MACHINE LEARNING

Linear Regression

Tharun Kumar Bettadalli Girish

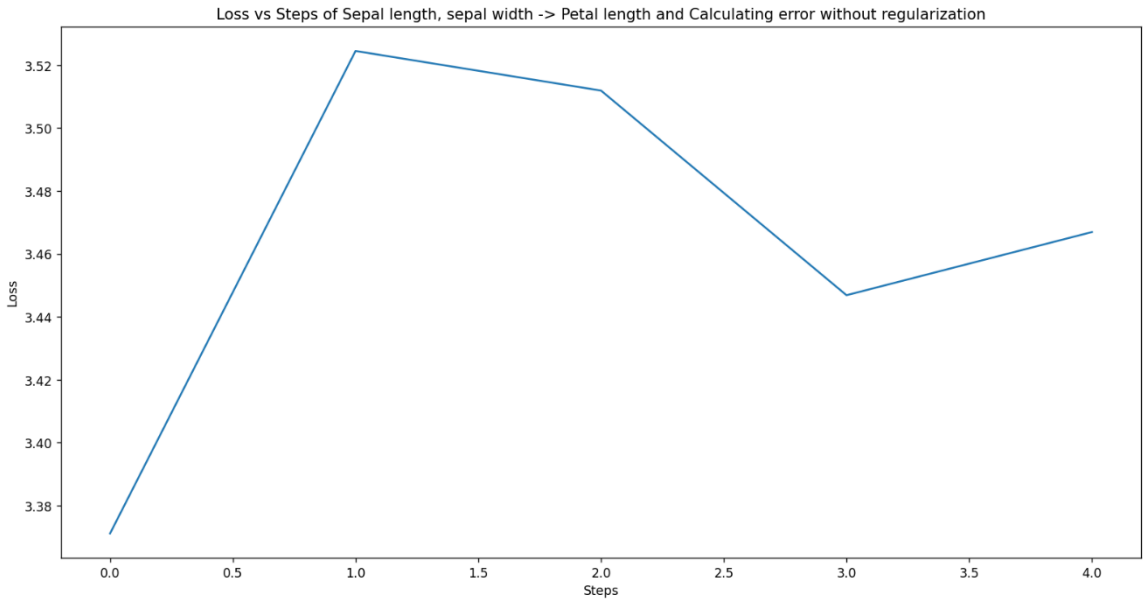
txb7785@mavs.uta.edu

Linear Regression

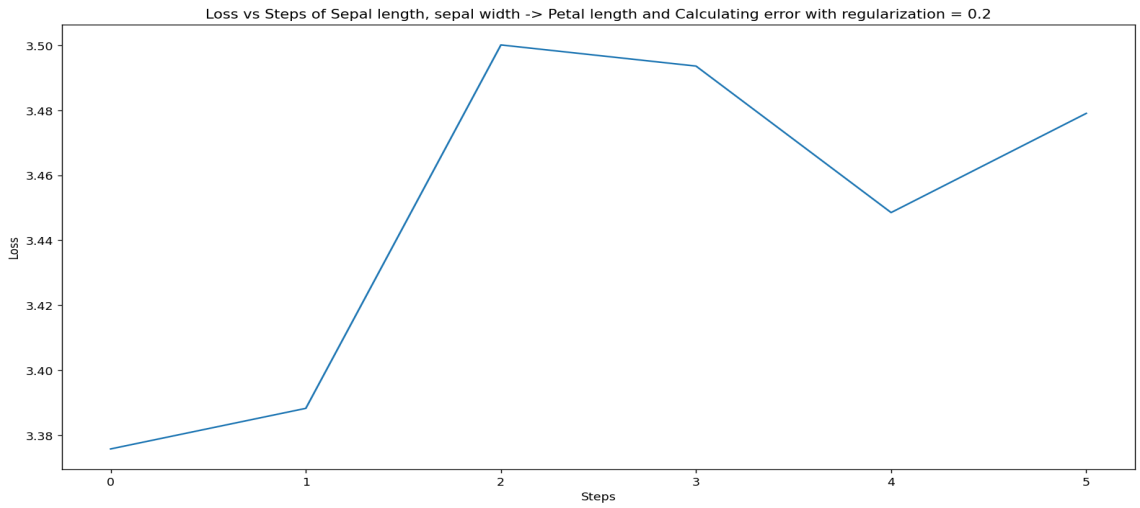
Linear Regression is a method of predicting real values given some input.

