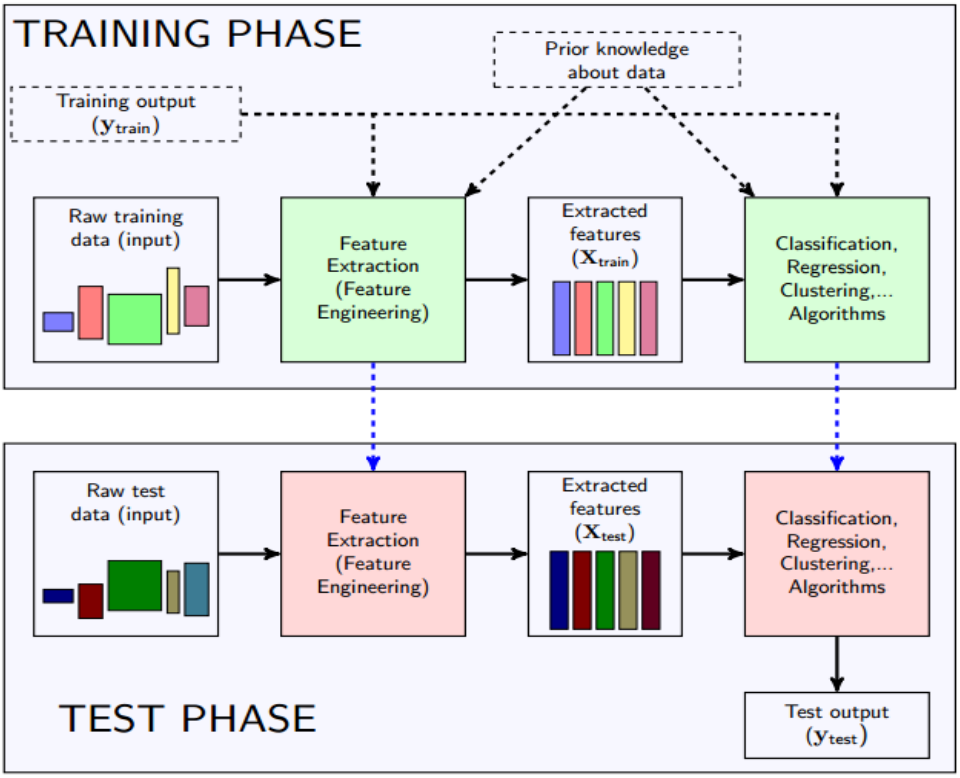
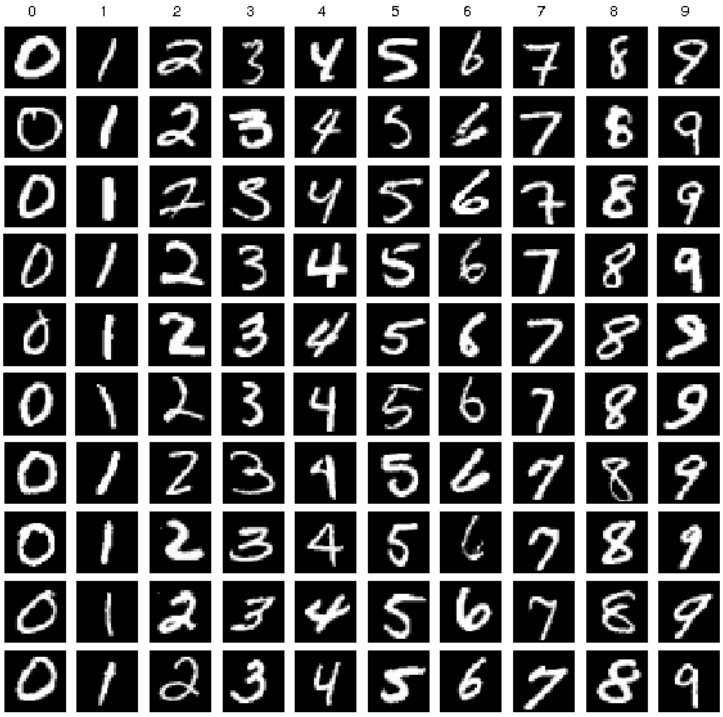
Mô hình chung cho bài toán Machine Learning



Thuật toán Softmax Regression

Ví dụ: Sử dụng thuật toán Softmax Regression để phân loại chữ số 0 đến 9 trong bộ dữ liệu MNIST



1. Sử dụng các thư viện: numpy, matplotlib, sklearn,… thông qua lệnh import

Import thư viện

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

2. **Load tập dữ liệu**

```
mnist = fetch_openml("mnist_784")
N, d = mnist.data.shape
print('Total {:d} samples, each has {:d} pixel.'.format(N,d))
```

```
Total 70000 samples, each has 784 pixel.
```

Có tổng cộng 70000 điểm dữ liệu trong tập dữ liệu MNIST, mỗi điểm là một mảng 784 phần tử tương ứng với 784 pixel. Mỗi chữ số từ 0 đến 9 chiếm khoảng 10%.

```
X = mnist.data
print(X)
y = mnist.target
print(y)
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
['5' '0' '4' ... '4' '5' '6']
```

3. **Phân chia dữ luyện huấn luyện, kiểm tra**
   Chúng ta sẽ lấy ra ngẫu nhiên 10000 điểm làm tập kiểm tra, phần còn lại dùng để huấn luyện.

```
#Split train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 10000)
print(X_train)
print(y_train)
print(X_test)
print(y_test)
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
['5' '9' '9' ... '6' '6' '3']
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
['1' '0' '6' ... '3' '0' '7']
```

4. **Xây dựng mô hình Logistic Regression trên tập huấn luyện và dự đoán nhãn của các điểm trong tập kiểm tra. Kết quả này được so sánh với nhãn thực sự của mỗi điểm dữ liệu để tính độ chính xác của bộ phân loại:**

```
model = LogisticRegression(C = 1e5, solver = 'lbfgs', multi_class = 'multinomial')
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Accuracy %.2f %%" % (100*accuracy_score(y_test, y_pred.tolist())))
```

```
Accuracy 92.62 %
```

5. **Tìm những ảnh bị phân loại sai và hiển thị chúng:**
   Hàm display_network tham khảo code theo linh này:
   https://github.com/tiepvupsu/tiepvupsu.github.io/blob/master/assets/kmeans/display_network.py
   Copy code => vào block ngay dưới. Sau đó thực hiện tiếp:

```
from matplotlib.backends.backend_pdf import PdfPages
mis = np.where((y_pred - y_test) !=0)[0]
print(mis)
Xmis = X_test[mis, :]
filename = 'mnist_mis.pdf'
with PdfPages(filename) as pdf:
    plt.axis('off')
    A = display_network(Xmis.T, 1, Xmis.shape[0])
    f2 = plt.imshow(A, interpolation='nearest')
    plt.gray()
    pdf.savefig(bbox_inches='tight')
    plt.show()
```

Kết quả hình bị phân loại sai: