PyTorch
Tutorial I

Course CS 419
Instructor Prof Abir De

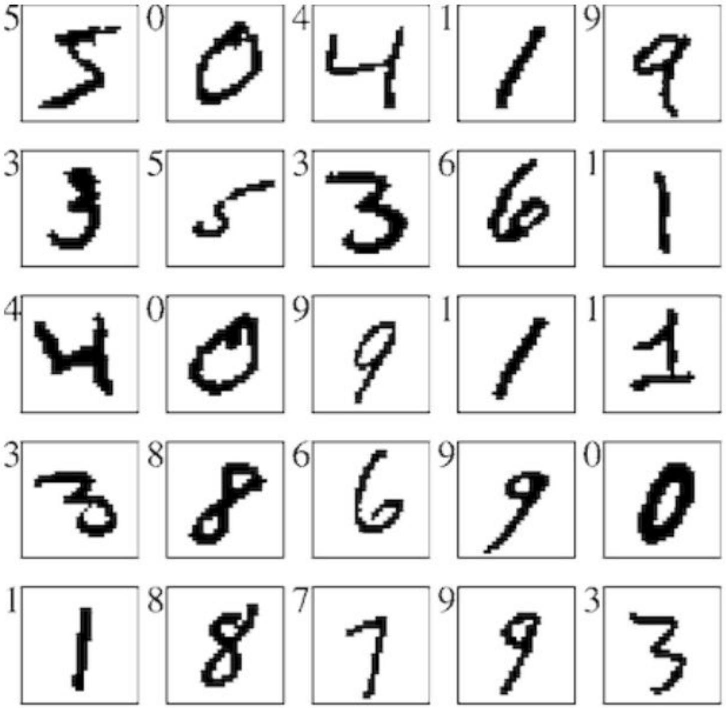
Aditya Pande
[22m2108 @iitb.ac.in]

Ninad Gandhi
[22m2151 @iitb.ac.in]

</ Problem Statement />

</ MNIST : dataset />

The MNIST (Modified National Institute of Standards and Technology)

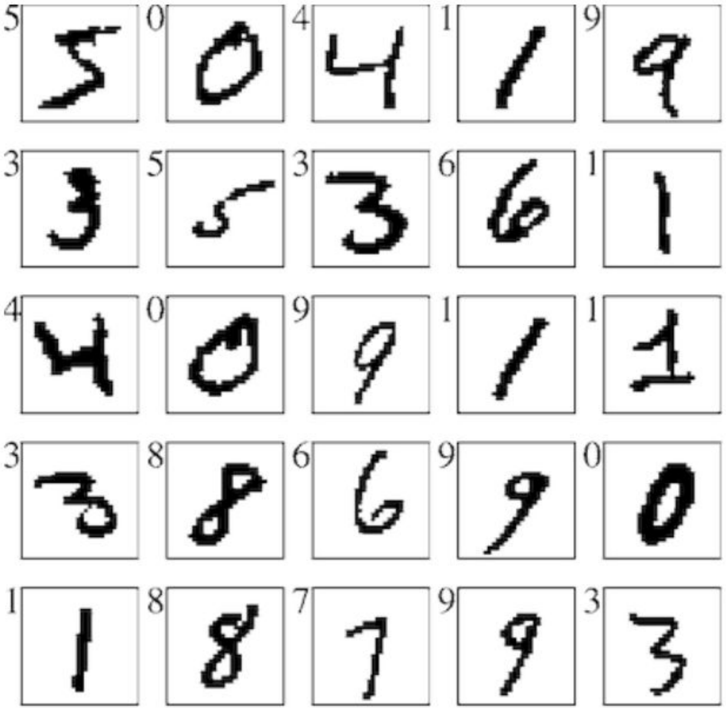


</ MNIST : dataset />

The MNIST (Modified National Institute of Standards and Technology)

dataset is a classic benchmark dataset in the field of machine learning and computer vision. It consists of:

- **Data:** 70,000 handwritten digits (60,000 for training and 10,000 for testing).
- **Image Size:** 28x28 pixels.
- **Labels:** Each image is labeled with the digit it represents (0-9)



</ MNIST : How does it look />

	0	1	2	3	4	5	6	7	8	9	...	18	19	20	21	22	23	24	25	26	27
0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
1	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
2	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
3	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
4	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
5	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
6	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
7	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
8	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	0.992	0.035	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-0.749	...	0.984	0.027	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
10	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-0.294	0.647	0.812	...	0.984	0.027	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
11	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.451	0.961	0.984	0.827	...	0.984	-0.796	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
12	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.114	0.365	-0.310	-0.937	...	0.529	-0.937	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
13	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-0.380	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
14	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-0.569	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
15	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-0.827	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
16	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
17	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
18	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
19	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
20	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
21	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
22	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
23	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
24	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
25	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
26	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
27	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000	-1.000	-1.000	-1.000	...	-1.000	-1.000	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0

- Normalise the data

  −1 to 1

- STACK THE ROWS

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

</ MNIST : How does it look />

A [?] dimensional
Real vector

$$(x^{(i)}, y^{(i)})$$

A categorical
variable with
[?] distinct values

</ INTERLUDE : TENSOR />

What?

