Assignment 5: Fashion MNIST

1.BASELINE MODEL:

1. **Code:**

```
import matplotlib.pyplot as plt
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
import tensorflow as tf

# Load and preprocess the MNIST dataset
mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images = train_images.reshape(60000, 28, 28, 1)
test_images = test_images.reshape(10000, 28, 28, 1)
train_images, test_images = train_images / 255.0, test_images / 255.0

# Define the model
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', input_shape=(28, 28,
1)),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(10, activation='softmax')
])

# Compile the model
model.compile(
    optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.9),
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

# Train the model and store the history
history = model.fit(
    train_images, train_labels,
    batch_size=32,
    epochs=10,
    verbose=1,
    validation_data=(test_images, test_labels)
)
```

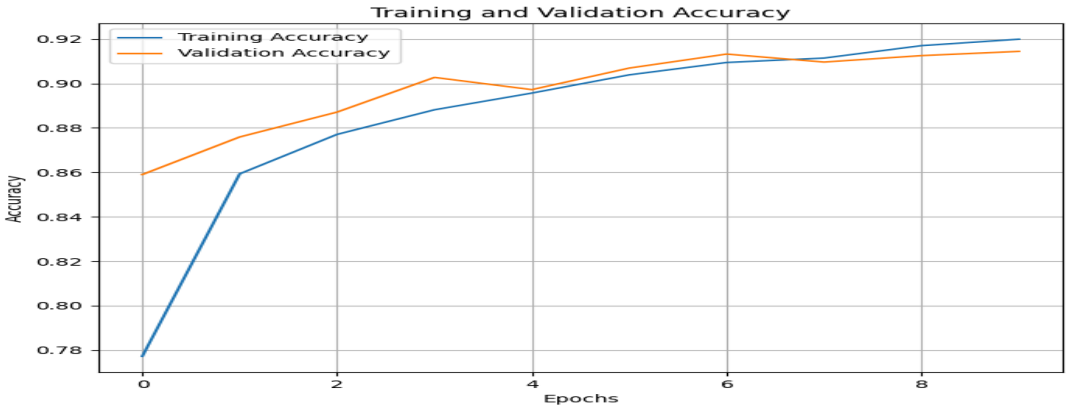
Assignment 5: Fashion MNIST

```
# Plot and save the training and validation accuracy
plt.figure(figsize=(8, 6))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()
plt.grid(True)
plt.savefig('accuracy_plot.png')  # Save the plot as a file
plt.show()
```

2. **Output:**

```
>>> exec(open("mnist.py").read())
Epoch 1/10
2024-11-25 16:32:44.619382: E tensorflow/core/grappler/optimizers/meta_optimizer.cc:1014] layout failed: INVALID_ARGUMENT: Size of values 0 does not match size of permutation
 4 @ fanin shape insequential_18/dropout_44/dropout/SelectV2-2-TransposeNHWCToNCHW-LayoutOptimizer
1875/1875 [==============================] - 5s 2ms/step - loss: 0.6018 - accuracy: 0.7772 - val_loss: 0.3835 - val_accuracy: 0.8590
Epoch 2/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3941 - accuracy: 0.8593 - val_loss: 0.3309 - val_accuracy: 0.8759
Epoch 3/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3395 - accuracy: 0.8771 - val_loss: 0.3106 - val_accuracy: 0.8871
Epoch 4/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3074 - accuracy: 0.8881 - val_loss: 0.2673 - val_accuracy: 0.9027
Epoch 5/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2847 - accuracy: 0.8957 - val_loss: 0.2749 - val_accuracy: 0.8972
Epoch 6/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2651 - accuracy: 0.9039 - val_loss: 0.2493 - val_accuracy: 0.9069
Epoch 7/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2484 - accuracy: 0.9094 - val_loss: 0.2319 - val_accuracy: 0.9132
Epoch 8/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2386 - accuracy: 0.9114 - val_loss: 0.2438 - val_accuracy: 0.9096
Epoch 9/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2236 - accuracy: 0.9170 - val_loss: 0.2360 - val_accuracy: 0.9125
Epoch 10/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2144 - accuracy: 0.9199 - val_loss: 0.2337 - val_accuracy: 0.9144
```

3. **Plot:**



Assignment 5: Fashion MNIST

2. DIFFERENT MODELS:

NO	CHANGES IN MODEL	VALIDATION ACCURACY
1	Baseline	0.9144
2	Reducing dropout to 0.125	0.9173
3	Increasing layer to 128 with learning rate 0.01	0.9194
4	Nadam Optimizer with 3 layers	0.9215
5	Nadam Optimizer with 2 layers	0.9246

3. BEST MODEL(NADAM OPTIMIZER WITH 2 LAYERS):

1. **Code:**

```
import matplotlib.pyplot as plt
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
import tensorflow as tf

# Load and preprocess the MNIST dataset
mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images = train_images.reshape(60000, 28, 28, 1)
test_images = test_images.reshape(10000, 28, 28, 1)
train_images, test_images = train_images / 255.0, test_images / 255.0

# Define the model
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(128, (3, 3), activation = 'relu', input_shape=(28,
28, 1)),
    #tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    #tf.keras.layers.Dropout(0.25),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(10, activation='softmax')
```

Assignment 5: Fashion MNIST

```
])

# Compile the model
model.compile(
    optimizer=tf.keras.optimizers.Nadam(
        learning_rate=0.001,
        beta_1=0.9,
        beta_2=0.999,
        epsilon=1e-07,
        name='nadam'
    ),
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

# Train the model and store the history
history = model.fit(
    train_images, train_labels,

    batch_size=32,
    epochs=10,
    verbose=1,
    validation_data=(test_images, test_labels)
)

# Plot and save the training and validation accuracy
plt.figure(figsize=(8, 6))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()
plt.grid(True)
plt.savefig('new_accuracy_plot.png')  # Save the plot as a file
plt.show()
```

Assignment 5: Fashion MNIST

2. Output:

```
>>> exec(open("new.py").read())
Epoch 1/10
2024-11-25 16:30:35.964772: E tensorflow/core/grappler/optimizers/meta_optimizer.cc:1014] layout failed: INVALID_ARGUMENT: Size of values 0 does not match size of permutation
 4 @ fanin shape insequential_17/dropout_41/dropout/SelectV2-2-TransposeNHWCToNCHW-LayoutOptimizer
1875/1875 [==============================] - 8s 4ms/step - loss: 0.5297 - accuracy: 0.8077 - val_loss: 0.3415 - val_accuracy: 0.8710
Epoch 2/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.3655 - accuracy: 0.8702 - val_loss: 0.2751 - val_accuracy: 0.8973
Epoch 3/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.3154 - accuracy: 0.8869 - val_loss: 0.2648 - val_accuracy: 0.9029
Epoch 4/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.2875 - accuracy: 0.8958 - val_loss: 0.2351 - val_accuracy: 0.9152
Epoch 5/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.2632 - accuracy: 0.9044 - val_loss: 0.2296 - val_accuracy: 0.9164
Epoch 6/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.2477 - accuracy: 0.9096 - val_loss: 0.2273 - val_accuracy: 0.9181
Epoch 7/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.2377 - accuracy: 0.9130 - val_loss: 0.2151 - val_accuracy: 0.9252
Epoch 8/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.2243 - accuracy: 0.9186 - val_loss: 0.2366 - val_accuracy: 0.9137
Epoch 9/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.2130 - accuracy: 0.9225 - val_loss: 0.2196 - val_accuracy: 0.9217
Epoch 10/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.2042 - accuracy: 0.9248 - val_loss: 0.2244 - val_accuracy: 0.9246
```

3. Plot: