

dl-lab-7

May 19, 2023

```
[2]: import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras
from tensorflow.keras import layers

# Load and preprocess the data
(X_train, y_train), (X_test, y_test) = keras.datasets.mnist.load_data()
X_train = X_train.reshape(-1, 28, 28, 1).astype("float32") / 255.0
X_test = X_test.reshape(-1, 28, 28, 1).astype("float32") / 255.0

# Define the CNN model
model = keras.Sequential([
    layers.Conv2D(32, kernel_size=(3, 3), activation="relu", input_shape=(28, 28, 1)),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Flatten(),
    layers.Dense(128, 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(X_train, y_train, batch_size=128, epochs=5, validation_split=0.1)

# Visualize the learned features
layer_outputs = [layer.output for layer in model.layers]
activation_model = keras.Model(inputs=model.input, outputs=layer_outputs)
activations = activation_model.predict(X_test[:1])

# Visualize the results
for layer_activation in activations:
    if len(layer_activation.shape) == 4:
```

```

num_filters = layer_activation.shape[-1]
rows = int(np.ceil(num_filters / 8)) # Round up the number of rows
fig, axs = plt.subplots(rows, 8, figsize=(12, rows * 2))
axs = axs.flatten()
for i in range(num_filters):
    axs[i].imshow(layer_activation[0, :, :, i], cmap="viridis")
    axs[i].axis("off")
plt.tight_layout()
plt.show()

```

Epoch 1/5
422/422 [=====] - 51s 120ms/step - loss: 0.2357 -
accuracy: 0.9305 - val_loss: 0.0779 - val_accuracy: 0.9765
Epoch 2/5
422/422 [=====] - 48s 115ms/step - loss: 0.0635 -
accuracy: 0.9804 - val_loss: 0.0542 - val_accuracy: 0.9852
Epoch 3/5
422/422 [=====] - 47s 112ms/step - loss: 0.0440 -
accuracy: 0.9863 - val_loss: 0.0481 - val_accuracy: 0.9863
Epoch 4/5
422/422 [=====] - 48s 115ms/step - loss: 0.0346 -
accuracy: 0.9893 - val_loss: 0.0451 - val_accuracy: 0.9887
Epoch 5/5
422/422 [=====] - 48s 114ms/step - loss: 0.0261 -
accuracy: 0.9920 - val_loss: 0.0383 - val_accuracy: 0.9900
1/1 [=====] - 0s 85ms/step







