i == sum_divisor:  
        print(i)
```

# ChatGPT補助付きPythonデバッグ入門（10分）



課題：以下のページの例題 1 を解け。 **その際にChatGPTに説明させよ。**

【要keio.jp】

<https://drive.google.com/file/d/1bhpxE--wcrus0xbhYW700FTPxcoPdO9I/view?usp=sharing>

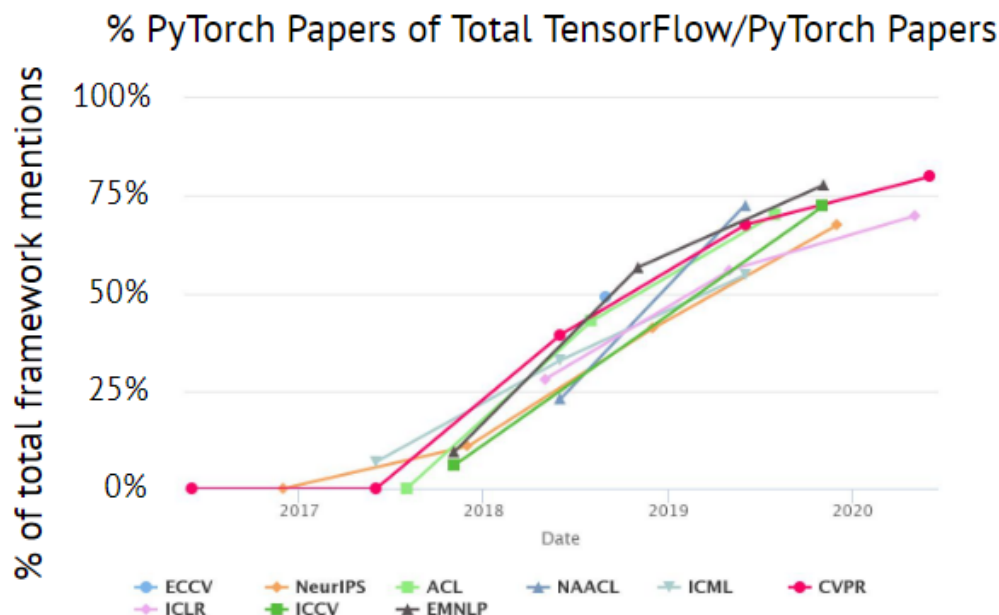
- 上級者：確認問題 2、確認問題 3 を解け

# PyTorchとは

<https://pytorch.org/>



- 主として深層学習に用いられるPython向け機械学習フレームワーク
    - 他にTensorFlow, JAX等がある
    - 2019年以降TensorFlowより優勢
- [State of AI Report 2020: <https://www.stateof.ai/2020>]



 PyTorch

# ChatGPT補助付きPyTorchチュートリアル



課題：以下のページの「テンソル」「データセットとデータローダー」の日本語解説を実行せよ。**その際にChatGPTに説明させよ。**

- [https://yutaroogawa.github.io/pytorch\\_tutorials\\_jp/](https://yutaroogawa.github.io/pytorch_tutorials_jp/)
- 上記はPyTorch公式チュートリアルが日本語訳されたものである
- 時間が余った場合は以下も入力してみよ
  - 他のチュートリアル
  - 機械学習基礎のスライド：Google Driveから入手可
  - 論文（例 <https://arxiv.org/abs/1706.03762>）



# 注意

---







# 注意



- 前提科目の単位未取得者は原則として履修できません
  - 小テストによる特別履修許可に対する回答送付済み
- 本講義の内容を既に習得済みの人 →講義後に申し出てください