学习和使用神经网络训练过程中常用的正则化方法

白锦琪

2025 年 3 月 6 日

以 MNIST 分类任务模型为例，展示 Weight decay、Dropout、Stochastic depth 四种正则化方法对模型分类效果和训练过程稳定性的影响。

一. 数学原理

1.Weight decay(权重衰减)

权重衰减的核心算法原理是通过在损失函数中添加一个正则项来约束模型的权重，从而防止模型过拟合。这个正则项通常是权重的平方和，加上一个正的超参数，这个超参数控制了正则项对损失函数的影响程度。具体来说，权重衰减的损失函数可以表示为：

$$L(\theta) = L_{data}(\theta) + \lambda L_{reg}(\theta)$$

其中，$L_{data}(\theta)$ 是原始损失函数，$L_{reg}(\theta)$ 是正则项，λ 是正则化超参数。

对于一个简单的线性回归模型，原始损失函数可以表示为：

$$L_{data}(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f_\theta(x_i) - y_i)^2$$

其中，$f_\theta(x_i)$ 是模型的预测值，y_i 是真实值，n 是训练样本数。

正则项通常是权重的平方和，加上一个正的超参数。对于一个简单的线性回归模型，正则项可以表示为：

$$L_{reg}(\theta) = \frac{\lambda}{2n}\sum_{i=1}^{m}\theta_i^2$$

其中，m 是权重的数量，λ 是正则化超参数。

将原始损失函数和正则项相加，我们可以得到权重衰减损失函数：

$$L(\theta) = L_{data}(\theta) + \lambda L_{reg}(\theta)$$

2.Dropout

Dropout 的核心思想是在训练过程中随机丢弃神经元，以防止模型过于依赖于某些特定的神经元。具体来说，Dropout 的算法原理可以分为以下几个步骤：

1）在训练过程中，每个神经元都有一定的概率被随机丢弃。这个概率通常被设为 0.5，但可以根据具体问题调整。

2）当一个神经元被丢弃时，它所连接的神经元将不能接收到来自该神经元的输出。

3）在每次训练迭代中，神经元的丢弃状态是随机的，即使同一个神经元可能在不同迭代中被丢弃或不被丢弃。

4）在测试过程中，所有的神经元都被保留，即使用训练好的模型进行预测时，不会随机丢弃神经元。

Dropout 的数学模型可以通过以下公式表示：

$$p(x) = \prod_{i=1}^{n} p\left(x_i\right)$$

其中，$p\left(x_i\right)$ 表示第 i 个神经元被丢弃的概率，n 表示神经元的数量。在训练过程中，我们可以通过以下公式计算每个神经元的丢弃概率：

$$p(x_i) = 1 - \frac{1}{1 + e^{-\alpha x_i}}$$

其中，α 是一个超参数，用于控制丢弃概率，x_i 是第 i 个神经元的输入值。在测试过程中，我们可以通过以下公式计算每个神经元的输出值：

$$y_i = \frac{1}{\sqrt{2^m - 1}} \sum_{j=1}^{2^m} \frac{x_j}{p(x_j)}$$

其中，m 是神经元的输入数量，x_j 是第 j 个输入神经元的输出值，$p(x_j)$ 是第 j 个输入神经元的丢弃概率。

3.Stochastic depth

Stochastic depth（随机深度）旨在通过随机丢弃网络的某些层来加速训练并提高模型的泛化能力。其核心思想是在训练过程中随机跳过某些层，而在测试阶段使用完整的网络。

假设网络有 L 层，第 l 层的输出为 $f_l(x)$，其中 x 是输入。Stochastic Depth 的数学模型可表示如下：

为每一层 l 定义一个丢弃概率 p_l，通常 p_l 随层数增加而线性增加：

$$p_l = 1 - \frac{l}{L}(1 - p_L)$$

其中 p_L 是最后一层的保留概率（通常设置为 0.5）。

在训练时，每一层以概率 p_l 被保留，或以概率 $1 - p_l$ 被丢弃。丢弃时，该层的输出直接跳过，使用恒等映射（identity mapping）：

$$f_l(x) = \begin{cases} \text{原层计算} & \text{以概率} p_l \\ x & \text{以概率} 1 - p_l \end{cases}$$

在测试阶段，所有层都被保留，但需要对每层的输出进行加权，以保持输出的期望值与训练时一致：

$$f_l(x) = p_l \cdot \text{原层计算} + (1 - p_l) \cdot x$$

二. 在 MNIST 分类任务上的应用

在此实验中，通过构建包含三层卷积和全连接层的深层网络，结合批量归一化（Batch Normalization）和动态学习率调整，对比分析了三种正则化方法（Weight Decay、Dropout、Stochastic Depth）对卷积神经网络在 MNIST 数据集上的分类效果及训练过程稳定性的影响。实验结果表明，Stochastic Depth 方法在测试集上表现最优（准确率 99.24%）。

