Ch : - 7 Deep Learning

Tensorflow

```
import tensorflow as tf
scalar = tf.constant(5)
print(scalar)
```

```
tf.Tensor(5, shape=(), dtype=int32)
```

```
var1 = tf.Variable([[1,2],[3,4]])
var2 = tf.Variable([[3,4],[3,4]])
result = tf.matmul(var1,var2)
print(result)
```

```
tf.Tensor(
[[ 9 12]
 [21 28]], shape=(2, 2), dtype=int32)
```

```
var1 = tf.constant([[1,2],[3,4]])
var2 = tf.constant([[3,4],[3,4]])
result = tf.add(var1,var2)
print(result)
```

```
tf.Tensor(
[[4 6]
 [6 8]], shape=(2, 2), dtype=int32)
```

```
pip install tensorflow
```

```
  Downloading libclang-18.1.1-py2.py3-none-manylinux2010_x86_64.whl.metadata (5.2 kB)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tensorflow) (25.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in /
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.3
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.7
Collecting tensorboard~=2.19.0 (from tensorflow)
  Downloading tensorboard-2.19.0-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: numpy<2.2.0,>=1.26.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Requirement already satisfied: ml-dtypes<1.0.0,>=0.5.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow)
Collecting tensorflow-io-gcs-filesystem>=0.23.1 (from tensorflow)
  Downloading tensorflow_io_gcs_filesystem-0.37.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadat
Collecting wheel<1.0,>=0.23.0 (from astunparse>=1.6.0->tensorflow)
  Downloading wheel-0.45.1-py3-none-any.whl.metadata (2.3 kB)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (14.0
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.1
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->te
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21
Requirement already satisfied: markdown>=2.6.8 in /usr/lib/python3/dist-packages (from tensorboard~=2.19.0->tensorfl
Collecting tensorboard-data-server<0.8.0,>=0.7.0 (from tensorboard~=2.19.0->tensorflow)
  Downloading tensorboard_data_server-0.7.2-py3-none-manylinux_2_31_x86_64.whl.metadata (1.1 kB)
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.5/57.5 kB 3.7 MB/s eta 0:00:00
Downloading libclang-18.1.1-py2.py3-none-manylinux2010_x86_64.whl (24.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.5/24.5 MB 40.0 MB/s eta 0:00:00
Downloading tensorboard-2.19.0-py3-none-any.whl (5.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.5/5.5 MB 49.9 MB/s eta 0:00:00
Downloading tensorflow_io_gcs_filesystem-0.37.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.1/5.1 MB 52.4 MB/s eta 0:00:00
Downloading tensorboard_data_server-0.7.2-py3-none-manylinux_2_31_x86_64.whl (6.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.6/6.6 MB 66.7 MB/s eta 0:00:00
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 224.5/224.5 kB 15.1 MB/s eta 0:00:00
Downloading wheel-0.45.1-py3-none-any.whl (72 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 72.5/72.5 kB 5.4 MB/s eta 0:00:00
Installing collected packages: libclang, flatbuffers, wheel, werkzeug, tensorboard-data-server, tensorflow-io-gcs-filesystem, tensorboard-dat
Successfully installed astunparse-1.6.3 flatbuffers-25.2.10 google-pasta-0.2.0 libclang-18.1.1 tensorboard-2.19.0 te
```

```
from tensorflow.keras.datasets import mnist
```

```
(x_train,y_train),(x_valid,y_valid)=mnist.load_data()
```

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

```
x_train.shape
```

```
(60000, 28, 28)
```

```
x_valid.shape
```

