

```

Commented out IPython magic to ensure Python compatibility.

# %load_ext tensorboard
import datetime, os
import tensorflow as tf
import tensorflow_datasets as tfds

# טעינת הנתונים
(ds_train, ds_test), ds_info = tfds.load(
    'mnist',
    split=['train', 'test'],
    shuffle_files=True,
    as_supervised=True,
    with_info=True,
)

# נוקעיית נירמול
def normalize_img(image, label):
    """Normalizes images: `uint8` -> `float32`."""
    return tf.cast(image, tf.float32) / 255., label

# נירמול הנתונים
ds_train = ds_train.map(normalize_img, num_parallel_calls=tf.data.AUTOTUNE)
ds_train = ds_train.batch(32).prefetch(tf.data.AUTOTUNE)

ds_test = ds_test.map(normalize_img, num_parallel_calls=tf.data.AUTOTUNE)
ds_test = ds_test.batch(32).prefetch(tf.data.AUTOTUNE)

#####
# תחילת ניסויים:
#####

#1 ניסוי
# בניית המודל
model_1= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='relu'), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='relu'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='relu'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='relu'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_1.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

# אימון המודל
model_1.fit(ds_train, epochs=5)

# בדיקת המודל
model_1.evaluate(ds_test)

#1 תוצאות פלט ניסוי מספר
Epoch 1/5
1875/1875 ██████████ 17s 8ms/step - loss: 0.6841 - sparse_categorical_accuracy: 0.7875
Epoch 2/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.1537 - sparse_categorical_accuracy: 0.9555
Epoch 3/5
1875/1875 ██████████ 7s 4ms/step - loss: 0.1066 - sparse_categorical_accuracy: 0.9689
Epoch 4/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.0847 - sparse_categorical_accuracy: 0.9769
Epoch 5/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.0729 - sparse_categorical_accuracy: 0.9801
313/313 ██████████ 1s 4ms/step - loss: 0.1191 - sparse_categorical_accuracy: 0.9687
[0.11359720677137375, 0.9692000150680542]

#####

#2 ניסוי
# בניית המודל
model_2= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='sigmoid'), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='sigmoid'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='sigmoid'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='sigmoid'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_2.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

```

```

# אימון המודל
model_2.fit(ds_train, epochs=5)

# בדיקת המודל
model_2.evaluate(ds_test)

#2 תוצאות פלט ניסוי מספר

Epoch 1/5
1875/1875 ██████████ 12s 6ms/step - loss: 1.9515 - sparse_categorical_accuracy: 0.2493
Epoch 2/5
1875/1875 ██████████ 17s 4ms/step - loss: 0.6994 - sparse_categorical_accuracy: 0.8061
Epoch 3/5
1875/1875 ██████████ 9s 4ms/step - loss: 0.3209 - sparse_categorical_accuracy: 0.9192
Epoch 4/5
1875/1875 ██████████ 11s 4ms/step - loss: 0.2196 - sparse_categorical_accuracy: 0.9432
Epoch 5/5
1875/1875 ██████████ 11s 5ms/step - loss: 0.1764 - sparse_categorical_accuracy: 0.9541
313/313 ██████████ 2s 4ms/step - loss: 0.1708 - sparse_categorical_accuracy: 0.9534

#####
#3 ניסוי
# בניית המודל

model_3= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='relu'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='relu'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='relu'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_3.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

# אימון המודל
model_3.fit(ds_train, epochs=5)
# בדיקת המודל
model_3.evaluate(ds_test)

#3 תוצאות פלט ניסוי מספר

1875/1875 ██████████ 12s 6ms/step - loss: 0.7148 - sparse_categorical_accuracy: 0.8130
Epoch 2/5
1875/1875 ██████████ 18s 5ms/step - loss: 0.2275 - sparse_categorical_accuracy: 0.9488
Epoch 3/5
1875/1875 ██████████ 9s 4ms/step - loss: 0.1815 - sparse_categorical_accuracy: 0.9616
Epoch 4/5
1875/1875 ██████████ 10s 4ms/step - loss: 0.1626 - sparse_categorical_accuracy: 0.9657
Epoch 5/5
1875/1875 ██████████ 12s 5ms/step - loss: 0.1534 - sparse_categorical_accuracy: 0.9684
313/313 ██████████ 2s 5ms/step - loss: 0.1658 - sparse_categorical_accuracy: 0.9634

#####
#4 ניסוי
# בניית המודל

model_4= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.002)), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='relu'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='relu'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='relu'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_4.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

# אימון המודל
model_4.fit(ds_train, epochs=5)
# בדיקת המודל
model_4.evaluate(ds_test)

#4 תוצאות פלט ניסוי מספר

Epoch 1/5
1875/1875 ██████████ 13s 6ms/step - loss: 0.7303 - sparse_categorical_accuracy: 0.8228
Epoch 2/5
1875/1875 ██████████ 17s 5ms/step - loss: 0.2409 - sparse_categorical_accuracy: 0.9486
Epoch 3/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.2011 - sparse_categorical_accuracy: 0.9569

