

训练卷积神经网络： 什么是机器学习？——第二部分

Ole Dreessen，现场应用工程师

摘要

本文是系列文章的第二部分，重点介绍卷积神经网络(CNN)的特性和应用。CNN主要用于模式识别和对象分类。在第一部分文章《卷积神经网络简介：什么是机器学习？——第一部分》中，我们比较了在微控制器中运行经典线性规划程序与运行CNN的区别，并展示了CNN的优势。我们还探讨了CIFAR网络，该网络可以对图像中的猫、房子或自行车等对象进行分类，还可以执行简单的语音识别。本文重点解释如何训练这些神经网络以解决实际问题。

神经网络的训练过程

本系列文章的第一部分讨论的CIFAR网络由不同层的神经元组成。如图1所示， 32×32 像素的图像数据被呈现给网络并通过网络层传递。CNN处理过程的第一步就是提取待区分对象的特性和结构，这需要借助滤波器矩阵实现。设计人员对CIFAR网络进行建模后，由于最初无法确定这些滤波器矩阵，因此这个阶段的网络无法检测模式和对象。

为此，首先需要确定滤波器矩阵的所有参数，以最大限度地提高检测对象的精度或最大限度地减少损失函数。这个过程就称为神经网络训练。本系列文章的第一部分所描述的常见应用在开发和测试期间只需对网络进行一次训练就可以使用，无需再调整参数。如果系统对熟悉的对象进行分类，则无需额外训练；当系统需要对全新的对象进行分类时，才需要额外进行训练。

进行网络训练需要使用训练数据集，并使用类似的一组测试数据集来测试网络的精度。例如CIFAR-10网络数据集为十个对象类的图像集合：飞机、汽车、鸟、猫、鹿、狗、青蛙、马、轮船和卡车。我们必须在训练CNN之前对这些图像进行命名，这也是人工智能应用开发过程中最为复杂的部分。本文讨论的训练过程采用反向传播的原理，即向网络连续展示大量图像，并且每次都同时传送一个目标值。本例的目标值为图像中相关的对象类。在每次显示图像时，滤波器矩阵都会被优化，这样对象类的目标值就会和实际值相匹配。完成此过程的网络就能够检测出训练期间从未看到过的图像中的对象。

