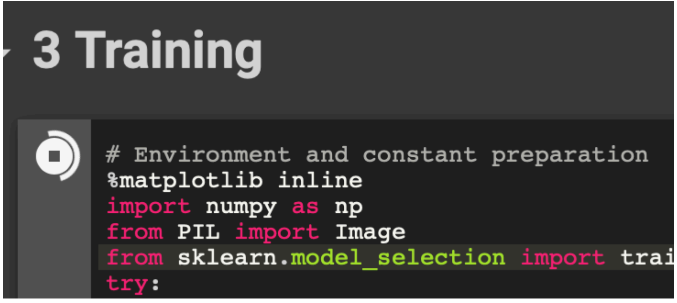
Prepare the environment and data

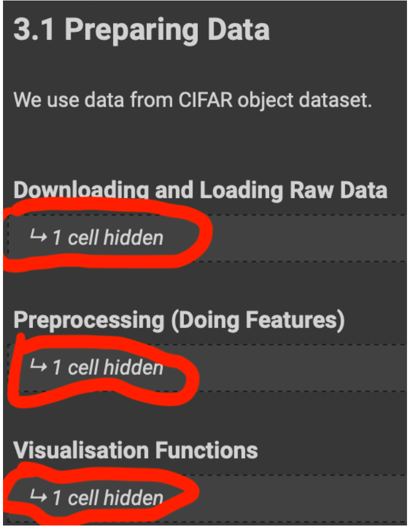
- Staring from Chapter 3 (now manually numbered), execute the first cell to have required libraries ready.



3 Training

```
# Environment and constant preparation
%matplotlib inline
import numpy as np
from PIL import Image
from sklearn.model_selection import trai
try:
```

- Prepare the data — we use the CIFAR-10 image dataset, and employ a few simplification steps. Section 3.1 contains the data preparation steps. Please study the function definition and experiment with the preprocessing steps if interested. To focus on the main topic of “model training”, you can simply execute all the cells



3.1 Preparing Data

We use data from CIFAR object dataset.

Downloading and Loading Raw Data

↳ 1 cell hidden

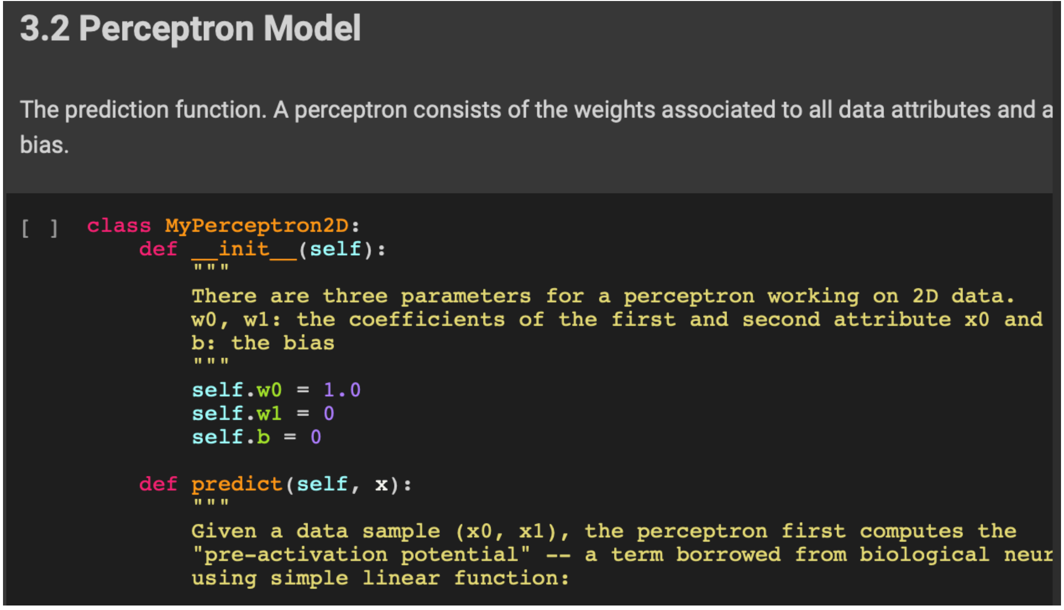
Preprocessing (Doing Features)

↳ 1 cell hidden

Visualisation Functions

↳ 1 cell hidden

- The main learning materials are in Section 3.2. The cell shown in the figure below, we define a perceptron class. Please build instances of the perceptron model using the class. Observe how the model works on the training data.



3.2 Perceptron Model

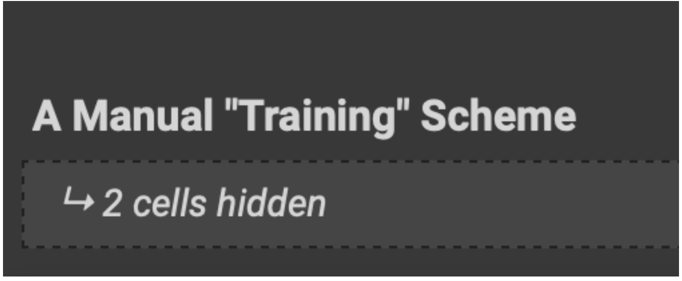
The prediction function. A perceptron consists of the weights associated to all data attributes and a bias.

```
class MyPerceptron2D:
    def __init__(self):
        """
        There are three parameters for a perceptron working on 2D data.
        w0, w1: the coefficients of the first and second attribute x0 and
        b: the bias
        """
        self.w0 = 1.0
        self.w1 = 0
        self.b = 0

    def predict(self, x):
        """
        Given a data sample (x0, x1), the perceptron first computes the
        "pre-activation potential" -- a term borrowed from biological neur
        using simple linear function:
```

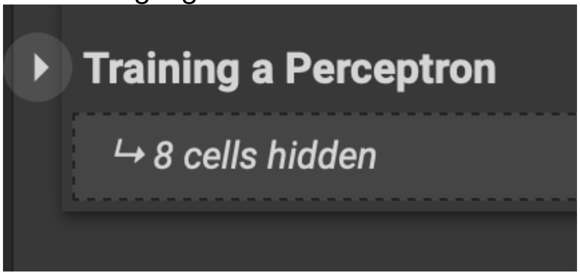
Practice Training Algorithm

- Try to improve the perceptron model by adjusting the parameters to make the E_in smaller.



A Manual "Training" Scheme

- Formulate the intuitive idea you developed in the above step. Design the main component of the training algorithm.



Training a Perceptron

- Construct a training algorithm for the perceptron model and compare your solution to `sklearn`. Hint: the central component of the training algorithm is a “main-loop”, where the model parameters are adjusted iteratively.

Further Study

- Following the similar procedure as in Week 2, test the model on test dataset (compute E_test). Adjust the number of samples in the training set and examine the effect. Record, discuss and report your findings.
- Is it sure that the algorithm you implemented will find the optimal hypothesis in the perceptron family? Why? If not, can you the problem settings to ensure the algorithm will reach the optimal hypothesis (Hint: see Abu-Mostafa et al. 2012 “Learning from Data” (book), Ch1.5, Problem 1.3)?
- You may notice that we simplified the object classification problem by considering only separable cases. Relax this simplification (take the data NOT processed by `prepare_cifar_two_class_separable_data` in notebook Ch3.1-Preprocessing). How does the perceptron training algorithm work? Discuss your findings in relation to your study of the previous question.
- Can you design another way of representing the data (not necessarily as 2D data, unless visualisation is a must)? How the representation affects the model performance?