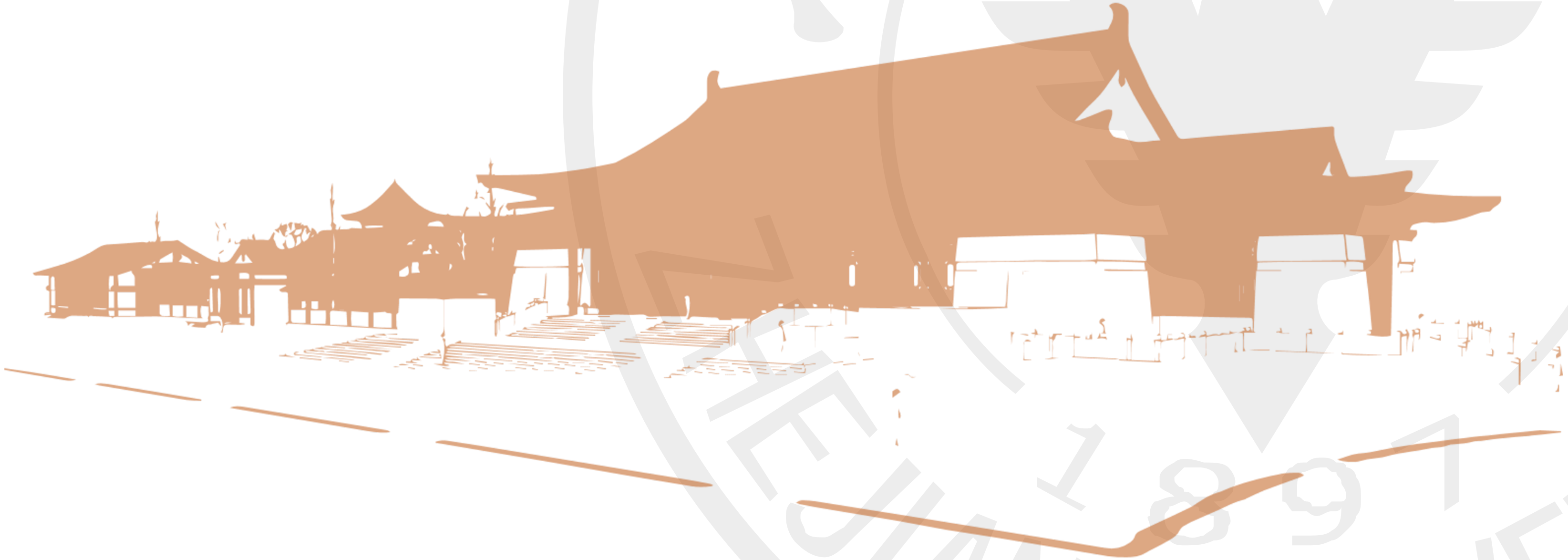
Practice: Supervised Learning



Scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license
- Scikit Provides: Classification, Regression, Clustering...

Scikit-learn

Installation

Scikit-learn requires:

Python (>= 2.7 or >= 3.3) with pip

Numpy (>= 1.8.2)

SciPy (>= 0.13.3)

Scikit-learn install:

```
pip install scikit-learn
pip install matplotlib
pip install pandas
```

Colab

https://colab.research.google.com/

Colab is a free **online** service that lets you run Python code with access to GPUs.

Forget the environment and focus on coding

API

Import the dataset:

datasets.fetch_openml([name, version, …])

Fetch dataset from openml by name or dataset id.

E.g. X, y = fetch_openml('mnist_784', version=1, return_X_y=True)

Select the model:

neural_network.MLPClassifier([…])

Multi-layer Perceptron classifier.

E.g. mlp = MLPClassifier(hidden_layer_sizes=(50,), max_iter=10, alpha=1e-4,
solver='sgd', verbose=10, random_state=1, learning_rate_init=.1)

API

Train the model:

MLPClassifier.fit(X, y)

Fit the model to data matrix X and target(s) y.

E.g. mlp.fit(X_train, y_train)

Make the prediction:

MLPClassifier.predict(X)

Predict using the multi-layer perceptron classifier.