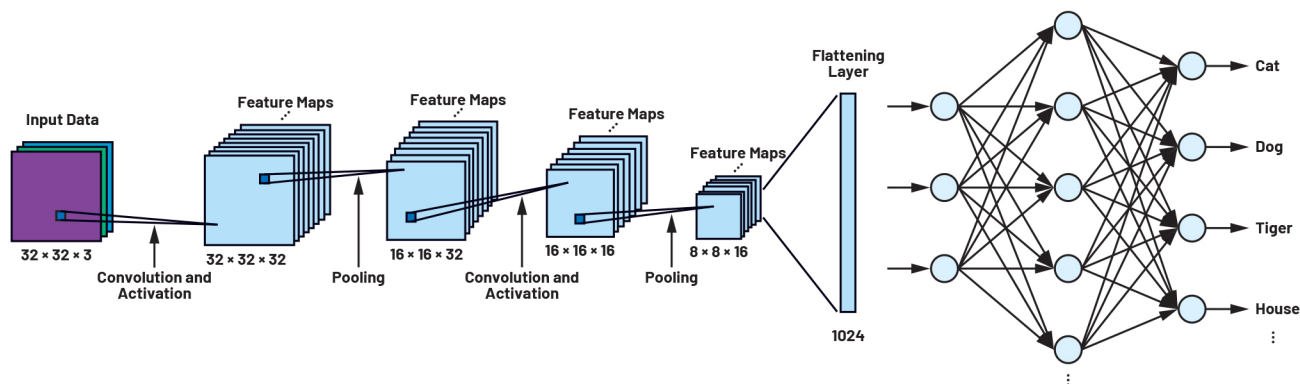


图1. CIFAR CNN架构。

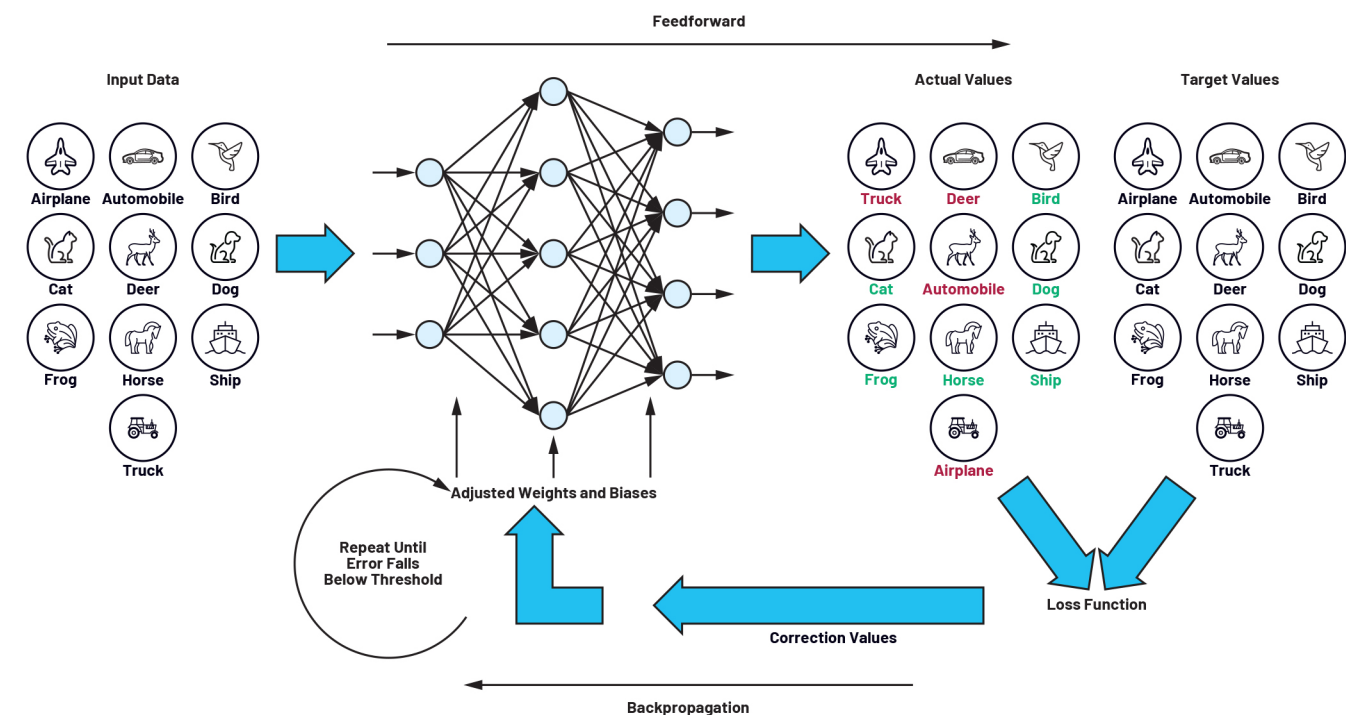


图2. 由前向传播和反向传播组成的训练循环。

过拟合和欠拟合

在神经网络的建模过程中经常会出现的问题是：神经网络应该有多少层，或者是神经网络的滤波器矩阵应该有多大。回答这个问题并非易事，因此讨论网络的过拟合和欠拟合至关重要。过拟合由模型过于复杂以及参数过多而导致。我们可以通过比较训练数据集和测试数据集的损失来确定预测模型与训练数据集的拟合程度。如果训练期间损失较低并且在向网络呈现从未显示过的测试数据时损失过度增加，这就强烈表明网络已经记住了训练数据而不是在实施模式识别。此类情况主要发生在网络的参数存储空间过大或者网络的卷积层过多的时候。这种情况下应当缩小网络规模。

损失函数和训练算法

学习分两个步骤进行。第一步，向网络展示图像，然后由神经元网络处理这些图像生成一个输出矢量。输出矢量的最大值表示检测到的对象类，例如示例中的“狗”，该值不一定是正确的。这一步称为前向传播。

目标值与输出时产生的实际值之间的差值称为损失，相关函数则称为损失函数。网络的所有要素和参数均包含在损失函数中。神经网络的学习过程旨在以最小化损失函数的方式定义这些参数。这种最小化可通过反向传播的过程实现。在反向传播

的过程中，输出产生的偏差（损失 = 目标值-实际值）通过网络的各层反馈，直至达到网络的起始层。

因此，前向传播和反向传播在训练过程中产生了一个可以逐步确定滤波器矩阵参数的循环。这种循环过程会不断重复，直至损失值降至一定程度以下。

优化算法、梯度和梯度下降法

为说明训练过程，图3显示了一个包含x和y两个参数的损失函数的示例，这里z轴对应于损失。如果我们仔细查看该损失函数的三维函数图，我们就会发现这个函数有一个全局最小值和一个局部最小值。

