Pytorch – data

黃志勝（Tommy）
義隆電子人工智慧研發部
國立陽明交通大學 AI學院 合聘助理教授
國立台北科技大學 電資學院 合聘助理教授

Database

- 1. Open Data

  UCI database,

  MNIST,

  ImageNet,

  MS-COCO,… etc.

- 2. Private Data

Database in Machine Learning

- 1. UCI database

Most Popular Data Sets (hits since 2007):

3825761: Iris

2070996: Adult

1601121: Wine

1442822: Heart Disease

1435469: Breast Cancer Wisconsin (Diagnostic)

1434425: Wine Quality

1397755: Bank Marketing

1326195: Car Evaluation

Link

Iris Data Set

Download: Data Folder, Data Set Description

Abstract: Famous database; from Fisher, 1936

Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	3825764

Link

Wine Data Set

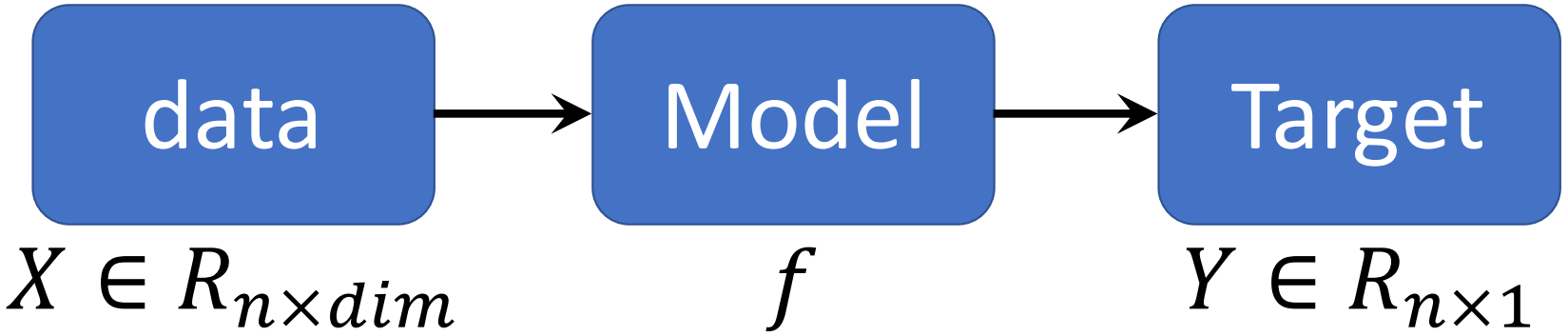
Download: Data Folder, Data Set Description

Abstract: Using chemical analysis determine the origin of wines

Data Set Characteristics:	Multivariate	**Number of Instances:**	178	**Area:**	Physical
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	13	**Date Donated**	1991-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1601130

Learning Model

Structure data-learning model



$$Y = f(X)$$

Database in Machine Learning
Structure data

IRIS

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
4.8,3.4,1.6,0.2,Iris-setosa
4.8,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.1,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.5,0.4,Iris-setosa
5.4,3.9,1.3,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
5.1,3.8,1.5,0.3,Iris-setosa
5.4,3.4,1.7,0.2,Iris-setosa
5.1,3.7,1.5,0.4,Iris-setosa
4.6,3.6,1.0,0.2,Iris-setosa
```

$$X \in R_{n \times dim} = R_{150 \times 4}$$
$$\mathrm{Y} \in R_{n \times 1} = R_{150 \times 1}$$

WINE

