INF8245E: Machine Learning | Assignment #1

Louis Plessis (1933334)

4 October 2021

Acknowledgement: I have briefly discussed about this assignment with Joseph Battesti and Marie-Christine Paré (parts 3 and 4).

1. Sampling

1.1. Pseudocode to sample the student's routine

Activity	***Distribution***
Movies	$0 \leq x < 0.2$
INF8245E	$0.2 \leq x < 0.6$
Playing	$0.6 \leq x < 0.7$
Studying	$0.7 \leq x < 1$

```
x = random(0,1)    # Random number between 0 and 1
activity = ""

if ( (x >= 0) and (x < 0.2) )
      activity = "Movies"

if ( (x >= 0.2) and (x < 0.6) )
      activity = "INF8245E"

if ( (x >= 0.6) and (x < 0.7) )
      activity = "Playing"

if ( (x >= 0.7) and (x < 1) )
      activity = "Studying"
```

1.2. Sampling *(results can vary as x is a random number)*

- **For 100 days:**

Movies:	23/100 days
INF8245E:	33/100 days
Playing:	11/100 days
Studying:	33/100 days

- **For 1000 days:**

Movies:	208/1000 days
INF8245E:	390/1000 days
Playing:	98/1000 days
Studying:	304/1000 days

- **Comparison with multinomial distribution**

Differences between multinomial distribution and sample:

100 days	*1000 days*
0.030	0.008
-0.070	-0.010
0.010	-0.002
0.030	0.004

The fractions are relatively close to the multinomial distribution. However, sampling for 1000 days seem t o cause less disparity on average.

2. Model selection

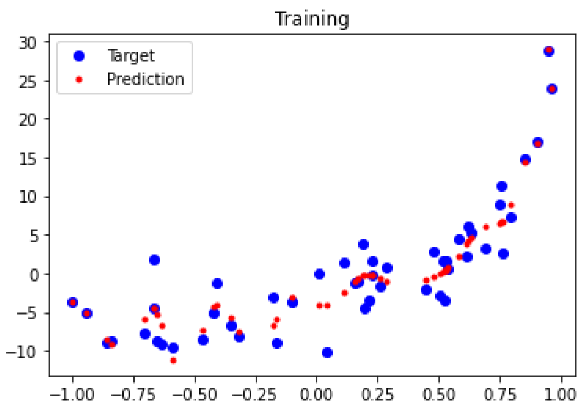
2.1. Fit a 20-degree polynomial to the data (no regularization)

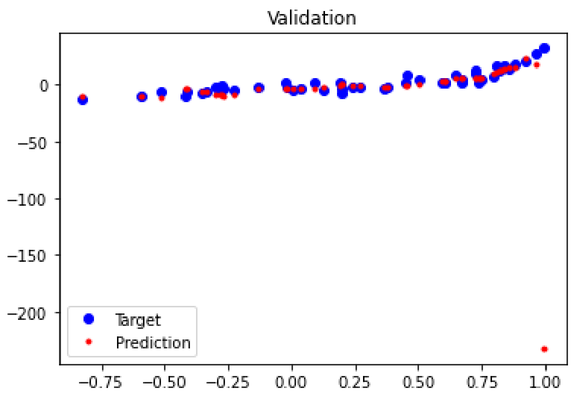
$$w^* = (X^T X)^{-1} X^T y$$
$$\hat{y} = w^T x$$

2.1.1. Training and validation RMSE

Training RMSE:	2.544590374303791
Validation RMSE:	37.6148458260915

2.1.2. Visualization



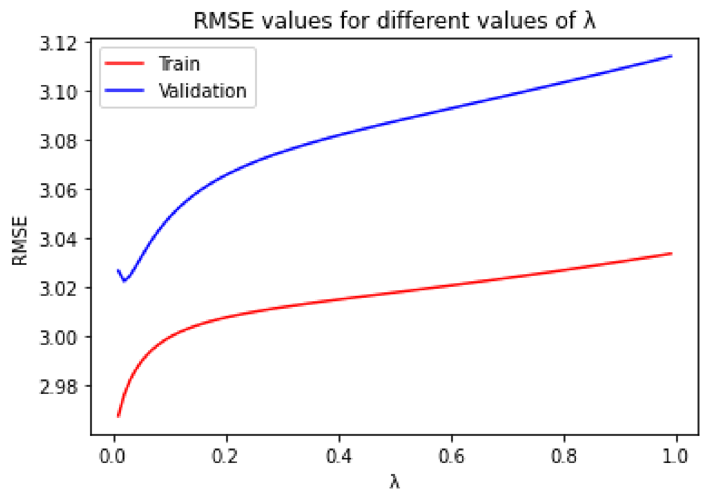


2.1.3. The test RMSE is 7.093, which is relatively high and means that the generalization performance is bad. Hence, the model is probably overfitting.

