Deep Learning with R

Francesca Lazzeri - @frlazzeri
Data Scientist II - Microsoft, AI Research
Agenda

- Deep Learning with R
- What is Deep Learning
- Demo

Better understanding of R DL tools
What is Deep Learning

Fundamental concepts in Deep Learning

- Forward Propagation Algorithm
- Activation Functions
- Gradient Descent
- Backpropagation
What is Deep Learning

Fundamental concepts in Deep Learning

- Forward Propagation Algorithm
- Activation Functions
- Gradient Descent
- Backpropagation
What is Deep Learning

Example as seen by linear regression

- Age
- Bank Balance
- Retirement Status
- ...

Number of Transactions
What is Deep Learning

Interactions

- Neural networks account for interactions really well

- Deep learning uses especially powerful neural networks for:
  - Text
  - Images
  - Videos
  - Audio
  - Source code
What is Deep Learning

Deep learning models capture interactions

Age
Bank Balance
Retirement Status
...

Number of Transactions
Interactions in neural networks

Input Layer:
- Age
- Bank Balance
- Retirement Status
- # Accounts

Hidden Layer

Output Layer:
- Number of Transactions
What is Deep Learning

Forward Propagation Algorithm

# Children
- 2

# Accounts
- 3

Hidden Layers:
- 1
- 5

Output Layer:
- 9
- 2

# Transactions
What is Deep Learning

Fundamental concepts in Deep Learning

- Forward Propagation Algorithm
- Activation Functions
- Gradient Descent
- Backpropagation
What is Deep Learning

Activation Functions

# Children
- 2

# Accounts
- 3

Input

Hidden
- \( \text{tanh} \ (2+3) \)
- \( \text{tanh} \ (-2+3) \)

Output
- 9

# Transactions
What is Deep Learning

ReLU Activation Function

Input 3

Hidden 26

Hidden 0

Hidden 0

Hidden 52

Output 364
Deep networks internally build representations of patterns in the data

- Partially replace the need for feature engineering
- Subsequent layers build increasingly sophisticated representations of raw data
- Modeler doesn’t need to specify the interactions
- When you train the model, the neural network gets weights that find the relevant patterns to make better predictions
What is Deep Learning

Fundamental concepts in Deep Learning

- Gradient Descent
- Forward Propagation Algorithm
- Activation Functions
- Backpropagation
What is Deep Learning

The Need for Optimization

- Predictions with multiple points
  - Making accurate predictions gets harder with more points
  - At any set of weights, there are many values of the error
  - Correspond to the many points we make predictions for

- Loss function
  - Aggregate errors in predictions from many data points into single number
  - Measure of model’s predictive performance
What is Deep Learning

The Need for Optimization

- Squared error loss function

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Actual</th>
<th>Error</th>
<th>Squared Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>20</td>
<td>-10</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>5</td>
<td>25</td>
</tr>
</tbody>
</table>

- Total Squared Error: 150
- Mean Squared Error: 50
- Lower loss function value means a better model
- Goal: find the weights that give the lowest value for the loss function
- Gradient descent!
What is Deep Learning

Gradient Descent

Loss(w)

w
Slope calculation example

- To calculate the slope for a weight, need to multiply:
  * Slope of the loss function w.r.t value at the node we feed into
  * The value of the node that feeds into our weight
  * Slope of activation function w.r.t value we feed into
What is Deep Learning

Fundamental concepts in Deep Learning

- Forward Propagation Algorithm
- Activation Functions
- Gradient Descent
- Backpropagation
What is Deep Learning

Backpropagation

Input: 3, 5
Hidden: 26, 0, 0, 52
Output: 364
What is Deep Learning

Backpropagation

- Allows gradient descent to update all weights in neural network (by getting gradients for all weights)

- Go back one layer at a time

- Important to understand the process, but you will generally use a library that implements this
## Deep Learning with R

