PyTorch vs. TensorFlow

```
import numpy as np
import matplotlib.pyplot as plt
epochs = 10
batch_size=64
import tensorflow as tf
import torch
import torchvision
import torch
import torchvision
```

Loading and preprocessing data in TF

```
(x_trainTF_, y_trainTF_), _ = tf.keras.datasets.mnist.load_data()
x_trainTF = x_trainTF_.reshape(60000, 784).astype('float32')/255
y_trainTF = tf.keras.utils.to_categorical(y_trainTF_,
 num_classes=10)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```

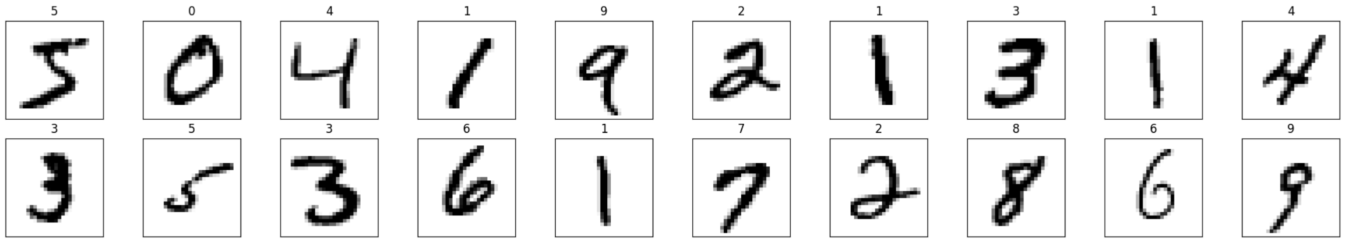
Loading and preprocessing data in PT

```
xy_trainPT = torchvision.datasets.MNIST(root='./data', train=True,
download=True,transform=torchvision.transforms.Compose([torchvision.
transforms.ToTensor()]))
xy_trainPT_loader = torch.utils.data.DataLoader(xy_trainPT, batch_size=batch_size)
```

```
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
Failed to download (trying next):
HTTP Error 404: Not Found

Downloading https://ossci-datasets.s3.amazonaws.com/mnist/train-images-idx3-ubyte.gz
Downloading https://ossci-datasets.s3.amazonaws.com/mnist/train-images-idx3-ubyte.gz to ./data/MNIST/raw/train-images-idx3-ubyte.gz
100%|██████████| 9.91M/9.91M [00:00<00:00, 36.3MB/s]
Extracting ./data/MNIST/raw/train-images-idx3-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
Failed to download (trying next):
HTTP Error 404: Not Found

Downloading https://ossci-datasets.s3.amazonaws.com/mnist/train-labels-idx1-ubyte.gz
Downloading https://ossci-datasets.s3.amazonaws.com/mnist/train-labels-idx1-ubyte.gz to ./data/MNIST/raw/train-labels-idx1-ubyte.gz
100%|██████████| 28.9k/28.9k [00:00<00:00, 1.17MB/s]
Extracting ./data/MNIST/raw/train-labels-idx1-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
Failed to download (trying next):
HTTP Error 404: Not Found

Downloading https://ossci-datasets.s3.amazonaws.com/mnist/t10k-images-idx3-ubyte.gz
Downloading https://ossci-datasets.s3.amazonaws.com/mnist/t10k-images-idx3-ubyte.gz to ./data/MNIST/raw/t10k-images-idx3-ubyte.gz
100%|██████████| 1.65M/1.65M [00:00<00:00, 10.2MB/s]
Extracting ./data/MNIST/raw/t10k-images-idx3-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
Failed to download (trying next):
HTTP Error 404: Not Found