2.2. L2 regularization

$$w = (X^T X + \lambda I)^{-1} X^T y$$

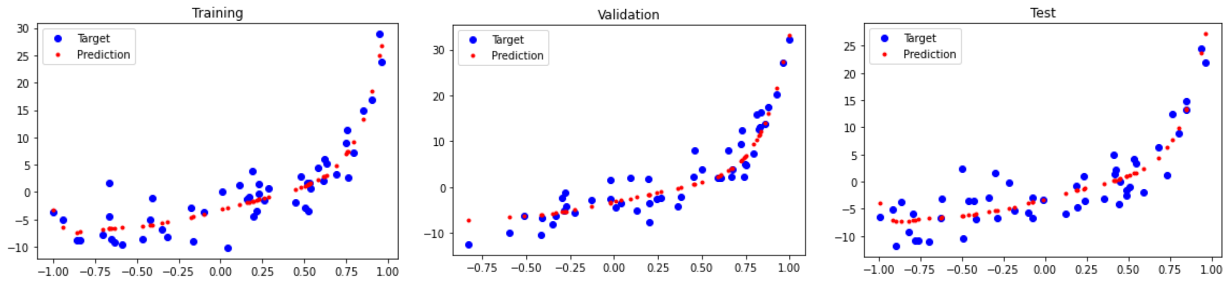
2.2.1. Training and Validation RMSE



2.2.2.

Best value of λ:	0.02
Train RMSE:	2.976181559923893
Valid RMSE:	3.0224325939041834
Test RMSE:	3.275701207516833

2.2.3. Visualization



2.2.4. The test RMSE is rather low compared to the model without regularization **(3.27 < 7.09)**. Also, the fitted curve does not oscillate like the one without regularization. Hence, there is no proof here that the model is overfitting or underfitting.

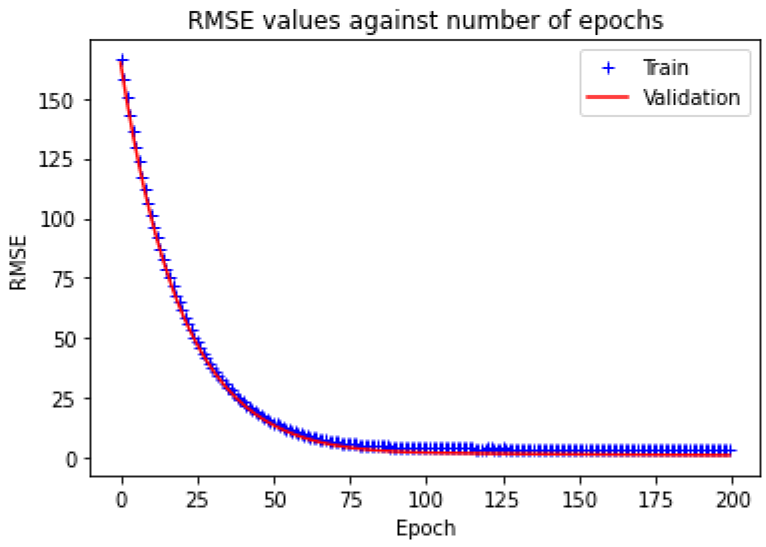
2.3. The model's curve seems to cross the x-axis 3 times, for both no-regularization and L2-regularization situations. Therefore, the degree of the source polynomial is probably around 4.

