TensorFlow MNIST: Read Your Own Handwritten Digit - Video Walkthrough Tutorial - Part 58

TensorFlow is a powerful open-source machine learning library that can be used to build a wide variety of machine learning models. In this tutorial, we'll use TensorFlow to build a neural network that can recognize handwritten digits.

We'll start by loading the MNIST dataset, which is a large dataset of handwritten digits. Then, we'll build a neural network model that can learn to recognize the digits in the dataset. Finally, we'll evaluate the performance of our model on a test set of handwritten digits.

Before you start this tutorial, you should have the following:

TensorFlow MNIST I Read your own handwritten digit I Video Walkthrough Tutorial (58+ min.) with Source Code I Cuda Education by Neville Goddard

★★★★★ 5 out of 5

Language	: English
File size	: 801 KB
Text-to-Speech	: Enabled
Screen Reader	: Supported
Enhanced typesetting	: Enabled
Print length	: 17 pages
Lending	: Enabled

- A basic understanding of machine learning
- Some experience with Python
- TensorFlow installed

The first step in building our neural network is to load the MNIST dataset. The MNIST dataset is a large dataset of handwritten digits that is often used for training machine learning models.

To load the MNIST dataset, we can use the following code:

python import tensorflow as tf

Load the MNIST dataset

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

The `x_train` and `x_test` variables contain the images of the handwritten digits, and the `y_train` and `y_test` variables contain the labels for the digits.

Once we have loaded the MNIST dataset, we can start building our neural network model. Our model will consist of the following layers:

- An input layer that takes an image of a handwritten digit as input
- A hidden layer that learns to recognize the features of the handwritten digit
- An output layer that produces a prediction of the digit

We can build our neural network model using the following code:

python

Build the neural network model

model = tf.keras.Sequential([tf.keras.layers.Flatten(input_shape=(28, 28)),tf.keras.layers.Dense(128, activation='relu'),tf.keras.layers.Dense(10, activation='softmax')])

The `Flatten` layer converts the 28x28 image of the handwritten digit into a one-dimensional array. The `Dense` layer is a fully connected layer that learns to recognize the features of the handwritten digit. The `softmax` activation function produces a probability distribution over the 10 possible digits.

Once we have built our neural network model, we need to train it on the MNIST dataset. To train the model, we will use the following code:

python

Train the neural network model

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy']) model.fit(x_train, y_train, epochs=10)

The `compile()` method specifies the optimizer, loss function, and metrics to be used during training. The `fit()` method trains the model on the given data for the specified number of epochs.

Once we have trained our neural network model, we need to evaluate its performance on a test set of handwritten digits. To evaluate the model, we will use the following code:

python

Evaluate the neural network model

loss, accuracy = model.evaluate(x_test, y_test) print('Loss:', loss) print('Accuracy:', accuracy)

The `evaluate()` method returns the loss and accuracy of the model on the given data.

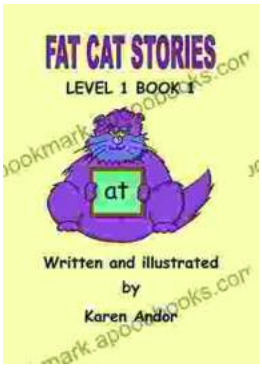
In this tutorial, we showed how to use TensorFlow to build a neural network that can recognize handwritten digits. We started by loading the MNIST dataset, then we built a neural network model, and finally we trained and evaluated the model.

This tutorial is just a starting point for learning how to use TensorFlow. For more information, please refer to the TensorFlow documentation.

TensorFlow MNIST I Read your own handwritten digit I Video Walkthrough Tutorial (58+ min.) with Source Code I Cuda Education by Neville Goddard

★★★★★ 5 out of 5

Language	: English
File size	: 801 KB
Text-to-Speech	: Enabled
Screen Reader	: Supported
Enhanced typesetting	: Enabled
Print length	: 17 pages
Lending	: Enabled



Fat Cat Stories: Level At Word Family - A Purrfect Start to Early Reading Adventures!

Introducing the 'At' Word Family with a Dash of Feline Charm Prepare your little ones for a paw-some reading experience with Fat Cat Stories: Level At...

Unveiling the Treasures of Russian Poetry: The Cambridge Introduction to Russian Poetry

Immerse yourself in the enchanting realm of Russian poetry, a literary treasure that has captivated hearts and minds for centuries. "The Cambridge to Russian...