1. 神经网络设计

实验采用自定义的深层卷积网络 DeeperCNN，结构如下：

1）输入层：MNIST 灰度图像（1x28x28）

2）三层卷积层：

Conv1：64 通道，3×3 卷积核，步长 =1，填充 =1

Conv2：128 通道，3×3 卷积核，步长 =1，填充 =1

Conv3：256 通道，3×3 卷积核，步长 =1，填充 =1

3）池化层：每层卷积后接 2×2 最大池化（步长 =2）

4）全连接层

2. 正则化方法实现

1. Weight Decay：优化器中设置 weight_decay=0.001。
2. Dropout：全连接层间插入丢弃概率为 0.3 的 Dropout 层。
3. Stochastic Depth：训练时以 20% 概率跳过第一层卷积（Conv1）。

3. 实验结果

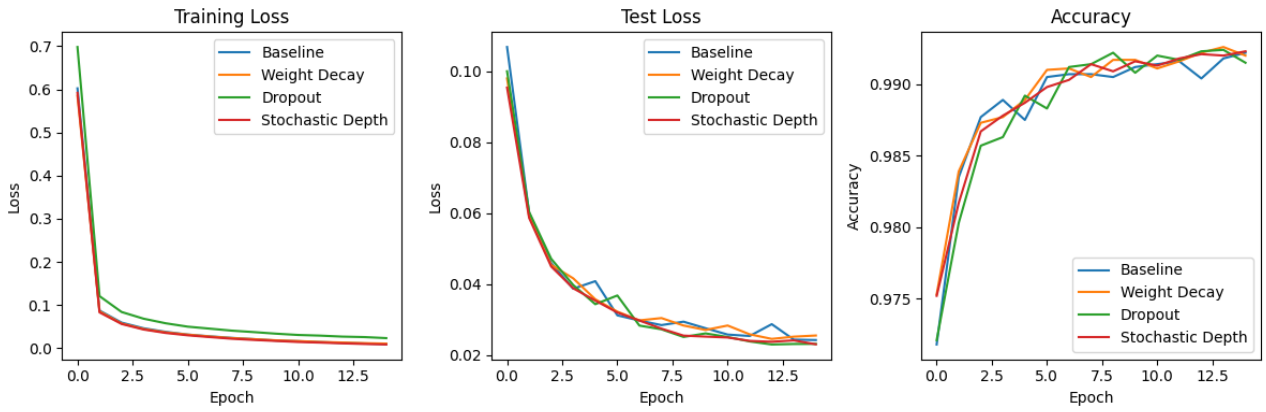

图 1: 使用不同正则化方法对神经网络训练的结果图

表 1: 不同正则化方法对比表

方法	训练损失	测试损失	准确率	稳定性
Baseline	最低	较高	99.22%	中
Weight Decay	中	中	99.22%	中
Dropout	中	较低	99.12%	最佳
Stochastic Depth	中	最低	99.24%	最佳

从上述结果图和对比表可以看出：Weight Decay 方法测试损失（0.0254）略高于 Baseline，可能因 L2 正则化强度不足；Stochastic Depth 方法测试损失最低（0.0230），泛化能力最优；使用 Stochastic Depth 方法，准确率达 99.24%，表现最佳；Dropout 和 Stochastic Depth 的训练曲线较为平滑，表明这两种方法能有效提升训练稳定性。

从三种方法的原理和实际实验结果也可得到：Stochastic Depth 优势在于通过随机跳过网络层，增强模型鲁棒性，显著提升泛化能力；Dropout 优势在于通过随机丢弃神经元，强制网络学习冗余特征，训练过程稳定；而 Weight Decay 因网络模型或参超数原因，或因 L2 正则化强度不足，效果并不是太好。当然这三种方法以及不使用优化方法都在 MNIST 数据集上表现出很好的损失率和测试准确率，想要得到更直观的不同优化方法对网络模型的影响结果，可能需要重新设计网络模型，如更深的卷积层、替换网络模型如使用 RestNet、替换更为复杂，数据量更大的数据集训练。

参考文献

[1] Srivastava N , Hinton G , Krizhevsky A ,et al.Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J].Journal of Machine Learning Research, 2014, 15(1):1929-1958.DOI:10.5555/2627435.2670313.

[2] Huang G , Sun Y , Liu Z ,et al.Deep Networks with Stochastic Depth[J].Springer International Publishing, 2016.DOI:10.1007/978-3-319-46493-0_39.