

Notes for install Keras on Anaconda3:

For some machines, you might find difficulties by following the official document of installing Keras for RStudio. There is the alternative to install Keras and Tensorflow for RStudio which works for most machines.

However, if you have a PC which has a CPU purchased before 2012, the current versions(1.10-2.1) of Tensorflow could not be loaded as AVX instructions set is needed from Tensorflow 1.6 and later.

Please ensure your machine was purchased in 2012 or later.

Step 1

Follow the document or video to install Anaconda3 and RStudio:

User guide: [Click here](#)

Video:

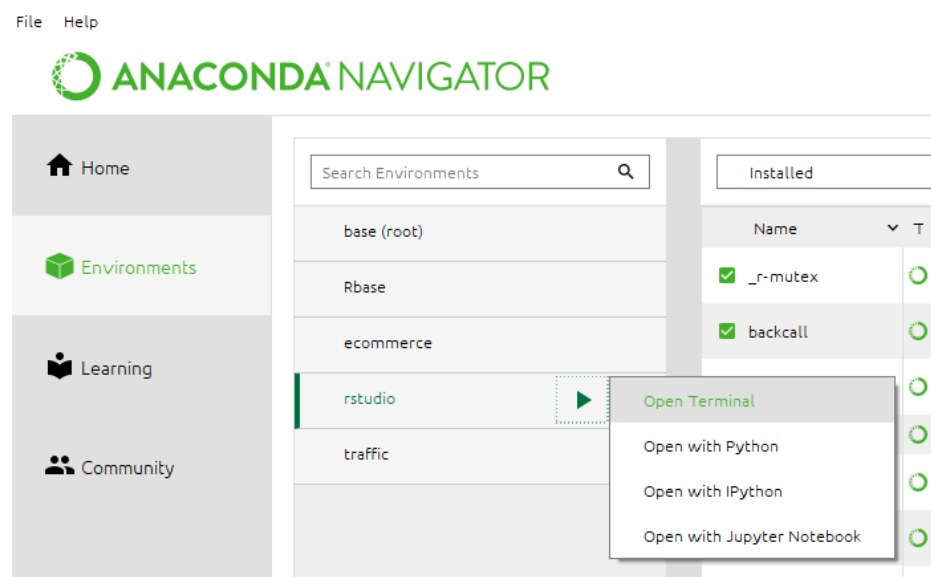
- A) Staff (<https://web.microsoftstream.com/video/6c1dea0f-a672-4f67-8983-ea5219200cee>)
- B) Students (<https://web.microsoftstream.com/video/33db8bdd-dd29-4c75-8cf9-4eaa0ac4dc08>)

Step 2

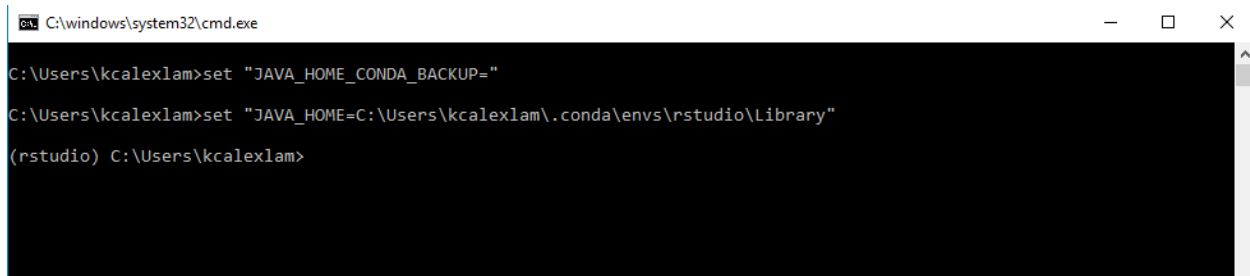
When finished, run these lines at Anaconda3 prompt.

For example, if your environment with RStudio installed called “rstudio”.

- A) Click “Open Terminal”:



B) The terminal as shown, input the following commands one by one using this prompt:



```
C:\windows\system32\cmd.exe
C:\Users\kcalexlam>set "JAVA_HOME_CONDA_BACKUP="
C:\Users\kcalexlam>set "JAVA_HOME=C:\Users\kcalexlam\.conda\envs\rstudio\Library"
(rstudio) C:\Users\kcalexlam>
```

- 1 conda install python=3.6
- 2 conda install -c conda-forge tensorflow
- 3 conda install -c r r-tensorflow
- 4 conda install -c conda-forge r-keras

C) Open the RStudio and run the following deep learning model to check the installation works:

```
library(keras)
mnist <- dataset_mnist()
train_images <- mnist$train$x
train_labels <- mnist$train$y
test_images <- mnist$test$x
test_labels <- mnist$test$y

#data structure checking
str(train_images)
str(train_labels)

str(test_images)
str(test_labels)

# reshape and rescale
train_images <- array_reshape(train_images, c(nrow(train_images), 28*28))
test_images <- array_reshape(test_images, c(nrow(test_images), 28*28))
train_images <- train_images / 255
test_images <- test_images / 255
train_labels <- to_categorical(train_labels, 10)
test_labels <- to_categorical(test_labels, 10)

# defining the model and layers
model <- keras_model_sequential()
```

```

model %>%
  layer_dense(units = 256, activation = 'relu', #Fully connected units
              input_shape = c(28*28)) %>%
  layer_dense(units = 128, activation = 'relu') %>%
  layer_dense(units = 10, activation = 'softmax')
summary(model)
# compile (define loss and optimizer)

model %>% compile(
  loss = 'categorical_crossentropy',
  optimizer = optimizer_rmsprop(),
  metrics = c('accuracy')
)
# train (fit)
model %>% fit(
  train_images, train_labels,
  epochs = 30, batch_size = 128,
)

```

D) The expected results are as follows:

