Regularization Techniques in Multi-Layer Perceptrons

1 Introduction

Neural networks with sufficient capacity are prone to overfitting, especially when trained on limited or noisy data. Regularization refers to a collection of techniques designed to improve generalization by constraining the effective complexity of a model. In the context of Multi-Layer Perceptrons (MLPs), regularization operates at multiple levels: parameter norms, training dynamics, stochasticity, and normalization of internal representations.

This report discusses the theoretical motivation and practical implementation of major regularization techniques used in MLPs.

2 Overfitting and Underfitting

2.1 Overfitting

Overfitting occurs when a model learns patterns specific to the training data, including noise, rather than the underlying data-generating process. This results in low training error but poor performance on unseen data.

Formally, overfitting is characterized by:

$$\mathcal{L}_{\text{train}} \ll \mathcal{L}_{\text{test}}$$

2.2 Underfitting

Underfitting arises when the model lacks sufficient capacity or is overly constrained, preventing it from capturing meaningful structure in the data. In this case, both training and test errors remain high.

3 L2 Regularization and Weight Decay

L2 regularization penalizes large weights by adding a quadratic penalty term to the loss function:

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \frac{\lambda}{2} \sum_{l} \|\mathbf{W}^{(l)}\|_2^2$$

The resulting gradient update becomes:

$$\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \eta \left(\nabla_{\mathbf{W}^{(l)}} \mathcal{L} + \lambda \mathbf{W}^{(l)} \right)$$

This effectively encourages smaller weights, leading to smoother functions and reduced sensitivity to input perturbations.

3.1 Weight Decay Interpretation

Weight decay can be interpreted as a continuous shrinking of weights during training. In SGD, L2 regularization and weight decay are equivalent, although they differ slightly in adaptive optimizers such as Adam.

4 L1 Regularization

L1 regularization imposes an absolute value penalty:

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \lambda \sum_{l} \|\mathbf{W}^{(l)}\|_1$$

Unlike L2 regularization, L1 promotes sparsity by driving many weights exactly to zero. This is particularly useful for feature selection and interpretability, although it is less commonly used in deep MLPs due to optimization difficulties.

5 Dropout

Dropout is a stochastic regularization technique that randomly disables neurons during training. For a hidden activation **a**:

$$\tilde{\mathbf{a}} = \mathbf{a} \odot \mathbf{m}, \quad m_i \sim \text{Bernoulli}(p)$$

At inference time, activations are scaled by p to preserve expected output magnitude.

Dropout can be viewed as:

- An implicit ensemble of subnetworks
- A mechanism that prevents co-adaptation of neurons

6 Batch Normalization

Batch normalization (BatchNorm) normalizes pre-activations within a mini-batch:

$$\mu_B = \frac{1}{m} \sum_{i=1}^{m} z_i, \quad \sigma_B^2 = \frac{1}{m} \sum_{i=1}^{m} (z_i - \mu_B)^2$$

$$\hat{z}_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i = \gamma \hat{z}_i + \beta$$

BatchNorm stabilizes training by reducing internal covariate shift and often acts as a regularizer by injecting batch-level noise.

7 Layer Normalization

Layer normalization computes statistics across features instead of batch samples:

$$\mu = \frac{1}{H}\sum_{j=1}^{H} z_j, \quad \sigma^2 = \frac{1}{H}\sum_{j=1}^{H}(z_j - \mu)^2$$

LayerNorm is independent of batch size and is particularly effective in recurrent and transformer-based architectures.

8 Early Stopping

Early stopping halts training once validation performance stops improving. This implicitly regularizes the model by limiting the effective training time, preventing convergence to overly specialized solutions.

From an optimization perspective, early stopping behaves similarly to L2 regularization by restricting parameter growth.

9 Regularization as Bias-Variance Tradeoff

Regularization increases bias while reducing variance. The optimal regularization strength balances these competing effects, minimizing expected generalization error.

10 Interaction of Regularization Techniques

In practice, multiple regularization methods are combined:

- L2 regularization for smoothness
- Dropout for robustness
- Normalization for stability
- Early stopping for generalization

However, excessive regularization can lead to underfitting, emphasizing the need for careful hyperparameter tuning.

11 Conclusion

Regularization is essential for training effective MLPs in real-world settings. By constraining model complexity, injecting stochasticity, and stabilizing internal representations, regularization techniques enable neural networks to generalize beyond their training data. A principled understanding of these methods allows practitioners to design models that are both expressive and robust.