关于神经网络中过拟合问题的解决方法综述

摘要

一个常见的现象是,我们所训练的神经网络模型在训练集上的表现总是比在验证集上的表现要好。这种现象称为过拟合，通常是由于训练数据量少，而不能满足模型的能力。为了解决这一问题，学者们提出了很多方法包括数据扩充、正则化、随机失活等。本文试图从数据和模型两方面解释过拟合的成因并综合研究解决过拟合的方法，并通过实验验证其性能。

关键词：过拟合，神经网络，数据增强

Abstract

A common phenomenon is that the neural network models we train always perform better on the training set than on the validation set. This phenomenon, known as overfitting, is usually due to the small amount of training data that cannot meet the capabilities of the model. In order to solve this problem, scholars put forward many methods including data expansion, regularization, random inactivation and so on. This paper attempts to explain the causes of overfitting from data and model, and comprehensively study the methods to solve overfitting, and verify its performance through experiments.

Keywords: overfitting, neural network,data augmentation

1 引言

日前，全球掀起了人工智能的热潮。而深度学习[1]作为人工智能最核心的驱动力，促进了语言识别、图像识别与检测以及智能推荐等众多领域的发展。最早提出的深度学习本质上是一种对深层神经网络学习的过程，但受限于当时的各种条件，并没有得到良好发展。然而近年以来，训练数据的增多和硬件能力的增强，使训练较好的深层神经网络模型成为了可能。

在训练深层神经网络模型的过程中，经常会遇到模型过拟合(over fitting[2])的问题。事实上已有早前学者提出预防过拟合问题的方法。李俭川[3]等针对反向传播学习算法及其改进算法中出现的过拟合问题,探讨了三种解决方法 :调整法、提前停止法和隐层节点自生成法 ,并用实例对三种方法进行了验证和比较。Patrice Simard 等人[4]提出在 MNIST 数据集上做各种数据扩增（Data augmentation）以防止过拟合。Hinton 等人[5]提出可以通过阻止特征检测器的共同作用来提高神经网络的性能的方法。Zhang, Hongyi 等人[6]提出了一种与数据无关的数据增强方法 mixup；通过 mixup 构建虚拟样本达到数据扩充的作用，进而抑制过拟合。Szegedy 等人在提出 inception v2 的模型的同时，也提到了一种 label smoothing(标签平滑)[7]的方法，用于解决训练数据不准确对模型泛化性能的影响……尽管解决过拟合方法层出不穷，但是少有能专门论述过拟合问题，以及通过实验具体评价各种方法实际表现的综述类文章。

本文将在现有研究的基础上，综合介绍过拟合的原因以及改善过拟合的策略。本文第二章将重点阐释过拟合的含义以及过拟合的成因。第三章将重点从数据增强和模型约束方面阐

释解决过拟合的方法原理与特性。第四章将通过对比实验的方式，测试各种方法的实际表现。文章最后总结防止过拟合方法的具体使用对策以及面临的挑战。

2 过拟合概念

2.1 过拟合含义

过拟合(overfitting)指的是模型过度拟合训练数据、导致其泛化性能差。具体的表现为：模型在训练集损失值小，准确率高；在验证集和测试集上损失值大，准确率低。而欠拟合(underfitting)则是过拟合的反面，它是指模型未能学习到训练数据的内在关系，具体表现为损失值在训练集上高、在验证集和测试集上也高。

假设用泛化误差$E(f;D)$衡量泛化性能，则有公式(1)：

$$E(f;D) = bias^2(x) + var(x) + \varepsilon^2 \tag{1}$$

其中$bias^2(x)$指偏差（bias），表示期望输出与真实标记的差别；$var(x)$指方差，刻画了数据扰动对模型的影响大小；ε^2指噪声，表示训练样本中不准确样本对模型的干扰。

偏差与训练时间、模型规模呈负相关；随着训练时间的增长、模型规模的提升，模型愈发适应训练数据，期望输出与真实标记差距即偏差逐渐减少。方差与训练时间、模型规模呈正相关；随着训练时间的增长、模型规模的提升，模型对训练数据敏感度提高，样本细微的差异引发模型显著变化，即方差提高。

当模型复杂度足够高、训练时间足够长时，即低偏差、高方差的情况下；模型学习到了一部分训练样本自身的、特殊的，而非整体的、一般的关系时，就发生了过拟合现象；泛化误差、(偏/方)差、(过/欠)拟合、(训练时间/模型规模)这几者的关系如图 1 所示：

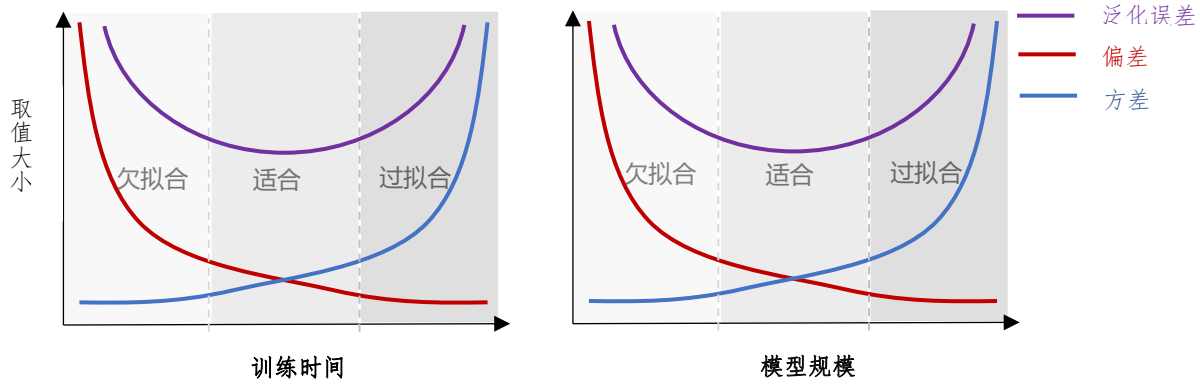

图 1 过拟合相关参数关系图

2.2 过拟合成因

可以从多个角度解释过拟合产生的原因，包括数据本身、模型复杂度、训练情况等；

(1) 数据方面：

- 样本数量少。在此情况下，即使样本足够准确，但是依然会存在过拟合的情况；这是因为少量样本不足以反应特征空间的所有关系。如下图 2(2) 所示
- 样本分布不均，与真实分布存在差距。在分类问题中表现为各个类别样本比例不平

衡；在回归问题中表现为，训练样本集中于特征空间的某个区域。如下图 2(3)所示

- 样本本身不够准确，存在异常或噪声。当模型过度拟合后，噪声也被学习，泛化性能降低。如下图 2(4)所示

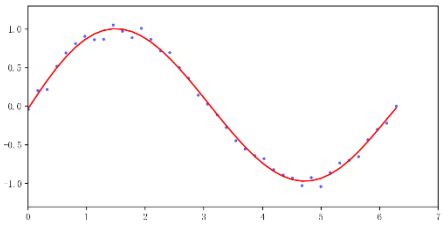

图 2(1) 正常情况

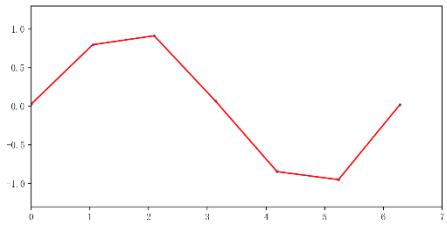

图 2(2) 样本过少

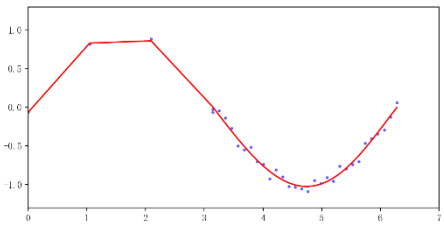

图 2(3) 分布不均

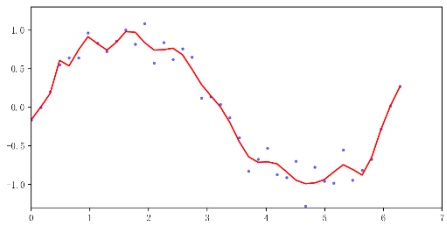

图 2(4) 存在噪声

(2) 模型方面：模型复杂度高，如 2.1 所述。
(3) 训练情况：模型被过度训练，如 2.1 所述。

3 解决过拟合的途径

针对第二节列举的过拟合的各种成因，本节将先从数据增强、模型约束两个方向介绍改善过拟合问题的方法，此部分涵盖了学术界至今为止大部分主流方法。另外对于没有涉及到的方法，也会在本节末尾补充。

3.1 数据增强

针对 2.2 中提及的训练样本少、分布不均匀、存在噪声三大数据方面的问题；在训练过程中，为了有效防止过拟合，我们自然期望训练样本数量多、分布均匀且足够准确；但是事与愿违，真实样本在数量、分布以及准确性上往往存在着的不小的缺陷。此时需要人为构造一些“假”数据，此节将从数据扩充和数据抗噪两方面介绍从数据角度应对过拟合的方法。

3.1.1 数据扩充

由于分布不均匀本质上可以通过扩充稀缺类别的样本来解决，所以两者都可以通过数据扩充的方式解决，数据扩充方法根据单次参与运算的样本数量可划分为单样本变换法和多样本合成法。

3.1.1.1 单样本变换法

单样本变换法指的是通过单一的样本变换得到新样本，从而达到数据扩充的目的。此方法主要应用在图像增强中，常见的图像增强方法包括：翻转、裁切、旋转、缩放、仿射变换、视觉变换、噪声、模糊、随机擦除等方法；其中一部分方法效果如下图 3 所示：

图 3(1) 原图像

图 3(2) 仿射变换

图 3(3) 高斯模糊

图 3(4) 高斯噪声

图 3(5) 剪裁

图 3(6) 水平翻转

图 3(7) 随机擦除

图 3(8) 缩放

图 3(9) 旋转

3.1.1.2 多样本合成法

单样本变换法虽然能增大样本数量，但是其扩充的数据与原数据存在极大的相似性，并不能从根本上解决有效样本少的问题；另外此方法目前还仅适用于图像，存在不通用的弊端。而多样本合成法则解决了上述问题，多样本合成法指多个样本线性组合、生成新的样本。多样本合成法包括 SMOTE[8]、SamplePairing[9]、mixup 三种具体方法。

(1) SMOTE

SMOTE 专门通过样本合成解决样本分布不平衡的问题。其算法核心在于：对于小样本中每一样本(x_a, y_a)，计算欧式距离得其k近邻；在k近邻中随机选取一个样本(x_b, y_b)，则根据公式(2)计算得到新样本点(x_{new}, y_{new})

$$(x_{new}, y_{new}) = (x_a, y_a) + rand(0,1) * (x_b - x_a, y_b - y_a) \quad (2)$$

后不断采样和计算新样本，直到各类别数量相同。

(2) SamplePairing

SamplePairing 方法是多样本合成法中最简单直接的。它的主要算法流程入下图所示:

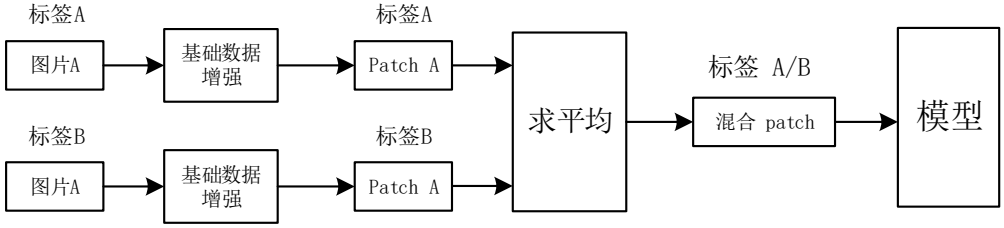

图 4 SamplePairing 算法流程示意图

在训练集中随机选取两个样本$(x_a,y_a),(x_b,y_b)$，通过简单的基础数据增强方法得到$(x_a',y_a),(x_b',y_b)$后直接求平均，而标签取其中之一。如公式(3)(4)：

$$x_{new}=\frac{x_a'+x_b'}{2} \tag{3}$$

$$y_{new}=y_a \tag{4}$$

在实际训练过程中，完成一定轮次迭代后间歇式禁止 SamplePairing；当损失值降低到一定值时，禁止 SamplePairing 完成后续精细化训练。

(3) Mixup

Mixup 是 SamplePairing 的进一步升级，前者相较于后者做了两个方面的改动；一方面特征向量x不再简单相加求平均，而是通过一组参数控制融合程度；另一方面标签 y 也随着特征向量 x 相融合。算法主要流程是，在训练集中随机选取两个样本$(x_a,y_a),(x_b,y_b)$,根据公式(5)(6)计算得到新样本(x_{new},y_{new}):

$$x_{new}=\lambda*x_a+(1-\lambda)*x_b \tag{5}$$

$$y_{new}=\lambda*y_a+(1-\lambda)*y_b \tag{6}$$

其中$\lambda\sim Beta(\alpha,\alpha)$取值范围介于 0-1 之间，超参数$\alpha$控制样本融合程度，当$\alpha\to 0$，样本不融合。Mixup 方法通过线性插值的方式大大扩充了样本数量，有效解决了样本不足的问题。

3.1.1.3 数据扩充方法小结

单样本变换法通过图像变换的手段扩充训练样本，本质是在特征空间中样本点的小领域内重复采样。而多样本合成法则可视为在样本之间线性插值，虽然合成的新样本不具备现实意义，缺乏解释性，甚至扰乱正常训练；但是客观上增大了训练样本的数量，从实践来看提高了泛化性能，总体上改善了过拟合的现象。

3.1.2 数据抗噪

基本解决了训练样本不足的问题后，样本存在噪声又该如何应对呢？最直接的方法是，找出异常样本并丢弃；然而现实情况是，我们往往既缺乏手动筛选数据的精力，又缺少分辨数据好坏的能力。能否在缺少先验知识的情况下，通过一种数据改造的方法提高数据的鲁棒性呢？

Label smoothing 使用了一种标签平滑的策略成功地达到此目的；对于分类问题，一个常规做法是将标签 one-hot 编码，并用交叉熵计算损失函数；假设有K个类别，样本x的真实标签为q、预测输出为p，则对于单个样本的损失函数计算公式如(7)所示：

$$loss = -\sum_{k=1}^{K} q(k|x)\log\left(p(k|x)\right) \tag{7}$$

机器学习实际上是一个不断让$p(k)$逼近$q(k)$的过程，当$p(x)$与$q(k)$一致时损失为 0；否则随着两者差距的提升趋向于无穷大；这引发了一个问题，当数据存在异常或者噪声时；模型强行拟合异常点会引发过拟合问题。

Label smoothing 做了以下变动，将原始 one-hot 编码后的标签$q(k|x)$与一种与样本无关的分布$u(k)$线性组合，得到新标签$q'(k|x)$，如公式(8)所示：

$$q'(k|x) = (1-\varepsilon) * q(k|x) + \varepsilon * u(k) \tag{8}$$

ε通常取得很小，也就相当于在原标签上增加一微小的惩罚因子；$u(k)$一种取法是第 k 个类别的先验概率。Label smoothing 一定程度上减小数据噪声对模型的影响；提高了模型的泛化性能。

3.2 模型约束

模型的复杂度过高也是导致过拟合问题的一大因素；但从限制模型规模角度，减少神经网络层数、降低神经元数量也是降低模型规模从而改善过拟合问题的有效手段。但过于删减可能会限制模型能力，使得模型不足以适应复杂数据而导致另外一个问题——欠拟合。此节将介绍两种隐式限制模型规模的方法，分别是 L2 正则化、dropout。

3.2.1 L2 正则化

L2 正则化是在原来损失函数上，增加 L2 正则化项；记L_0为原始损失，L为更新后损失，λ为正则化系数，w为权重大小，n为样本数量；则 L2 正则化公式如(9)所示：

$$L = L_0 + \frac{\lambda}{2n} * \sum w^2 \tag{9}$$

将上式求w偏导得公式(10)：

$$\frac{\partial L}{\partial w} = \frac{\partial L_0}{\partial w} + \frac{\lambda}{n} w \tag{10}$$

因此，w将以公式(11)更新：

$$w \to \left(1 - \frac{\lambda}{n}\right) w - \frac{\partial L_0}{\partial w} \tag{11}$$

不难发现，w随迭代次数逐渐减少，称作权值衰减；权值越小，模型复杂度也相应减小。此外，L1 正则化也能达到相似的效果；与 L2 不同的是，L1 正则化的惩罚因子为$|w|$，更适合作为特征选择器使用。

3.2.2 dropout

Dropout 随机失活指的是网络中神经元将以一定概率失去作用，使得网络不严重依赖于某个特定的神经元，增加了模型的泛化性能。对于不采用 dropout 的网络，前向传播如公式(12)(13)所示：

$$z^{l+1} = w^{l+1}y^{l} + b^{l+1} \tag{12}$$

$$y^{l+1} = f(z^{l+1}) \tag{13}$$

其中上标l表示第l层网络参数，z^{l+1}、y^{l+1}层分别为第$l+1$层网络的未经激活、和激活后的的输出。而增加了 dropout 过程后，网络以公式 14~17 传递：