```
1,14.23,1.71,2.43,15.6,127,2.8,3.06,.28,2.29,5.64,1.04,3.92,1065
1,13.2,1.78,2.14,11.2,100,2.65,2.76,.26,1.28,4.38,1.05,3.4,1050
1,13.16,2.36,2.67,18.6,101,2.8,3.24,.3,2.81,5.68,1.03,3.17,1185
1,14.37,1.95,2.5,16.8,113,3.85,3.49,.24,2.18,7.8,.86,3.45,1480
1,13.24,2.59,2.87,21,118,2.8,2.69,.39,1.82,4.32,1.04,2.93,735
1,14.2,1.76,2.45,15.2,112,3.27,3.39,.34,1.97,6.75,1.05,2.85,1450
1,14.39,1.87,2.45,14.6,96,2.5,2.52,.3,1.98,5.25,1.02,3.58,1290
1,14.06,2.15,2.61,17.6,121,2.6,2.51,.31,1.25,5.05,1.06,3.58,1295
1,14.83,1.64,2.17,14,97,2.8,2.98,.29,1.98,5.2,1.08,2.85,1045
1,13.86,1.35,2.27,16,98,2.98,3.15,.22,1.85,7.22,1.01,3.55,1045
1,14.1,2.16,2.3,18,105,2.95,3.32,.22,2.38,5.75,1.25,3.17,1510
1,14.12,1.48,2.32,16.8,95,2.2,2.43,.26,1.57,5,1.17,2.82,1280
1,13.75,1.73,2.41,16,89,2.6,2.76,.29,1.81,5.6,1.15,2.9,1320
1,14.75,1.73,2.39,11.4,91,3.1,3.69,.43,2.81,5.4,1.25,2.73,1150
1,14.38,1.87,2.38,12,102,3.3,3.64,.29,2.96,7.5,1.2,3,1547
1,13.63,1.81,2.7,17.2,112,2.85,2.91,.3,1.46,7.3,1.28,2.88,1310
1,14.3,1.92,2.72,20,120,2.8,3.14,.33,1.97,6.2,1.07,2.65,1280
1,13.83,1.57,2.62,20,115,2.95,3.4,.4,1.72,6.6,1.13,2.57,1130
1,14.19,1.59,2.48,16.5,108,3.3,3.93,.32,1.86,8.7,1.23,2.82,1680
1,13.64,3.1,2.56,15.2,116,2.7,3.03,.17,1.66,5.1,.96,3.36,845
1,14.06,1.63,2.28,16,126,3,3.17,.24,2.1,5.65,1.09,3.71,780
1,12.93,3.8,2.65,18.6,102,2.41,2.41,.25,1.98,4.5,1.03,3.52,770
1,13.71,1.86,2.36,16.6,101,2.61,2.88,.27,1.69,3.8,1.11,4,1035
```

$$X \in R_{n \times dim} = R_{178 \times 13}$$
$$\mathrm{Y} \in R_{n \times 1} = R_{178 \times 1}$$

Database in Machine Learning

- scikit-learn is a good python package in machine learning.
- Sklearn database

`sklearn.datasets`: Datasets

The `sklearn.datasets` module includes utilities to load datasets, including methods to load and fetch popular reference datasets. It also features some artificial data generators.

User guide: See the Dataset loading utilities section for further details.

Loaders

`datasets.clear_data_home`([data_home])	Delete all the content of the data home cache.
`datasets.dump_svmlight_file`(X, y, f, *[, ...])	Dump the dataset in svmlight / libsvm file format.
`datasets.fetch_20newsgroups`(*[, data_home, ...])	Load the filenames and data from the 20 newsgroups dataset (classification).
`datasets.fetch_20newsgroups_vectorized`(*[, ...])	Load and vectorize the 20 newsgroups dataset (classification).
`datasets.fetch_california_housing`(*[, ...])	Load the California housing dataset (regression).
`datasets.fetch_covtype`(*[, data_home, ...])	Load the covertype dataset (classification).
`datasets.fetch_kddcup99`(*[, subset, ...])	Load the kddcup99 dataset (classification).
`datasets.fetch_lfw_pairs`(*[, subset, ...])	Load the Labeled Faces in the Wild (LFW) pairs dataset (classification).
`datasets.fetch_lfw_people`(*[, data_home, ...])	Load the Labeled Faces in the Wild (LFW) people dataset (classification).
`datasets.fetch_olivetti_faces`(*[, ...])	Load the Olivetti faces data-set from AT&T (classification).
`datasets.fetch_openml`([name, version, ...])	Fetch dataset from openml by name or dataset id.
`datasets.fetch_rcv1`(*[, data_home, subset, ...])	Load the RCV1 multilabel dataset (classification).
`datasets.fetch_species_distributions`(*[, ...])	Loader for species distribution dataset from Phillips et.
`datasets.get_data_home`([data_home])	Return the path of the scikit-learn data dir.
`datasets.load_boston`(*[, return_X_y])	Load and return the boston house-prices dataset (regression).
`datasets.load_breast_cancer`(*[, return_X_y, ...])	Load and return the breast cancer wisconsin dataset (classification).
`datasets.load_diabetes`(*[, return_X_y, as_frame])	Load and return the diabetes dataset (regression).
`datasets.load_digits`(*[, n_class, ...])	Load and return the digits dataset (classification).
`datasets.load_files`(container_path, *[, ...])	Load text files with categories as subfolder names.
`datasets.load_iris`(*[, return_X_y, as_frame])	Load and return the iris dataset (classification).
`datasets.load_linnerud`(*[, return_X_y, as_frame])	Load and return the physical excercise linnerud dataset.
`datasets.load_sample_image`(image_name)	Load the numpy array of a single sample image
`datasets.load_sample_images`()	Load sample images for image manipulation.
`datasets.load_svmlight_file`(f, *[, ...])	Load datasets in the svmlight / libsvm format into sparse CSR matrix
`datasets.load_svmlight_files`(files, *[, ...])	Load dataset from multiple files in SVMlight format
`datasets.load_wine`(*[, return_X_y, as_frame])	Load and return the wine dataset (classification).

Database in Machine Learning

Implement

Pytorch dataset

- Deep learning courses usually use MNIST as sample data. In this class, we also use MNIST.
- Pytorch provides torchvision as interface for datasets.
- Example:

MNIST

CIFAR10

TORCHVISION.DATASETS

All datasets are subclasses of `torch.utils.data.Dataset` i.e, they have `__getitem__` and `__len__` methods implemented. Hence, they can all be passed to a `torch.utils.data.DataLoader` which can load multiple samples parallelly using `torch.multiprocessing` workers. For example:

```
imagenet_data = torchvision.datasets.ImageNet('path/to/imagenet_root/')
data_loader = torch.utils.data.DataLoader(imagenet_data,
                                          batch_size=4,
                                          shuffle=True,
                                          num_workers=args.nThreads)
```

The following datasets are available:

Datasets

- CelebA
- CIFAR
- Cityscapes
- COCO
  - Captions
  - Detection
- DatasetFolder
- EMNIST
- FakeData
- Fashion-MNIST
- Flickr
- HMDB51
- ImageFolder
- ImageNet
- Kinetics-400
- KMNIST
- LSUN
- MNIST
- Omniglot
- PhotoTour
- Places365
- QMNIST
- SBD
- SBU
- STL10
- SVHN
- UCF101
- USPS
- VOC

MNIST

- MNIST: http://yann.lecun.com/exdb/mnist/

THE MNIST DATABASE

of handwritten digits

Yann LeCun, Courant Institute, NYU
Corinna Cortes, Google Labs, New York
Christopher J.C. Burges, Microsoft Research, Redmond

Training set: 60,000 (10,000 images per class)
Testing set:10,000 (1000 images per class)

Kaggle: https://www.kaggle.com/c/digit-recognizer

CIFAR-10

- CIFAR10: https://www.cs.toronto.edu/~kriz/cifar.html
- Training set: 50,000 with 10 classes (5000 images per class)
- Testing set:10,000 (1000 images per class)

< Back to Alex Krizhevsky's home page

The CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset. They were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.

The CIFAR-10 dataset

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from
remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain e

Here are the classes in the dataset, as well as 10 random images from each:

CIFAR-100

- **The CIFAR-100 dataset**
- This dataset is just like the CIFAR-10, except it has 100 classes containing 600 images each.
- There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 superclasses.
- Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs).

Superclass	Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

Pytorch datasets

Implement

Other open data

Kaggle: https://www.kaggle.com

PhysioNet: https://www.physionet.org/

Private Data

- How to load private data in pytorch.

- Implement:

Sample data from kaggle.

https://www.kaggle.com/datasets?topic=imageDataset

Car Brands Images: https://www.kaggle.com/yamaerenay/100-images-of-top-50-car-brands

Car Brands Images

- Build a Computer Vision model to predict whether a car image belongs to a luxury or mass-market car brand