深度学习综述

摘要

深度学习是机器学习和人工智能研究的最新趋势之一。它也是当今最流行的科学研究趋势之一。深度学习方法为计算机视觉和机器学习带来了革命性的进步。新的深度学习技术正在不断诞生，超越最先进的机器学习甚至是现有的深度学习技术。近年来，全世界在这一领域取得了许多重大突破。由于深度学习正快度发展，导致了它的进展很难被跟进，特别是对于新的研究者。在本文中，我们介绍了四种主流深度学习模型，分别是深度前馈网络，卷积神经网络，循环神经网络，生成对抗网络。最后本文介绍了神经网络训练技术。

关键词：深度学习，神经网络，卷积神经网络，循环神经网络，生成对抗神经网络

Overview of deep lerning

Abstract

Deep learning is one of the latest trends in machine learning and artificial intelligence research. It is also one of the most popular scientific research trends today. The deep learning approach has revolutionized computer vision and machine learning. New deep learning technologies are constantly being born, surpassing state-of-the-art machine learning and even existing deep learning techniques. In recent years, the world has made many major breakthroughs in this field. Due to the rapid development of deep learning, its progress is difficult to follow, especially for new researchers. In this paper, we introduce four mainstream deep learning models, namely deep feedforward networks, convolutional neural networks, and cyclic neural networks to generate confrontational networks. Finally, this paper introduces neural network training techniques.

Key words:deep learning,CNN,RNN,GAN

分工：孙天柠（引言和相关研究+CNN） 王伟华（前馈神经网络+RNN） 王起帆（GAN+神经网络训练技术）

**1.引言**

深度学习（DL）一词最初在 1986 被引入机器学习（ML），后来在 2000 年时被用于人工神经网络（ANN）。深度学习方法由多个层组成，以学习具有多个抽象层次的数据特征。DL 方法允许计算机通过相对简单的概念来学习复杂的概念。对于人工神经网络（ANN），深度学习（DL）（也称为分层学习（Hierarchical Learning））是指在多个计算阶段中精确地分配信用，以转换网络中的聚合激活。为了学习复杂的功能，深度架构被用于多个抽象层次，即非线性操作；例如 ANNs，具有许多隐藏层。用准确的话总结就是，深度学习是机器学习的一个子领域，它使用了多层次的非线性信息处理和抽象，用于有监督或无监督的特征学习、表示、分类和模式识别。

深度学习即表征学习是机器学习的一个分支或子领域，大多数人认为近代深度学习方法是从 2006 开始发展起来的。本文是关于最新的深度学习技术的综述，主要推荐给即将涉足该领域的研究者。本文包括 DL 的基本思想、网络优化、最新进展以及应用。

综述论文是非常有益的，特别是对某一特定领域的新研究人员。一个研究领域如果在不久的将来及相关应用领域中有很大的价值，那通常很难被实时跟踪到最新进展。现在，科学研究是一个很有吸引力的职业，因为知识和教育比以往任何时候都更容易分享和获得。对于一种技术研究的趋势来说，唯一正常的假设是它会在各个方面有很多的改进。几年前对某个领域的概述，现在可能已经过时了。考虑到近年来深度学习的普及和推广，我们简要概述了深度学习和神经网络（NN），以及它的主要进展和几年来的重大突破。

我们希望这篇文章将帮助许多新手研究者在这一领域全面了解最近的深度学习的研究和技术，并引导他们以正确的方式开始。同时，我们希望通过这项工作，向这个时代的顶级 DL 和 ANN 研究者们致敬：Geoffrey Hinton（Hinton）、Juergen Schmidhuber（Schmidhuber）、Yann LeCun（LeCun）、Yoshua Bengio（Bengio）和许多其他研究学者，他们的研究构建了现代人工智能（AI）。跟进他们的工作，以追踪当前最佳的 DL 和 ML 研究进展对我们来说也至关重要。在本论文中，我们首先简述过去的研究论文，对深度学习的模型和方法进行研究。然后，我们将开始描述这一领域的最新进展，以及应用。我们将讨论主流深度学习模型，以及对应的优化和应用等后续研究。此外，我们还介绍了深度学习训练技术，并对深度学习未来进行展望。

**2.相关研究**

在过去的几年中，有许多关于深度学习的综述论文。他们以很好的方式描述了 DL 方法、方法论以及它们的应用和未来研究方向。这里，我们简要介绍一些关于深度学习的优秀综述论文。

Young 等人（2018）[1]讨论了 DL 模型和架构，主要用于自然语言处理（NLP）。他们在不同的 NLP 领域中展示了 DL 应用，比较了 DL 模型，并讨论了可能的未来趋势。

为了纪念人工智能60周年 LeCun、Bengio和Hinton首次合作在nature上发表综述文章“Deep Learning”[2] 介绍了深度学习的基本原理和核心优势，详细介绍了CNN、分布式特征表示、RNN及其不同的应用，并对深度学习技术的未来发展进行展望。

ZIXING Zhang 等人（2017）[3]讨论了用于前端和后端语音识别系统的的单通道和多通道技术，以及联合前端和后端培训框架。讨论了各种方法的优缺点，并在基准数据库中提供了实验结果。

Xiao Xiang Zhu等人（2017）[4]分析了使用深度学习进行遥感数据分析，研究了最新进展以及未来的挑战。

Jiuxiang Gu等人(2015-2017)[5]对卷积神经网络的最新进展进行了广泛的调查。 详细介绍了CNN在不同方面的改进，包括层设计，激活功能，损耗函数，正则化，优化和快速计算。 此外，我们还介绍了卷积神经网络在计算机视觉，语音和自然语言处理中的各种应用。

H Wang等人(2017)[6]对深度学习模型演化历史的回顾。它涵盖了当研究大脑的关联模型时神经网络的起源，以及主导过去十年深度学习研究的模型，如卷积神经网络，深度信念网络和递归神经网络。除了对这些模型的回顾之外，主要关注上述模型的先例，研究如何组合初始构思以构建早期模型以及如何将这些初步模型发展为其当前形式。文中回顾了这些演化路径，并提供了如何开发这些模型的简明思想流程，旨在为深度学习提供全面的背景。更重要的是，随着这条道路，本文总结了这些里程碑背后的要点，并提出了许多指导未来深度学习研究的方向。

Goodfellow在NIPS2016主题演讲[7]（1）为什么生成建模是值得研究的主题，（2）生成模型如何工作，GAN如何与其他生成模型进行比较，（3）GAN如何工作的细节，（4）GAN中的研究前沿 ，（5）将GAN与其他方法结合起来的最先进的图像模型。

J Schmidhuber(2015)[8]回顾了深度监督学习（也概括了反向传播的历史），无监督学习，强化学习和进化计算，以及间接搜索编码深度和大型网络的短程序。

最近所有关于深度学习（DL）的论文都从多个角度讨论了深度学习重点。这对 DL 的研究人员来说是非常有必要的。然而，DL 目前是一个蓬勃发展的领域。在最近的 DL 概述论文发表之后，仍有许多新的技术和架构被提出。为此，我们将努力为新研究人员和任何对这一领域感兴趣的人提供一个深度学习的基础和清晰的概念。

**3.深度前馈网络**

深度前馈网络，也叫作前馈神经网络或者多层感知机（multilayer perceptron, MLP），是典型的深度学习模型。前馈网络的目标是近似某个函数。例如，对于分类器，y = 将输入 x 映射到一个类别 y。前馈网络定义了一个映射 y = f(x; θ)，并且学习参数θ的值，使它能够得到最佳的函数近似。

**3.1深度前馈网络的发展**

由神经科学家麦卡洛克(W.S.McCilloch)和数学家皮兹（W.Pitts）在《数学生物物理学公告》上发表论文《神经活动中内在思想的逻辑演算》（A Logical Calculus of the Ideas Immanent in Nervous Activity）[10]。建立了神经网络和数学模型，称为**MCP**模型。人工神经网络的大门由此开启。

1958年计算机科学家罗森布拉特（Rosenblatt）提出了两层神经元组成的神经网络，称之为“感知器”(Perceptrons)[11]。第一次将MCP用于机器学习（machine learning）分类(classification)。“感知器”算法算法使用MCP模型对输入的多维数据进行二分类，且能够使用梯度下降法从训练样本中自动学习更新权值。1962年,该方法被证明为能够收敛，理论与实践效果引起第一次神经网络的浪潮。

1969年，美国数学家及人工智能先驱Marvin Minsky在其著作中证明了感知器本质上是一种**线性模型（linear model）[12]**，只能处理线性分类问题，就连最简单的XOR（亦或）问题都无法正确分类。这等于直接宣判了感知器的死刑，神经网络的研究也陷入了将近20年的停滞。

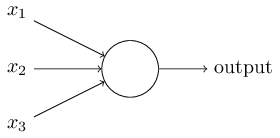
由神经网络之父Geoffrey Hinton 在1986年发明了适用于多层感知器（MLP）的**BP（Backpropagation）**算法[13]，并采用**Sigmoid**进行非线性映射，有效解决了非线性分类和学习的问题。该方法引起了神经网络的第二次热潮。

**万能近似定理（universal approximation theorem (Hornik et al.， 1989;Cybenko, 1989) [14]**表明，一个前馈神经网络如果具有线性输出层和至少一层具有任何一种“挤压”性质的激活函数（例如logistic sigmoid激活函数）的隐藏层，只要给予网络足够数量的隐藏单元，它可以以任意的精度来近似任何**从一个有限维空间到另一个有限维空间**的Borel 可测函数。

**3.2深度前馈网络简介**

感知机模型

感知机由科学家Frank Rosenblatt发明于1950至1960年代，他受到了来自Warren McCulloch 和Walter Pitts的更早工作的启发。感知机的输入是多个二进制，输出是一位单独的二进制。模型如下图：

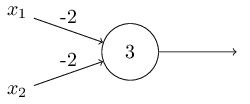


感知机模型

即：

Rosenblatt[15]提出了一种计算输出的简单的规则。他引入了权重（weight），，等实数来表示各个输入对于输出的重要程度。神经元的输出是0还是1，由加权和是否小于或者大于某一个阈值（threshold value）。和感知机通过调整权重和阈值的大小，可以得到不同的决策模型。

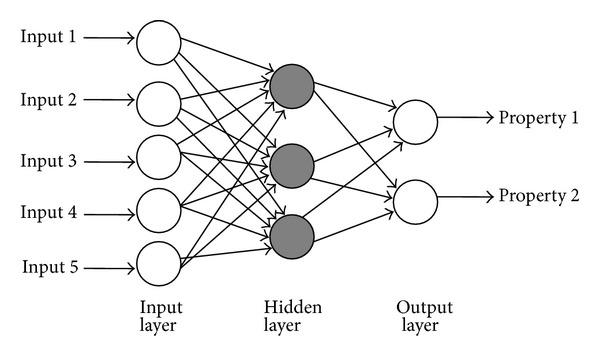
感知机的最初一种用途是计算初等逻辑函数，例如AND、OR和NAND。例如，假如一个感知机有两个输入，每一个权重都是，偏置为，如图：



感知机模型NAND电路

多层感知机模型

多层感知机由感知机组成一个有向无环图相关联，前一层感知机的输出是后一层的输入，第一层为输入层，最后一层是输出层，中间层为隐藏层。网络中的每个隐藏层通常都是向量值的。这些隐藏层的维数决定了模型的宽度



深度前馈神经网络结构图

网络中最左边的一层被称作输入层，其中的神经元被称为输入神经元（input neurons）。最右边的一层是输出层（output layer），包含的神经元被称为输出神经元(output neurons)。理论上只要隐含的节点足够多，就可以拟合任意函数，同时，隐含层越多，越容易拟合更复杂的函数。

**4.卷积神经网络**

1962年, 生物学家Hubel和Wiesel[21]通过对猫脑视觉皮层的研究, 发现在视觉皮层中存在一系列复杂构造的细胞, 这些细胞对视觉输入空间的局部区域很敏感, 它们被称为"感受野". 感受野以某种方式覆盖整个视觉域, 它在输入空间中起局部作用, 因而能够更好地挖掘出存在于自然图像中强烈的局部空间相关性. 文献[22]将这些被称为感受野的细胞分为简单细胞和复杂细胞两种类型． 根据Hubel-Wiesel的层级模型, 在视觉皮层中的神经网络有一个层级结构: 外侧膝状体→简单细胞→复杂细胞→低阶超复杂细胞→高阶超复杂细胞[22]. 低阶超复杂细胞与高阶超复杂细胞之间的神经网络结构类似于简单细胞和复杂细胞间的神经网络结构. 在该层级结构中, 处于较高阶段的细胞通常会有这样一个倾向: 选择性地响应刺激模式更复杂的特征; 同时还具有一个更大的感受野, 对刺激模式位置的变化更加不敏感[23]. 1980年, Fukushima根据Hubel和Wiesel的层级模型提出了结构与之类似的神经认知机(Neocognitron)[23]. 神经认知机采用简单细胞层(S-layer, S层)和复杂细胞层(C-layer, C层)交替组成, 其中S层与Hubel-Wiesel层级模型中的简单细胞层或者低阶超复杂细胞层相对应, C层对应于复杂细胞层或者高阶超复杂细胞层. S层能够最大程度地响应感受野内的特定边缘刺激, 提取其输入层的局部特征, C层对来自确切位置的刺激具有局部不敏感性. 尽管在神经认知机中没有像BP算法那样的全局监督学习过程可利用, 但它仍可认为是CNN的第一个工程实现网络, 卷积和池化(也称作下采样)分别受启发于Hubel-Wiesel概念的简单细胞和复杂细胞, 它能准确识别具有位移和轻微形变的输入模式[23,23]. 随后, LeCun等人基于Fukushima的研究工作使用BP算法设计并训练了CNN(该模型称为LeNet-5), LeNet-5是经典的CNN结构, 后续有许多工作基于此进行改进, 它在一些模式识别领域中取得了良好的分类效果[7].