$$r^{l} \sim Bernouli(p) \tag{14}$$

$$\tilde{y}^{l} = r^{l} * y^{l} \tag{15}$$

$$z^{l+1} = w^{l+1}\tilde{y}^{l} + b^{l+1} \tag{16}$$

$$y^{l+1} = f(z^{l+1}) \tag{17}$$

与原始网络相比，dropout 先生成以p为参数伯努利分布的随机向量，接着与上层输出相乘得到实际输入；在此过程中，约有占比为p的神经元被置 0，从而失去作用。

对于 Dropout，可以从多种角度理解。在训练过程中，由于随机因素的存在，数据的局部特征不容易被传播；网络对数据的学习转而向更全局、更共性、更鲁棒的方向发展，这大大提升了模型的泛化性能。另外在实际预测中，由于存在神经元失活这个不确定因素；网络预测并不依赖单一的、特定的神经元，而是共同决策，这实际上包含了集成学习的思想。最后从生物进化学角度来说，环境突变（指随机失活）将导致优胜劣汰；这促使每个神经元都往最优的情况发展。

3.2 其他方法

针对 2.2 提及的过度训练问题，可以使用早停法（early-stop [10]），提早结束模型训练；但在实际操作中，提前停止的时间很难把握；过早停止模型的训练会使模型不够拟合训练样本，导致欠拟合；一种解决方法是设置回调函数监测每轮训练情况，保存至今为止最优的参数。

此外集成学习[11]，包括 bagging[12]、boosting [13]等；通过综合多个模型的意见，能有效防止一两个模型过拟合的问题。由于属于通用机器学习方法，此文不展开赘述。

4 实验验证

4.1 实验设置

本次实验将依旧从数据增强和模型约束两个方向分别进行;实验的第一个阶段先从各个方向内部选取代表性的算法进行横向比较,第二个阶段再根据第一阶段的成绩选取部分纵向对比，并尝试将两个方向的算法融合，以探究其实际性能。

本次实验将选取经典数据集 cifar-10 作为本次实验的研究对象，它是由 Alex Krizhevsky 等人收集，包含 50000 张训练图像，10000 张测试图像，共有 10 个类别。本次实验所有算法都将在数据集上训练 100 个 epoch,每次迭代的数据量相同，以保证每个算法总共迭代相同数量的数据；同时每个 epoch 结束时，记录 loss 和 val-loss。

4.2 实验结果

4.2.1 第一阶段

第一组实验考察数据增强方面的相关算法，选取了最具代表性的基础数据增强方法（记为 simplea）、mixup 方法、labelsmoothing 方法作为实验组，另外设置一组不采用任何增强方法作为对照组（记为 noa）。基础数据增强选用了随机旋转(40° 以内)、水平垂直移动(比例在 0.2 以内)、水平翻转；mixup 中 λ 设定为默认值 0.2；labelsmoothing 中 ε设定为默认 0.1；实验结果如图 5(1)、图 5(2)：

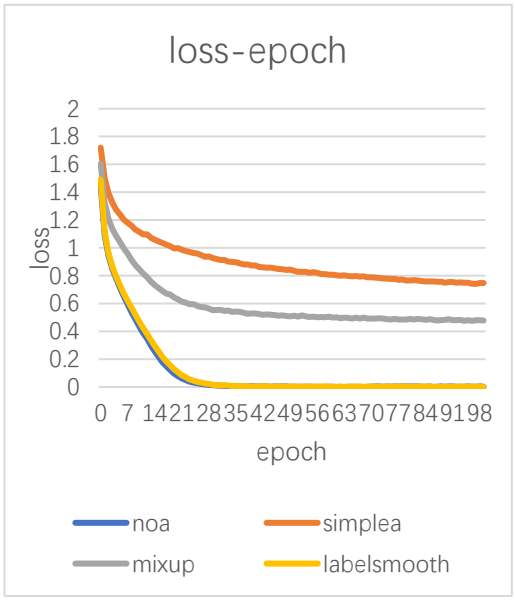

图 5(1) 数据增强方法 loss-epoch 图

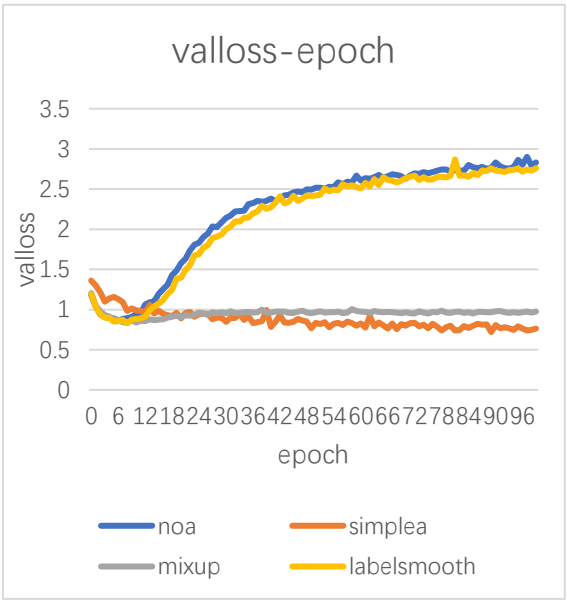

图 5(1) 数据增强方法 valloss-epoch 图

从实验结果上看基础增强方法和 mixup 方法都取得了不错的效果，基础增强方法在验证集上损失值最小，可谓泛化性能最佳，这也和本次实验使用了多种基础增强的手段有关；mixup 略有逊色，不过泛化效果提升依旧明显，并且相比基础数据增强在训练集拟合效果更佳；而未做任何增强操作的对照组发生了非常严重的过拟合现象；labelsmoothing 在本次实验中对泛化性能的提升并不明显，这可能和训练集本身精度高、样本类别过少有关系。

第二组实验考查模型约束相关方面的算法，选取了 12 正则化、dropout 方法；分别对 12 正则化的超参数 λ 取 0.1 和 0.25(分别记为 12-01 和 12-025)，对 dropout 中超参数 p 取 0.1 和 0.25(分别记为 12-01 和 12-025)，同时设置对照组作为实验参照。实验结果如图 6(1)、图 6(2)所示：

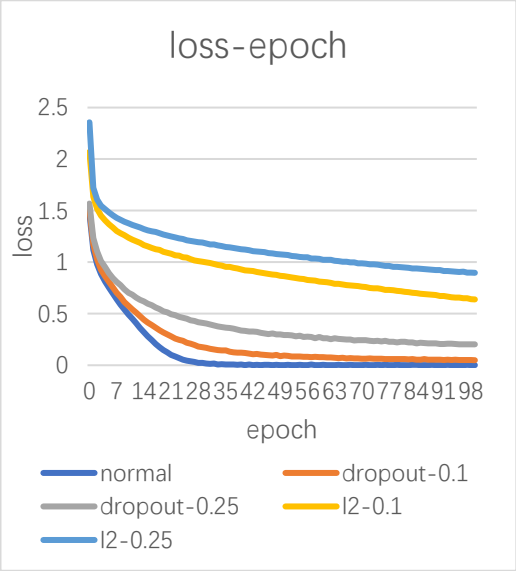

图 6（1）模型约束方法 loss-epoch 图

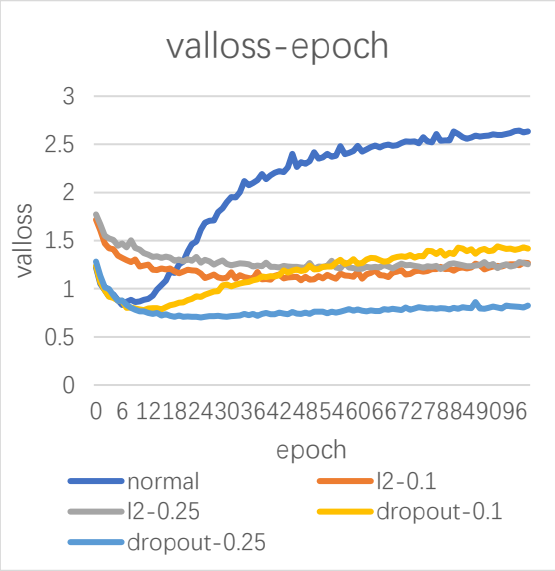

图 6（2）模型约束方法 valloss-epoch 图

从实验结果上看，无论是 l2 正则化还是 dropout 都有效地改善了正则化现象，l2 正则化超参数的取值对验证集结果影响不大。dropout-0.25 表现最佳，且领先优势十分明显；dropout-0.1 与 dropout-0.25 差异明显。最后无论何种算法，适当增加模型约束的程度都会导致训练误差的增大，而验证误差则会相应减小。