Downloading https://ossci-datasets.s3.amazonaws.com/mnist/t10k-labels-idx1-ubyte.gz
Downloading https://ossci-datasets.s3.amazonaws.com/mnist/t10k-labels-idx1-ubyte.gz to ./data/MNIST/raw/t10k-labels-idx1-ubyte.gz
100%|██████████| 4.54k/4.54k [00:00<00:00, 6.40MB/s]Extracting ./data/MNIST/raw/t10k-labels-idx1-ubyte.gz to ./data/MNIST/raw
```

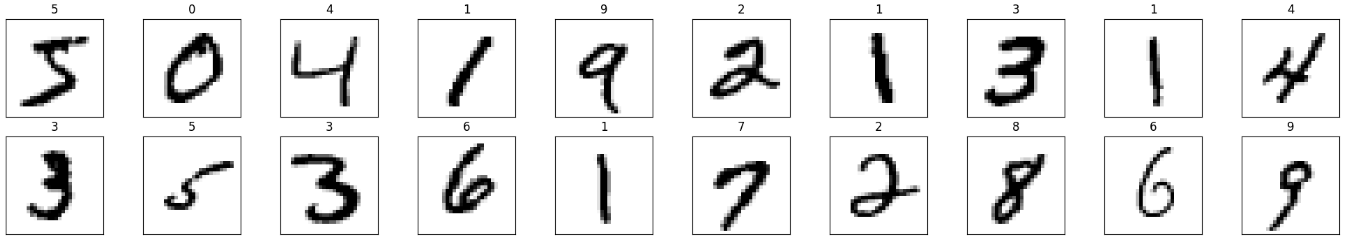
```
print("TensorFlow:")
fig = plt.figure(figsize=(25, 4))
for idx in np.arange(20):
 ax = fig.add_subplot(2, 10, idx+1, xticks=[], yticks=[])
 ax.imshow(x_trainTF_[idx], cmap=plt.cm.binary)
 ax.set_title(str(y_trainTF_[idx]))
```

```
TensorFlow:
```



```
print("PyTorch:")
fig = plt.figure(figsize=(25, 4))
for idx in np.arange(20):
 ax = fig.add_subplot(2, 10, idx+1, xticks=[], yticks=[])
 image, label = xy_trainPT [idx]
 ax.imshow(torch.squeeze(image, dim = 0).numpy(),
 cmap=plt.cm.binary)
 ax.set_title(str(label))
```

```
PyTorch:
```



Define Model

```
modelTF = tf.keras.Sequential([
tf.keras.layers.Dense(10,activation='sigmoid',input_shape=(784,)),
tf.keras.layers.Dense(10,activation='softmax')
])


modelPT= torch.nn.Sequential(
 torch.nn.Linear(784,10),
 torch.nn.Sigmoid(),
 torch.nn.Linear(10,10),
 torch.nn.LogSoftmax(dim=1)
 )
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` ar
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Define the optimizer and loss function

```
modelTF.compile(
 loss="categorical_crossentropy",
 optimizer=tf.optimizers.SGD(learning_rate=0.01),
 metrics = ['accuracy']
 )
```

```
criterion = torch.nn.NLLLoss()
optimizer = torch.optim.SGD(modelPT.parameters(), lr=0.01)
```

Train the model

```
_ = modelTF.fit(x_trainTF, y_trainTF, epochs=epochs,
 batch_size=batch_size, verbose = 0)


for e in range(epochs):
 for images, labels in xy_trainPT_loader:
  images = images.view(images.shape[0], -1)
  loss = criterion(modelPT(images), labels)
  loss.backward()
  optimizer.step()
  optimizer.zero_grad()
```

Evaluate the model

```
_, (x_testTF, y_testTF)= tf.keras.datasets.mnist.load_data()
x_testTF = x_testTF.reshape(10000, 784).astype('float32')/255
y_testTF = tf.keras.utils.to_categorical(y_testTF, num_classes=10)
_ , test_accTF = modelTF.evaluate(x_testTF, y_testTF)
print('\n TensorFlow model Accuracy =', test_accTF)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8734 - loss: 0.4783

 TensorFlow model Accuracy = 0.8909000158309937
```

```
xy_testPT = torchvision.datasets.MNIST(root='./data', train=False,
download=True,

transform=torchvision.transforms.Compose([torchvision.transforms.ToTensor()]))
xy_test_loaderPT = torch.utils.data.DataLoader(xy_testPT)
correct_count, all_count = 0, 0
for images,labels in xy_test_loaderPT:
 for i in range(len(labels)):
  img = images[i].view(1, 784)
  logps = modelPT(img)
  ps = torch.exp(logps)
  probab = list(ps.detach().numpy()[0])
  pred_label = probab.index(max(probab))
  true_label = labels.numpy()[i]
  if(true_label == pred_label):
    correct_count += 1
  all_count += 1
print("\n PyTorch model Accuracy =", (correct_count/all_count))
```

```

 PyTorch model Accuracy = 0.8907
```