CNN的基本结构由输入层, 卷积层(convolutional layer), 池化层(pooling layer, 也称为取样层), 全连接层及输出层构成. 卷积层和池化层一般会取若干个, 采用卷积层和池化层交替设置, 即一个卷积层连接一个池化层, 池化层后再连接一个卷积层, 依此类推. 由于卷积层中输出特征面的每个神经元与其输入进行局部连接, 并通过对应的连接权值与局部输入进行加权求和再加上偏置值, 得到该神经元输入值, 该过程等同于卷积过程, CNN也由此而得名[24].

4.3.1卷积层

卷积层由多个特征面(Feature Map)组成, 每个特征面由多个神经元组成, 它的每一个神经元通过卷积核与上一层特征面的局部区域相连. 卷积核是一个权值矩阵(如对于二维图像而言可为3×3或5×5矩阵)[24,25]. CNN的卷积层通过卷积操作提取输入的不同特征, 第1层卷积层提取低级特征如边缘, 线条, 角落, 更高层的卷积层提取更高级的特征. 为了能够更好地理解CNN, 下面以一维CNN(1DCNN)为例,二维和三维CNN可依此进行拓展. 图3所示为一维CNN 的卷积层和池化层结构示意图, 最顶层为池化层, 中间层为卷积层, 最底层为卷积层的输入层.

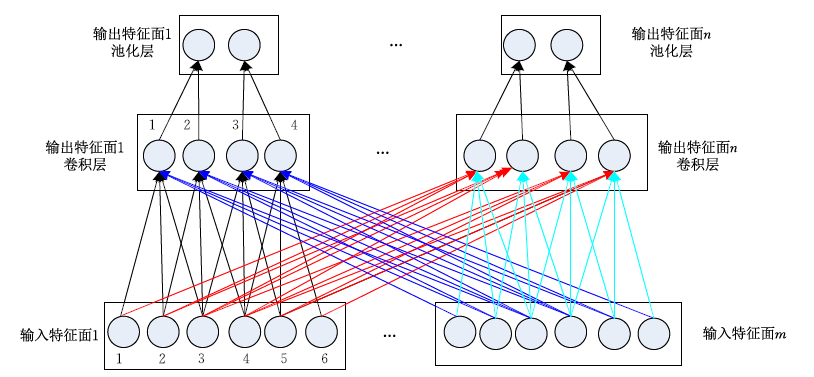


图3 卷积层与池化层结构示意图

由图3可看出卷积层的神经元被组织到各个特征面中, 每个神经元通过一组权值被连接到上一层特征面的局部区域, 即卷积层中的神经元与其输入层中的特征面进行局部连接[26]. 然后将该局部加权和传递给一个非线性函数如ReLU函数即可获得卷积层中每个神经元的输出值. 在同一个输入特征面和同一个输出特征面中, CNN的权值共享, 如图3所示, 权值共享发生在同一种颜色当中, 不同颜色权值不共享. 通过权值共享可以减小模型复杂度, 使得网络更易于训练. 以图3中卷积层的输出特征面1和其输入层的输入特征面1为例, w1(1)1(1)=w1(2)1(2)=w1(3)1(3)=w1(4)1(4),而w1(1)1(1)≠w1(2)1(1)≠w1(3)1(1), 其中w*m*(*i*)*n*(*j*)表示输入特征面*m*第*i*个神经元与输出特征面*n*第*j*个神经元的连接权值. 此外卷积核的滑动步长即卷积核每一次平移的距离也是卷积层中一个重要的参数. 在图3中, 设置卷积核在上一层的滑动步长为1, 卷积核大小为1×3.CNN中每一个卷积层的每个输出特征面的大小(即神经元的个数)满足如下关系[27]

 (6)

其中: 表示每一个输入特征面的大小; 为卷积核的大小; 表示卷积核在其上一层的滑动步长. 通常情况下, 要保证式(6)能够整除, 否则需对CNN网络结构作额外处理. 每个卷积层可训练参数数目满足式(7)[27]

 (7)

其中: 为每个卷积层输出特征面的个数; 为输入特征面个数. 1表示偏置, 在同一个输出特征面中偏置也共享. 假设卷积层中输出特征面*n*第*k*个神经元的输出值为, 而表示其输入特征面*m*第*h*个神经元的输出值, 以图3为例, 则[27]

 (8)

式(8)中, 为输出特征面*n*的偏置值. 为非线性激励函数. 在传统的CNN中, 激励函数一般使用饱和非线性函数(saturating nonlinearity)如sigmoid函数, tanh函数等. 相比较于饱和非线性函数, 不饱和非线性函数(non-saturating nonlinearity)能够解决梯度爆炸/梯度消失问题, 同时也能够加快收敛速度[281]. Jarrett等人[29]探讨了卷积网络中不同的纠正非线性函数(rectified nonlinearity, 包括max(0,x)非线性函数), 通过实验发现它们能够显著提升卷积网络的性能, Nair等人[1]也验证了这一结论. 因此在目前的CNN结构中常用不饱和非线性函数作为卷积层的激励函数如ReLU函数.ReLU函数的计算公式如下所示[1,13]

 (9)

图4中实线为ReLU曲线, 虚线为tanh曲线. 对于ReLU而言, 如果输入大于0, 则输出与输入相等, 否则输出为0. 从图4可以看出，使用ReLU函数, 输出不会随着输入的逐渐增加而趋于饱和. Chen在其报告中分析了影响CNN性能的3个因素: 层数, 特征面的数目及网络组织. 该报告使用9种结构的CNN进行中文手写体识别实验, 通过统计测试结果得到具有较小卷积核的CNN结构的一些结论: (1)增加网络的深度能够提升准确率; (2)增加特征面的数目也可以提升准确率; (3)增加一个卷积层比增加一个全连接层更能获得一个更高的准确率. Bengio等人[14]指出深度网络结构具有两个优点: (1)可以促进特征的重复利用; (2)能够获取高层表达中更抽象的特征, 由于更抽象的概念可根据抽象性更弱的概念来构造, 因此深度结构能够获取更抽象的表达, 例如在CNN中通过池化操作来建立这种抽象, 更抽象的概念通常对输入的大部分局部变化具有不变性. He等人[15]探讨了在限定计算复杂度和时间上如何平衡CNN网络结构中深度, 特征面数目, 卷积核大小等因素的问题. 该文献首先研究了深度(Depth)与卷积核大小间的关系, 采用较小的卷积核替代较大的卷积核, 同时增加网络深度来增加复杂度, 通过实验结果表明网络深度比卷积核大小更重要; 当时间复杂度大致相同时, 具有更小卷积核且深度更深的CNN结构比具有更大卷积核同时深度更浅的CNN结构能够获得更好的实验结果. 其次, 该文献也研究了网络深度和特征面数目间的关系, CNN网络结构设置为: 在增加网络深度时适当减少特征面的数目, 同时卷积核的大小保持不变, 实验结果表明, 深度越深, 网络的性能越好; 然而随着深度的增加, 网络性能也逐渐达到饱和. 此外, 该文献还通过固定网络深度研究了特征面数目和卷积核大小间的关系, 通过实验对比, 发现特征面数目和卷积核大小的优先级差不多, 其发挥的作用均没有网络深度大.

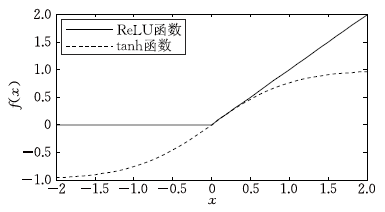


图4 ReLU与tanh函数曲线图

在CNN结构中, 深度越深, 特征面数目越多, 则网络能够表示的特征空间也就越大, 网络学习能力也越强, 然而也会使网络的计算更复杂, 极易出现过拟合的现象. 因而, 在实际应用中应适当选取网络深度, 特征面数目, 卷积核的大小及卷积时滑动的步长, 以使在训练能够获得一个好的模型的同时还能减少训练时间.

1.3.2 池化层

池化层紧跟在卷积层之后, 同样由多个特征面组成, 它的每一个特征面唯一对应于其上一层的一个特征面, 不会改变特征面的个数. 如图3, 卷积层是池化层的输入层, 卷积层的一个特征面与池化层中的一个特征面唯一对应, 且池化层的神经元也与其输入层的局部接受域相连, 不同神经元局部接受域不重叠. 池化层旨在通过降低特征面的分辨率来获得具有空间不变性的特征[16]. 池化层起到二次提取特征的作用, 它的每个神经元对局部接受域进行池化操作. 常用的池化方法有最大池化即取局部接受域中值最大的点, 均值池化即对局部接受域中的所有值求均值, 随机池化[17,18]. Boureau等人[19]给出了关于最大池化和均值池化详细的理论分析, 通过分析得出以下一些预测: (1)最大池化特别适用于分离非常稀疏的特征; (2)使用局部区域内所有的采样点去执行池化操作也许不是最优的, 例如均值池化就利用了局部接受域内的所有采样点. Boureau等人[20]比较了最大池化和均值池化两种方法, 通过实验发现: 当分类层采用线性分类器如线性SVM时, 最大池化方法比均值池化能够获得一个更好的分类性能. 随机池化方法是对局部接受域采样点按照其值大小赋予概率值, 再根据概率值大小随机选择, 该池化方法确保了特征面中不是最大激励的神经元也能够被利用到[16]. 随机池化具有最大池化的优点, 同时由于随机性它能够避免过拟合. 此外, 还有混合池化, 空间金字塔池化, 频谱池化等池化方法[16]. 在通常所采用的池化方法中, 池化层的同一个特征面不同神经元与上一层的局部接受域不重叠, 然而也可以采用重叠池化的方法. 所谓重叠池化方法就是相邻的池化窗口间有重叠区域. Krizhevsky等人［13］采用重叠池化框架使top-1 和top-5的错误率分别降低了0.4%和0.3%, 与无重叠池化框架相比, 其泛化能力更强, 更不易产生过拟合. 设池化层中第*n*个输出特征面第*l*个神经元的输出值为, 同样以图3为例, 则有[10]:

 (10)

其中：表示池化层的第*n*个输入特征面第*q*个神经元的输出值; 可为取最大值函数, 取均值函数等.

池化层在上一层滑动的窗口也称为池化核. 事实上, CNN中的卷积核与池化核相当于Hubel-Wiesel模型[4]中感受野在工程上的实现, 卷积层用来模拟Hubel-Wiesel理论的简单细胞, 池化层模拟该理论的复杂细胞. CNN中每个池化层的每一个输出特征面的大小(神经元个数)为[10]

 (11)

其中, 池化核的大小为, 在图3中. 池化层通过减少卷积层间的连接数量, 即通过池化操作使神经元数量减少, 降低了网络模型的计算量.

4.3.3 全连接层

在CNN结构中, 经多个卷积层和池化层后, 连接着1个或1个以上的全连接层. 与MLP类似, 全连接层中的每个神经元与其前一层的所有神经元进行全连接. 全连接层可以整合卷积层或者池化层中具有类别区分性的局部信息[21]. 为了提升CNN网络性能, 全连接层每个神经元的激励函数一般采用ReLU函数[22]. 最后一层全连接层的输出值被传递给一个输出层, 可以采用softmax逻辑回归(softmax regression)进行分类, 该层也可称为softmax层(softmax layer). 对于一个具体的分类任务, 选择一个合适的损失函数是十分重要的, Gu等人[16]介绍了CNN几种常用的损失函数并分析了它们各自的特点. 通常, CNN的全连接层与MLP结构一样, CNN的训练算法也多采用BP算法.

当一个大的前馈神经网络训练一个小的数据集时, 由于它的高容量, 它在留存测试数据(held-out test data, 也可称为校验集)上通常表现不佳[6].为了避免训练过拟合, 常在全连接层中采用正则化方法: 丢失数据(dropout)技术, 即使隐层神经元的输出值以0.5的概率变为0, 通过该技术部分隐层节点失效, 这些节点不参加CNN的前向传播过程, 也不会参加后向传播过程[6,13], 对于每次输入到网络中的样本, 由于dropout技术的随机性, 它对应的网络结构不相同, 但是所有的这些结构共享权值[13]. 由于一个神经元不能依赖于其它特定神经元而存在, 所以这种技术降低了神经元间相互适应的复杂性, 使神经元学习能够得到更鲁棒的特征[13].目前,关于CNN的研究大都采用ReLU+dropout技术, 并取得了很好的分类性能[13,23,24].

4.3.4 特征面

特征面数目作为CNN的一个重要参数, 它通常是根据实际应用进行设置的, 如果特征面个数过少, 可能会使一些有利于网络学习的特征被忽略掉, 从而不利于网络的学习; 但是如果特征面个数过多, 可训练参数个数及网络训练时间也会增加, 这同样不利于学习网络模型. Chu等人[25]提出了一种理论方法用于确定最佳的特征面数目, 然而该方法仅对极小的接受域有效, 它不能够推广到任意大小的接受域. 该文献通过实验发现: 与每层特征面数目均相同的CNN结构相比, 金字塔架构(该网络结构的特征面数目按倍数增加)更能有效利用计算资源. 目前, 对于CNN网络特征面数目的设定通常采用的是人工设置方法, 然后进行实验并观察所得训练模型的分类性能, 最终根据网络训练时间和分类性能来选取特征面数目.

1.3.5 CNN结构的进一步说明

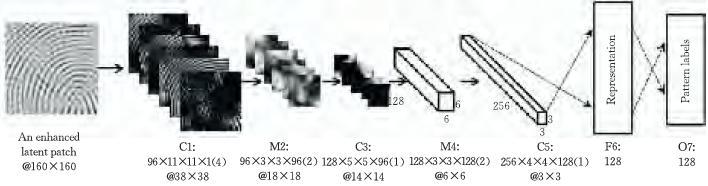


图5 指纹经过CNN的中间层输出特征[26]

CNN的实现过程实际上已经包含了特征提取过程, 以图5, 图6为例直观地显示CNN提取的特征. Cao等人[26]采用CNN进行指纹方向场评估, 图5为其模型结构. 图5共有3个卷积层(C1, C3, C5), 2个池化层(M2, M4), 1个全连接层(F6)和1个输出层(O7).输入的大小为160×160, C1中96×11×11×1(4)表示C1层有96个大小为11×11的卷积核, 1为它的输入特征面个数, 4是卷积核在其输入特征面上的滑动步长, 38×38为每个输出特征面的大小. 卷积层通过卷积操作提取其前一层的各种不同的局部特征, 由图5可看出, C1层提取输入图像的边缘, 轮廓特征, 可看成是边缘检测器. 池化层的作用是在语义上把相似的特征合并起来, 池化层通过池化操作使得特征对噪声和变形具有鲁棒性[9]. 从图上可看出, 各层所提取的特征以增强的方式从不同角度表现原始图像, 并且随着层数的增加, 其表现形式越来越抽象[27]. 全连接层F6中的每个神经元与其前一层进行全连接, 该层将前期所提取的各种局部特征综合起来, 最后通过输出层得到每个类别的后验概率. 从模式分类角度来说, 满足Fisher判别准则的特征最有利于分类, 通过正则化方法(dropout方法),网络参数得到有效调整, 从而使全连接层提取的特征尽量满足Fisher判别准则, 最终有利于分类[27]. 图6给出了CNN提取心电图(electrocardiogram, ECG)特征的过程, 首先通过卷积单元A1, B1, C1(其中每个卷积单元包括一个卷积层和一个池化层)提取特征, 最后由全连接层汇总所有局部特征. 由图中也可以看出, 层数越高, 特征的表现形式也越抽象. 显然, 这些特征并没有临床诊断的物理意义, 仅仅是数理值[27].

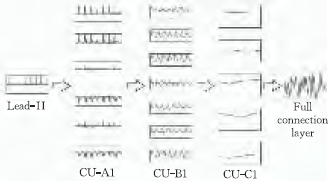


图6 ECG经过CNN的中间层[27]

1.3.6 CNN与传统算法的比较

CNN的本质就是每一个卷积层包含一定数量的特征面或者卷积核[25]. 与传统MLP相比, CNN中卷积层的权值共享使网络中可训练的参数变少, 降低了网络模型复杂度, 减少过拟合, 从而获得了一个更好的泛化能力[28]. 同时, 在CNN结构中使用池化操作使模型中的神经元个数大大减少, 对输入空间的平移不变性也更具有鲁棒性[28]. 而且CNN结构的可拓展性很强, 它可以采用很深的层数. 深度模型具有更强的表达能力, 它能够处理更复杂的分类问题. 总的来说, CNN的局部连接, 权值共享和池化操作使其比传统MLP具有更少的连接和参数, 从而更易于训练.

2.改进算法

2.1网中网结构

CNN中的卷积滤波器是一种广义线性模型(GeneralizedLinearModel, GLM), GLM的抽象水平比较低, 但通过抽象却可以得到对同一概念的不同变体保持不变的特征[50]. Lin等人[50]提出了一种网中网(NetworkinNetwork, NIN)模型, 该模型使用微型神经网络(microneuralnetwork)代替传统CNN的卷积过程, 同时还采用全局平均池化层来替换传统CNN的全连接层, 它可以增强神经网络的表示能力. 微神经网络主要是采用MLP模型, 如图7所示.

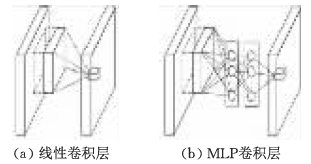


图7 线性卷积层与MLP卷积层对比

图7中, 图7(a)是传统CNN的线性卷积层, 图7(b)是NIN模型的非线性卷积层, 用MLP来取代原来的GLM. NIN通过在输入中滑动微型神经网络得到卷积层的特征面. 与卷积的权值共享类似, MLP对同一个特征面的所有局部感受野也共享, 即对于同一个特征面MLP相同. 文献[1]之所以选择MLP, 考虑到MLP采用BP算法进行训练, 能与CNN结构融合, 同时MLP也是一种深度模型, 具有特征重用的思想. MLP卷积层能够处理更复杂的非线性问题, 提取更加抽象的特征. 在传统的CNN结构中全连接层的参数过多, 易于过拟合, 因此它严重依赖于dropout正则化技术. NIN模型采用全局平均池化代替原来的全连接层, 使模型的参数大大减少. 它通过全局平均池化方法对最后一个MLP卷积层的每个特征面求取均值, 再将这些数值连接成向量, 最后输入到softmax分类层中. 全局平均池化可看成是一个结构性的正则化算子(structural regularizer), 它可以增强特征面与类别的一致性. 在全局平均池化层中没有需要优化的参数, 因此能够避免过拟合. 此外,全局平均池化层对空间信息进行求和, 因此对输入的空间变化具有更强的鲁棒性. Lin等人[1]将该算法应用于MNIST及SVHN等数据集中, 验证了该算法的有效性. Xu等人[2]结合NIN模型提出了ML-DNN模型, 使用与文献[1]相同的数据库, 将其与稀疏编码等方法比较, 表明了该模型的优越性.

2.2空间变换网络

尽管CNN已经是一个能力强大的分类模型, 但是它仍然会受到数据在空间上多样性的影响. Jaderberg等人[3]采用一种新的可学习模块-空间变换网络(Spatial Transformer Networks, STNs)来解决此问题, 该模块由3个部分组成: 本地化网络(localisation network), 网格生成器(grid generator)及采样器(sampler). STNs可用于输入层, 也可插入到卷积层或者其它层的后面, 不需要改变原CNN模型的内部结构. STNs能够自适应地对数据进行空间变换和对齐, 使得CNN模型对平移, 缩放, 旋转或者其它变换等保持不变性. 此外, STNs的计算速度很快, 几乎不会影响原有CNN模型的训练速度.

2.3反卷积

由Zeiler 等人[4]提出的反卷积网络(Decon-volutional Networks)模型与CNN的思想类似, 只是在运算上有所不同. CNN是一种自底而上的方法, 其输入信号经过多层的卷积, 非线性变换和下采样处理. 而反卷积网络中的每层信息是自顶而下的, 它对由已学习的滤波器组与特征面进行卷积后得到的特征求和就能重构输入信号. 随后, Zeiler[5]采用反卷积网络可视化CNN中各网络层学习得到的特征, 从而有利于分析并改进它的网络结构. 反卷积网络也可看成是一个卷积模型, 它同样需要进行卷积和池化过程, 不同之处在于与CNN是一个逆过程. 文献[5]模型中的每一个卷积层都加上一个反卷积层. 在卷积, 非线性函数, 最大池化后, 不仅将输出的特征作为下一层的输入, 也将它送给对应的反卷积层. 反卷积层需要依次进行unpooling(采用一种近似的方法求最大池化的逆过程), 矫正(使用非线性函数来保证所有输出均为非负数)及反卷积操作(利用卷积过程中卷积核的转置作为核, 与矫正后的特征作卷积运算), 然后形成重构特征. 通过反卷积技术可视化CNN各网络层学习到的特征, Zeiler[5]还得出以下结论: CNN学习到的特征对于平移和缩放具有不变性, 但是对于旋转操作一般不具有该特性, 除非被识别对象具有很强的对称性. Zhao等人[6]提出了一个新的称为SWWAE(Stacked What-Where Auto-Encoders)的结构, SWWAE模型由卷积结构及反卷积结构组成, 采用卷积结构对输入进行编码, 而反卷积结构用来进行重构. SWWAE的每一个阶段是一个“内容-位置”(what-where)自动编码机, 编码机由一个卷积层及紧随其后的一个最大池化层组成, 通过最大池化层产生两个变量集: 最大池化的输出记为what变量, 它作为下一层的输入; 将最大池化的位置信息记为where变量, where变量要横向传递到反卷积结构中. SWWAE的损失函数包含3个部分(判别损失, 重构损失及中间重构损失). SWWAE在各种半监督和有监督任务中取得了很高的准确率, 它特别适用于具有大量无标注类别而有标注类别相对少的数据集的情况, 该模型也可能适用于与视频相关的任务[6].

3.实际应用-图像处理

近年来, CNN已经被广泛应用于图像处理领域.



图8 风格化的图像

我们考虑图像变换问题，其中输入图像被变换为输出图像. 用于这些问题的最新方法通常使用输出和标定真实图像之间的每像素损失来训练前馈卷积神经网络. 其他研究表明, 通过基于从预训练网络提取的高级特征定义和优化感知损失函数, 可以生成高质量图像. Justin Johnson[7]将两种方法的优点进行结合, 并提出使用感知损失函数来训练前馈网络以进行图像转换任务. 他的工作展示了图像样式转换的结果, 其中训练前馈网络以解决Gatys等人提出的实时优化问题. 与基于优化的方法相比, 这一研究提供了类似的定性结果, 但速度提高了三个数量级[7]. 这一研究还尝试使用单图像超分辨率, 其中用感知损失替换每像素损耗给出视觉上令人愉悦的结果.



图9 具有实例规范化的模型

**CNN 在语音识别中的应用**

目前语音识别的发展现状，dnn、rnn/lstm和cnn算是语音识别中几个比较主流的方向。2012年，微软邓力和俞栋老师将前馈神经网络FFDNN（Feed Forward Deep Neural Network）引入到声学模型建模中，将FFDNN的输出层概率用于替换之前GMM-HMM中使用GMM计算的输出概率，引领了DNN-HMM混合系统的风潮。长短时记忆网络（LSTM，LongShort Term Memory）可以说是目前语音识别应用最广泛的一种结构。

CNN 被用在语音识别中由来已久，在 12、13 年的时候 Ossama Abdel-Hamid 就将 CNN 引入了语音识别中。那时候的卷积层和 pooling 层是交替出现的，并且卷积核的规模是比较大的，CNN 的层数也并不多， 主要是用来对特征进行加工和处理，使其能更好的被用于 DNN 的分类。随着CNN在图像领域的发光发热，VGGNet，GoogleNet和ResNet的应用，为cnn在语音识别提供了更多思路，比如多层卷积之后再接 pooling 层，减小卷积核的尺寸可以使得我们能够训练更深的、效果更好的 CNN 模型。

CLDNN

在CLDNN中有两层CNN的应用，算是浅层CNN应用的代表。CNN 和 LSTM 在语音识别任务中可以获得比DNN更好的性能提升，对建模能力来说，CNN擅长减小频域变化，LSTM可以提供长时记忆，所以在时域上有着广泛应用，而DNN适合将特征映射到独立空间。而在CLDNN中，作者将CNN，LSTM和DNN串起来融合到一个网络中，获得比单独网络更好的性能。

deep CNN

IBM、微软、百度等多家机构相继推出了自己的Deep CNN模型，提升了语音识别的准确率。Residual/Highway网络的提出使我们可以把神经网络训练的更深。尝试Deep CNN的过程中，大致也分为两种策略：一种是HMM 框架中基于Deep CNN结构的声学模型，CNN可以是VGG、Residual 连接的 CNN 网络结构、或是CLDNN结构。另一种是近两年非常火的端到端结构，比如在 CTC 框架中使用CNN或CLDNN实现端对端建模，或是最近提出的Low Frame Rate、Chain 模型等粗粒度建模单元技术。对于输入端，大体也分为两种：输入传统信号处理过的特征，采用不同的滤波器处理，然后进行左右或跳帧扩展。第二种是直接输入原始频谱，将频谱图当做图像处理。

**百度deep speech**

百度将 Deep CNN 应用于语音识别研究，使用了 VGGNet ，以及包含Residual 连接的深层CNN等结构，并将 LSTM 和 CTC 的端对端语音识别技术相结合，使得识别错误率相对下降了10% (原错误率的90%)以上。此前，百度语音每年的模型算法都在不断更新，从 DNN ，到区分度模型，到 CTC 模型，再到如今的 Deep CNN 。基于 LSTM-CTC的声学模型也于 2015 年底已经在所有语音相关产品中得到了上线。比较重点的进展如下：1)2013 年，基于美尔子带的 CNN 模型;2)2014年，Sequence Discriminative Training(区分度模型);3)2015 年初，基于 LSTM-HMM的语音识别;4)2015 年底，基于 LSTM-CTC的端对端语音识别;5)2016 年，Deep CNN 模型，目前百度正在基于Deep CNN 开发deep speech3，据说训练采用大数据，调参时有上万小时，做产品时甚至有 10 万小时。

**微软**

2016年9月在产业标准 Switchboard 语音识别任务上，微软研究者取得了产业中最低的6.3% 的词错率（WER）。基于神经网络的声学和语言模型的发展，数个声学模型的结合，把ResNet 用到语音识别。而在2016年的10月，微软人工智能与研究部门的团队报告出他们的语音识别系统实现了和专业速录员相当甚至更低的词错率（WER），达到了5.9%。5.9% 的词错率已经等同于人速记同样一段对话的水平，而且这是目前行Switchboard 语音识别任务中的最低记录。这个里程碑意味着，一台计算机在识别对话中的词上第一次能和人类做得一样好。系统性地使用了卷积和LSTM 神经网络，并结合了一个全新的空间平滑方法（spatial smoothing method）和lattice-free MMI 声学训练。虽然在准确率的突破上都给出了数字基准，微软的研究更加学术，是在标准数据库——口语数据库 switchboard 上面完成的，这个数据库只有 2000 小时。

