Thomas Dinh

DSE6211

Week 1

Sept 3, 2023

Exercises

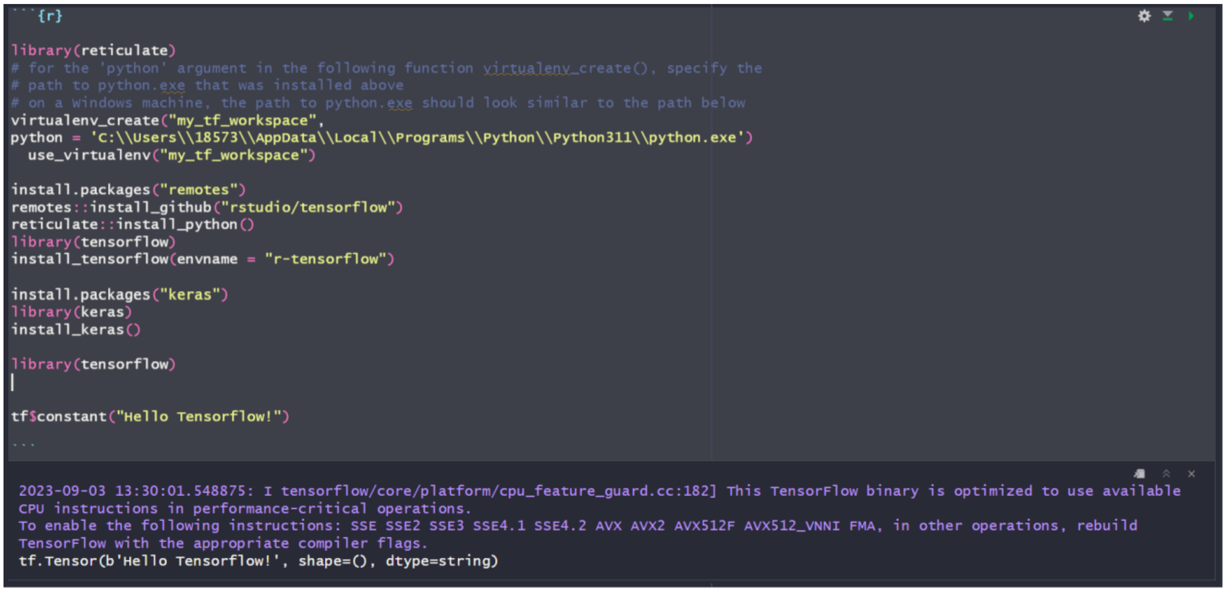
1) What is the main difference between supervised and unsupervised learning? Was the type of learning performed in the 'First Keras Example' section supervised or unsupervised learning? Why?

The three major differences between supervised and unsupervised learning are as follows:

- In supervised learning, input data is provided to the model along with the corresponding output, whereas in unsupervised learning, only input data is provided.
- Supervised learning algorithms are trained using labeled data, while unsupervised learning algorithms are trained using unlabeled data.
- In supervised learning, the model predicts the output, whereas unsupervised learning is used to discover hidden patterns within the data.

In the first Keras example, we observe supervised learning because labels are present in the data, and we are predicting the mpg.

2) Take a screenshot of the output of 'tf$constant("Hello Tensorflow!")' as described above.



````
```{r}

library(reticulate)
# for the 'python' argument in the following function virtualenv_create(), specify the
# path to python.exe that was installed above
# on a Windows machine, the path to python.exe should look similar to the path below
virtualenv_create("my_tf_workspace",
python = 'C:\\Users\\18573\\AppData\\Local\\Programs\\Python\\Python311\\python.exe')
  use_virtualenv("my_tf_workspace")

install.packages("remotes")
remotes::install_github("rstudio/tensorflow")
reticulate::install_python()
library(tensorflow)
install_tensorflow(envname = "r-tensorflow")

install.packages("keras")
library(keras)
install_keras()

library(tensorflow)


tf$constant("Hello Tensorflow!")

```

2023-09-03 13:30:01.548875: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
tf.Tensor(b'Hello Tensorflow!', shape=(), dtype=string)
````

3) Briefly describe the fields of machine learning and deep learning, as well as the main difference(s) between both fields.

Machine learning is a subfield of artificial intelligence that centers on developing algorithms and statistical models. These enable computers to learn and make predictions or decisions without requiring explicit programming. It entails training algorithms on extensive datasets to discern patterns and relationships. Subsequently, these identified patterns are employed to make predictions or decisions regarding new data.

Conversely, deep learning, a subset of machine learning, employs neural networks with multiple layers. These layers are utilized to scrutinize intricate patterns and relationships within data. Deep learning draws inspiration from the structure and function of the human brain and has proven highly successful in various tasks, including computer vision, natural language processing, and speech recognition.

Delving into some key distinctions between the two approaches. In machine learning, data is typically structured, resembling neatly organized rows and columns of information. In contrast, deep learning employs neural networks. Furthermore, machine learning applications tend to be less complex compared to deep learning and can be executed on standard computers. Deep learning systems, on the other hand, demand significantly more potent hardware and resources. Therefore models undergo training using substantial amounts of data and algorithms that continually learn and enhance their accuracy as they process more data.

4) What type of transformation is performed by each layer in a neural network? What is the purpose of the loss function in training a neural network?

The Neural Network is constructed from 3 types of layers:

1. Input layer — Initial data for the neural network. No transformation is done in this layer.
2. Hidden layers — Intermediate layer between input and output layer and place where all the computation is done. Each neuron in the hidden layer performs a linear transformation followed by a non-linear activation function.
    a. The linear transformation involves multiplying the input values from the previous layer by associated weights, summing these products, and adding a bias term. After an activation function (i.e. the Rectified Linear Unit (ReLU) or Sigmoid function) is applied element-wise to introduce non-linearity into the network.
3. Output layer — Produces the results for given inputs.

The output layer also performs a linear transformation followed by an activation function. However, the choice of activation function and the number of neurons in the output layer depend on the specific task the neural network is designed for. There are three types of tasks for neural networks:

1. Regression task
    a. A common choice is to have a single neuron in the output layer with no activation function
2. Binary classification task
    a. A sigmoid activation function is often used in a single-neuron output layer
3. Multi-class classification task
    a. Typically employs a softmax activation function, and the number of neurons in the output layer matches the number of classes.

A loss function is a function that compares the target and predicted output values; measures how well the neural network models the training data. When training, we aim to minimize this loss between the predicted and target outputs.

The hyperparameters are adjusted to minimize the average loss — we find the weights, and biases, that minimize the value of J (average loss).