1.

Here is the output of train_regression1.py where the graph 1 tells loss versus steps of sepal length and sepal width, here calling the fit function we have used regularization as 0. Hence, we got the below graph



Here is the output of train_regression1.py where the graph 2 tells loss versus steps of sepal length and sepal width, here calling the fit function we have used regularization as 0.2. Hence, we got the below graph



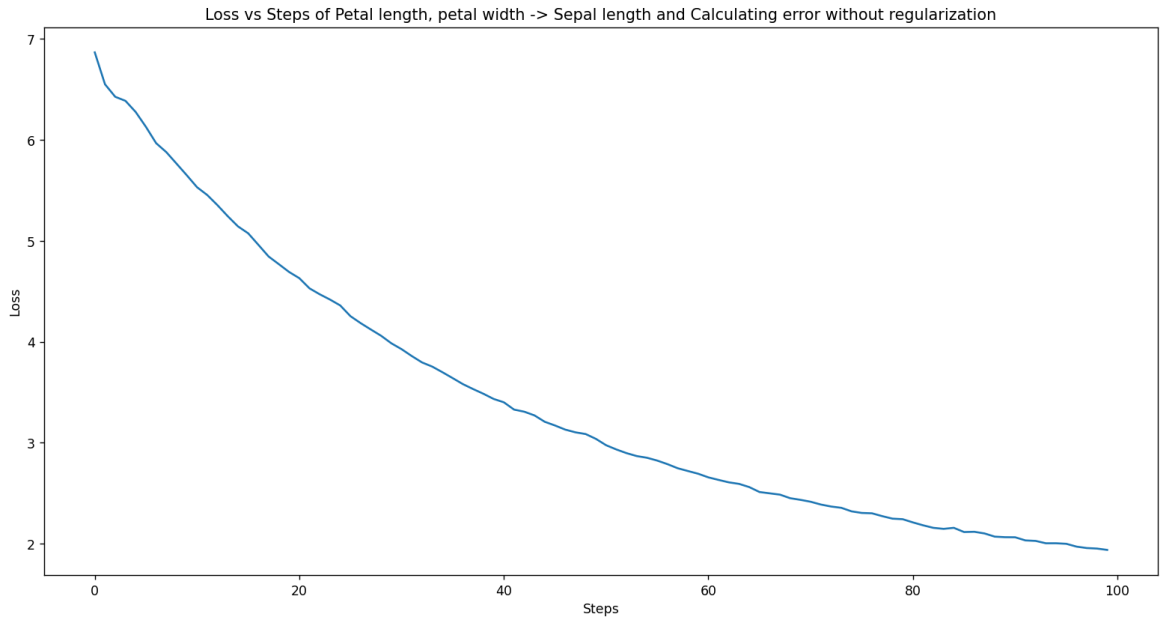

```
PS C:\Users\USER\OneDrive\Desktop\UTA\2nd SEM\Tharunproject> & C:/Users/USER/anaconda3/python.exe "c:/Users/
USER/OneDrive/Desktop/UTA/2nd SEM/Tharunproject/train_regression1.py"
Early stopping at epoch is : 8

Early stopping at epoch is : 4
```

Here is the output of eval_regression1.py where we got the below value for mean squared error

