

从数据空间变换和集成算法角度浅谈神经网络

田青¹, 田广军²

中文摘要

本文首先从数据空间变换的角度解读(深度)神经网络的作用, 即将数据从复杂的原始空间转换到简单可分的空间(比如线性空间)。将神经网络与日常生活中的折纸相联系。通俗易懂地诠释了神经网络在数据空间变换和分类中的作用。除此之外, 本文还探讨了神经网络与集成算法的相似性。

English title: Random thoughts on neural networks from the views of data space transformation and ensemble classification

Authors: Qing Tian¹, Guangjun Tian²

English abstract

This manuscript sketch first describes neural networks' effect from the perspective of data space transformation, which is transforming data in a complicated raw space into an easily (e.g. linearly) separable space. We use a simple paper wrapping example to illustrate this point. In addition, this sketch also discusses some similarities between neural networks and ensemble classification.

[中文正文]

近年来, 深度学习神经网络技术在人类生活的各个领域得到越来越多的青睐, 然而, 许多初学者觉得高深莫测。怎样通俗地理解深度神经网络在目标分类方面的功用呢? 我们在本文从数据空间变换和集成算法角度给出我们对(深度)神经网络的解读, 仅供初学者参考。

一、神经网络的空间变换作用

众所周知, 无论是时间、空间还是数值域空间, 线性边界是最简单的边界, 很多传统的分类算法都依赖线性假设。然而现实当中数据很少是线性可分的。学习神经网络无非就是学习如何把数据转变到简单的(尽可能的)线性空间。在此我们不妨把神经网络学习与简单的折纸过程联系起来理解。假设空间分布有两种不同的样本点, 如图 1(a)所示, 分别用圆圈和方块标识, 显然用一个线性分类器是不可能把两个类完美分开的。但是, 对这个简单示例, 我们可以通过图 1(a)~(d)逐次折叠降低非线性, 最后通过在适当位置进行直线(线性)剪纸式分割, 实现了不同标识样本点的完全分类。神经网络的学习可以简单理解为学习如何折

纸从而把线性不可分数据点转换到线性可分空间的过程。不管是一般的全连接网络还是卷积神经网络，每一层的权重或卷积滤波器决定了对当前数据空间的投影。在一个训练好的网络中，它们保证本层折叠沿着正确路径方案执行。神经网络的复杂度，比如层数，直接影响到我们折纸的自由度和灵活度。更多的层数意味着更复杂的折纸变换可能。在当今物联网和数据爆炸的时代，要完成复杂数据的‘完美’分割，我们需要更复杂的折纸方法,这就是我们需要更多层数的原因。深度神经网络的名称就应运而生了。深度学习无非是学习更多层的神经网络罢了。当然，这个看似简单的过程在过去很长的一段时间并不容易实现。近年来更多的计算力（如 GPU）以及有效的训练方法对训练深层神经网络起着至关重要的作用。

