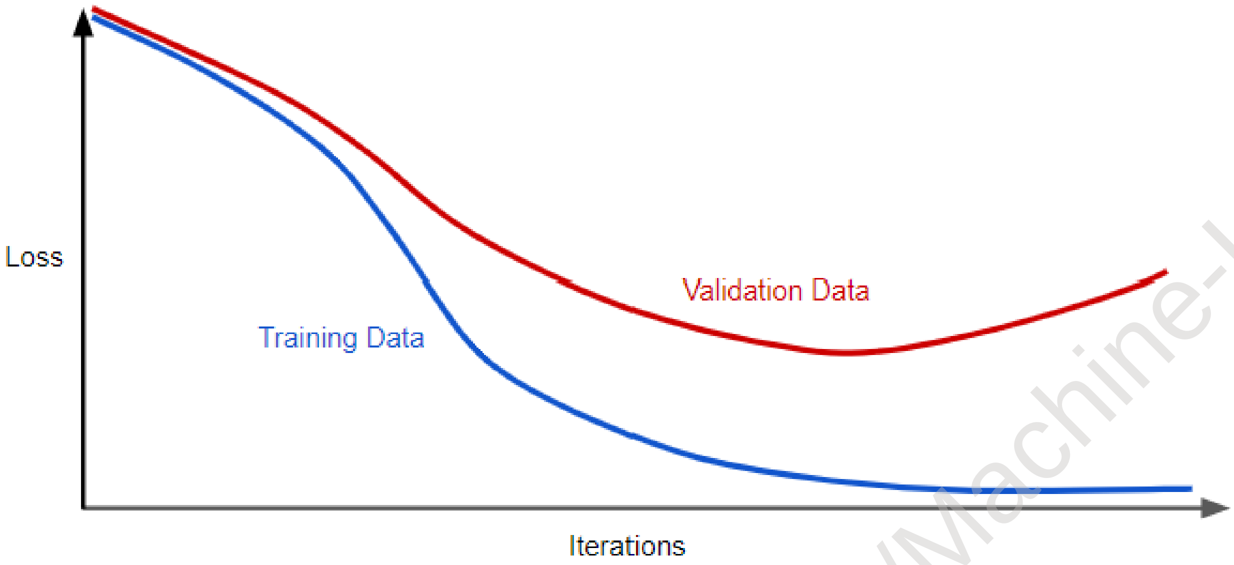
ML Concepts - Regularization



The generalization curve illustrates the relationship between loss and the number of training iterations for both the training set and the validation set. In this case, the training loss gradually decreases, while the validation loss initially decreases but then starts to rise.

This pattern suggests that the model is overfitting to the training data, emphasizing the need for regularization to prevent overfitting. Regularization involves minimizing both the loss term (which measures how well the model fits the data) and the regularization term (which measures model complexity).

1. Complexity based on feature weights in the model.
2. Complexity based on the total number of features with nonzero weights.

If complexity is defined by feature weights, higher absolute values indicate greater complexity. The L2 regularization formula quantifies complexity by summing the squares of all feature weights. Weights close to zero have minimal impact on complexity, while outlier weights exert significant influence.

*How do you define Complexity?

- Favor smaller weights.
- This concept can be implemented using L2 regularization, also known as ridge regularization which penalizes excessively large weights..
- Complexity of a model can be quantified as the sum of the squares of the weights.
- **Bayesian Prior for Model Complexity**:
  - Suggests that weights should be centered around zero.
  - Emphasizes that weights should follow a normal (bell-shaped) distribution.

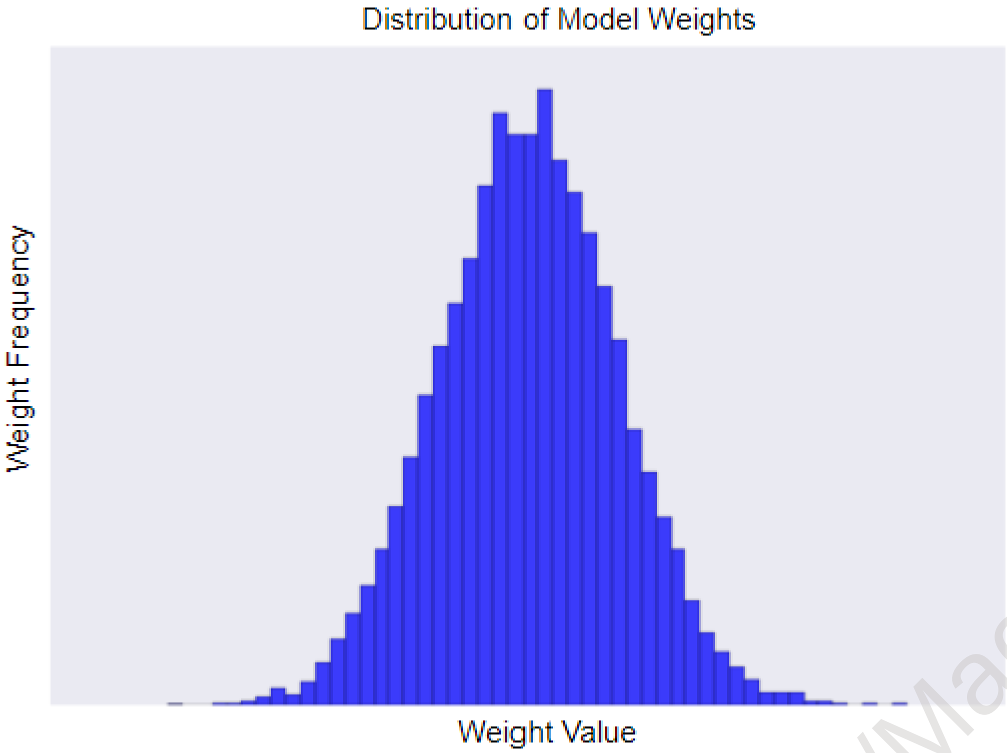
$$Loss(Data|Model) + \lambda\left(w_1^2 + \ldots + w_n^2\right)$$

Where:

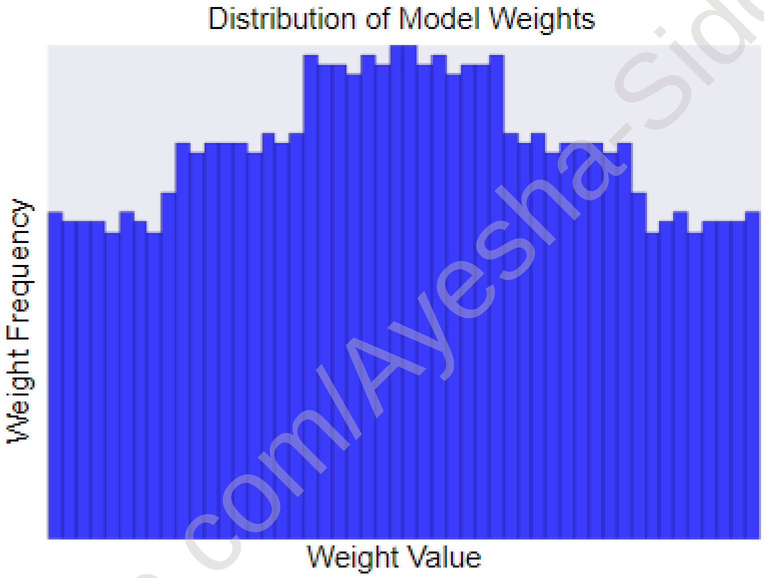
$Loss$: Aims for low training error
λ: Scalar value that controls how weights are balanced
$w_1^2 + \ldots + w_n^2$: Square of L_2 norm

*Selecting the right Lambda

- Model developers adjust the impact of the regularization term using a scalar known as lambda or regularization rate.
- L2 regularization encourages weight values towards 0, but not exactly 0, and it also encourages the mean of the weights to approach 0 with a normal distribution.
- Increasing lambda strengthens the regularization effect, leading to weight distributions with higher concentration around 0.
- Lowering lambda leads to flatter weight distributions with less concentration around 0.
- Choosing the right lambda value is crucial to strike a balance between model simplicity and fitting the training data.
- A too high lambda may result in underfitting, where the model doesn't learn enough from the training data.
- A too low lambda may lead to overfitting, where the model learns too much about the specifics of the training data and struggles to generalize to new data.
- Setting lambda to zero removes regularization completely, which can lead to the highest risk of overfitting.
- The ideal lambda value depends on the specific dataset, so some tuning may be necessary.
- There is a connection between learning rate and lambda. Strong L2 regularization and lower learning rates can have similar effects on driving feature weights closer to 0.
- Early stopping involves ending training before the model fully converges, which can be useful in preventing overfitting or when training in an online, continuous fashion.
- Implicit early stopping can occur in online training when some trends haven't had enough data to converge.
- It's advisable to use a sufficient number of iterations in training to minimize the impact of early stopping when training on a fixed batch of data.

