Double-click (or enter) to edit

Implement Neural Network

by **Sanjeev Gupta**

Train a DNN using the sequntial API on the MNIST fashion dataset by following the instructions given below:

Import all necessary modules

Use the sequential API to make a model with the following dense layers: (3)

i. layer-1: 128 neurons, relu activation

ii. layer-2: X neurons, softmax activation (Deduce the value of X based on the number of classes in the MNIST fashion dataset)

Define the input_shape=(None, Y) for the model. Deduce Y from the dataset

Please plot the model (plot_model())

Load the mnist fashion data from keras.datasets and perform necessary preprocessing (like reshaping and normalizing) on the train and test sets. (2)

Split the original training set into train and validation (10%) sets (1)

Compile the model using apropriate loss, any optimizer, and "accuracy" metric (3)

Define CallbackList with EarlyStopping (patience=2) and Tensorboard callbacks. (2)

Fit the model on training data for 10 epochs. (2)

Predict the labels of the first 5 images in the test set. (2)

```
import tensorflow as tf
from tensorflow import keras
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from keras.layers import Dense, Flatten
from keras import Input
from tensorflow.keras.utils import plot_model

from tensorflow.keras.datasets import fashion_mnist
from keras.callbacks import EarlyStopping, ModelCheckpoint, TensorBoard
```

```
(fashion_train_imgs, fashion_train_labels), (fashin_test_imgs, fashion_test_labels) = fashion_mnist.load_data()
```

```
fashion_train_imgs = fashion_train_imgs.reshape((len(fashion_train_imgs),
                                    28*28)).astype("float32")/255
fashin_test_imgs = fashin_test_imgs.reshape((len(fashin_test_imgs),
                                    28*28)).astype("float32")/255

print(fashin_test_imgs.shape)

train_x = fashion_train_imgs[6000:]
train_y = fashion_train_labels[6000:]
print(train_y[:5])
val_x = fashin_test_imgs[:6000]
val_y = fashion_test_labels[:6000]

agginment_model = keras.Sequential([
                          Dense(128, activation="relu"),
                          Dense(10, activation="softmax")
])


agginment_model.build(input_shape=(None, 28*28))

agginment_model.summary()


plot_model(agginment_model,show_shapes=True)

callbacks_list = [EarlyStopping(monitor="val_loss", patience=2)
```

```
callbacks_list = [EarlyStopping(monitor='val_loss', patience=2),
                  ModelCheckpoint("mnist_model_checkpoint",save_best_only=True),
                  TensorBoard(log_dir="/tensorboard_files")]

agginment_model.compile(optimizer =keras.optimizers.RMSprop(),
              loss = keras.losses.SparseCategoricalCrossentropy(),
              metrics = ["accuracy"])


agginment_model.fit(x=train_x, y=train_y, epochs=10,
                    validation_data=(val_x, val_y),
                    callbacks=callbacks_list)

predictions = agginment_model.predict(fashin_test_imgs[:5])
predicted_labels = np.argmax(predictions,axis=1)
print(predicted_labels)
```

```
(10000, 784)
[8 6 4 4 6]
Model: "sequential_26"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_52 (Dense)            (None, 128)               100480

 dense_53 (Dense)            (None, 10)                1290

=================================================================
Total params: 101,770
Trainable params: 101,770
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
1687/1688 [============================>.] - ETA: 0s - loss: 0.5110 - accuracy: 0.8177WARNING:absl:Found untraced functions such as
1688/1688 [==============================] - 8s 5ms/step - loss: 0.5110 - accuracy: 0.8177 - val_loss: 0.4564 - val_accuracy: 0.837
Epoch 2/10
1675/1688 [============================>.] - ETA: 0s - loss: 0.3845 - accuracy: 0.8617WARNING:absl:Found untraced functions such as
1688/1688 [==============================] - 7s 4ms/step - loss: 0.3839 - accuracy: 0.8618 - val_loss: 0.4199 - val_accuracy: 0.851
Epoch 3/10
1683/1688 [============================>.] - ETA: 0s - loss: 0.3483 - accuracy: 0.8745WARNING:absl:Found untraced functions such as
1688/1688 [==============================] - 7s 4ms/step - loss: 0.3480 - accuracy: 0.8745 - val_loss: 0.4073 - val_accuracy: 0.866
Epoch 4/10
1688/1688 [==============================] - 6s 4ms/step - loss: 0.3276 - accuracy: 0.8812 - val_loss: 0.4152 - val_accuracy: 0.857
Epoch 5/10
1686/1688 [============================>.] - ETA: 0s - loss: 0.3129 - accuracy: 0.8880WARNING:absl:Found untraced functions such as
1688/1688 [==============================] - 8s 4ms/step - loss: 0.3128 - accuracy: 0.8881 - val_loss: 0.3778 - val_accuracy: 0.872
Epoch 6/10
1688/1688 [==============================] - 6s 3ms/step - loss: 0.3020 - accuracy: 0.8931 - val_loss: 0.4352 - val_accuracy: 0.869
Epoch 7/10
1688/1688 [==============================] - 7s 4ms/step - loss: 0.2924 - accuracy: 0.8959 - val_loss: 0.3827 - val_accuracy: 0.867
WARNING:tensorflow:6 out of the last 318 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7f07f29c4d30>
1/1 [==============================] - 0s 64ms/step
[9 2 1 1 6]
```