Homework 5

March 6, 2022

0.1 Problem 1

0.2 MSE Linear Regression

0.2.1 Aim:

- Build a Linear regressor
- Investigate different Learning rates
- Effect of learning rate on the convegence in GD optimization

Given dataset has the following 4 features: **T**: temperature; **AP**: ambient pressure; **RH**: relative humidity; **V**: exhaust vacuum

Output: **EP**: energy output per hr.

the given data points are already normalized.

0.2.2 Solution 1(a)

As we know, the weight update/gradient descent formula for both MSE Regression and MSE Classification is as follows:

MSE REGRESSION:

$$\underline{w}(i+1) = \underline{w}(i) - \eta(i)(\underline{w}^T(i)\underline{x}_n - \underline{y}_n)\underline{x}_n$$

MSE CLASSIFICATION:

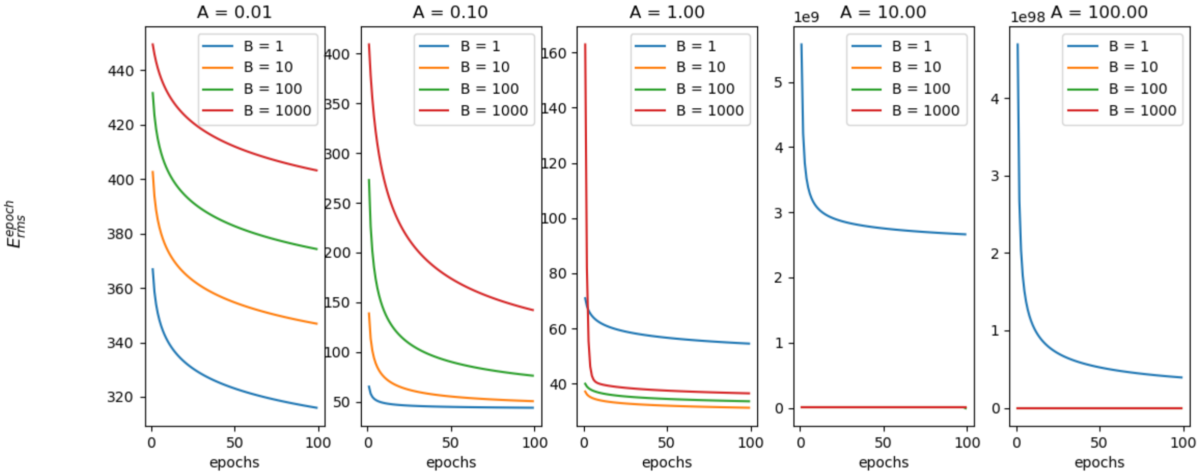
$$\underline{w}(i+1) = \underline{w}(i) - \eta(i)(\underline{w}^T(i)\underline{x}_n - \underline{b}_n)\underline{x}_n$$

with the only difference being that $\underline{y}_n$ is the actual outputs (usually continuous) and $\underline{b}_n$ is the actual labels (discrete). Therefore the following learning rate conditions also apply for convergence in MSE Regression case

$$\boxed{\lim_{m\to\infty}\sum_{i=1}^{m}\eta(i) = +\infty \lim_{m\to\infty}\sum_{i=1}^{m}\eta_2(i) < \infty}$$

0.2.3 Solution 1(b)

Learning Curves $E_{RMS}^{(epoch)}$ vs. *epochs*



0.2.4 Solution 1(c)

Comment on the dependence of the learning curves on A and B.

As seen in the above figure, as the learning rate gets smaller and smaller, the convergence of the gradient descent algorithm gets slower. So, that means even in the 100 epochs, the gradient would not converge and hence, the error would still be high enough (this can be seen in the 1st plot, for A = 0.01, B = 1000). The error slowly decreases as A increases to 1 indicating lower RMS rates and as the learning rate gets bigger and bigger, instead of the convergence, the gradients will now fluctuate from one point to another to a larger extent with every weight update and this will cause the error to further shoot up as seen in the last two plots here (when A = 10 and 100; B = 1). Hence, we can say that these plots give a practical justification of the above argument that even in the case of MSE Regression, the learning rates can neither be too small nor too large.

DISCLAIMER: As seen in the figure above, the plots for the last two are not clear because the error rates are out of scale, i.e., too large. Hence, please check the *Miscellaneous* Section below for zoomed-in plots for further clarification.

0.2.5 Solution 1(d)

The best pair (A, B) are as follows:

$$\boxed{(A, B) = (100, 1000)}$$

and the corresponding weight vector approximated to 3 decimal places is given as:

$$\boxed{w_{best} \approx [257.245, -56.396, -23.798, 255.210, -6.139]^T}$$

The above regressor is run on the test set and the train set and the following MSE was observed for each -

$$\boxed{E_{RMS}^{test} = 4.6536\% | E_{RMS}^{train} = 4.6442\%}$$

correct to 4 decimals.

0.2.6 Solution 1(e)

I have run a trivial regressor that always predicts $\hat{y}(\underline{x}) = y_{mean}$. The RMS error over the test set with this trivial regressor is as follows:

$$\boxed{E_{RMS}^{test} = 18.8670\%}$$

correct to 4 decimals

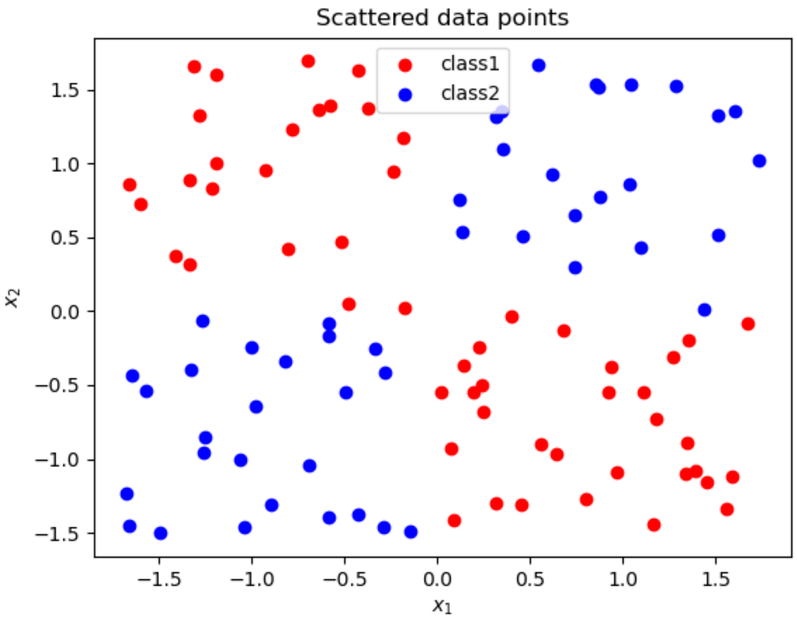
The Regressor error is substantially lower than the error of this trivial regressor -

As seen from Solution 1(d), the E_{RMS} using the regressor is much lower than the trivial error. This because in the trivial regressor we have a straight line that passes through y_{mean} and this curve fit clearly underfits the test data and hence, the trivial error is larger than the regressor error above.

0.3 Problem 3

0.3.1 Solution 3(a)

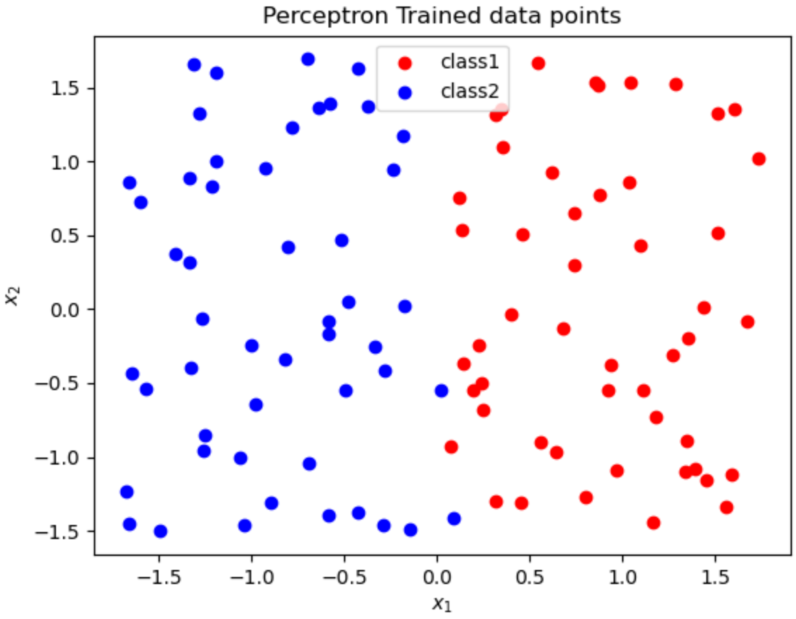
Scatter plot of the data points in non-augmented feature space:



