CNN.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

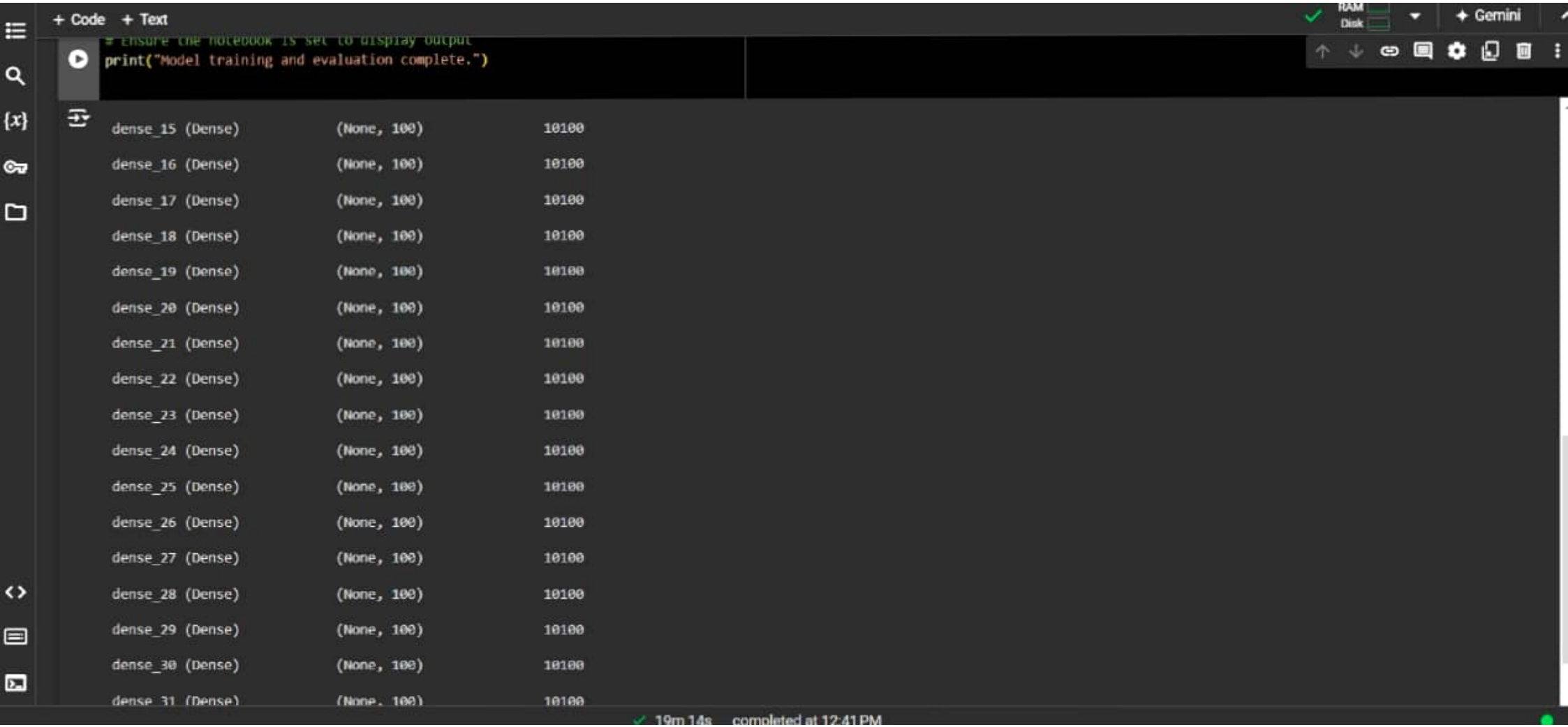
Comment Share

+ Code + Text

RAM Disk Gemini

```python
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, models

# Set random seed for reproducibility
np.random.seed(123)
tf.random.set_seed(123)

# Load the data and normalize
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

class_names = ["T-shirt/top", "Trouser", "Pullover", "Dress", "Coat", "Sandal", "Shirt", "Sneaker", "Bag", "Ankle boot"]

# Print to confirm data is loaded
print("Data loaded successfully")

# Build the model
model = models.Sequential()

# Add normalization layer
model.add(layers.Input(shape=(28, 28)))
model.add(layers.Reshape((28, 28, 1)))
model.add(layers.Conv2D(32, (3, 3), activation='selu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='selu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='selu'))
```

```
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='selu'))
model.add(layers.Flatten())

# Add 100 hidden layers each with 100 neurons using SELU activation function
for i in range(100):
    model.add(layers.Dense(100, activation='selu'))

model.add(layers.Dense(10, activation='softmax'))

# Print to confirm model is built
print("Model built successfully")

# Compile the model
model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Print to confirm model is compiled
print("Model compiled successfully")

# Train the model with verbose set to 1 for detailed output
history = model.fit(train_images, train_labels, epochs=10, validation_data=(test_images, test_labels), verbose=1)

# Evaluate the model with verbose set to 2 for detailed output
test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)
print(f'Test accuracy: {test_acc}')

# Print the model summary
model.summary()

# Ensure the notebook is set to display output
print("Model training and evaluation complete.")
```



CNN.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

Gemini

```
# Ensure the notebook is set to display output
print("Model training and evaluation complete.")
```

```
dense_23 (Dense)            (None, 100)               10100
dense_24 (Dense)            (None, 100)               10100
dense_25 (Dense)            (None, 100)               10100
dense_26 (Dense)            (None, 100)               10100
dense_27 (Dense)            (None, 100)               10100
dense_28 (Dense)            (None, 100)               10100
dense_29 (Dense)            (None, 100)               10100
dense_30 (Dense)            (None, 100)               10100
dense_31 (Dense)            (None, 100)               10100
dense_32 (Dense)            (None, 100)               10100
dense_33 (Dense)            (None, 100)               10100
dense_34 (Dense)            (None, 100)               10100
dense_35 (Dense)            (None, 100)               10100
dense_36 (Dense)            (None, 100)               10100
dense_37 (Dense)            (None, 100)               10100
dense_38 (Dense)            (None, 100)               10100
dense_39 (Dense)            (None, 100)               10100
```

CNN.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Gemini

```
# Ensure the notebook is set to display output
print("Model training and evaluation complete.")
```

```
dense_32 (Dense)        (None, 100)        10100
dense_33 (Dense)        (None, 100)        10100
dense_34 (Dense)        (None, 100)        10100
dense_35 (Dense)        (None, 100)        10100
dense_36 (Dense)        (None, 100)        10100
dense_37 (Dense)        (None, 100)        10100
dense_38 (Dense)        (None, 100)        10100
dense_39 (Dense)        (None, 100)        10100
dense_40 (Dense)        (None, 100)        10100
dense_41 (Dense)        (None, 100)        10100
dense_42 (Dense)        (None, 100)        10100
dense_43 (Dense)        (None, 100)        10100
dense_44 (Dense)        (None, 100)        10100
dense_45 (Dense)        (None, 100)        10100
dense_46 (Dense)        (None, 100)        10100
dense_47 (Dense)        (None, 100)        10100
dense_48 (Dense)        (None, 100)        10100
```