数据集

在之前的部分，我们都用的是简单的数据，也就是：

In [1]:
```
import torch
X = torch.tensor([0.2, 0.5, 0.7, 0.4, 0.6, 0.8, 0.3, 0.6, 0.9]).view(3,3)
Y = torch.tensor([0.8, 0.8, 0.7, 0.9, 0.9, 0.9]).view(3,2)
print(X)
print(Y)
```

```
tensor([[0.2000, 0.5000, 0.7000],
        [0.4000, 0.6000, 0.8000],
        [0.3000, 0.6000, 0.9000]])
tensor([[0.8000, 0.8000],
        [0.7000, 0.9000],
        [0.9000, 0.9000]])
```

也就是一共有$E=3$个数据，每个数据有$M=3$个输入特征和$N=2$个目标数据。这个{X,Y}就是一个数据集。

然而在大数据时代，数据的个数可能成千上万甚至数百万，数据的特征也可能有几十个甚至几百个。我们不可能手动输入这些数据，至少不可能轻易的输入这些数据。而且这些数据往往都是直接从传感器纪录在文档里面，因此可以相对容易的调用。在这一点上，科研和工业有相当大的区别：

- 科研：科研的目的往往是设计新的模型，因此，为了体现模型的优越性并且增加实验的说服力，模型必须在一些公认的、著名的数据集上测试。（由于数据集难度的不同，如果自己找一些简单的数据集，结果就不具备和其他工作的可比性。）因此，科研用的数据集往往比较容易获得，而且比较容易使用，一般来说从网上下载下来，用几行代码就可以读取并且使用。比较著名的数据集网站是https://archive.ics.uci.edu/ml/datasets.php
- 工业：工业上的数据质量往往不堪入目。例如一个机器需要用50个传感器来判断机器的运行状态，一共需要采集10000个数据，也就是说，理想情况下X是一个10000×50的矩阵。然而在现实中会出现以下问题：
  - 项目初期并不知道所需传感器的个数，许多初期采集的数据没有50个特征
  - 项目初期并不知道所需传感器安装的位置，许多初期采集的数据的安装位置都不一样（不是同一个特征）
  - 实际中安装传感器的位置不精确
  - 传感器在运行过程中损坏
  - 由于迭代，传感器类型（生产厂商、标准）不同
  - 记录文件丢失、重复
  - 数据没有存在一起，而是在不同的文件夹里

  在这种情况下，就要有大量的工作用于整理数据、筛选数据等等。所以绝大多数工业AI都会失败*。

* Sculley, David, et al. "Hidden technical debt in machine learning systems." Advances in neural information processing systems 28 (2015).

* https://www.pacteraedge.com/pactera-white-paper-reveals-85-percent-ai-projects-ultimately-fail-0

数据

数据维度

即便在科研领域，数据集也有复杂和简单之分：

- 最简单的就是MLP(Multi Layer Perception)的数据集，这种数据集的输入数据往往就是$X\in\mathbb{R}^{E\times M}$，$Y\in\mathbb{R}^{E\times N}$，表示有$E$个数据，每个数据有$M$个特征和$N$个输出。
- 还有CNN(Convulutional Neural Network)，数据集的输入往往是图片信息，也就是$X\in\mathbb{R}^{E\times M_1\times M_2}$，$Y\in\mathbb{R}^{E\times N}$，其中$M_1$和$M_2$是图片的尺寸。当然图片信息也可以经过转换后用MLP解决。
- 时间序列和NLP，这类数据往往有顺序信息，而且输入的序列很长，需要滑动窗口（sliding window）将数据切片，因此比MLP的数据复杂许多。

当然，对于初学者，应该从最基础的数据集入门。

数据类型

机器学习的数据类型大致分为2种：定性数据和定量数据。

- 定性数据，没有大小之分，例如男人/女人，苹果/桃子/梨。人们无法比较它们的区别
- 定量数据，有大小之分，例如考试成绩、年龄等等。

很多数据需要具体情况具体分析，例如时间 **8:00** 和 **9:30** 有前后之分，但是有没有大小之分需要具体问题具体分析。但是持续的时间段往往有长短（大小）之分。而"大"，"中"，"小"尽管是定性数据，某些情况下也可以当作定量数据（比如用3，2，1）来描述。