```

```

Epoch 4/5
1875/1875 ██████████ 9s 4ms/step - loss: 0.1876 - sparse_categorical_accuracy: 0.9608
Epoch 5/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.1797 - sparse_categorical_accuracy: 0.9633
313/313 ██████████ 2s 4ms/step - loss: 0.1736 - sparse_categorical_accuracy: 0.9663

#####
#5 ניסוי
#בניית המודל

model_5= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l2(0.001)), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='sigmoid'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='sigmoid'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='sigmoid'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_5.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

# אימון המודל
model_5.fit(ds_train, epochs=5)

# בדיקת המודל
model_5.evaluate(ds_test)

# תוצאות פלט ניסוי מספר 5

Epoch 1/5
1875/1875 ██████████ 13s 6ms/step - loss: 2.0902 - sparse_categorical_accuracy: 0.2342
Epoch 2/5
1875/1875 ██████████ 18s 5ms/step - loss: 0.9426 - sparse_categorical_accuracy: 0.7576
Epoch 3/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.5887 - sparse_categorical_accuracy: 0.8598
Epoch 4/5
1875/1875 ██████████ 9s 4ms/step - loss: 0.4609 - sparse_categorical_accuracy: 0.8996
Epoch 5/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.3994 - sparse_categorical_accuracy: 0.9162
313/313 ██████████ 2s 4ms/step - loss: 0.4051 - sparse_categorical_accuracy: 0.9074

#####
#6 ניסוי
#בניית המודל

model_6= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l2(0.002)), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='sigmoid'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='sigmoid'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='sigmoid'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_6.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

# אימון המודל
model_6.fit(ds_train, epochs=5)

# בדיקת המודל
model_6.evaluate(ds_test)

# תוצאות פלט ניסוי מספר 6

Epoch 1/5
1875/1875 ██████████ 12s 6ms/step - loss: 2.1968 - sparse_categorical_accuracy: 0.1899
Epoch 2/5
1875/1875 ██████████ 20s 6ms/step - loss: 1.1541 - sparse_categorical_accuracy: 0.6474
Epoch 3/5
1875/1875 ██████████ 17s 4ms/step - loss: 0.7355 - sparse_categorical_accuracy: 0.8285
Epoch 4/5
1875/1875 ██████████ 10s 4ms/step - loss: 0.5634 - sparse_categorical_accuracy: 0.8761
Epoch 5/5
1875/1875 ██████████ 11s 4ms/step - loss: 0.4889 - sparse_categorical_accuracy: 0.8928
313/313 ██████████ 2s 4ms/step - loss: 0.5104 - sparse_categorical_accuracy: 0.8826

#####
#7 ניסוי
#בניית המודל

model_7= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),