3. Gradient descent for regression

3.1. Stochastic gradient descent

For i in 1 to N:

$$w_0 = w_0 - \alpha(\hat{y(x^{(i)}; w)} - y^{(i)})$$
$$w_1 = w_1 - \alpha(\hat{y(x^{(i)}; w)} - y^{(i)})x^{(i)}$$



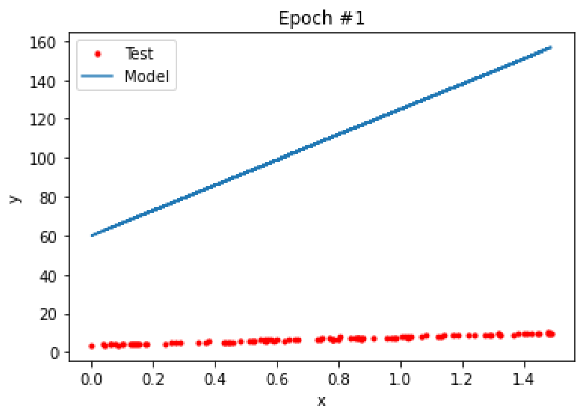
3.2. Steps *(results can vary as data points are shuffled at every iteration)*

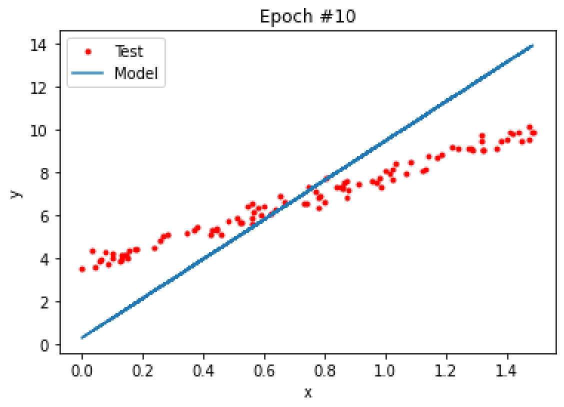
Step size (α)	***Validation RMSE***
10^{-1}	2.081
10^{-2}	0.576
10^{-3}	**0.466**
10^{-4}	1.103
10^{-5}	63.994
10^{-6}	156.831
10^{-7}	171.525
10^{-8}	173.067
10^{-9}	173.222
10^{-10}	173.238

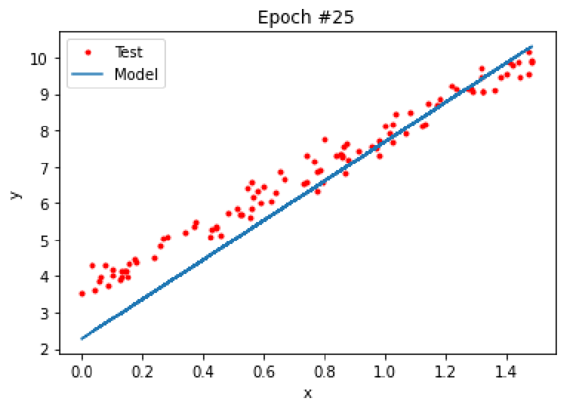
The lowest validation RMSE was **0.466** and found when **α = 0.001**.

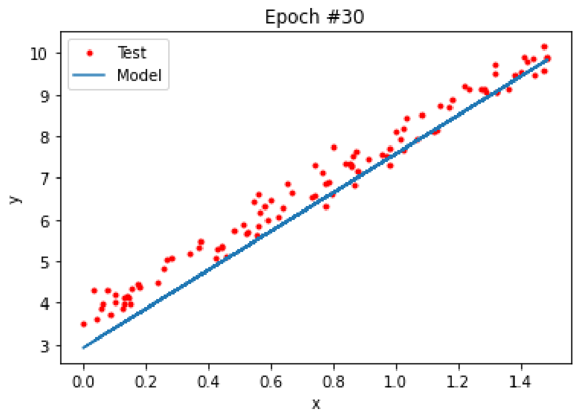
The test RMSE of this model is **0.449**