A torch tensor is a multi-dimensional array used for data and computations in PyTorch

```python
import torch

# Create a tensor
x = torch.tensor([1.0, 2.0, 3.0])

# Print the tensor
print("Tensor x:", x)

# Perform basic operations
y = x + 2.0          # Addition
z = x * 3.0          # Multiplication

# Print results
print("x + 2:", y)   # Output: tensor([3.0, 4.0, 5.0])
print("x * 3:", z)   # Output: tensor([3.0, 6.0, 9.0])
```

</ Neural Network />

</ Neural Network: **Prereqs/Neuron** />

- Let us take an aside and look at the building blocks of neural networks. This is a cool website from Jared Wilber. [Website]

</ Neural Network: Activations />

Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-(x)}}$$

Tanh

$$\sigma(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

ReLU

$$\sigma(x) = max(x, 0)$$

LeakyReLU

$$\sigma(x) = max(0.1x, x)$$

Visualization

</ Neural Network: Mathematical Form />

$$y_i = f(x_i)$$

</ Neural Network: Mathematical Form />

$$y_i = f(x_i)$$

$$y_i = W_1^T \cdot x_i + b_1$$

</ Neural Network: Mathematical Form />

$$y_i = f(x_i)$$

$$y_i = W_1^T \cdot x_i + b_1$$

$$h_1 = \sigma_1(W_1^T \cdot x_i + b_1)$$

</ Neural Network: Mathematical Form />

$$y_i = f(x_i)$$

$$y_i = W_1^T \cdot x_i + b_1$$

$$h_1 = \sigma_1(W_1^T \cdot x_i + b_1)$$

$$h_2 = \sigma_2(W_2^T \cdot (h_1) + b_2)$$

</ Neural Network: Mathematical Form />

$$y_i = f(x_i)$$

$$y_i = W_1^T \cdot x_i + b_1$$

$$h_1 = \sigma_1(W_1^T \cdot x_i + b_1)$$

$$h_2 = \sigma_2(W_2^T \cdot (h_1) + b_2)$$

$$\vdots$$

$$h_l = \sigma_l(W_l^T \cdot (h_{l-1}) + b_l)$$

</ Neural Network: Mathematical Form />

$$y_i = f(x_i)$$

$$y_i = W_1^T \cdot x_i + b_1$$

$$h_1 = \sigma_1(W_1^T \cdot x_i + b_1)$$

$$h_2 = \sigma_2(W_2^T \cdot (h_1) + b_2)$$

$$\vdots$$

$$h_l = \sigma_l(W_l^T \cdot (h_{l-1}) + b_l)$$

$$logits = W_{out}^T \cdot (h_l) + b_{out}$$

</ PROBABILITY : Softmax />

softmax function $\mathrm{softmax} : \mathbb{R}^k \to \mathbb{R}^k$ as

$$\mathrm{softmax}(t_1, \ldots, t_k) = \begin{bmatrix} \frac{\exp(t_1)}{\sum_{j=1}^{k} \exp(t_j)} \\ \vdots \\ \frac{\exp(t_k)}{\sum_{j=1}^{k} \exp(t_j)} \end{bmatrix}.$$

This has the characteristics of a _ _ _

</ PROBABILITY : _•_•_ />

- It has all positive values
- It sums up to 1

</ PROBABILITY : MLE />

It can be used as a P.D.F

$$\begin{bmatrix} P(y=1 \mid x;\theta) \\ \vdots \\ P(y=k \mid x;\theta) \end{bmatrix} = \mathrm{softmax}(t_1, \cdots, t_k)$$

YOU CAN NOW TAKE ARGMAX

$$\begin{bmatrix} P(y = 1 \mid x;\theta) \\ \vdots \\ P(y = k \mid x;\theta) \end{bmatrix} = \operatorname{softmax}(t_1, \cdots, t_k)$$

BUT WHY DO THAT ?????

MLE : Maximum Likelihood Estimation

</ PART 1 : FIN />