```

```

tf.keras.layers.Dense(93, activation='sigmoid'), # שכבה ראשונה
tf.keras.layers.Dense(16, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l2(0.001)), # שכבה שנייה
tf.keras.layers.Dense(92, activation='sigmoid'), # שכבה שלישית
tf.keras.layers.Dense(14, activation='sigmoid'), # שכבה רביעית
tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_7.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

# אימון המודל
model_7.fit(ds_train, epochs=5)

# בדיקת המודל
model_7.evaluate(ds_test)

# תוצאות פלט ניסוי מספר 7

Epoch 1/5
1875/1875 ██████████ 15s 7ms/step - loss: 2.0448 - sparse_categorical_accuracy: 0.2633
Epoch 2/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.7928 - sparse_categorical_accuracy: 0.8086
Epoch 3/5
1875/1875 ██████████ 9s 4ms/step - loss: 0.4447 - sparse_categorical_accuracy: 0.9014
Epoch 4/5
1875/1875 ██████████ 11s 4ms/step - loss: 0.3217 - sparse_categorical_accuracy: 0.9254
Epoch 5/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.2636 - sparse_categorical_accuracy: 0.9373
313/313 ██████████ 1s 4ms/step - loss: 0.2623 - sparse_categorical_accuracy: 0.9427

#####
#8 ניסוי
# בניית המודל

model_8= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='sigmoid'), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='sigmoid'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l2(0.001)), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='sigmoid'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_8.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

# אימון המודל
model_8.fit(ds_train, epochs=5)

# בדיקת המודל
model_8.evaluate(ds_test)

# תוצאות פלט ניסוי מספר 8

Epoch 1/5
1875/1875 ██████████ 12s 6ms/step - loss: 2.0772 - sparse_categorical_accuracy: 0.2246
Epoch 2/5
1875/1875 ██████████ 16s 4ms/step - loss: 0.9296 - sparse_categorical_accuracy: 0.8003
Epoch 3/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.4916 - sparse_categorical_accuracy: 0.9067
Epoch 4/5
1875/1875 ██████████ 7s 4ms/step - loss: 0.3514 - sparse_categorical_accuracy: 0.9307
Epoch 5/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.2806 - sparse_categorical_accuracy: 0.9431
313/313 ██████████ 2s 4ms/step - loss: 0.2578 - sparse_categorical_accuracy: 0.9470

#####
#9 ניסוי
# בניית המודל

model_9= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='sigmoid'), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='sigmoid'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='sigmoid'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l2(0.003)), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])

# קימפול המודל
model_9.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)

```

```
# אימון המודל
model_9.fit(ds_train, epochs=5)

# בדיקת המודל
model_9.evaluate(ds_test)

#9 תוצאות פלט ניסוי מספר

Epoch 1/5
1875/1875 ██████████ 13s 6ms/step - loss: 1.9907 - sparse_categorical_accuracy: 0.2913
Epoch 2/5
1875/1875 ██████████ 7s 4ms/step - loss: 0.8747 - sparse_categorical_accuracy: 0.8225
Epoch 3/5
1875/1875 ██████████ 11s 4ms/step - loss: 0.5340 - sparse_categorical_accuracy: 0.9056
Epoch 4/5
1875/1875 ██████████ 11s 5ms/step - loss: 0.3896 - sparse_categorical_accuracy: 0.9300
Epoch 5/5
1875/1875 ██████████ 7s 4ms/step - loss: 0.3079 - sparse_categorical_accuracy: 0.9444
313/313 ██████████ 2s 5ms/step - loss: 0.2887 - sparse_categorical_accuracy: 0.9461
```

```
#####
#10 ניסוי
#בניית המודל
```

```
model_10= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(93, activation='relu'), # שכבה רגילה
    tf.keras.layers.Dropout(0.4), # Dropout שכבת Dense
    tf.keras.layers.Dense(16, activation='sigmoid'),
    tf.keras.layers.Dropout(0.4), # Dropout נוסף
    tf.keras.layers.Dense(92, activation='relu'),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

```
# קימפול המודל
model_10.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)
```