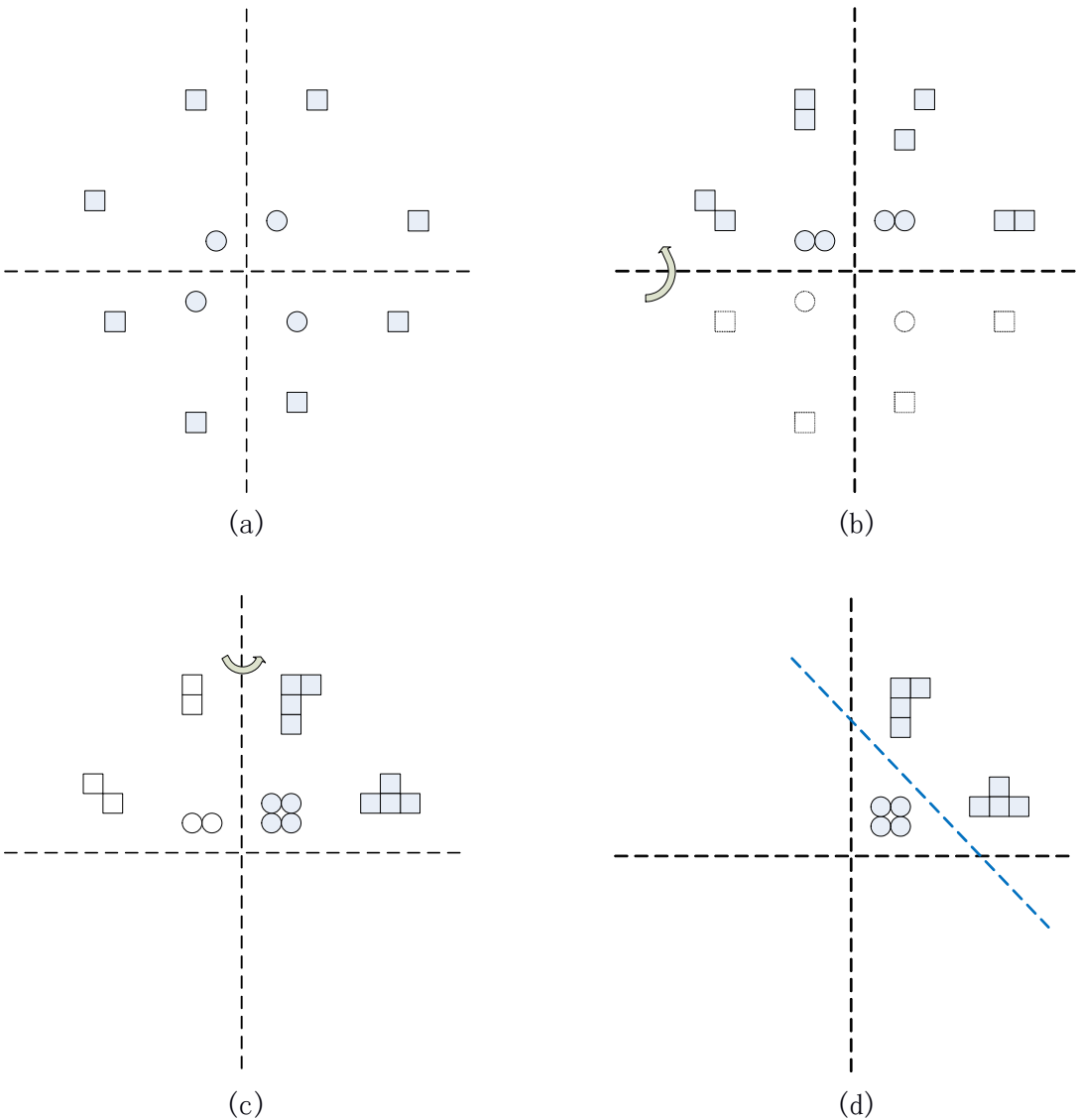


图 1. 神经网络的空间变换作用

相比之下，传统手工制作 (handcrafted) 方法就像是按照一些约定俗成的或是专家方案进行折纸。当数据像图 1 那样简单的时候，人类很容易根据自己经验尝试不同的折纸方式，对

数据进行完美分类。但是在复杂的大数据时代，由于实际问题中数据空间的多样性和折纸过程的复杂性，按照一些人类手工的折纸方法往往比不过机器学习来的方法。如图 2 所示为一个卷积神经网络迭代训练过程曲线，对应机器学习折纸反复尝试(trial-and-error)的过程。卷积神经网络一直在快速精细地调整和寻找最佳‘折叠’位置，探索通达目标的演进路径。

