

localhost:8888/notebooks/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 22 minutes ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

```
[ ]: import tensorflow as tf
from tensorflow.keras import layers, models, datasets

# Load CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()

# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0

# Define the CNN architecture
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax')
])

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

# Train the model
model.fit(train_images, train_labels, epochs=10, batch_size=64, validation_data=(test_images, test_labels))

# Evaluate the model
test_loss, test_acc = model.evaluate(test_images, test_labels)
print("Test Accuracy:", test_acc)
```

Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>  
40198144/170498071 — 8:42 4us/step

```
[1]: import tensorflow as tf
from tensorflow.keras import layers, models, datasets

[2]: # Load CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()
```

Home x Untitled1 x +

localhost:8888/notebooks/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 27 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
[6]: # Train the model
model.fit(train_images, train_labels, epochs=10, batch_size=64, validation_data=(test_images, test_labels))

Epoch 1/10
782/782 ————— 9s 10ms/step - accuracy: 0.3383 - loss: 1.7711 - val_accuracy: 0.5507 - val_loss: 1.2671
Epoch 2/10
782/782 ————— 7s 9ms/step - accuracy: 0.5474 - loss: 1.2644 - val_accuracy: 0.6024 - val_loss: 1.1134
Epoch 3/10
782/782 ————— 7s 9ms/step - accuracy: 0.6150 - loss: 1.0994 - val_accuracy: 0.6387 - val_loss: 1.0233
Epoch 4/10
782/782 ————— 7s 9ms/step - accuracy: 0.6510 - loss: 0.9996 - val_accuracy: 0.6389 - val_loss: 1.0187
Epoch 5/10
782/782 ————— 7s 9ms/step - accuracy: 0.6830 - loss: 0.9096 - val_accuracy: 0.6654 - val_loss: 0.9593
Epoch 6/10
782/782 ————— 7s 9ms/step - accuracy: 0.7117 - loss: 0.8316 - val_accuracy: 0.6835 - val_loss: 0.9224
Epoch 7/10
782/782 ————— 7s 9ms/step - accuracy: 0.7239 - loss: 0.8014 - val_accuracy: 0.6957 - val_loss: 0.8995
Epoch 8/10
782/782 ————— 7s 9ms/step - accuracy: 0.7404 - loss: 0.7518 - val_accuracy: 0.7113 - val_loss: 0.8538
Epoch 9/10
782/782 ————— 7s 9ms/step - accuracy: 0.7513 - loss: 0.7111 - val_accuracy: 0.7029 - val_loss: 0.8668
Epoch 10/10
782/782 ————— 7s 9ms/step - accuracy: 0.7646 - loss: 0.6724 - val_accuracy: 0.7138 - val_loss: 0.8417

[6]: <keras.src.callbacks.history.History at 0x1a4ddce0510>

[7]: # Evaluate the model
test_loss, test_acc = model.evaluate(test_images, test_labels)
print("Test Accuracy:", test_acc)

313/313 ————— 1s 2ms/step - accuracy: 0.7103 - loss: 0.8368
Test Accuracy: 0.7138000130653381

[ ]:
```