```
# אימון המודל
model_10.fit(ds_train, epochs=5)
```

```
# בדיקת המודל
model_10.evaluate(ds_test)
```

```
#10 תוצאות פלט ניסוי מספר

Epoch 1/5
1875/1875 ██████████ 12s 6ms/step - loss: 1.2663 - sparse_categorical_accuracy: 0.5701
Epoch 2/5
1875/1875 ██████████ 17s 4ms/step - loss: 0.5724 - sparse_categorical_accuracy: 0.8108
Epoch 3/5
1875/1875 ██████████ 11s 5ms/step - loss: 0.4939 - sparse_categorical_accuracy: 0.8399
Epoch 4/5
1875/1875 ██████████ 11s 5ms/step - loss: 0.4307 - sparse_categorical_accuracy: 0.8662
Epoch 5/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.4002 - sparse_categorical_accuracy: 0.8793
313/313 ██████████ 2s 7ms/step - loss: 0.1791 - sparse_categorical_accuracy: 0.9511
```

```
#####
#11 ניסוי
#בניית המודל
```

```
model_11= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='relu'),# שכבה ראשונה
    tf.keras.layers.Dropout(0.3),# שכבת Dropout
    tf.keras.layers.Dense(16, activation='relu'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='relu'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation='relu'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])
```

```
# קימפול המודל
model_11.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)
```

```
# אימון המודל
model_11.fit(ds_train, epochs=5)
```

```
# בדיקת המודל
model_11.evaluate(ds_test)
```

```
#11 תוצאות פלט ניסוי מספר
```

```
Epoch 1/5
```

```
1875/1875 ██████████ 13s 6ms/step - loss: 0.7492 - sparse_categorical_accuracy: 0.7547
Epoch 2/5
1875/1875 ██████████ 7s 4ms/step - loss: 0.2252 - sparse_categorical_accuracy: 0.9344
Epoch 3/5
1875/1875 ██████████ 10s 4ms/step - loss: 0.1833 - sparse_categorical_accuracy: 0.9469
Epoch 4/5
1875/1875 ██████████ 12s 5ms/step - loss: 0.1675 - sparse_categorical_accuracy: 0.9529
Epoch 5/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.1508 - sparse_categorical_accuracy: 0.9581
313/313 ██████████ 2s 4ms/step - loss: 0.1068 - sparse_categorical_accuracy: 0.9683
```

```
#####
```

```
#12 ניסוי
בניית המודל
```

```
model_12= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)), # התחונה
    tf.keras.layers.Dense(93, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)), # שכבה ראשונה עם רגוליציה L2
    tf.keras.layers.Dense(16, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l2(0.001)), # שכבה שנייה עם רגוליציה L2
    tf.keras.layers.Dense(92, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.002)), # שכבה שלישית עם רגוליציה L2
    tf.keras.layers.Dense(14, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l2(0.003)), # שכבה רביעית עם רגוליציה L2
    tf.keras.layers.Dense(10, activation='softmax') # שכבה אחרונה עם Softmax
])
```

```
# קימפול המודל
model_12.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)
```

```
# אימון המודל
model_12.fit(ds_train, epochs=5)
```

```
# בדיקת המודל
model_12.evaluate(ds_test)
```

```
# תוצאות פלט ניסוי מספר 12
```

```
Epoch 1/5
1875/1875 ██████████ 13s 6ms/step - loss: 1.5807 - sparse_categorical_accuracy: 0.6094
Epoch 2/5
1875/1875 ██████████ 16s 4ms/step - loss: 0.5889 - sparse_categorical_accuracy: 0.9229
Epoch 3/5
1875/1875 ██████████ 11s 4ms/step - loss: 0.4772 - sparse_categorical_accuracy: 0.9369
Epoch 4/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.4343 - sparse_categorical_accuracy: 0.9431
Epoch 5/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.4088 - sparse_categorical_accuracy: 0.9479
313/313 ██████████ 2s 5ms/step - loss: 0.4005 - sparse_categorical_accuracy: 0.9477
```

```
#####
```

```
#13 ניסוי
בניית המודל
```

```
model_13= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='relu'), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='relu'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation='relu'), # שכבה שלישית
    tf.keras.layers.Dropout(0.3), # שכבת Dropout
    tf.keras.layers.Dense(14, activation='relu'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])
```

```
# קימפול המודל
model_13.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)
```

```
# אימון המודל
model_13.fit(ds_train, epochs=5)
```

```
# בדיקת המודל
model_13.evaluate(ds_test)
```

```
# תוצאות פלט ניסוי מספר 13
```

```
Epoch 1/5
1875/1875 ██████████ 12s 6ms/step - loss: 0.7066 - sparse_categorical_accuracy: 0.7734
Epoch 2/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.1584 - sparse_categorical_accuracy: 0.9559
Epoch 3/5
1875/1875 ██████████ 10s 5ms/step - loss: 0.1172 - sparse_categorical_accuracy: 0.9684
Epoch 4/5
1875/1875 ██████████ 7s 4ms/step - loss: 0.0994 - sparse_categorical_accuracy: 0.9735
Epoch 5/5
1875/1875 ██████████ 12s 5ms/step - loss: 0.0833 - sparse_categorical_accuracy: 0.9781
313/313 ██████████ 1s 4ms/step - loss: 0.1236 - sparse_categorical_accuracy: 0.9735
```

#####

#14 ניסוי  
בניית המודל

```
model_14= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='relu'), # שכבה ראשונה
    tf.keras.layers.Dropout(0.6),# שכבת Dropout
    tf.keras.layers.Dense(16, activation='relu'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation= 'relu'), # שכבה שלישית
    tf.keras.layers.Dense(14, activation= 'relu'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])
```

```
# קימפול המודל
model_14.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)
```

```
# אימון המודל
model_14.fit(ds_train, epochs=5)
```

```
# בדיקת המודל
model_14.evaluate(ds_test)
```

#14 תוצאות פלט ניסוי מספר

```
Epoch 1/5
1875/1875 ██████████ 15s 7ms/step - loss: 0.9726 - sparse_categorical_accuracy: 0.6721
Epoch 2/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.3617 - sparse_categorical_accuracy: 0.8932
Epoch 3/5
1875/1875 ██████████ 9s 5ms/step - loss: 0.3217 - sparse_categorical_accuracy: 0.9050
Epoch 4/5
1875/1875 ██████████ 11s 5ms/step - loss: 0.2918 - sparse_categorical_accuracy: 0.9147
Epoch 5/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.2820 - sparse_categorical_accuracy: 0.9172
313/313 ██████████ 2s 5ms/step - loss: 0.1606 - sparse_categorical_accuracy: 0.9534
```

#####

#15 ניסוי  
בניית המודל

```
model_15= tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(93, activation='relu'), # שכבה ראשונה
    tf.keras.layers.Dense(16, activation='relu'), # שכבה שנייה
    tf.keras.layers.Dense(92, activation= 'relu'), # שכבה שלישית
    tf.keras.layers.Dropout(0.5),# שכבת Dropout
    tf.keras.layers.Dense(14, activation= 'relu'), # שכבה רביעית
    tf.keras.layers.Dense(10, activation='softmax'), # הגדרה של 10 קטגוריות
])
```

```
# קימפול המודל
model_15.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=[tf.keras.metrics.SparseCategoricalAccuracy()],
)
```

```
# אימון המודל
model_15.fit(ds_train, epochs=5)
```

```
# בדיקת המודל
model_15.evaluate(ds_test)
```

#15 תוצאות פלט ניסוי מספר

```
Epoch 1/5
1875/1875 ██████████ 13s 6ms/step - loss: 0.8887 - sparse_categorical_accuracy: 0.7052
Epoch 2/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.2036 - sparse_categorical_accuracy: 0.9459
Epoch 3/5
1875/1875 ██████████ 13s 6ms/step - loss: 0.1468 - sparse_categorical_accuracy: 0.9622
Epoch 4/5
1875/1875 ██████████ 10s 5ms/step - loss: 0.1244 - sparse_categorical_accuracy: 0.9685
Epoch 5/5
1875/1875 ██████████ 8s 4ms/step - loss: 0.1120 - sparse_categorical_accuracy: 0.9736
313/313 ██████████ 2s 4ms/step - loss: 0.1384 - sparse_categorical_accuracy: 0.9665
[0.13854868710041046, 0.9664999842643738]
```