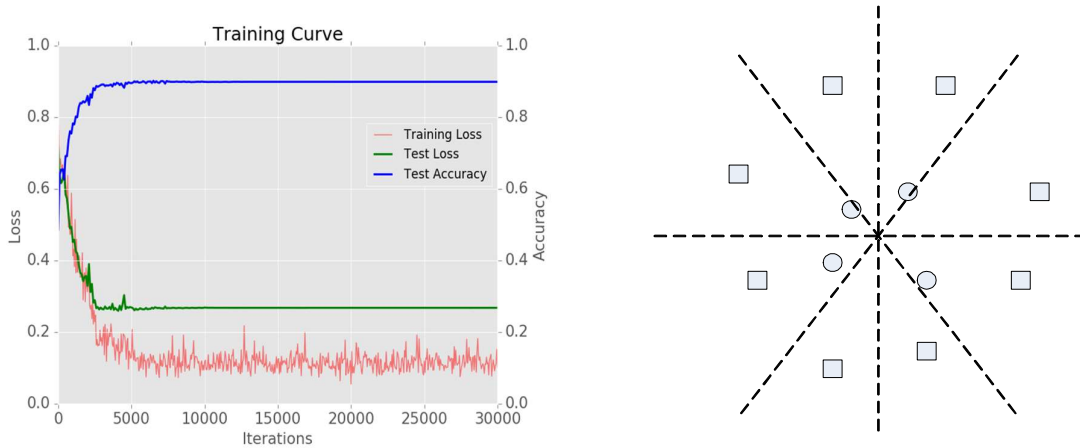


图 2. 学习网络参数 v.s. 学习如何折纸

二、神经网络和集成算法的相似性

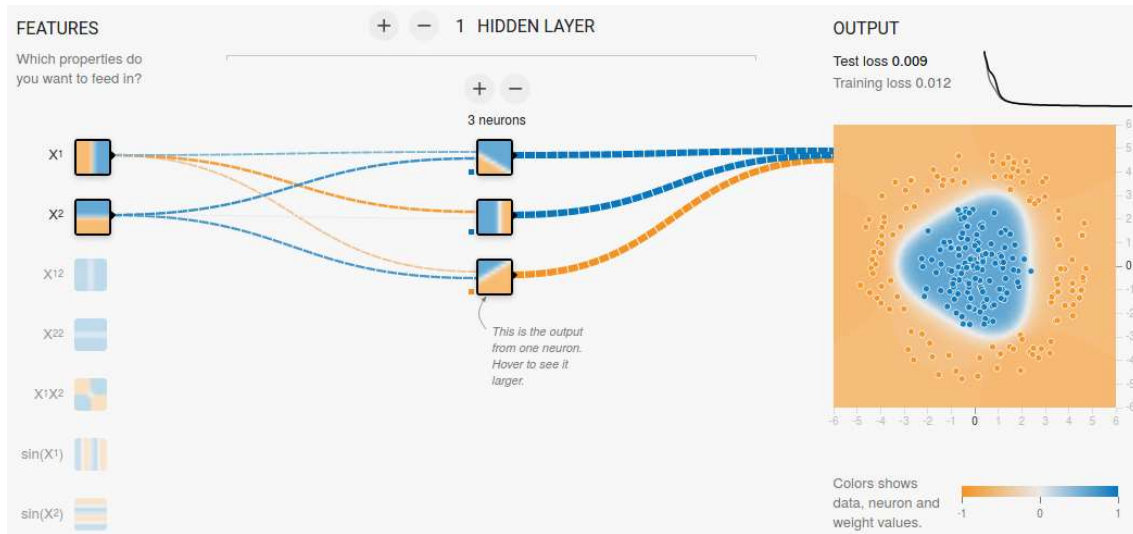


图 3. 神经网络，一种特殊的集成分类器

神经网络可以看作是分层次的集成分类器。在每一层，每个神经元都可以被视为一个有特定分工的个体学习器（或基学习器）。与平均法、投票法等传统结合策略不同，所有同层神经元的结合方式由下一层的神经元权重决定，而这些权重又由学习而来。因此，我们可以把这种结合方法看成是学习集成法中的 stacking 方法。这里我们不是对基学习器的结果做简单的人为逻辑处理，而是再加上另一层学习器。图 3 是一个典型的例子（实验平台：[1]）。

在隐藏层中我们可以清楚的看到三个神经元的作用。由于第三个神经元对下一层贡献为负，所以可以颠倒其空间划分（变成左上橙色，右下蓝色）。这样的话中间的蓝色区域相当于三个空间划分的交集：只有三个空间划分都是蓝色的中间区域才是蓝色。在这个简单例子中，从集成算法的角度分析甚至可以让我们知道针对这个问题最佳网络结构。在这个例子中我们只需要一层隐藏神经网络，三个神经元。

当然实际问题要复杂的很多，本短文只是作者的随笔，未完待续。

References

1. <https://playground.tensorflow.org>