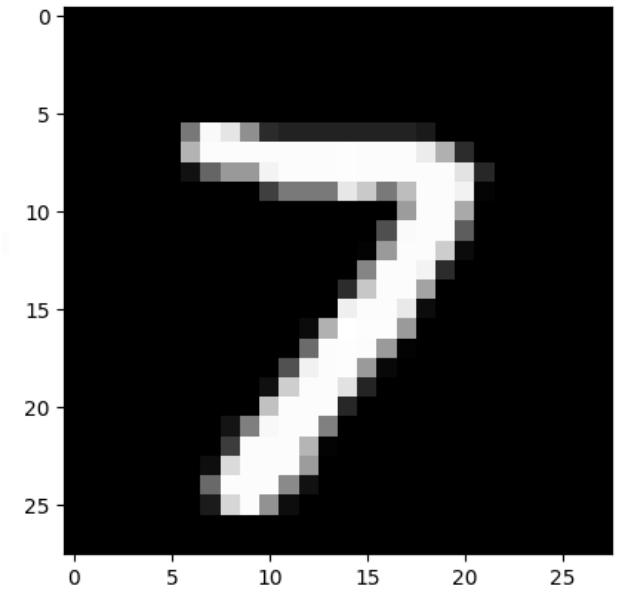
```
(10000, 28, 28)
```

```
x_train[4000]
```

```
ndarray (28, 28) show data
```

```
import matplotlib.pyplot as plt
image = x_train[4000]
plt.imshow(image,cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x7896e3fc2050>
```




```
y_train[1]
```

```
np.uint8(0)
```

```
y_train[4000]
```

```
np.uint8(7)
```

```
y_train[0]
```

```
np.uint8(5)
```

```
x_train = x_train.reshape(60000,784)
x_valid = x_valid.reshape(10000,784)
```

```
x_train.shape
```

```
(60000, 784)
```

```
x_valid.shape
```

```
(10000, 784)
```

```
x_train[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,   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,   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,   0,   0,   0,   0,   0,   0,   3,  18,  18,  18,
        126, 136, 175,  26, 166, 255, 247, 127,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170, 253,
        253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253,
        253, 253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  18, 219, 253,
        253, 253, 253, 253, 198, 182, 247, 241,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         80, 156, 107, 253, 253, 205,  11,   0,  43, 154,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,  14,   1, 154, 253,  90,   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, 139, 253, 190,   2,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190, 253,  70,
          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,  35,
        241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,  81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,  45, 186, 253, 253, 150,  27,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,  16,  93, 252, 253, 187,
          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, 249,
        253, 249,  64,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  46, 130,
        183, 253, 253, 207,   2,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39, 148,
        229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114,
        221, 253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  23,  66,
        213, 253, 253, 253, 253, 198,  81,   2,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  18, 171,
        219, 253, 253, 253, 253, 195,  80,   9,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  55, 172,
        226, 253, 253, 253, 253, 244, 133,  11,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
        136, 253, 253, 253, 212, 135, 132,  16,   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,   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], dtype=uint8)
```

```
x_train = x_train/255
x_valid = x_valid/255
# normalization
```

```
x_train.min()
```

```
np.float64(0.0)
```

```
x_train.dtype
```

```
dtype('float64')
```

```
# y_train
```

```
# y_train[0]
```

```
import tensorflow.keras as keras
num_categories = 10
```

```
y_train = keras.utils.to_categorical(y_train,num_categories)
y_valid = keras.utils.to_categorical(y_valid,num_categories)
```

```
y_train[0]
```

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.])
```

```
y_train[1]
```

```
array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
from tensorflow.keras.models import Sequential
model = Sequential()
from tensorflow.keras.layers import Dense
```

```
model.add(Dense(units=512,activation="relu",input_shape=(784,)))
model.add(Dense(units=512,activation="relu"))
model.add(Dense(units=10,activation="softmax"))
```

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

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	401,920
dense_1 (Dense)	(None, 512)	262,656
dense_2 (Dense)	(None, 10)	5,130

Total params: 669,706 (2.55 MB)
Trainable params: 669,706 (2.55 MB)

```
model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=["accuracy"])
```

```
history = model.fit(x_train,y_train,epochs=5,verbose=1,validation_data=(x_valid,y_valid))
```

```
Epoch 1/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 21s 11ms/step - accuracy: 0.9022 - loss: 0.3176 - val_accuracy: 0.9700 - val_loss: 0.0
Epoch 2/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 20s 11ms/step - accuracy: 0.9754 - loss: 0.0800 - val_accuracy: 0.9731 - val_loss: 0.0
Epoch 3/5
```

```
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 20s 11ms/step - accuracy: 0.9841 - loss: 0.0519 - val_accuracy: 0.9788 - val_loss: 0.06
Epoch 4/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 20s 11ms/step - accuracy: 0.9892 - loss: 0.0358 - val_accuracy: 0.9741 - val_loss: 0.09
Epoch 5/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 20s 11ms/step - accuracy: 0.9910 - loss: 0.0303 - val_accuracy: 0.9811 - val_loss: 0.07
```

