Regularization

Mohammed Luqmaan Akhtar

December 2025

Introduction

Since MLPs are so complex with many different layers and nodes they are highly prone to overfitting, thus regularization techniques are used.

Overfitting and Underfitting

Overfitting occurs when a model performs well on training data but poorly on unseen data, typically due to excessive model complexity. Underfitting, on the other hand, arises when a model is too simple to capture underlying patterns in the data. Regularization aims to find a balance between bias and variance.

Regularization Concept

Regularization introduces penalties into the loss function to discourage overly complex models. This is typically achieved by penalizing large weights, encouraging sparsity, or adding stochasticity during training.

L1 and L2 Regularization

L1 Regularization

L1 regularization adds the absolute value of weights to the loss function:

$$L = L_{\text{data}} + \lambda \sum_i |w_i|$$

This encourages sparsity in the learned parameters, effectively performing feature selection by driving some weights to zero.

L2 Regularization

L2 regularization penalizes the squared magnitude of weights:

$$L = L_{\text{data}} + \lambda \sum_i w_i^2$$

The gradient becomes:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L_{\text{data}}}{\partial w_i} + 2\lambda w_i$$

L2 regularization discourages large weights and leads to smoother models.

Weight Decay

Weight decay is an alternative interpretation of L2 regularization where weights are explicitly scaled down during optimization:

$$w := (1 - \eta\lambda)w - \eta \nabla L_{\text{data}}$$

It effectively reduces model complexity during training.

Dropout

Dropout randomly deactivates neurons during training with probability p. This prevents neurons from co-adapting and forces the network to learn redundant representations.

$$\tilde{a}^{(l)} = a^{(l)} \cdot r^{(l)}, \quad r^{(l)} \sim \text{Bernoulli}(p)$$

Batch Normalization

Batch Normalization normalizes layer activations using mini-batch statistics:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

Some patterns are ignored when using large batches, this helps in solving the problem.

Layer Normalization

Unlike Batch Normalization, Layer Normalization normalizes across features within a single sample, making it suitable for recurrent and small-batch settings.

Early Stopping

Early stopping halts training when validation loss stops improving. This prevents overfitting by avoiding excessive parameter updates.