对于定性数据，人们往往用One-Hot Encoding来处理，例如男人和女人用2个特征来描述，比如男的就是$[0,1]$，女的就是$[1,0]$。这样的话既可以区分二者，又不会有"误导性"。相反把男的记为1，女的记为2，就有了大小之分。而且1-2之间的意义也很难解释。

对于定量的数据，一般可以直接使用。

机器学习任务

在机器学习任务中，最具有代表性的2种任务就是回归（regression）和分类（classification）。

对于分类任务，输出的是类型，比如猫/狗/猪。那么输出就有N个，这里的N指的就是一共有多少类。然后把输出最大的一个当作分类。

对于回归任务，就是正常的使用

代码

下面的代码展示了一个读若干取整理好的数据集的过程：

In [2]:
```
import pickle
import os
```

In [3]:
```
datasets = os.listdir('./others/datasets/')
datasets = [f for f in datasets if (f.startswith('Dataset') and f.endswith('.p'))]
datasets.sort()
```

In [4]:
```
for dataset in datasets:
    datapath = os.path.join(f'./others/datasets/{dataset}')
    with open(datapath, 'rb') as f:
        data = pickle.load(f)
    X_train    = data['X_train']
    y_train    = data['y_train']
    X_valid    = data['X_valid']
    y_valid    = data['y_valid']
    X_test     = data['X_test']
    y_test     = data['y_test']
    data_name  = data['name']

    N_class    = data['n_class']
    N_feature  = data['n_feature']
    N_train    = X_train.shape[0]
    N_valid    = X_valid.shape[0]
    N_test     = X_test.shape[0]

    print(f'Dataset "{data_name}" has {N_feature} input features and {N_class} classes.\nThere are {N_train} training e
```

```
Dataset "acuteinflammation" has 6 input features and 2 classes.
There are 70 training examples, 23 valid examples, and 25 test examples in the dataset.

Dataset "acutenephritis" has 6 input features and 2 classes.
There are 70 training examples, 23 valid examples, and 25 test examples in the dataset.

Dataset "balancescale" has 4 input features and 3 classes.
There are 373 training examples, 124 valid examples, and 126 test examples in the dataset.

Dataset "blood" has 4 input features and 2 classes.
There are 447 training examples, 149 valid examples, and 150 test examples in the dataset.

Dataset "breastcancer" has 9 input features and 2 classes.
There are 170 training examples, 56 valid examples, and 58 test examples in the dataset.

Dataset "breastcancerwisc" has 9 input features and 2 classes.
There are 418 training examples, 139 valid examples, and 140 test examples in the dataset.

Dataset "breasttissue" has 9 input features and 6 classes.
There are 62 training examples, 20 valid examples, and 22 test examples in the dataset.

Dataset "ecoli" has 7 input features and 8 classes.
There are 200 training examples, 66 valid examples, and 68 test examples in the dataset.

Dataset "energyy1" has 8 input features and 3 classes.
There are 459 training examples, 153 valid examples, and 154 test examples in the dataset.

Dataset "energyy2" has 8 input features and 3 classes.
There are 459 training examples, 153 valid examples, and 154 test examples in the dataset.

Dataset "fertility" has 9 input features and 2 classes.
There are 58 training examples, 19 valid examples, and 21 test examples in the dataset.

Dataset "glass" has 9 input features and 6 classes.
There are 127 training examples, 42 valid examples, and 43 test examples in the dataset.

Dataset "habermansurvival" has 3 input features and 2 classes.
There are 182 training examples, 60 valid examples, and 62 test examples in the dataset.

Dataset "hayesroth" has 3 input features and 3 classes.
There are 93 training examples, 31 valid examples, and 32 test examples in the dataset.

Dataset "ilpdindianliver" has 9 input features and 2 classes.
There are 348 training examples, 116 valid examples, and 117 test examples in the dataset.

Dataset "iris" has 4 input features and 3 classes.
There are 88 training examples, 29 valid examples, and 31 test examples in the dataset.

Dataset "mammographic" has 5 input features and 2 classes.
There are 575 training examples, 191 valid examples, and 193 test examples in the dataset.

Dataset "monks1" has 6 input features and 2 classes.
There are 331 training examples, 110 valid examples, and 111 test examples in the dataset.

Dataset "monks2" has 6 input features and 2 classes.
There are 358 training examples, 119 valid examples, and 120 test examples in the dataset.

Dataset "monks3" has 6 input features and 2 classes.
There are 330 training examples, 110 valid examples, and 110 test examples in the dataset.

Dataset "Pendigits" has 16 input features and 10 classes.
There are 6595 training examples, 2198 valid examples, and 2199 test examples in the dataset.

Dataset "pima" has 8 input features and 2 classes.
There are 459 training examples, 153 valid examples, and 154 test examples in the dataset.

Dataset "pittsburgbridgesMATERIAL" has 7 input features and 3 classes.
There are 62 training examples, 20 valid examples, and 22 test examples in the dataset.

Dataset "pittsburgbridgesRELL" has 7 input features and 3 classes.
There are 60 training examples, 20 valid examples, and 21 test examples in the dataset.

Dataset "pittsburgbridgesSPAN" has 7 input features and 3 classes.
There are 54 training examples, 18 valid examples, and 18 test examples in the dataset.

Dataset "pittsburgbridgesTORD" has 7 input features and 2 classes.
There are 60 training examples, 20 valid examples, and 20 test examples in the dataset.

Dataset "pittsburgbridgesTYPE" has 7 input features and 6 classes.
There are 61 training examples, 20 valid examples, and 22 test examples in the dataset.

Dataset "postoperative" has 8 input features and 3 classes.
There are 52 training examples, 17 valid examples, and 19 test examples in the dataset.

Dataset "seeds" has 7 input features and 3 classes.
There are 124 training examples, 41 valid examples, and 43 test examples in the dataset.

Dataset "teaching" has 5 input features and 3 classes.
There are 89 training examples, 29 valid examples, and 31 test examples in the dataset.

Dataset "tictactoe" has 9 input features and 2 classes.
There are 573 training examples, 191 valid examples, and 192 test examples in the dataset.

Dataset "vertebralcolumn2clases" has 6 input features and 2 classes.
There are 184 training examples, 61 valid examples, and 63 test examples in the dataset.

Dataset "vertebralcolumn3clases" has 6 input features and 3 classes.
There are 184 training examples, 61 valid examples, and 63 test examples in the dataset.
```

