Convolutional Neural Networks in R

Mengli Xiao

Division of Biostatistics
University of Minnesota

February 21, 2018

Outline

Why we need Python?

- Most deep learning algorithms are written in Python
- Based on some intermediate packages, R can call Python to implement the deep neural networks written in Python

Create Python virtual environment

- Download Python 2.7 at *https://www.python.org/downloads/*
- Download Anaconda Python 2.7 version
  https://www.anaconda.com/download/?lang=en-us
- The python 3.6 can also be used

Packages in R

- ▶ Package "keras": Keras is a high-level neural network API written in Python (*https://keras.rstudio.com/*)

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

  - ▶ The commands automatically install tensorflow with keras

- ▶ **Optional:** Package "tensorflow" – Tensorflow is a low-level API and Keras is a wrapper to it (*https://tensorflow.rstudio.com/*)

```
install.packages("tensorflow")
```

  - ▶ Faster than Keras
  - ▶ Harder to use at the expense of having more control

- ▶ It's **necessary** to create Python virtual environment to ensure you to load the R packages successfully
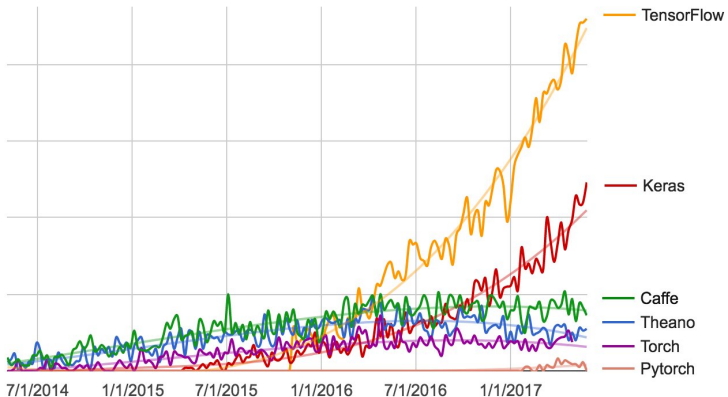
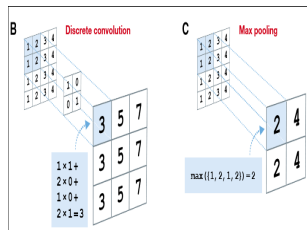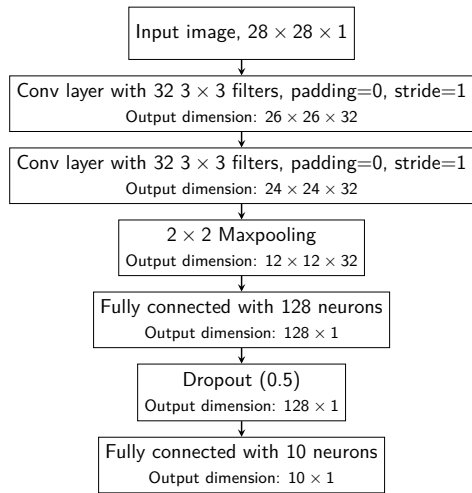**Deep learning framework search interest**

Figure: Popularity of deep learning framework by year

Convolutional neural network

A very simple CNN structure

Input image, $28 \times 28 \times 1$

Conv layer with 32 3×3 filters, padding=0, stride=1
Output dimension: $26 \times 26 \times 32$

Conv layer with 32 3×3 filters, padding=0, stride=1
Output dimension: $24 \times 24 \times 32$

2×2 Maxpooling
Output dimension: $12 \times 12 \times 32$

Fully connected with 128 neurons
Output dimension: 128×1

Dropout (0.5)
Output dimension: 128×1

Fully connected with 10 neurons
Output dimension: 10×1

Data: MNIST

Steps in the implementation

1. Load the data from the Keras pacakage
2. Construct the model structure
3. Compile the model
4. Evaluate the model

Reference I

- https://keras.rstudio.com/
- https://tensorflow.rstudio.com/