

EX NO: 1 Working of CNN architecture to classify images

Aim:

To build and evaluate a Convolutional Neural Network (CNN) model using TensorFlow and Keras to classify handwritten digits from the MNIST dataset.

Algorithm:

1. Import required libraries and load the MNIST dataset.
2. Normalize the images and convert labels to one-hot encoding.
3. Build a CNN model with Conv2D, MaxPooling2D, Flatten, and Dense layers.
4. Compile the model using 'adam' optimizer and 'categorical_crossentropy' loss.
5. Train the model on the training data for 10 epochs.
6. Evaluate the model using test data.
7. Display a sample prediction using matplotlib.

Code:

```
import tensorflow as tf

from tensorflow import keras

import matplotlib.pyplot as plt

import numpy as np

(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()

x_train, x_test = x_train[..., None] / 255.0, x_test[..., None] / 255.0

y_train, y_test = map(lambda y: keras.utils.to_categorical(y, 10), [y_train, y_test])

model = keras.Sequential([
```

```
keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
keras.layers.MaxPooling2D((2, 2)),
keras.layers.Flatten(),
keras.layers.Dense(64, activation='relu'),
keras.layers.Dense(10, activation='softmax')
])

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(x_train, y_train, batch_size=128, epochs=10, validation_split=0.1)
test_loss, test_acc = model.evaluate(x_test, y_test)
print('Test accuracy:', test_acc)
predicted_label = np.argmax(model.predict(x_test[:1])[0])
plt.imshow(x_test[0].squeeze(), cmap='gray')
plt.title(f'True: {np.argmax(y_test[0])}, Predicted: {predicted_label}')
plt.axis('off')
plt.show()
```

Output:

Test accuracy: 0.9875

True: 7, Predicted: 7

Result:

The CNN model achieved a high accuracy on the MNIST test dataset. A sample digit image was displayed with its true and predicted label using matplotlib.

