```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
from tensorflow.keras.optimizers import RMSprop


# Load CIFAR-10 dataset
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# Normalization
x_train, x_test = x_train / 255.0, x_test / 255.0

# one-hot encoding
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)


model = models.Sequential()

# Flatten the input image dimensions to 1D
model.add(layers.Flatten(input_shape=(32, 32, 3)))
#adding layers
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(128, activation='relu'))

# Output layer
model.add(layers.Dense(10, activation='relu'))
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_2 (Flatten)         (None, 3072)              0
```

```
 dense_8 (Dense)             (None, 512)               1573376

 dense_9 (Dense)             (None, 256)               131328

 dense_10 (Dense)            (None, 128)               32896

 dense_11 (Dense)            (None, 10)                1290

=================================================================
Total params: 1738890 (6.63 MB)
Trainable params: 1738890 (6.63 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(x_train, y_train, epochs=50, batch_size=64, validation_data=(x_test, y_test))
```

```
Epoch 1/50
782/782 [==============================] - 28s 35ms/step - loss: 4.6955 - accuracy: 0.2143 - val_loss: 4.7411 - val_
Epoch 2/50
782/782 [==============================] - 27s 35ms/step - loss: 4.6857 - accuracy: 0.2192 - val_loss: 4.7207 - val_
Epoch 3/50
782/782 [==============================] - 26s 33ms/step - loss: 4.6530 - accuracy: 0.2427 - val_loss: 4.6316 - val_
Epoch 4/50
782/782 [==============================] - 27s 34ms/step - loss: 4.6376 - accuracy: 0.2508 - val_loss: 4.6014 - val_
Epoch 5/50
782/782 [==============================] - 27s 35ms/step - loss: 4.6135 - accuracy: 0.2559 - val_loss: 4.5818 - val_
Epoch 6/50
782/782 [==============================] - 27s 35ms/step - loss: 4.7667 - accuracy: 0.1932 - val_loss: 4.8858 - val_
Epoch 7/50
782/782 [==============================] - 29s 37ms/step - loss: 4.8176 - accuracy: 0.1608 - val_loss: 4.9472 - val_
Epoch 8/50
782/782 [==============================] - 29s 37ms/step - loss: 4.8157 - accuracy: 0.1481 - val_loss: 4.8028 - val_
Epoch 9/50
782/782 [==============================] - 28s 36ms/step - loss: 4.7603 - accuracy: 0.1636 - val_loss: 4.7088 - val_
```

```
Epoch 10/50
782/782 [==============================] - 28s 36ms/step - loss: 4.6975 - accuracy: 0.2069 - val_loss: 4.6619 - val_
Epoch 11/50
782/782 [==============================] - 28s 36ms/step - loss: 4.6642 - accuracy: 0.2285 - val_loss: 4.6448 - val_
Epoch 12/50
782/782 [==============================] - 27s 34ms/step - loss: 4.7893 - accuracy: 0.1705 - val_loss: 4.8822 - val_
Epoch 13/50
782/782 [==============================] - 28s 36ms/step - loss: 4.7517 - accuracy: 0.1876 - val_loss: 4.6736 - val_
Epoch 14/50
782/782 [==============================] - 29s 37ms/step - loss: 4.6520 - accuracy: 0.2394 - val_loss: 4.6279 - val_
Epoch 15/50
782/782 [==============================] - 27s 34ms/step - loss: 4.6199 - accuracy: 0.2523 - val_loss: 4.6008 - val_
Epoch 16/50
782/782 [==============================] - 30s 38ms/step - loss: 4.6145 - accuracy: 0.2559 - val_loss: 4.6373 - val_
Epoch 17/50
782/782 [==============================] - 29s 37ms/step - loss: 4.6695 - accuracy: 0.2127 - val_loss: 4.6417 - val_
Epoch 18/50
782/782 [==============================] - 28s 36ms/step - loss: 4.6411 - accuracy: 0.2546 - val_loss: 4.6046 - val_
Epoch 19/50
782/782 [==============================] - 31s 40ms/step - loss: 4.6164 - accuracy: 0.2536 - val_loss: 4.5992 - val_
Epoch 20/50
782/782 [==============================] - 31s 40ms/step - loss: 4.5890 - accuracy: 0.2682 - val_loss: 4.5706 - val_
Epoch 21/50
782/782 [==============================] - 30s 39ms/step - loss: 4.5763 - accuracy: 0.2707 - val_loss: 4.5696 - val_
Epoch 22/50
782/782 [==============================] - 31s 40ms/step - loss: 4.6638 - accuracy: 0.2241 - val_loss: 4.5909 - val_
Epoch 23/50
782/782 [==============================] - 27s 35ms/step - loss: 4.5702 - accuracy: 0.2728 - val_loss: 4.5548 - val_
Epoch 24/50
782/782 [==============================] - 28s 35ms/step - loss: 4.5579 - accuracy: 0.2777 - val_loss: 4.5520 - val_
Epoch 25/50
782/782 [==============================] - 31s 40ms/step - loss: 4.5493 - accuracy: 0.2816 - val_loss: 4.5284 - val_
Epoch 26/50
782/782 [==============================] - 32s 41ms/step - loss: 3.8427 - accuracy: 0.2324 - val_loss: 3.3337 - val_
Epoch 27/50
782/782 [==============================] - 29s 38ms/step - loss: 3.4911 - accuracy: 0.1435 - val_loss: 3.4300 - val_
Epoch 28/50
782/782 [==============================] - 26s 34ms/step - loss: 3.3732 - accuracy: 0.2072 - val_loss: 3.3075 - val_
Epoch 29/50
782/782 [==============================] - 28s 36ms/step - loss: 3.2859 - accuracy: 0.2568 - val_loss: 3.2532 - val_
```

```
test_loss, test_acc = model.evaluate(x_test, y_test, verbose=2)
print(f'Test accuracy: {test_acc}')
```

```
313/313 - 2s - loss: 3.1254 - accuracy: 0.3170 - 2s/epoch - 6ms/step
Test accuracy: 0.31700000166893005
```