4.2.2 第二阶段

此阶段选择第一阶段表现优异的算法进行综合比较，以探究解决过拟合的两大方向是否存在明显的优劣之分；此阶段只进行 valloss 的对比，结果如图 7 所示：

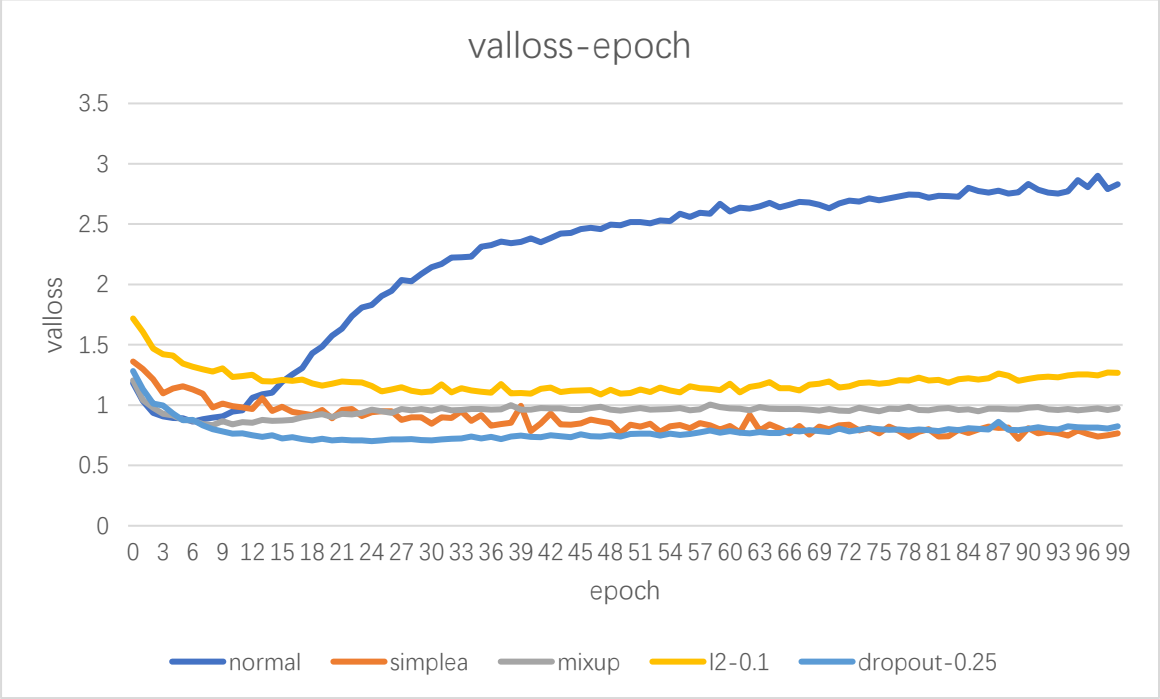

图 7 纵向对比 valloss-epoch 图

实验结果表明,数据增强和模型约束对解决过拟合问题具有相似的效果,可谓殊途同归,并不存在明显的优劣之分。也可看出，在 cifar-10 数据集上，dropout 和基础数组增强相较于其他方法而言具有最好的泛化效果；mixup 方法和 l2 正则化次之，不过相比于未经过任何处理的对照组提升依旧明显。

最后我们尝试将两个方向的最优算法融合，既采用基础数据增强、又采用 dropout 方法随机失活，以探究其泛化性能是否更加出色，实验结果如图 8：

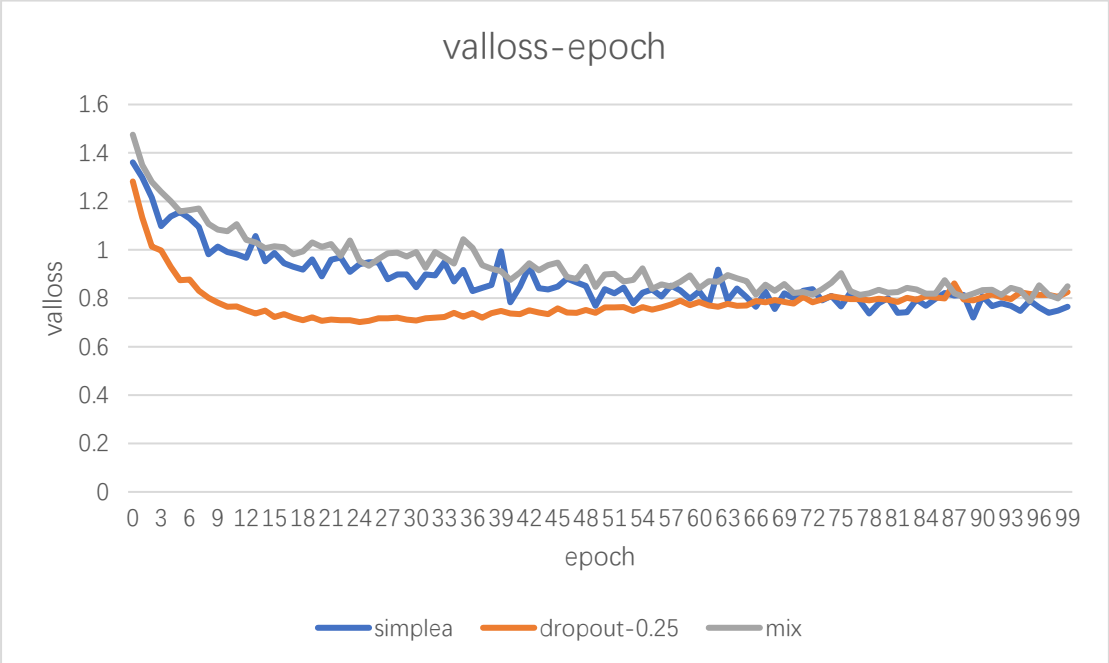

图 8 融合情况对比图

实验结果表明，简单将两个方向的方法融合在此次实验中并不能产生更优的表现。

4.3 实验总结

此次实验结果基本验证了理论假设的正确性,上述方法均在不同程度上改善了过拟合问题;数据增强和模型约束均是应对过拟合现象的有效手段,且这两个方向并不存在明显优劣。训练误差和验证误差往往不可兼得，防止过拟合需要保证训练不能过分低；dropout 和基础数据增强在此次实验中表现优异，但是并不能说明其一定优于其他方式。

5 总结与展望

在介绍完过拟合的含义、成因和解决方式,以及通过实验实际验证过拟合的解决方法后,我们对过拟合问题有了更全面、更深刻的认识。需要注意的是，因为样本空间的不同，应对过拟合问题的方法也不是一层不变的；我们往往需要根据实际问题，广泛尝试各种方法，以选择更适合的应对策略。

事实上，机器学习的发展始终绕不开过拟合这个问题；但从分类器的发展上看，从最初的逻辑回归，发展到支持向量机，最后发展到神经网络；模型的能力一步步提升，过拟合问题也愈发突出。而自从跨入深度学习以来，少量样本和复杂模型便成为了更加尖锐的矛盾。研究人员提出各种改善过拟合的方法，但没有一种方法能真正解决这个问题。这是因为，真

实世界的对象再被数字化后，本身就不可知；神经网络可以看做是人们创造出来探索数字化世界的工具；然而人们无论对数据蕴含的规律还是对网络的内在逻辑都知之甚少。因为没有真正了解，所以也不能真正解决数据和模型的矛盾。因此，人们需要从更高维度上真正解开数据之谜，我们期待这天的早日到来。

参考文献

[1] Lecun Y , Bengio Y , Hinton G . Deep learning.[J]. Nature, 2015, 521(7553):436.

[2] Schaffer C . Overfitting Avoidance as Bias[J]. Machine Learning, 1970, 10(2):153-178.

[3]李俭川, 秦国军, 温熙森, et al. 神经网络学习算法的过拟合问题及解决方法[J]. 振动、测试与诊断, 2002(04):16-20+76.

[4] Simard P , Steinkraus D , Platt J C . Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis[C]// 7th International Conference on Document Analysis and Recognition (ICDAR 2003), 2-Volume Set, 3-6 August 2003, Edinburgh, Scotland, UK. IEEE Computer Society, 2003.

[5] Hinton G E , Srivastava N , Krizhevsky A , et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science, 2012.

[6] Zhang H , Cisse M , Dauphin Y N , et al. mixup: Beyond Empirical Risk Minimization[J]. 2017.

[7] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818－2826, 2016.

[8] Chawla N V , Bowyer K W , Hall L O , et al. SMOTE: Synthetic Minority Over-sampling Technique[J]. Journal of Artificial Intelligence Research, 2002, 16(1):321-357.

[9] Inoue H . Data Augmentation by Pairing Samples for Images Classification[J]. 2018.

[10] Prechelt L . Early Stopping - But When?[J]. neural networks tricks of the trade.

[11] 周志华. 机器学习. 北京：清华大学出版社，2016 年

[12] Leo Breiman. Bagging Predictors[J]. Machine Learning, 1996, 24(2):123-140.

[13] Bauer E , Kohavi R . An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants[J]. Machine Learning, 1999, 36(1-2):105-139.