这里的数据集已经由我处理好，并且分成了train，valid和test数据集。还有learning/test分类，其中learning集就是train和valid的和。我们可以看一下数据

In [5]:
```
X_train
```

Out[5]:
```
tensor([[0.1207, 0.3491, 0.2327, 0.1139, 0.5896, 0.0320],
        [0.5573, 0.8546, 0.4296, 0.2710, 0.4832, 0.0876],
        [0.1812, 0.4285, 0.1233, 0.1307, 0.5151, 0.0387],
        ...,
        [0.1360, 0.3657, 0.0995, 0.1199, 0.6479, 0.0309],
        [0.5859, 0.7053, 0.3024, 0.3757, 0.6990, 0.2625],
        [0.4321, 0.4771, 0.4373, 0.3463, 0.4958, 0.1014]])
```

In [6]:
```
y_train
```

Out[6]:
```
tensor([0, 2, 0, 0, 2, 1, 0, 2, 2, 1, 2, 1, 2, 0, 0, 2, 2, 2, 1, 1, 2, 1, 0, 0,
        2, 0, 1, 1, 0, 2, 0, 2, 2, 2, 2, 2, 2, 1, 2, 2, 0, 1, 2, 1, 1, 2, 2, 1,
        0, 2, 2, 0, 2, 2, 2, 0, 1, 1, 0, 0, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        1, 2, 1, 2, 1, 2, 0, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 0, 2, 2, 1, 0, 1, 2, 1,
        2, 1, 1, 2, 2, 2, 2, 2, 1, 0, 1, 1, 2, 0, 2, 2, 2, 1, 0, 2, 2, 2, 0, 1, 1,
        0, 1, 0, 1, 1, 2, 2, 2, 1, 1, 1, 2, 0, 2, 0, 2, 0, 2, 1, 2, 1, 2, 2, 1, 1,
        2, 0, 2, 2, 0, 1, 2, 2, 0, 1, 0, 1, 2, 2, 0, 2, 0, 2, 0, 2, 1, 0, 2, 1,
        2, 1, 2, 2, 0, 0, 1, 1, 2, 2, 1, 2, 2, 2, 0, 2, 2, 2])
```

现实里的数据集往往不会这么完善，而是存在excel,cvs以及txt文件中，大家需要自己处理，有时候必须自己分割数据集。

下一部分我们将搭建一个神经网络，并且用上数据集

In [ ]:
```

```