Sharon Suryawanshi Roll No 40

AIM - Convolutional neural network (CNN) Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories.

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

```
fashion_mnist = keras.datasets.fashion_mnist
(train_images,train_labels), (test_images,test_labels) = fashion_mnist.load_data()
train_images = train_images / 255.0
test_images = test_images  / 255.0
```

```
model = keras.Sequential([
    keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28, 1)),
    keras.layers.MaxPooling2D((2,2)),
    keras.layers.Flatten(),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

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

```
model.fit(train_images, train_labels, epochs=10)
```

```
Epoch 1/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 78s 41ms/step - accuracy: 0.8143 - loss: 0.5171
Epoch 2/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 79s 39ms/step - accuracy: 0.9038 - loss: 0.2592
Epoch 3/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 83s 40ms/step - accuracy: 0.9203 - loss: 0.2151
Epoch 4/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 77s 41ms/step - accuracy: 0.9366 - loss: 0.1745
Epoch 5/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 81s 40ms/step - accuracy: 0.9456 - loss: 0.1469
Epoch 6/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 80s 39ms/step - accuracy: 0.9551 - loss: 0.1220
Epoch 7/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 83s 39ms/step - accuracy: 0.9622 - loss: 0.1024
Epoch 8/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 74s 39ms/step - accuracy: 0.9703 - loss: 0.0830
Epoch 9/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 81s 39ms/step - accuracy: 0.9761 - loss: 0.0670
Epoch 10/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 86s 41ms/step - accuracy: 0.9807 - loss: 0.0558
<keras.src.callbacks.history.History at 0x784b5cf52dd0>
```

```
test_loss, test_acc = model.evaluate(test_images, test_labels)
print("Test accuracy:", test_acc)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 12ms/step - accuracy: 0.9123 - loss: 0.3634
Test accuracy: 0.9171000123023987
```

```
predictions = model.predict(test_images)
predicted_labels = np.argmax(predictions, axis=1)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 12ms/step
```

```
import numpy as np
import matplotlib.pyplot as plt

num_rows = 5
num_cols = 5
num_images = num_rows * num_cols

plt.figure(figsize=(12, 10))  # Increased figure size for better visibility

for i in range(num_images):
```

```
    # Image subplot
    plt.subplot(num_rows, num_cols * 2, 2 * i + 1)
    plt.imshow(test_images[i], cmap='gray')
    plt.axis('off')

    # Bar chart subplot
    plt.subplot(num_rows, num_cols * 2, 2 * i + 2)
    plt.bar(range(10), predictions[i])

    # Set x-axis ticks at intervals of 5
    plt.xticks(np.arange(0, 10, 5), fontsize=8, rotation=45)

    # Set y-axis ticks at intervals of 0.25
    plt.yticks(np.arange(0, 1.1, 0.25))

    plt.ylim([0, 1])
    plt.title(f"Pred: {predicted_labels[i]}", fontsize=10)

plt.tight_layout()
plt.show()
```