As seen, the data is **not linearly separable** here

0.3.2 Solution 3(b)

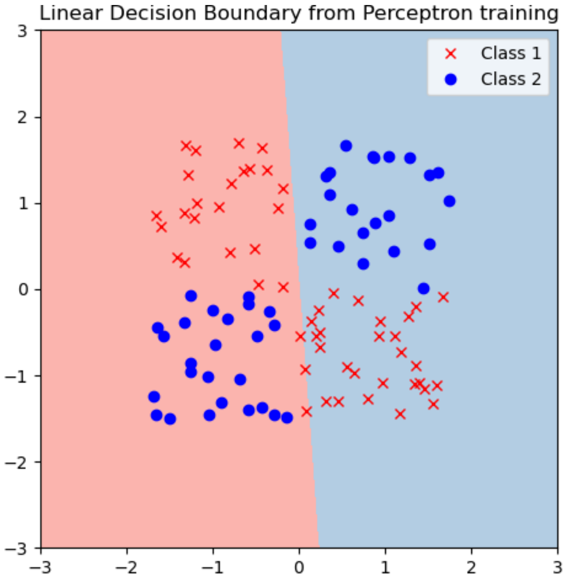
Scatter plot of the perceptron trained data points:



classification accuracy in feature space = $\mathbf{0.53} = \mathbf{53\%}$

0.3.3 Solution 3(c)

Learned linear decision boundaries from the perceptron above:



0.3.4 Solution 3(d)

Expanded Feature Space

$$g(x) = w_0\mathbf{x_1} + w_1\mathbf{x_2} + w_2\mathbf{x_1}\mathbf{x_2} + w_3\mathbf{x_1^2} + w_4\mathbf{x_2^2}$$

where the original feature space is mapped into an expanded feature space given as:

$$(x_1, x_2) \rightarrow (x_1, x_2, x_1x_2, x_1^2, x_2^2)$$

The classification error in the expanded feature space after running the perceptron: $\mathbf{1.0 = 100\%}$

In the expanded feature space, the data is linearly separable. the linear hyperplane in this new expanded space will be a non-linear hypersurface in the original space. This shall be seen as follows

0.3.5 Solution 3(e) EXTRA CREDIT

The learned weight vector from 3(d): $\boxed{[-0.10562188, -0.14390327, 7.616127, 0.05743614, -0.28634851]^T}$

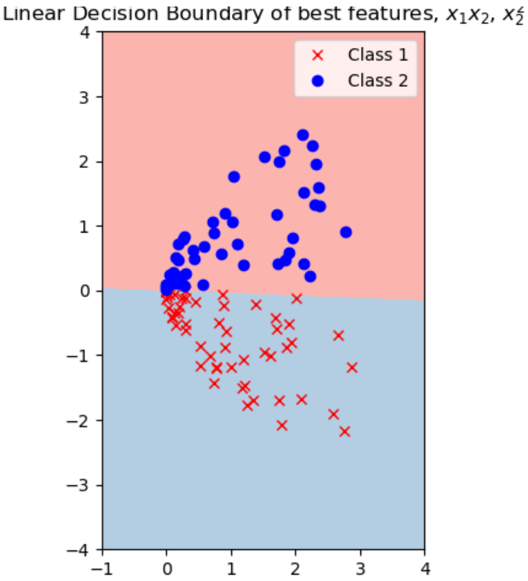
Finding best two features with the highest absolute weight values -

$$(x_1x_2, x_2^2)$$

decision function:

$$g(x) = w_2x_1x_2 + w_4x_2^2$$

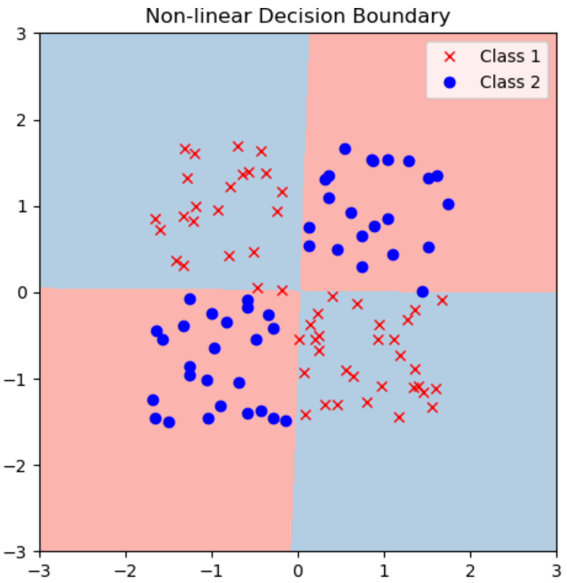
Decision Boundary plot with the 2 best features and corresponding weight vectors:



As seen with these two best features, **the data is linearly separable.**

0.3.6 Solution 3(f) EXTRA CREDIT

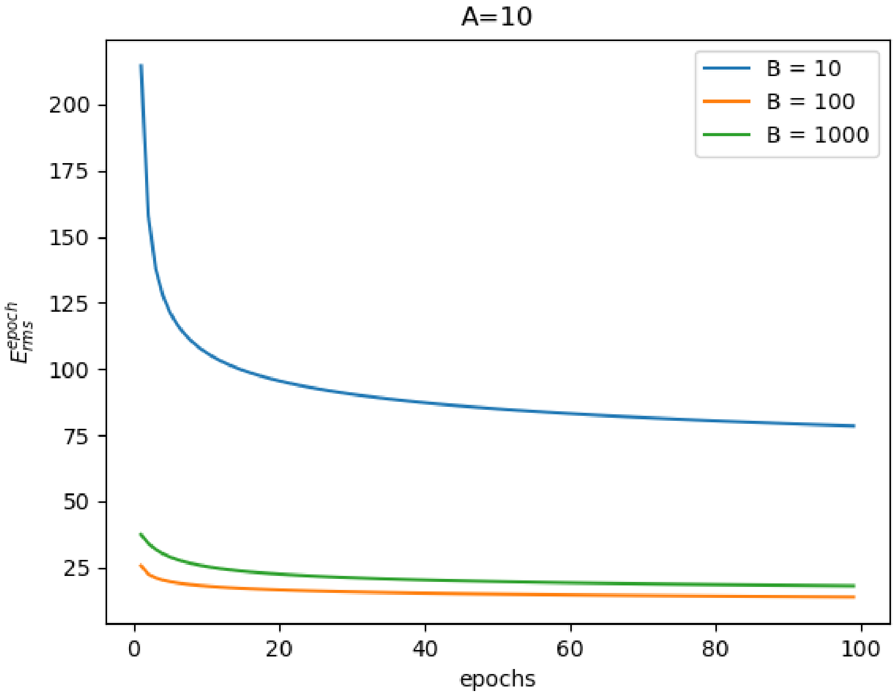
Non-Linear decision boundary and regions in original feature space:

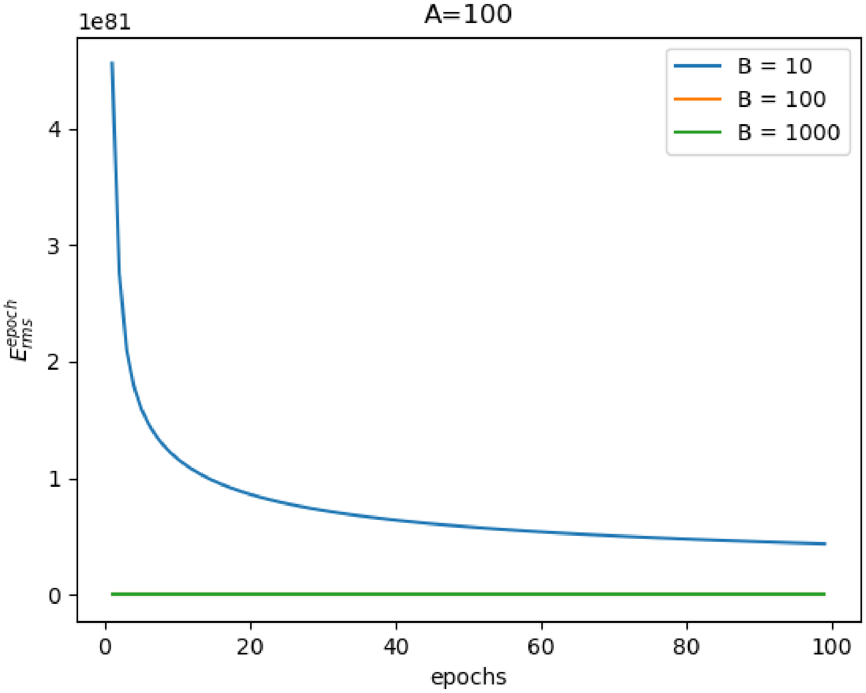


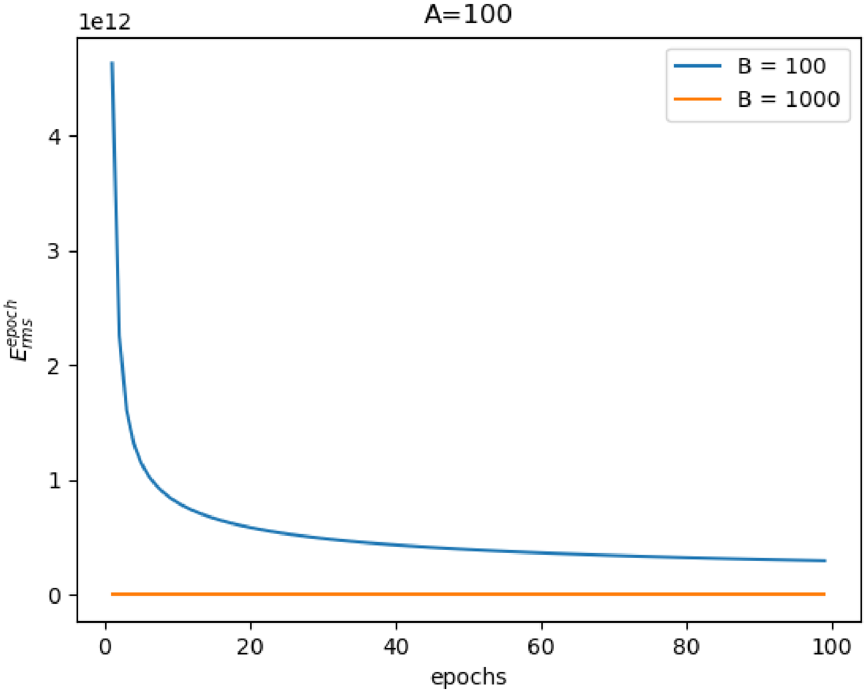
As seen the data is very nicely separable

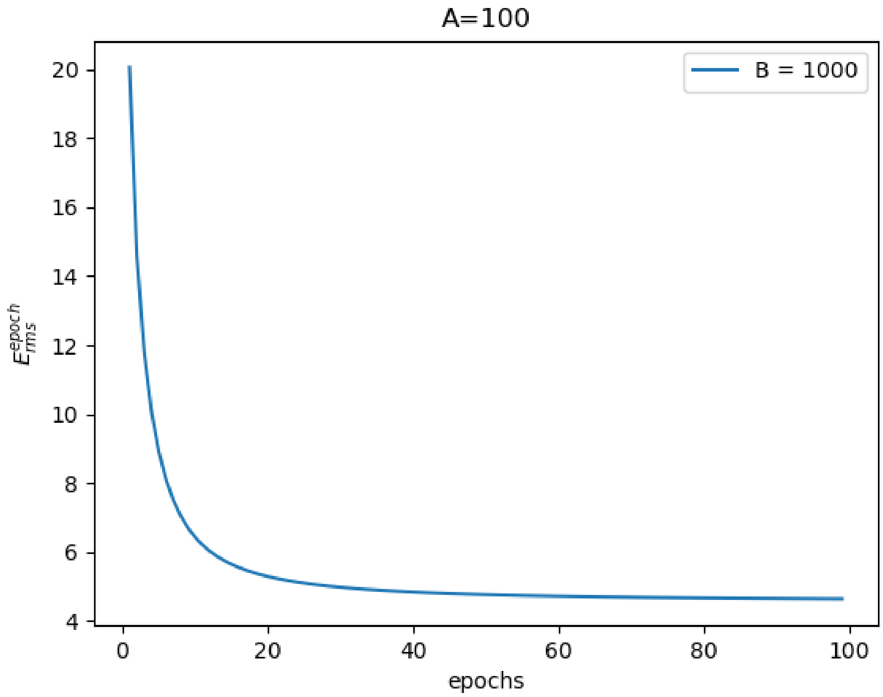
0.4 Miscallaneous

I have here attached the zoomed-in plots for the last two plots in the figure given in Solution 1(b) for general clarity.




A=100
1e81
4
3
2
1
0
E_{rms}^{epoch}
0
20
40
60
80
100
epochs
B = 10
B = 100
B = 1000


A=100
1e12
4
3
2
1
0
E_{rms}^{epoch}
0
20
40
60
80
100
epochs
B = 100
B = 1000



The last figure clearly shows a low RMS error and hence justifies our choice of best pair of (A,B)

```
[ ]:
```