Adeh Pawar

BECOB265

Problem Statement: Perform classification using deep learning convolution network. Compare and comment on the result by using different activation function and optimizers.

Dataset used: **fashion-mnist.csv** (kaggle)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import tensorflow as tf
from tensorflow.keras import layers
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
data = pd.read_csv('/content/drive/MyDrive/Deep Learning BE/fashion-mnist_test.csv')
X = data.iloc[:, 1:].values
y = data.iloc[:, 0].values
```

```
#split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
#reshape data
X_train = X_train.reshape(-1, 28, 28, 1) / 255.0
X_test = X_test.reshape(-1, 28, 28, 1) / 255.0
```

```
#CNN model definition 'relu' as activation function
model = tf.keras.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(10, activation='softmax')
])
```

chosen optimizers:

- adam
- sgd
- rmsprop

chosen activation functions:

- relu
- sigmoid
- tanh

```
optimizers = ['adam', 'sgd', 'rmsprop']
activation_functions = ['relu', 'sigmoid', 'tanh']
results = []

for optimizer in optimizers:
    for activation_func in activation_functions:
        # Clear previous model session and compile the model
        tf.keras.backend.clear_session()
        model.compile(optimizer=optimizer, loss='sparse_categorical_crossentropy', metrics=['accuracy'])

        # Train the model
        history = model.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), verbose=0)
        accuracy = history.history['val_accuracy'][-1]
        results.append((optimizer, activation_func, accuracy))


df_results = pd.DataFrame(results, columns=['Optimizer', 'Activation Function', 'Accuracy'])
print(df_results)
```

```
  Optimizer Activation Function  Accuracy
0      adam                relu    0.8855
1      adam             sigmoid    0.8840
2      adam                tanh    0.8795
3       sgd                relu    0.8825
4       sgd             sigmoid    0.8820
5       sgd                tanh    0.8820
6   rmsprop                relu    0.8820
7   rmsprop             sigmoid    0.8795
8   rmsprop                tanh    0.8790
```

- ## Conclusion:

> From above result, it is observed that the **highest** accuracy is **0.8855** where actiavtion function is **'relu'** and the suitable optimizer is **'adam'** where as, the activation function 'rmsprop' with 'tanh' gives **lowest** accuracy.