

```

X_test_pd = pd.DataFrame(X_test)
X_test_pd.columns = columns_names

# train model
estimator.train(input_fn=lambda:input_fn(dict(X_train_pd), y_train),
steps=2000)

# evaluate model
metrics = estimator.evaluate(input_fn=lambda:input_fn(dict(X_test_pd),
y_test, training=False))

# print model metrics
metrics

```

Neural Networks with Keras

In this section, we will use the Sequential and Functional Keras API to build a simple neural network model. A Sequential API is the most commonly used method to build deep neural network models by stacking one layer on another. The Functional API offers more flexibility to build more complex neural network architectures. Both API methods are relatively easy to construct in Keras as we will see in the examples.

Subclassing a model as we did in the preceding examples provides even more flexibility for building and inspecting complex models. However, the code is more verbose and may be prone to errors. This technique should be used when it makes the most sense to, depending on the problem use case. We used them previously to serve as an illustration.

The following examples will use the Iris Dataset to build a neural network with one hidden layer as illustrated in Figure 30-11.

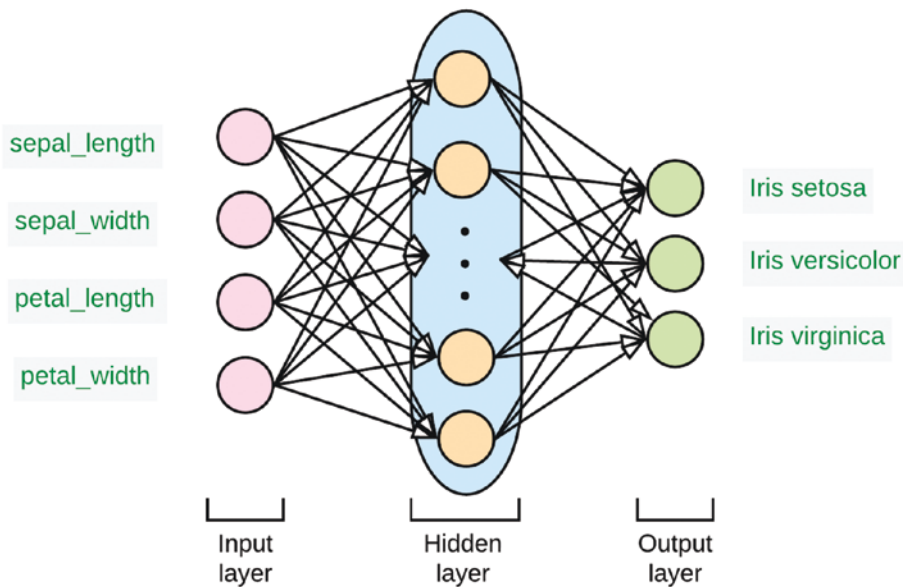


Figure 30-11. *Iris dataset – neural network architecture*

Using the Keras Sequential API

This code segment will construct a neural network model with the Sequential API using the method `'tf.keras.Sequential()'` to stack layers on each other. The model creates a hidden layer with 32 neurons and an output layer with 3 output units because the Iris target contains 3 classes.

```
!pip install -q tensorflow==2.0.0-beta0

# import packages
import tensorflow as tf
import pandas as pd
from sklearn.preprocessing import OneHotEncoder

# dataset url
train_data_url = "https://storage.googleapis.com/download.tensorflow.org/
data/iris_training.csv"
test_data_url = "https://storage.googleapis.com/download.tensorflow.org/
data/iris_test.csv"
```