目前，有大量数值优化算法可用于确定权重和偏差。其中，梯度下降法最为简单。梯度下降法的理念是使用梯度算子在逐步训练的过程中找到一条通向全局最小值的路径，该路径的起点从损失函数中随机选择。梯度算子是一个数学运算符，它会在损失函数的每个点生成一个梯度矢量。该矢量的方向指向函数值变化最大的方向，幅度对应于函数值的变化程度。在图3的函数中，右下角（红色箭头处）由于表面平坦，因此梯度矢量的幅度较小。而接近峰值时的情况则完全不同。此处矢量（绿色箭头）的方向急剧向下，并且由于此处高低差明显，梯度矢量的幅度也较大。

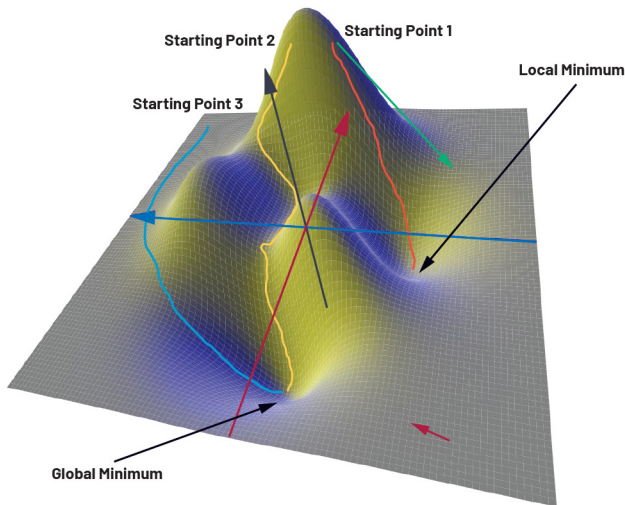


图3. 使用梯度下降法确定到最小值的不同路径。

因此我们可以利用梯度下降法从任意选定的起点开始以迭代的方式寻找下降至山谷的最陡峭路径。这意味着优化算法会在起点计算梯度，并沿最陡峭的下降方向前进一小步。之后算法会重新计算该点的梯度，继续寻找创建一条从起点到山谷的路径。这种方法的问题在于起点并非是提前定义的，而是随机选择的。在我们的三维地图中，某些细心的读者会将起点置于函数图左侧的某个位置，以确保路径的终点为全局最小值（如蓝色路径所示）。其他两个路径（黄色和橙色）要么非常长，要么终点位于局部最小值。但是，算法必须对成千上万个参数进行优化，显然起点的选择不可能每次都碰巧正确。在具体实践中，这种方法用处不大。因为所选择的起点可能会导致路径

（即训练时间）较长，或者目标点并不位于全局最小值，导致网络的精度下降。

因此，为避免上述问题，过去几年已开发出大量可作为替代的优化算法。一些替代的方法包括随机梯度下降法、动量法、AdaGrad方法、RMSProp方法、Adam方法等。鉴于每种算法都有其特定的优缺点，实践中具体使用的算法将由网络开发人员决定。

训练数据

在训练过程中，我们会向网络提供标有正确对象类的图像，如汽车、轮船等。本例使用了已有的CIFAR-10数据集。当然，在具体实践中，人工智能可能会用于识别猫、狗和汽车之外的领域。这可能需要开发新应用，例如检测制造过程中螺钉的质量必须使用能够区分好坏螺钉的训练数据对网络进行训练。创建此类数据集极其耗时费力，往往是开发人工智能应用过程中成本最高的一步。编译完成的数据集分为训练数据集和测试数据集。训练数据集用于训练，而测试数据则用于在开发过程的最后检查训练好的网络的功能。

结论

本系列文章的第一部分《人工智能简介：什么是机器学习？——第一部分》介绍了神经网络并对其设计和功能进行了详细探讨。本文则定义了函数所需的所有权重和偏置，因此现在可以假定网络能够正常运行。在后续第三部分的文章中，我们将通过硬件运行神经网络以测试其识别猫的能力。这里我们将使用ADI公司开发的带硬件CNN加速器的MAX78000人工智能微控制器来进行演示。



作者简介

Ole Dreessen是ADI公司的现场应用工程师，于2014年加入ADI公司，此前曾在Avnet Memec和Macnica任职，负责支持通信产品和高性能微处理器。Ole在微控制器和安全方面拥有广泛的专业知识，拥有丰富的会议主讲经验。在业余时间，他是混沌计算机俱乐部的活跃成员，主要研究逆向工程和嵌入式安全等概念。