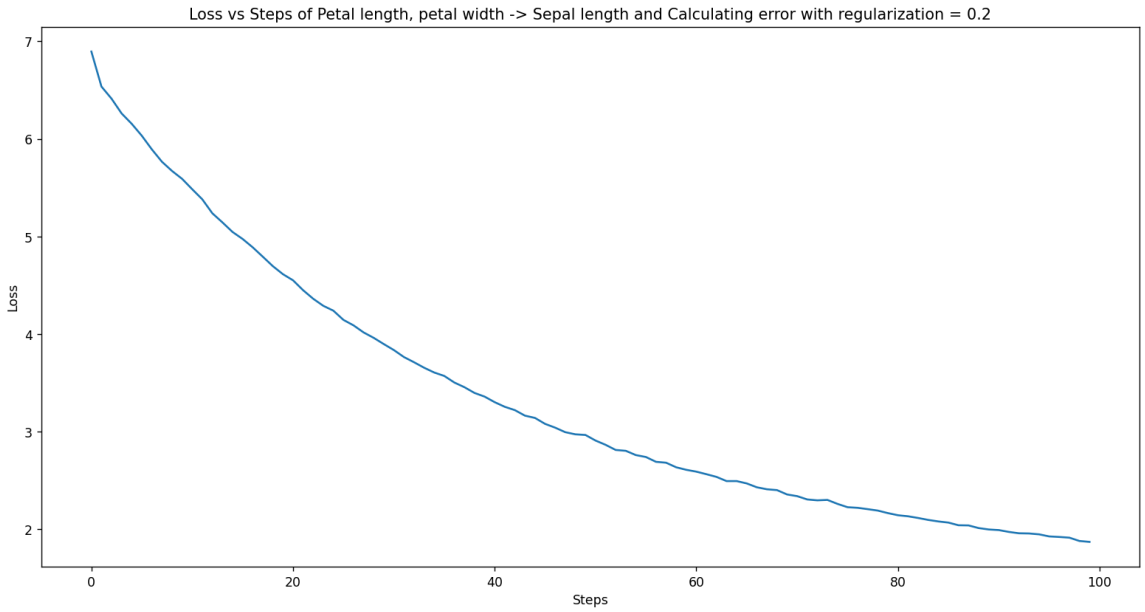
```
PS C:\Users\USER\OneDrive\Desktop\UTA\2nd SEM\Tharunproject> & C:/Users/USER/anaconda3/python.exe "c:/Users/USER
/OneDrive/Desktop/UTA/2nd SEM/Tharunproject/eval_regression2.py"
27.387315584244345
Mean Squared Error: 27.387315584244345
```

2.

Here is the output of train_regression2.py where the graph 1 tells loss versus steps of sepal length and sepal width, here calling the fit function we have used regularization as 0. Hence, we got the below graph



Here is the output of train_regression2.py where the graph 1 tells loss versus steps of sepal length and sepal width, here calling the fit function we have used regularization as 0.2. Hence, we got the below graph.

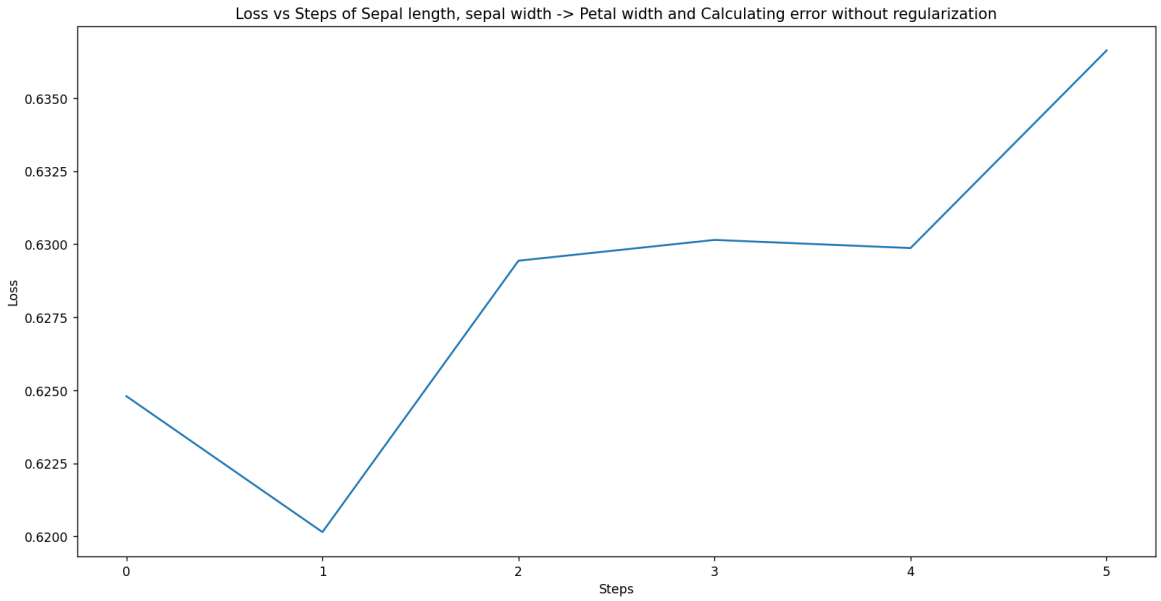


Here is the output of eval_regression2.py where we got the below value for mean squared error

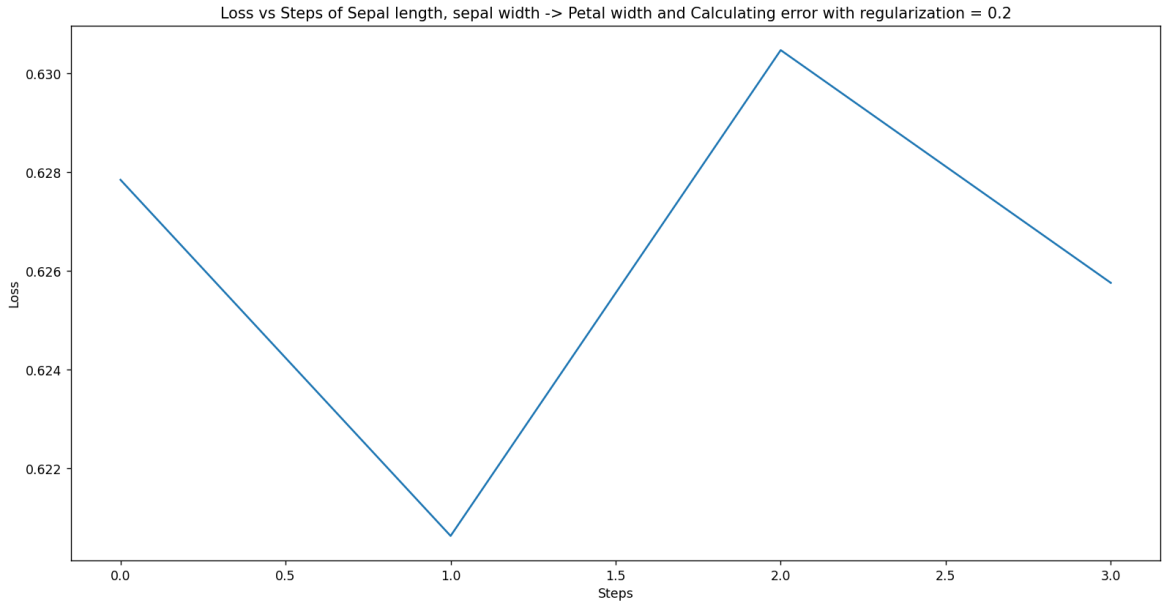
```
PS C:\Users\USER\OneDrive\Desktop\UTA\2nd SEM\Tharunproject> & C:/Users/USER/anaconda3/python.exe "c:/Users/
USER/OneDrive/Desktop/UTA/2nd SEM/Tharunproject/eval_regression2.py"
1.9103119741126928
Mean Squared Error: 1.9103119741126928
```

3.