<table>
<thead>
<tr>
<th>Model/Dataset</th>
<th>MNIST</th>
<th>Iris</th>
<th>Forest Cover Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>Runtime (sec)</td>
<td>Accuracy (%)</td>
</tr>
<tr>
<td>MXNetR (CPU)</td>
<td>98.33</td>
<td>147.78</td>
<td>83.04</td>
</tr>
<tr>
<td>MXNetR (GPU)</td>
<td>98.27</td>
<td>336.94</td>
<td>84.77</td>
</tr>
<tr>
<td>darch 100</td>
<td>92.09</td>
<td>1368.31</td>
<td>69.12</td>
</tr>
<tr>
<td>darch 500/300</td>
<td>95.88</td>
<td>4706.23</td>
<td>54.78</td>
</tr>
<tr>
<td>deepnet DBN</td>
<td>97.85</td>
<td>6775.4</td>
<td>30.43</td>
</tr>
<tr>
<td>deepnet DNN</td>
<td>97.05</td>
<td>2183.92</td>
<td>78.26</td>
</tr>
<tr>
<td>H2O</td>
<td>98.08</td>
<td>543.14</td>
<td>89.56</td>
</tr>
<tr>
<td>Random Forest</td>
<td>96.77</td>
<td>125.28</td>
<td>91.3</td>
</tr>
</tbody>
</table>
Overview

Keras is a high-level neural networks API developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research. Keras has the following key features:

- Allows the same code to run on CPU or on GPU, seamlessly.
- User-friendly API which makes it easy to quickly prototype deep learning models.
- Built-in support for convolutional networks (for computer vision), recurrent networks (for sequence processing), and any combination of both.
- Supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, etc. This means that Keras is appropriate for building essentially any deep learning model, from a memory network to a neural Turing machine.
- Is capable of running on top of multiple back-ends including TensorFlow, CNTK, or Theano.

This website provides documentation for the R interface to Keras. See the main Keras website at https://keras.io for additional information on the project.
Demo

Deep Learning with R on Azure with Keras and CNTK

DSVM

CNTK

R & Keras

Azure
Keras Workflow Steps to Build your Model

- Specify architecture
- Compile the model
- Fit the model
- Predict
library(keras)
install_keras()

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)
Demo

Defining the Model

```r
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')
```
### Defining the Model

```python
summary(model)
```

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>dense_1 (Dense)</td>
<td>(None, 256)</td>
<td>200960</td>
</tr>
<tr>
<td>dropout_1 (Dropout)</td>
<td>(None, 256)</td>
<td>0</td>
</tr>
<tr>
<td>dense_2 (Dense)</td>
<td>(None, 128)</td>
<td>32896</td>
</tr>
<tr>
<td>dropout_2 (Dropout)</td>
<td>(None, 128)</td>
<td>0</td>
</tr>
<tr>
<td>dense_3 (Dense)</td>
<td>(None, 10)</td>
<td>1290</td>
</tr>
</tbody>
</table>

---

Total params: 235,146  
Trainable params: 235,146  
Non-trainable params: 0
Defining the Model

```r
model %>% compile(
    loss = 'categorical_crossentropy',
    optimizer = optimizer_rmsprop(),
    metrics = c('accuracy')
)
```
```r
history <- model %>% fit(
  x_train, y_train,
  epochs = 30, batch_size = 128,
  validation_split = 0.2
)

model %>% evaluate(x_test, y_test)

$loss
[1] 0.1149

$acc
[1] 0.9807

model %>% predict_classes(x_test)

[1] 7 2 1 0 4 1 4 9 5 9 0 6 9 0 1 5 9 7 3 4 9 6 6 5 4 0 7 4 0 1 3 1 3 4 7 2 7 1 2
[40] 1 1 7 4 2 3 5 1 2 4 4 6 3 5 5 6 0 4 1 9 5 7 8 9 3 7 4 6 4 3 0 7 0 2 9 1 7 3 2
[79] 9 7 6 2 7 8 4 7 3 6 1 3 6 9 3 1 4 1 7 6 9
[ reached getOption("max.print") -- omitted 9900 entries ]
```

http://www.rblog.uni-freiburg.de

https://keras.rstudio.com/

https://campus.datacamp.com/courses/deep-learning

http://gluon.mxnet.io
Thank You!

Francesca Lazzeri - @frlazzeri
Data Scientist II - Microsoft, AI Research