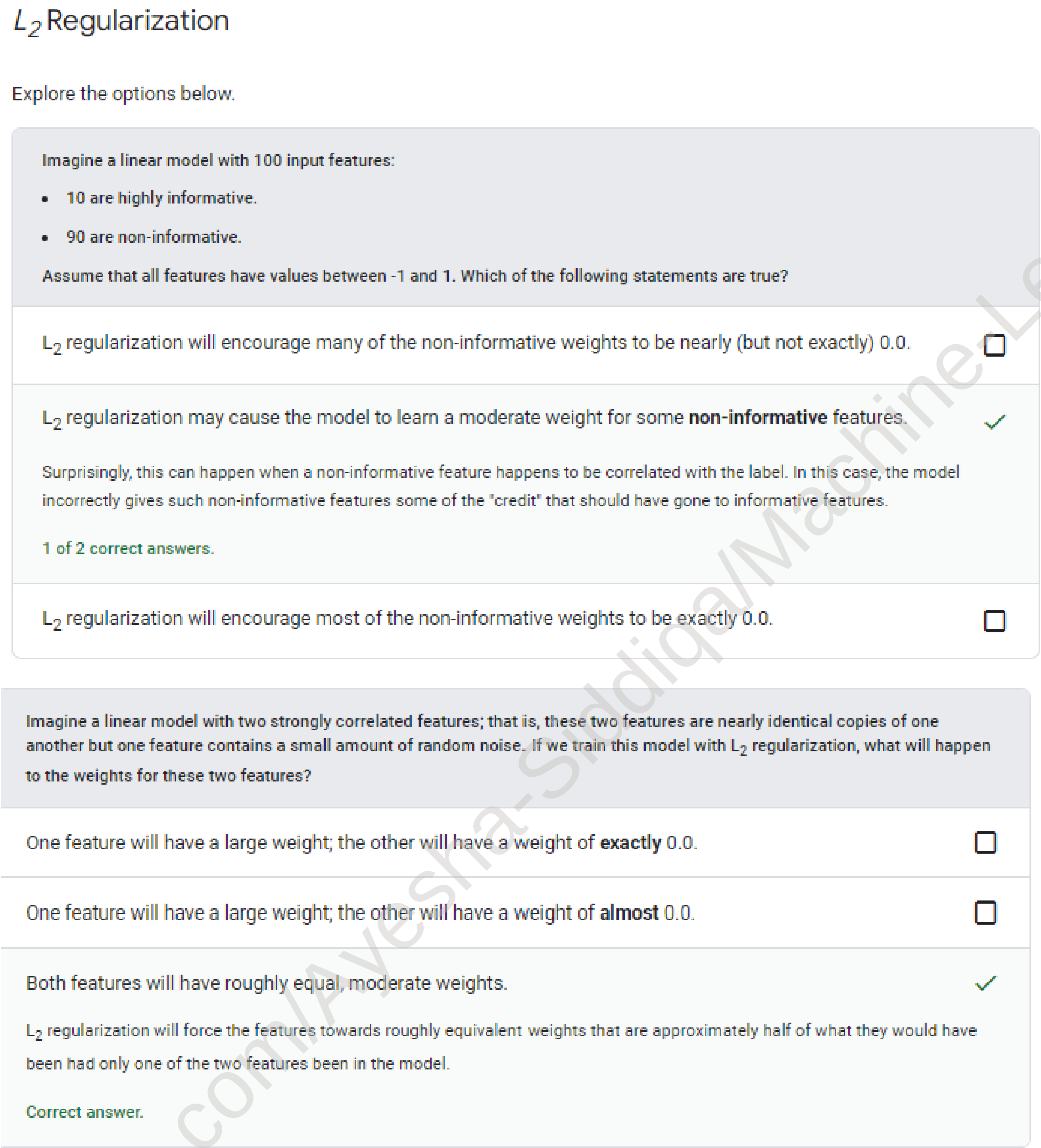


Increasing the lambda value strengthens the regularization effect. For example, the histogram of weights for a high value of lambda might look as shown in Figure



Lowering the value of lambda tends to yield a flatter histogram

Quiz:



L_2 Regularization

Explore the options below.

Imagine a linear model with 100 input features:

- 10 are highly informative.
- 90 are non-informative.

Assume that all features have values between -1 and 1. Which of the following statements are true?

L_2 regularization will encourage many of the non-informative weights to be nearly (but not exactly) 0.0. □

L_2 regularization may cause the model to learn a moderate weight for some **non-informative** features. ✓

Surprisingly, this can happen when a non-informative feature happens to be correlated with the label. In this case, the model incorrectly gives such non-informative features some of the "credit" that should have gone to informative features.

1 of 2 correct answers.

L_2 regularization will encourage most of the non-informative weights to be exactly 0.0. □

Imagine a linear model with two strongly correlated features; that is, these two features are nearly identical copies of one another but one feature contains a small amount of random noise. If we train this model with L_2 regularization, what will happen to the weights for these two features?

One feature will have a large weight; the other will have a weight of **exactly** 0.0. □

One feature will have a large weight; the other will have a weight of **almost** 0.0. □

Both features will have roughly equal, moderate weights. ✓

L_2 regularization will force the features towards roughly equivalent weights that are approximately half of what they would have been had only one of the two features been in the model.

Correct answer.

*What is Regularization Sparsity?

- High-dimensional sparse vectors and feature crosses can lead to large model sizes and high RAM requirements.
- Encouraging weights to drop to exactly 0 in high-dimensional sparse vectors can save RAM and reduce noise in the model.
- L2 regularization encourages small weights but does not force them to exactly 0.
- L0 regularization, which penalizes the count of non-zero coefficients, is intuitive but turns the optimization problem into a non-convex one, making it impractical.
- L1 regularization serves as an approximation to L0, is convex, and efficiently encourages many uninformative coefficients to be exactly 0, leading to RAM savings at inference time.

- L2 penalizes weight^2, while L1 penalizes |weight|.
- The derivative of L2 is 2 * weight, while the derivative of L1 is a constant independent of weight.
- L2 gradually reduces weights, but they don't reach exactly zero. L1, due to absolute values, can force weights to exactly zero when the subtraction results cross 0.
- L1 regularization, which penalizes the absolute value of weights, is efficient for wide models.