Deep Learning with R

Chenshu Liu

April 2022

Contents

Introduction from Simple Linear Regression

Why is deep learning important and useful?

```
# Scenario: predicting sales performance
sales <- c(3, 4, 2, 4, 5, 6, 3, 9, 1, 12)
```

In the simplest way possible, we can simple find the mean of all the observations, and use the mean as our prediction for future sales performances

$$Performance = \frac{\Sigma_{i=1}^{n} x_i}{n}$$

```
mean.sales <- mean(sales)
mean.sales
```

```
## [1] 4.9
```

However, using only the mean as future prediction is not close to accurate, we are bound to make errors. Thus, to measure the magnitude of the error we are making using the mean, we introduce the concept of *sum of squared errors (SSE)*:

$$SSE = \Sigma_{i=1}^{n} (x_i - \bar{X})^2$$

Where $\bar{X}$ is the mean of the observations.

NOTE: There the differences are squared because there can be positive and negative numbers in the differences, and summing positive and negative differences together will make 0. Thus, we need to **square** the differences to make them significant.

```
SSE = sum((sales - mean.sales)^2)
SSE
```

```
## [1] 100.9
```

However, the measurement of deviation from actual observations has some limitations:

1. In the calculation of SSE, we summed all the differences together, which means the sample size plays a role in the calculation of the deviation. Thus, in order to rule out the effect of sample size on SSE, we introduce the idea of **variance**: $var = \frac{\Sigma_{i=1}^{n} (x_i - \bar{X})^2}{n-1}$, where $n-1$ is the degrees of freedom. Now, the standard deviation is a square-root version of the measurement of deviation, which makes more sense.

2. Because the differences are squared, it is hard for us to interpret the deviation, thus, we further introduce the idea of *standard deviation*: $s = \sqrt{\frac{\Sigma_{i=1}^{n} (x_i - \bar{X})^2}{n-1}}$

Both the variance and standard deviation describes how bad the model's prediction is

```
variance <- (sum((sales - mean.sales)^2))/(length(sales) - 1)
variance
```

```
## [1] 11.21111
```

```r
# base R function
var(sales)
```

```
## [1] 11.21111
```

```r
standard.deviation <- sqrt((sum((sales - mean.sales)^2))/(length(sales) - 1))
standard.deviation
```

```
## [1] 3.3483
```

```r
# base R function
sd(sales)
```

```
## [1] 3.3483
```

So, predictions are bound to have errors:

$$y_i = \bar{X} + \epsilon_i$$
$$target = model + error$$

Take away: After learning about how to measure the accuracy of a prediction model, we now need to know how to optimize the model and letting it to generate more accurate predictions. The task of optimizing the model that minimizes prediction error is the in the scope of Deep Learning!