Lecture Notes for session conducted on June 10, 2022

https://www.scaler.com/academy/mentee-dashboard/class/28569/session

Content

- Regularisation:
  - What are Regularisation techniques?
  - How do they address Bias-Variance Trade-off?
  - Lasso and Ridge Regression.
  - Advantages and Dis-advantages.

Regularisation Techniques:

- Regularisation techniques are used in Linear Regression along with the Loss function (be it Square Loss, Log loss, Hinje Loss and Huber Loss) to ***reduce overfitting*** and try to balance the Bias-Variance Trade-off.

L1 - Regularisation:

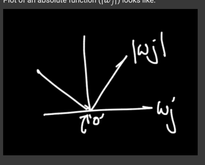
- L1 regularisation is also known as *Lasso* or *Tikhonov* regularisation or L1-norm.
- Cost function while using L1 regularisation looks like:

$$\min_{w_j} \square(Loss function) + \lambda \sum_{j=0}^{d} |w_j|$$

where:

  - Loss function could be either of Squared Loss, Log Loss, Hinge Loss, Huber Loss.
  - d: Number of features. ($j : 0 \rightarrow d$ so as to include w_0 as well)

- Weights of useless features becomes zero.
- Plot of an absolute function ($|w_j|$) looks like:



L2 - Regularisation:

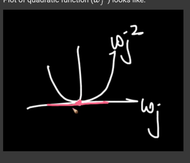
- L2 regularisation is also known as *Ridge* regularisation or L2-norm.
- Cost function while using L2 regularisation looks like:

$$\min_{w_j} \square(Loss function) + \lambda \sum_{j=0}^{d} w_j{}^2$$

where:

  - Loss function could be either of Squared Loss, Log Loss, Hinge Loss, Huber Loss).
  - d: Number of features. ($j : 0 \rightarrow d$ so as to include w_0 as well)

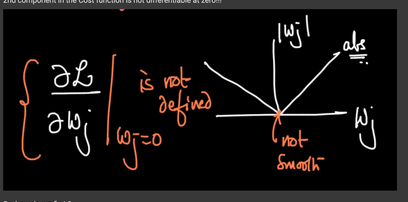
- Weights of useless features becomes very small (Close to Zero).
- Plot of quadratic function ($w_j{}^2$) looks like:



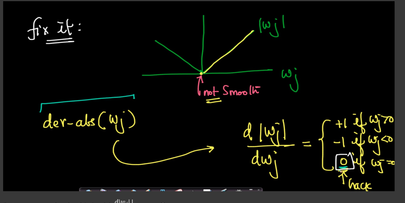
Question: Do you notice any problem with L1 regularisation?

Mathematical Explanation:

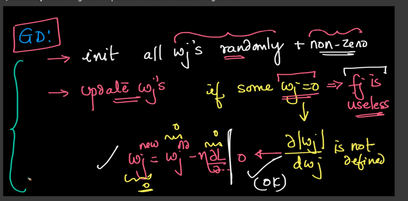
- In Gradient Descent, we update the weights by calculating derivative of Loss function wrt to weights.
- 2nd component in the Cost function is not differentiable at zero!!!



- So, how do we fix it?



- But one may ask, why this is ok? $\frac{d|w_j|}{dt}\Big|_{w_j=0} = 0$
- The answer to above equation lies in how weights are updated in Gradient Descent:
- ***Quick Recap:*** How weights are updated in Gradient Descent Algorithm.



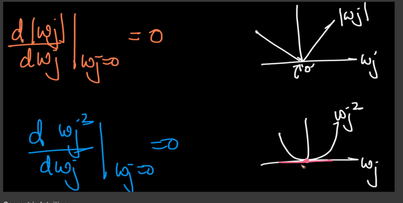
- In the update step we do the following:
  - $w_j{}^{new} = w_j{}^{old} - \eta \frac{\partial \square}{\partial w_j}$
  - If $\frac{\partial \square}{\partial w_j}\Big|_{w_j=0} = 0$, then $w_j{}^{new}$ also becomes zero in subsequent update.
- To summarise, $\frac{d|w_j|}{dt}\Big|_{w_j=0} = 0$ is ok $\because w_j = 0$ means that f_j(feature) is useless.

Interview Alert!!! Asked in MAANG. Interviewer would ask why $\frac{d|w_j|}{dt}\Big|_{w_j=0} = 0$. Why not some other value other than 0?

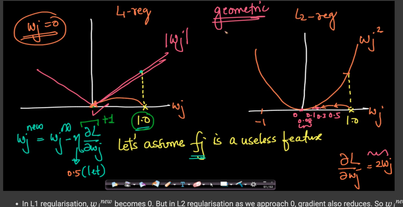
Interesting property of L1 regularisation:

- L1 regularisation results in sparse solution. i.e. all less useful feature's weight becomes 0.
- On other hand in L2 regularisation, all less useful feature's weight becomes close to 0, but often not exactly 0.

Can you guess why this happens? Let's try to find the answer intuitively.Again the answer lies in how the weights are updated in L1 and L2 regularisation.



Geometric Intuition:



- In L1 regularisation, $w_j{}^{new}$ becomes 0. But in L2 regularisation as we approach 0, gradient also reduces. So $w_j{}^{new}$ approaches 0, but may not becomes 0.

Disadvantages of L1 regularisation:

- If d > n (d is dimensions and n is #datapoints.):
  - Typically a situation in GNOME data.
  - L1 regularisation will lead to atmost "n" non-zero features even if more than "n" features are useful.
- When we have lot of correlated features:
  - If L1 regularisation is used, then one of the feature will be non-zero and other correlated features become 0.
  - This is sometimes okay. But while using weights for feature importance, it will lead to wrong interpretation and messup feature importance.