# Installation of Keras for with Tensorflow GPU Support

[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 R.

[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)