```
import tensorflow.keras as keras
```

```
import pandas as pd
```

```
train_df = pd.read_csv('sign_mnist_train.csv')
```

```
test_df = pd.read_csv('sign_mnist_valid.csv')
```

```
y_train = train_df['label']
```

```
y_valid = test_df['label']
```

```
del train_df['label']
del test_df['label']
```

```
x_train = train_df.values
x_valid = test_df.values
```

```
num_classes = 25
y_train = keras.utils.to_categorical(y_train,num_classes)
y_valid = keras.utils.to_categorical(y_valid,num_classes)
```

```
x_train = x_train/255
x_valid = x_valid/255
```

```
x_train.shape,x_valid.shape
```

```
((27455, 784), (7172, 784))
```

```
x_train = x_train.reshape(-1,28,28,1)
x_valid = x_valid.reshape(-1,28,28,1)
```

```
x_train.shape
```

```
(27455, 28, 28, 1)
```

```
x_valid.shape
```

```
(7172, 28, 28, 1)
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (
    Conv2D,
    Dense,
    MaxPool2D,
    Flatten,
    Dropout,BatchNormalization
)
model = Sequential()
model.add(Conv2D(75,(3,3),strides=1,padding="same",activation="relu",input_shape=(28,28,1)))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2),strides=2,padding="same"))
model.add(Conv2D(50,(3,3),strides=1,padding="same",activation="relu"))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2),strides=2,padding="same"))
model.add(Conv2D(25,(3,3),strides=1,padding="same",activation="relu"))
model.add(BatchNormalization())
```

```
model.add(MaxPool2D((2,2),strides=2,padding="same"))
model.add(Flatten())
model.add(Dense(units=512,activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(units=num_classes,activation="softmax"))
```

```
model.summary()
```

Model: "sequential_5"

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 28, 28, 75)	750
batch_normalization_6 (BatchNormalization)	(None, 28, 28, 75)	300
max_pooling2d_6 (MaxPooling2D)	(None, 14, 14, 75)	0
conv2d_9 (Conv2D)	(None, 14, 14, 50)	33,800
dropout_4 (Dropout)	(None, 14, 14, 50)	0
batch_normalization_7 (BatchNormalization)	(None, 14, 14, 50)	200
max_pooling2d_7 (MaxPooling2D)	(None, 7, 7, 50)	0
conv2d_10 (Conv2D)	(None, 7, 7, 25)	11,275
batch_normalization_8 (BatchNormalization)	(None, 7, 7, 25)	100
max_pooling2d_8 (MaxPooling2D)	(None, 4, 4, 25)	0
flatten_2 (Flatten)	(None, 400)	0
dense_7 (Dense)	(None, 512)	205,312
dropout_5 (Dropout)	(None, 512)	0
dense_8 (Dense)	(None, 25)	12,825

Total params: 264,562 (1.01 MB)

```
model.compile(loss="categorical_crossentropy",metrics=["accuracy"],optimizer='adam')
```

```
model.fit(x_train,y_train,epochs=5,verbose=1,validation_data=(x_valid,y_valid))
```

```
Epoch 1/5
858/858 ━━━━━━━━━━━━━━━━━━━━ 101s 115ms/step - accuracy: 0.7458 - loss: 0.8721 - val_accuracy: 0.9427 - val_loss: 0.19
Epoch 2/5
858/858 ━━━━━━━━━━━━━━━━━━━━ 97s 113ms/step - accuracy: 0.9948 - loss: 0.0178 - val_accuracy: 0.9274 - val_loss: 0.241
Epoch 3/5
858/858 ━━━━━━━━━━━━━━━━━━━━ 97s 113ms/step - accuracy: 0.9949 - loss: 0.0162 - val_accuracy: 0.9387 - val_loss: 0.205
Epoch 4/5
858/858 ━━━━━━━━━━━━━━━━━━━━ 97s 113ms/step - accuracy: 0.9963 - loss: 0.0130 - val_accuracy: 0.7023 - val_loss: 1.476
Epoch 5/5
858/858 ━━━━━━━━━━━━━━━━━━━━ 97s 113ms/step - accuracy: 0.9894 - loss: 0.0372 - val_accuracy: 0.9336 - val_loss: 0.246
<keras.src.callbacks.history.History at 0x7895b6881190>
```

Start coding or generate with AI.