Here is the output of train_regression3.py where the graph 1 tells loss versus steps of sepal length and sepal width, here calling the fit function we have used regularization as 0. Hence, we got the below graph



Here is the output of train_regression3.py where the graph 1 tells loss versus steps of sepal length and sepal width, here calling the fit function we have used regularization as 0.2. Hence, we got the below graph



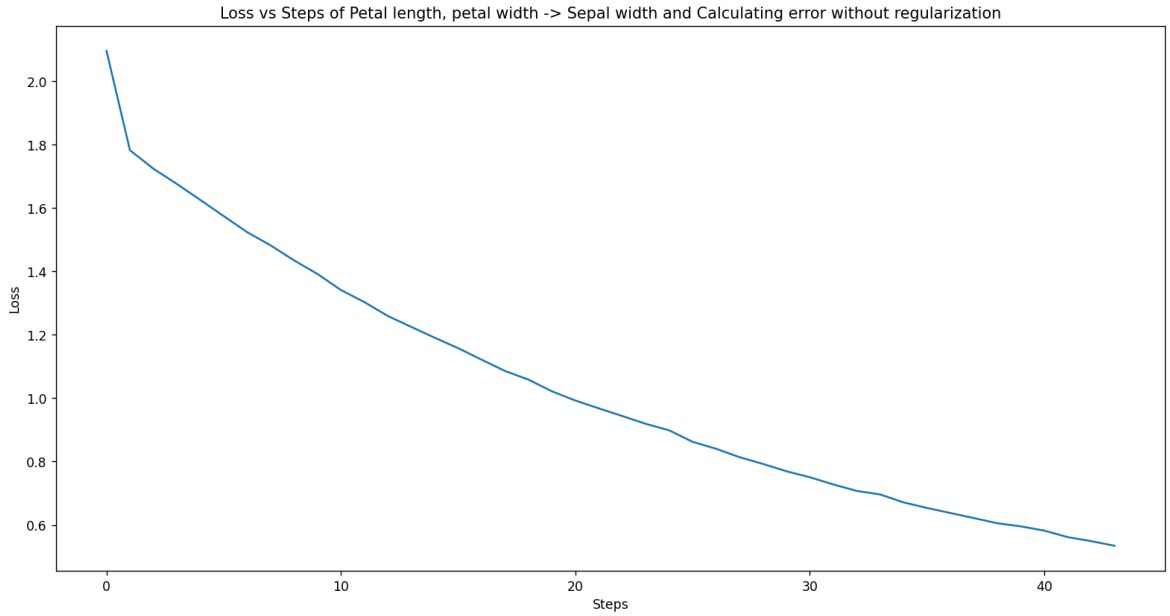
```
PS C:\Users\USER\OneDrive\Desktop\UTA\2nd SEM\Tharunproject> & C:/Users/USER/anaconda3/python.exe "c:/Users/
USER/OneDrive/Desktop/UTA/2nd SEM/Tharunproject/train_regression3.py"
Early stopping at epoch is : 4
Early stopping at epoch is : 5

Early stopping at epoch is : 10
```

Here is the output of eval_regression3.py where we got the below value for mean squared error