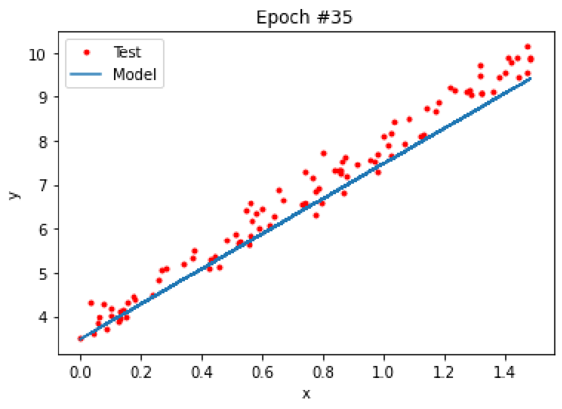
3.3. Visualizations











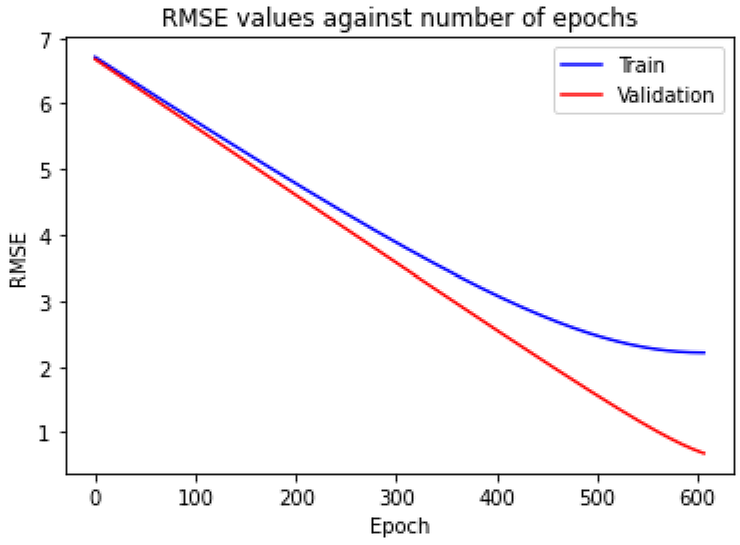
It seems like we are getting the best fit at around 30 epochs. However, after 35 epochs, the model gradually moves away from the test data.

3.4. Full-batch gradient descent

Repeat until convergence:

$$w_0 = w_0 - \alpha \frac{1}{N} \sum_{i=1}^{N} (\hat{y}(x^{(i)}; w) - y^{(i)})$$

$$w_1 = w_1 - \alpha \frac{1}{N} \sum_{i=1}^{N} (\hat{y}(x^{(i)}; w) - y^{(i)}) x^{(i)}$$



3.5. Full batch vs. Stochastic gradient descent

Full-batch gradient descent seems to work better as it stops automatically when the model converges. Stochastic gradient descent doesn't guarantee that the best fit will be found at the last epoch (it will probably be found before, as seen on graphs in section 3.3).

4. Real life dataset

Column nos.	*# Missing attributes*	*% of data missing*
1	1174	58.88%
2	1177	59.03%
30	1	0.05%
101-117	1675	84.00%
121-124	1675	84.00%
126	1675	84.00%

4.1. Completing the data set

4.1.1.Using sample mean of each column to fill in the missing attributes

Using sample mean of each column is not a good way to fill in missing attributes (especially if a large proportion of them are missing) as it creates false data that'd be interpreted as true data when using ML algorithms. In our case, we've got between 58% and 84% of the data missing for some columns. Additionally, we don't know if these numbers are quantities or categories (e.g., column 3 values are categories). Using the sample mean for missing categories would not make sense at all. Hence, in our case, this method is not the best.

4.1.2.Other ways to fill in the missing attributes

Other ways to deal with missing data would be to:

- Delete columns with too much data missing
- Delete rows with missing data if they represent only a small proportion of the column
- For columns with categories, consider missing attributes as part of an extra category "N/A".

4.1.3.Completing data set

Using the *"communities.names"* file, we can suppose that columns 1 (county code), 2 (community code) and 3 (community name) are categories. Let's replace "?" values of columns 1 and 2 by "0" values (new county and community "0").

Column no. 30 only has one missing attribute: let's delete the corresponding row.

Columns 101 to 117, 121 to 124 and 126 have more than 84% of their attributes missing: let's delete these columns.

This method is better because we are making sure that only "real" data is going to be used in the next sections. Even though we've deleted one row and a few columns, we are still left with plenty of data to work with. I also think it was important not to delete columns 1, 2 and 3 as their values are categories and thus very important for the next steps.

4.1.4.Completed data set

The completed data set can be found in "communities_data_completed.csv".

4.2. 5-fold cross-validation (80/20 split)

1	Test	Train			
2	Train	Test	Train		
3	Train		Test	Train	
4	Train			Test	Train
5	Train				Test

4.3. Ridge-regression

X