DSE6211 Module 01 Lab 01

Joseph Annand

2024-01-21

1.) Supervised learning is an approach to machine learning where labeled data is used to train a model that predicts an outcome. Regression and classification problems can be solved using supervised learning. The “First Keras Example” in the lab is an example of supervised learning because the neural network is fitted with a labeled training data set to create a linear regression model. In this exercise, the input data, the mtcars training dataset, is linked to a desired output, the linear regression model to predict mpg. Unsupervised learning does not use a labeled dataset and the input data is not linked to a desired output. In the unsupervised learning approach, the machine learning algorithm attempts to determine patterns and relationships in unsorted, unlabeled datasets.

2.)

A screenshot of a computer screen

Description automatically generated

3.) Machine learning is a branch of artificial intelligence in which a machine is trained using input data and its corresponding answers, or correct outputs, and creates a set of rules to automate a task. Machine learning models take input data and expected outputs, find appropriate representations of the data for the task, and use some method of how well the algorithm it created works at accomplishing the task. Deep learning is a subset of machine learning that follows the same basic steps in solving problems, but it uses neural networks that may contain tens to hundreds of layers of representations. Contrastingly, other machine learning models learn only one or two layers of representations.

4.) Each layer of a neural network performs a nonlinear transformation on linear combinations of the inputs from the previous layer. The purpose of the loss function is to measure how well the network is performing by determining the difference between the predictions by the network and the true values.

## Create virtual environment and use Tensorflow

reticulate::virtualenv\_create("my\_tf\_workspace",  
 python = 'C:\\Users\\janna\\AppData\\Local\\Programs\\Python\\Python311\\python.exe')

## virtualenv: my\_tf\_workspace

reticulate::virtualenv\_install(envname = "my\_tf\_workspace", packages = "tensorflow")

## Using virtual environment "my\_tf\_workspace" ...

## + "C:/Users/janna/OneDrive/Documents/.virtualenvs/my\_tf\_workspace/Scripts/python.exe" -m pip install --upgrade --no-user tensorflow

reticulate::use\_virtualenv(virtualenv = "my\_tf\_workspace", required = TRUE)  
  
library(tensorflow)

## Warning: package 'tensorflow' was built under R version 4.3.2

tf$constant("Hello Tensorflow!")

## tf.Tensor(b'Hello Tensorflow!', shape=(), dtype=string)

## First Keras Example

library(keras)

## Warning: package 'keras' was built under R version 4.3.2

mtcars <- mtcars  
  
# Create an array with values for cylinders, displacement, and horsepower from  
# mtcars dataset  
mtcars\_x <- mtcars[, c("cyl", "disp", "hp")]  
mtcars\_x <- array(data = unlist(mtcars\_x),  
 dim = c(32, 3),  
 dimnames = list(rownames(mtcars\_x),  
 colnames(mtcars\_x)))  
# Create a vector with miles per gallon values from mtcars dataset  
mtcars\_y <- mtcars[, "mpg"]  
  
# Specify architecture of neural network model to predict mpg  
nn\_model <- keras\_model\_sequential() %>%  
 layer\_dense(units = 1, input\_shape = 3, activation = "linear")  
  
# Specify optimization algorithm as Adam algorithm, learning rate,  
# and loss function as MSE   
nn\_model %>% compile(optimizer = optimizer\_adam(learning\_rate = 0.2),  
 loss = "mean\_squared\_error")  
  
# Fit the neural network using the   
nn\_model\_training <- nn\_model %>% fit(x = mtcars\_x,  
 y = mtcars\_y,  
 epoch = 10000,  
 verbose = FALSE)  
  
get\_weights(nn\_model)

## [[1]]  
## [,1]  
## [1,] -1.22745621  
## [2,] -0.01887140  
## [3,] -0.01471268  
##   
## [[2]]  
## [1] 34.1849

lr\_fit <- lm(mpg ~ cyl + disp + hp, data = mtcars)  
lr\_fit$coefficients

## (Intercept) cyl disp hp   
## 34.18491917 -1.22741994 -0.01883809 -0.01467933