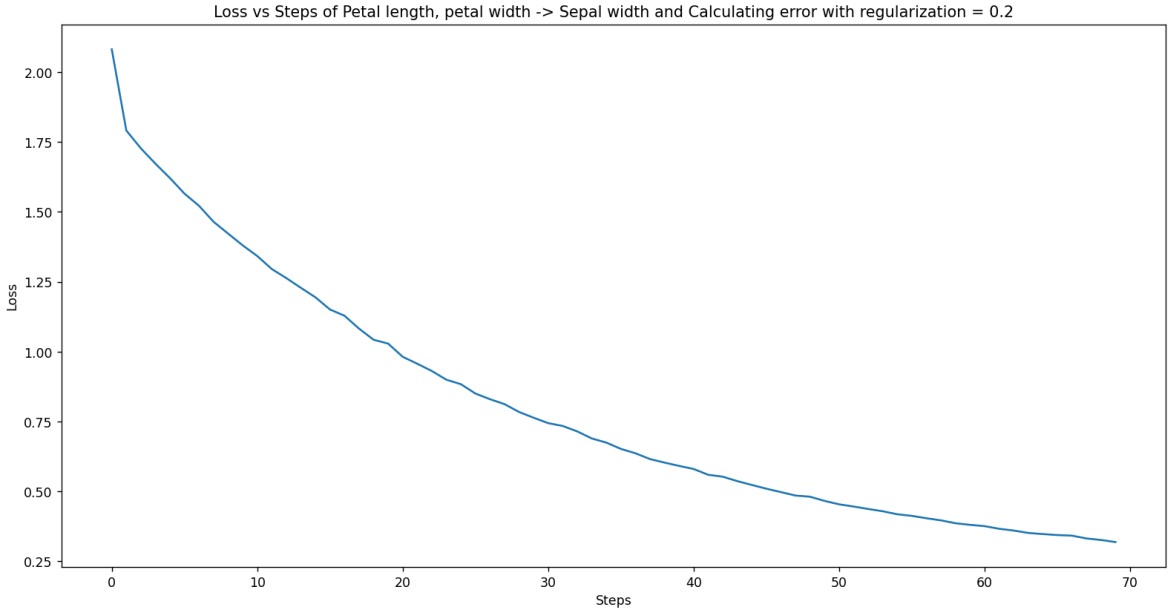
```
PS C:\Users\USER\OneDrive\Desktop\UTA\2nd SEM\Tharunproject> & C:/Users/USER/anaconda3/python.exe "c:/Users/
USER/OneDrive/Desktop/UTA/2nd SEM/Tharunproject/eval_regression3.py"
0.6271637767034545
Mean Squared Error: 0.6271637767034545
```

4.

Here is the output of train_regression4.py where the graph 1 tells loss versus steps of sepal length and sepal width, here calling the fit function we have used regularization as 0. Hence, we got the below graph



Here is the output of train_regression4.py where the graph 1 tells loss versus steps of sepal length and sepal width, here calling the fit function we have used regularization as 0.2. Hence, we got the below graph



```
PS C:\Users\USER\OneDrive\Desktop\UTA\2nd SEM\Tharunproject> & C:/Users/USER/anaconda3/python.exe "c:/Users/
USER/OneDrive/Desktop/UTA/2nd SEM/Tharunproject/train_regression4.py"
Early stopping at epoch is : 60
Early stopping at epoch is : 79

Early stopping at epoch is : 74
```

Here is the output of eval_regression4.py where we got the below value for mean squared error

```
PS C:\Users\USER\OneDrive\Desktop\UTA\2nd SEM\Tharunproject> & C:/Users/USER/anaconda3/python.exe "c:/Users/
USER/OneDrive/Desktop/UTA/2nd SEM/Tharunproject/eval_regression4.py"
0.3387197566195627
Mean Squared Error: 0.3387197566195627
```

Linear Regression

1. What are the pros and cons of using the normal equation to solve for the weights in linear regression as opposed to using gradient descent?

- The Normal Equation offers a precise solution, but it may be computationally costly for big datasets. No need to adjust the hyperparameters.
- Normal equation works well with small number of features.
- No need for feature scaling.

Gradient Descent:

- Requires careful feature scaling and modifying; it scales better with large datasets and can involve regularization.
- Gradient descent works well with large number of features.
- feature scaling can be used.