Installation of Keras for with Tensorflow GPU Support

See also Howto Install Tensorflow-GPU with Keras in R - A manual that worked on 2021.02.20 (and likely will work in future) (letyourmoneygrow.com) . Its information is more recent (February 20, 2021) than our installation outline below (September 29,2019)

[1] Check whether the specific NVIDIA graphics card of the computer supports a CUDA version > 3.5 at:

https://developer.nvidia.com/cuda-gpus

[2] Follow the steps 1 and 2 in https://www.pugetsystems.com/labs/hpc/How-to-Install-TensorFlow-with-GPU-Support-on-Windows-10-Without-Installing-CUDA-UPDATED-1419/ .

[3] For steps 3 and 4 use the virtual environment **`r-reticulate`** rather than the proposed environment **`tf-gpu`**. Note: This is the key trick of getting Keras running smoothly in R.

[4] Check the installation by issuing within the **`r-reticulate`** environment the python commands in step 5 .

[4] Make sure that the **`rtools`** (see https://cran.r-project.org/bin/windows/Rtools/) are installed and then open RStudio.

[5] Run the R installation commands:

```
install.packages("remotes")
remotes::install_github("rstudio/keras")
keras::install_keras()
```

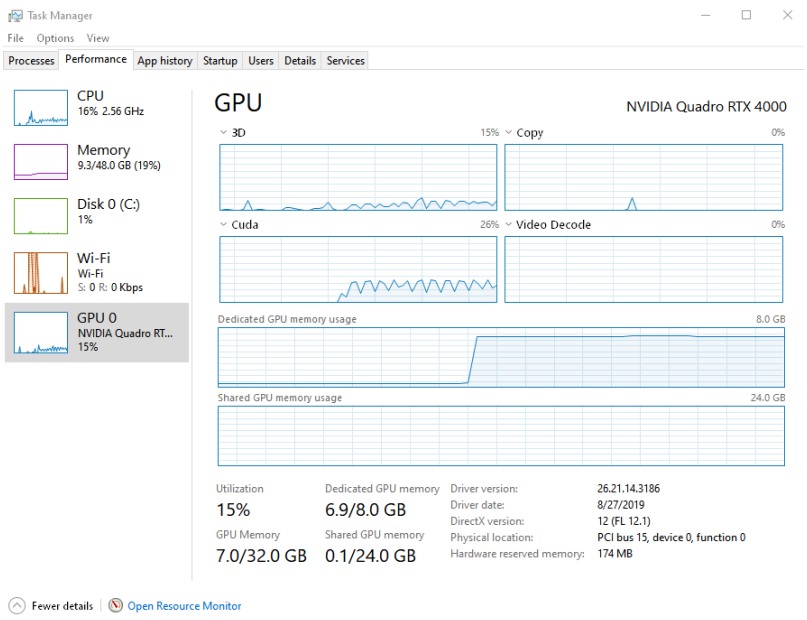
Note: Don't use `keras::install_keras(tensorflow="gpu")` because the **`r-reticulate`** environment is already setup for GPU support.

[6] Open Window's **TASK MANAGER** ▶ **PERFORMANCE** ▶ **GPU** to monitor the GPU usage

[7] Run the test script for the mnist dataset, which is documented in Chollet and Allaire, 2018. *Deep Learning with R.* Manning. (see listings 2.1 to 2.5):

```
library(keras)
mnist <- dataset_mnist()
x_train <- mnist$train$x
y_train <- mnist$train$y
x_test <- mnist$test$x
y_test <- mnist$test$y

## reshape
x_train <- array_reshape(x_train, c(nrow(x_train), 784))
x_test <- array_reshape(x_test, c(nrow(x_test), 784))

## rescale
x_train <- x_train / 255
x_test <- x_test / 255

## ------------------------------------------------------------------------
```

```
y_train <- to_categorical(y_train, 10)
y_test <- to_categorical(y_test, 10)

## ------------------------------------------------------------------------
model <- keras_model_sequential()
model %>%
  layer_dense(units = 256, activation = 'relu', input_shape = c(784)) %>%
  layer_dropout(rate = 0.4) %>%
  layer_dense(units = 128, activation = 'relu') %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 10, activation = 'softmax')

model <- keras_model_sequential() %>%
  layer_dense(units=32, input_shape = c(784)) %>%
  layer_dense(units=10, activation = "softmax")

## ------------------------------------------------------------------------
summary(model)

## ------------------------------------------------------------------------
model %>% compile(
  loss = 'categorical_crossentropy',
  optimizer = optimizer_rmsprop(),
  metrics = c('accuracy')
)

## ---- results='hide'-----------------------------------------------------
history <- model %>% fit(
  x_train, y_train,
  epochs = 30, batch_size = 128,
  validation_split = 0.2
)

## ------------------------------------------------------------------------
plot(history)

## ---- results = 'hide'---------------------------------------------------
model %>% evaluate(x_test, y_test)

## ---- results = 'hide'---------------------------------------------------
model %>% predict_classes(x_test)
```


Task Manager
File Options View
Processes Performance App history Startup Users Details Services
CPU
16% 2.56 GHz
Memory
9.3/48.0 GB (19%)
Disk 0 (C:)
1%
Wi-Fi
Wi-Fi
S: 0 R: 0 Kbps
GPU 0
NVIDIA Quadro RT...
15%
GPU
NVIDIA Quadro RTX 4000
3D
15%
Copy
0%
Cuda
26%
Video Decode
0%
Dedicated GPU memory usage
8.0 GB
Shared GPU memory usage
24.0 GB
Utilization
15%
Dedicated GPU memory
6.9/8.0 GB
GPU Memory
7.0/32.0 GB
Shared GPU memory
0.1/24.0 GB
Driver version:
26.21.14.3186
Driver date:
8/27/2019
DirectX version:
12 (FL 12.1)
Physical location:
PCI bus 15, device 0, function 0
Hardware reserved memory:
174 MB
Fewer details
Open Resource Monitor