E.g. mlp.predict(X)

API

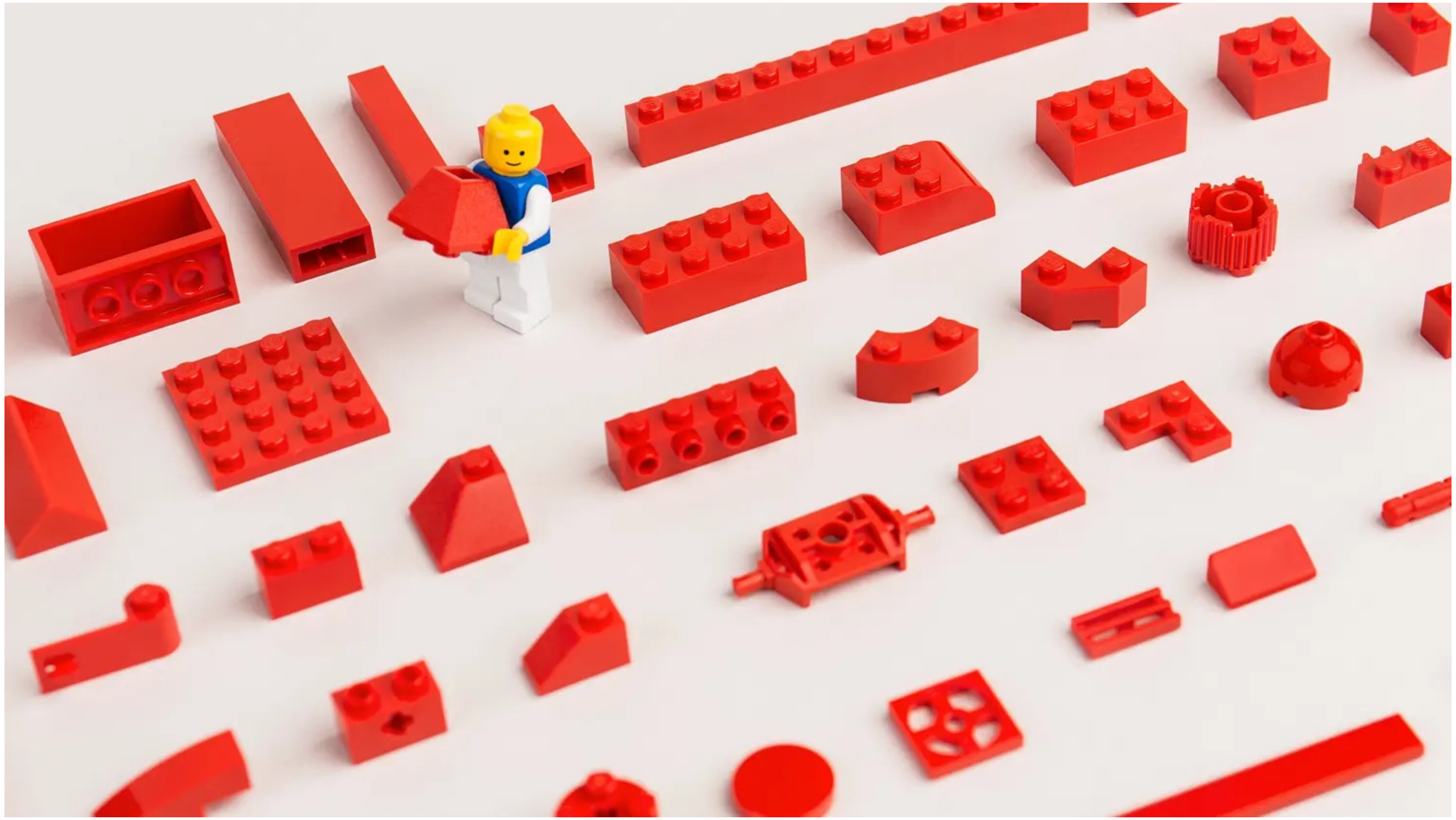
Other useful methods of MLPClassifier:

get_metadata_routing()	Get metadata routing of this object.
get_params([deep])	Get parameters for this estimator.
partial_fit(X, y[, classes])	Update the model with a single iteration over the given data.
predict_log_proba(X)	Return the log of probability estimates.
predict_proba(X)	Probability estimates.
score(X, y[, sample_weight])	Return the mean accuracy on the given test data and labels.

