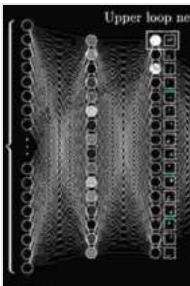


Work With Keras MNIST Data Sets And Advanced Neural Networks

The MNIST data set is a large collection of handwritten digits that is often used to train and test machine learning models. The data set consists of 70,000 grayscale images of handwritten digits, with 60,000 images in the training set and 10,000 images in the test set.

To load the MNIST data set into Keras, we can use the following code:

```
python from keras.datasets import mnist
```



Beginning Deep Learning with TensorFlow: Work with Keras, MNIST Data Sets, and Advanced Neural Networks

by Hans Fallada

★★★★☆ 4.3 out of 5

Language : English
File size : 49198 KB
Text-to-Speech : Enabled
Enhanced typesetting : Enabled
Print length : 785 pages
Screen Reader : Supported



```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

The `x_train` and `y_train` variables will contain the training data, while the `x_test` and `y_test` variables will contain the test data.

Once we have loaded the MNIST data set into Keras, we can create a simple neural network model to classify the digits in the data set. The following code creates a simple neural network model with two hidden layers:

```
python from keras.models import Sequential from keras.layers import  
Dense, Dropout
```

```
model = Sequential() model.add(Dense(512, activation='relu',  
input_shape=(784,))) model.add(Dropout(0.2)) model.add(Dense(512,  
activation='relu')) model.add(Dropout(0.2)) model.add(Dense(10,  
activation='softmax'))
```

The first layer in the model is a dense layer with 512 units and a relu activation function. The second layer is a dropout layer with a dropout rate of 0.2. The third layer is a dense layer with 512 units and a relu activation function. The fourth layer is a dropout layer with a dropout rate of 0.2. The fifth layer is a dense layer with 10 units and a softmax activation function.

Once we have created our neural network model, we can train it on the MNIST data set. The following code trains the model for 10 epochs with a batch size of 128:

```
python model.compile(loss='sparse_categorical_crossentropy',  
optimizer='adam', metrics=['accuracy'])
```

```
model.fit(x_train, y_train, epochs=10, batch_size=128)
```

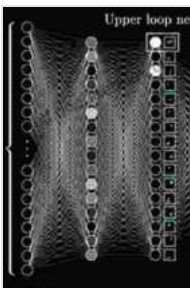
The `compile()` method compiles the model by specifying the loss function, the optimizer, and the metrics to be evaluated during training. The

`fit()` method trains the model on the training data.

Once we have trained our model, we can evaluate its performance on the test data. The following code evaluates the model on the test data and prints the accuracy:

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

The `evaluate()` method evaluates the model on the test data and ret



Beginning Deep Learning with TensorFlow: Work with Keras, MNIST Data Sets, and Advanced Neural Networks

by Hans Fallada

★★★★☆ 4.3 out of 5

Language : English
File size : 49198 KB
Text-to-Speech : Enabled
Enhanced typesetting : Enabled
Print length : 785 pages
Screen Reader : Supported

FREE

DOWNLOAD E-BOOK





Later Political Writings: A Window into the Evolution of Political Thought

Political thought, like the ever-changing tapestry of human history, has undergone a continuous process of evolution, with each era contributing its...



The Essential Guide to Family School Partnerships: Building a Strong Foundation for Student Success

: The Importance of Family School Partnerships Family school partnerships are essential for student success. When schools and families work...