More details at https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier.predict

API

Now we can build our neural network like building blocks



Introduction

Pytorch

Deep Learning Framework

- Known for its flexibility and ease-of-use
- Developed by Facebook’s AI Research lab
- It leverages the power of GPUs
- Automatic computation of gradients
- Make it easier to test and develop new ideas

Installation

Getting started with Pytorch

Via Anaconda/Miniconda:

```
conda install pytorch -c pytorch
```

Via pip:

```
pip3 install torch
```

https://colab.research.google.com/

API

Import the dataset:

torchvision.datasets

torchvision.datasets contains the following data sets: MNIST, COCO, CIFAR10 and CIFAR100, and so on.

E.g. datasets.MNIST('data', train=True, download=True, transform= transforms.ToTensor())

Define the dataloader:

torch.utils.data.DataLoader([...])

At the heart of PyTorch data loading utility is the torch.utils.data.DataLoader class.

E.g. train_loader = torch.utils.data.DataLoader(datasets.MNIST([...])),
batch_size=BATCH_SIZE, shuffle=True)

API

Define our model:

torch.nn.Module(*args, **kwargs)

Base class for all neural network modules. Our models should also subclass this class.

Modules can also contain other Modules, allowing to nest them in a tree structure.

We can assign the submodules as regular attributes:

```
class Network(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(1, 20, 5)
        self.conv2 = nn.Conv2d(20, 20, 5)
    def forward(self, x):
        x = F.relu(self.conv1(x))
        return F.relu(self.conv2(x))
```

API

Other useful APIs:

nn.Linear(in_features, out_features, bias=True, device=None, dtype=None)

nn.ReLU(inplace=False)

nn.Conv2d(in_channels, out_channels, kernel_size, stride=1, padding=0, ...)

nn.CrossEntropyLoss(weight=None, ...)

nn.MaxPool2d(kernel_size, stride=None,padding=0,...)

nn.Flatten(start_dim=1, end_dim=-1)

More details at https://pytorch.org/docs/stable/nn.html

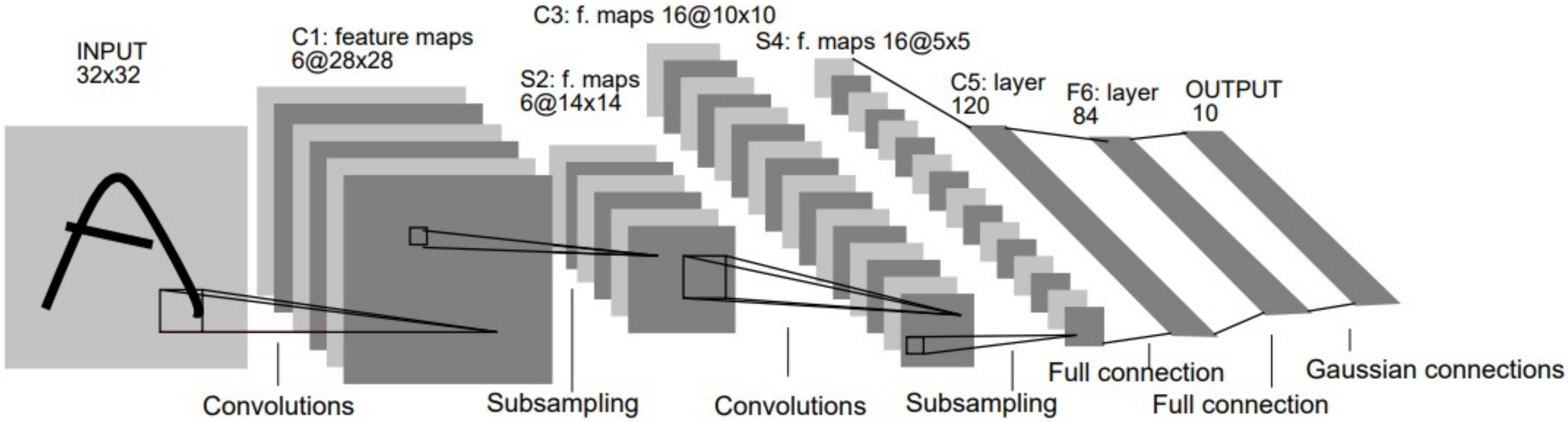
Next Lecture

Natural Language Processing(NLP)

1. Word2Vec
2. Embedding
3. Recurrent Neural Network

Mid-term Examination & Assignment2

1. Use Scikit-learn for classification on CIFAR10.
2. Use Pytorch to reimplement LeNet for classification on CIFAR10.
3. Submit your experimental report in pdf format



浙江大學
ZHEJIANG UNIVERSITY

Thank you!