**谷歌**

根据 Mary Meeker 年度互联网报告，Google以机器学习为背景的语音识别系统，2017年3月已经获得英文领域95%的字准确率，此结果逼近人类语音识别的准确率。如果定量的分析的话，从2013年开始，Google系统已经提升了20%的性能。

**科大讯飞DFCNN**

2016年,在提出前馈型序列记忆网络FSMN (Feed-forward Sequential Memory Network) 的新框架后，科大讯飞又提出了一种名为深度全序列卷积神经网络（Deep Fully Convolutional Neural Network，DFCNN）的语音识别框架，使用大量的卷积层直接对整句语音信号进行建模，更好地表达了语音的长时相关性。

由于CNN本身卷积在频域上的平移不变性，同时VGG、残差网络等深度CNN网络的提出，给CNN带了新的新的发展，使CNN成为近两年语音识别最火的方向之一。用法也从最初的2-3层浅层网络发展到10层以上的深层网络，从HMM-CNN框架到端到端CTC框架，各个公司也在deep CNN的应用上取得了令人瞩目的成绩。

4.递归神网络

4.1 RNN的概念

RNN（Recurrent Neural Network）即递归神经网络，是神经网络的一种，主要分为时间递归神经网络（recurrent neural network）和结构递归神经网络（recursive neural network）。RNN一般指代时间递归神经网络，可以描述动态时间行为，因为和前馈神经网络（feedforward neural network）接受较特定结构的输入不同，RNN将状态在自身网络中循环传递，因此可以接受更广泛的时间序列结构输入。单纯的RNN因为无法处理随着递归，权重指数级爆炸或消失的问题（Vanishing gradient problem），难以捕捉长期时间关联；而结合不同的LSTM可以很好解决这个问题[1][2]。结构递归神经网络是一类用结构递归的方式构建的网络，比如说递归自编码机（Recursive Autoencoder），在自然语言处理的神经网络分析方法中用于解析语句。

4.2 RNN的特点

RNN主要用于处理序列数据。序列数据是指在不同时间点上收集到的数据，这类数据反映了某一事物、现象等随时间的变化状态或程度。比如文字序列，其特点是后面的数据跟前面的数据有关系。基础的神经网络只在层与层之间建立了权连接，RNN最大的不同之处就是在层之间的神经元之间也建立的权连接。

4.3 RNN的结构

一个简单的循环神经网络如，它由输入层、一个隐藏层和一个输出层组成：

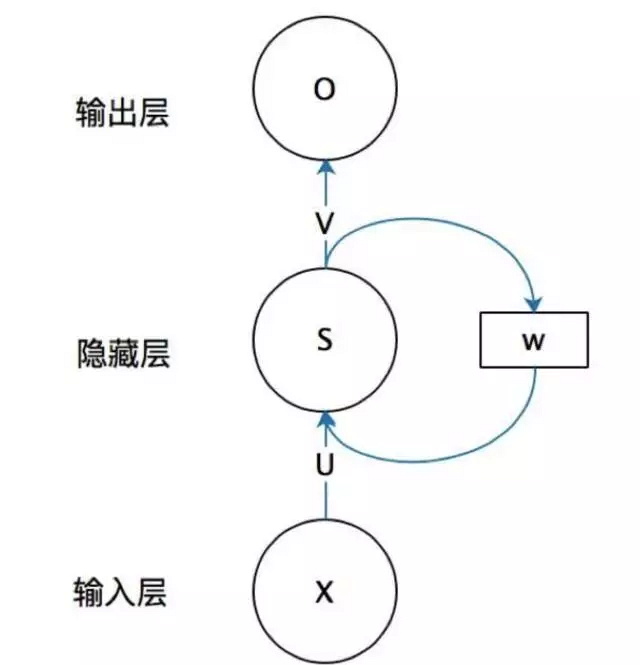


图1 简单的循环神经网络结构

RNN被称为递归神经网络是因为它对一个序列中的每个节点执行相同的任务，而且输出依赖于前面的计算。可以认为RNN存在一个“记忆”，它能够捕获已经计算出的信息。理论上，RNN可以利用任意长序列中的信息，但是实际上它们仅仅能够利用前面极少步的信息。经典的RNN如下所示:

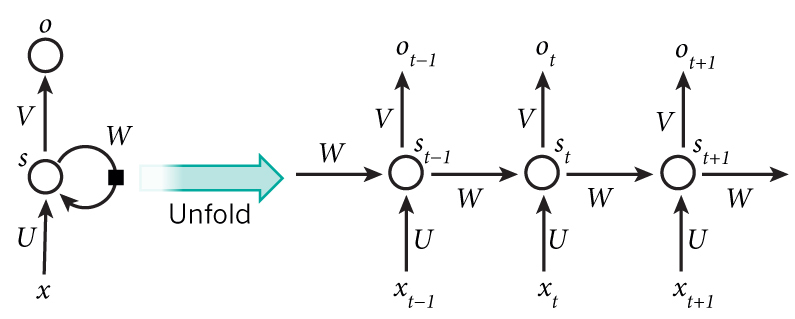


图2 递归神经网络以及

它在时间步长上展开的前向计算图

RNN模型有比较多的变种，这里介绍最主流的RNN模型结构如图3。

图3中左边是RNN模型没有按时间展开的图，如果按时间序列展开，则是上图中的右边部分。我们重点观察右边部分的图。

这幅图描述了在序列索引号t附近RNN的模型。其中：

1）x(t)代表在序列索引号t时训练样本的输入。同样的，x(t−1)和x(t+1)代表在序列索引号t−1和t+1时训练样本的输入。

2）h(t)代表在序列索引号t时模型的隐藏状态。h(t)由x(t)和h(t−1)共同决定。

3）o(t)代表在序列索引号t时模型的输出。o(t)只由模型当前的隐藏状态h(t)决定。

4）L(t)代表在序列索引号t时模型的损失函数。

5）y(t)代表在序列索引号t时训练样本序列的真实输出。

6）U,W,V这三个矩阵是我们的模型的线性关系参数，它在整个RNN网络中是共享的，

这点和DNN很不相同。 也正因为是共享了，它体现了RNN的模型的“循环反馈”的思想。

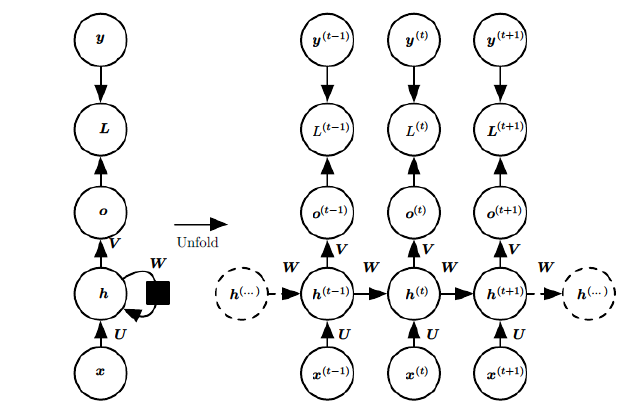


图3主流的RNN模型结构

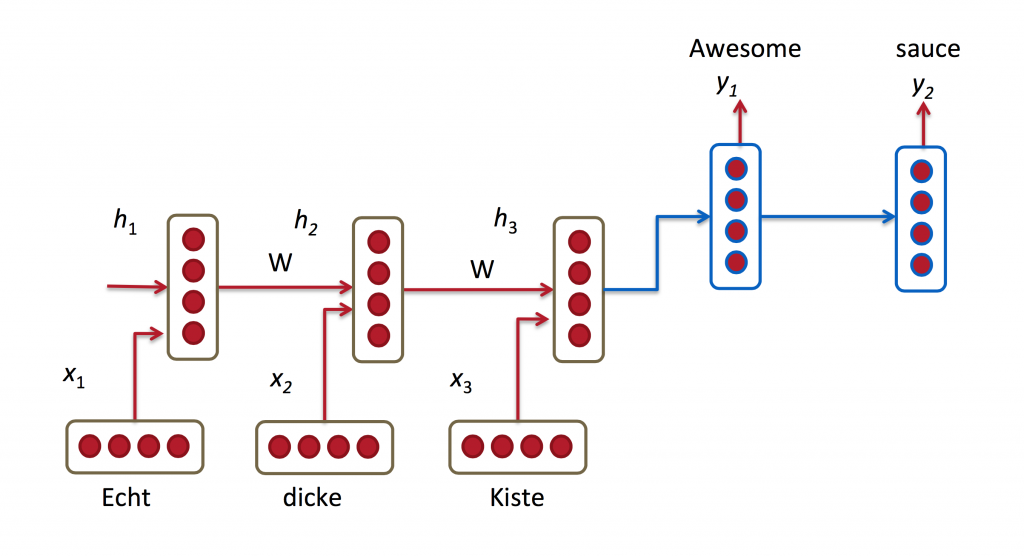


图4 RNN用于机器翻译

4.4 RNN的应用

4.4.1 语言建模和生成文本

给定一个单词序列，我们想预测每个单词在给定前面的单词后的概率。语言模型允许我们衡量句子的可能性,这是机器翻译的一个重要输入（由于高概率的句子一般正确）。能预测下一个单词带来一个好处是我们可以得到一个生成模型，这允许我们可以通过输出概率进行抽样而生成新的文本。训练数据不同，我们可以得到各种各样的模型。语言建模的输入通常是单词序列（例如使用one-hot向量编码），输出是预测单词的概率。训练网络时，我们令ot=xt+1

4.4.2 机器翻译

机器翻译与语言建模的相同之处是源语言（比如德语）的输入也是单词序列。我们想要的输出是对应目标语言（比如英语）的单词序列。一个关键差别在于当我们看到了所有的输入后才开始输出，因为在翻译的句子中的第一个单词要依赖于从整个输入序列中捕获的信息。encoder-decoder是现在广泛使用的利用神经网络的翻译模型，和过去的方法相比也能够达到很高精度。

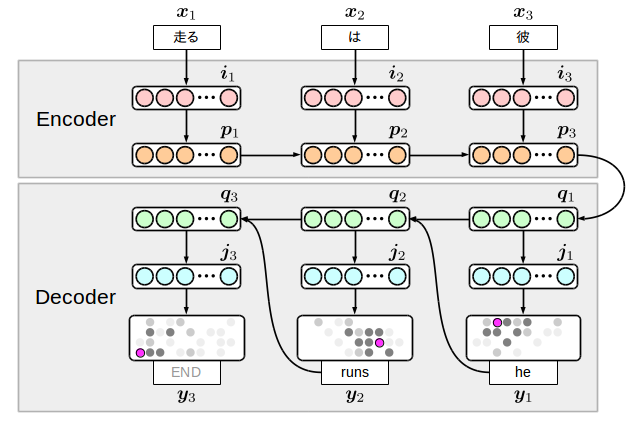
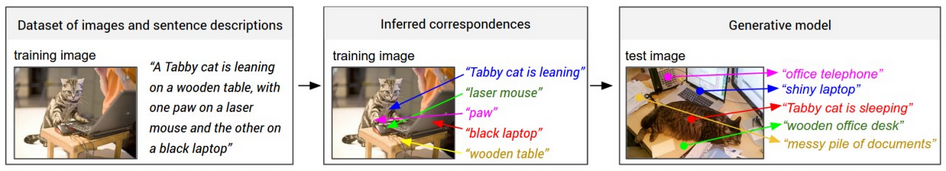


图4 encoder-decoder神经网络的翻译模型



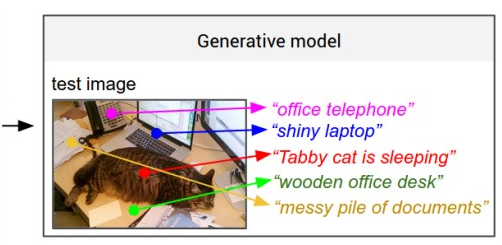


图5 深度神经网络模型用于推断语句片段与图像区域的对应关系

4.4.3语音识别

输入一个声波的声信号序列，我们可以预测一个语音序列以及它们的概率。

4.4.4生成图像描述

结合卷积神经网络，RNN已经被用于无标记图像描述生成模型的一部分。模型的最终目标是生成图像区域的文本描述。如图5所示，在训练阶段，模型的输入是图像集合和对应的语句。这里有两个模型，第一个模型通过多模嵌入来对齐语句片段和视觉区域。然后，我们将这些对齐好的视觉区域和语句片段作为训练数据，训练一个Multimodal RNN模型，这个模型可以根据输入图像自动生成对应的文本描述。

4.4 RNN训练

训练RNNs与训练传统神经网络类似。我们同样使用反向传播算法（BP），但是需要一些变化。因为网络中的参数在所有的时间步上是共享的，每个输出的梯度不仅仅与当前时间步的计算相关，而且依赖于前面的时间步。比如，为了计算t=4

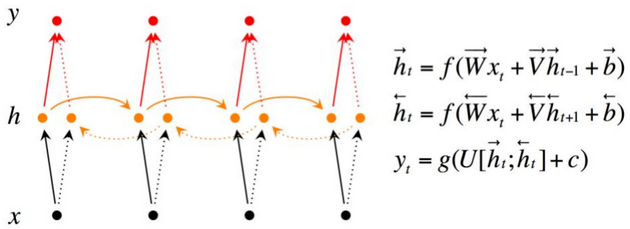
时的梯度我们必须反向传播到前面的3个时间步，然后总和这些梯度。这被成为通过时间的反向传播（BPTT）。如果这些介绍还很难理解，不用担心，后面会有一整篇文章详细介绍BPTT。现在,你要知道普通RNNs通过BPTT训练很难学习到长期依赖（比如在很远时间步间的依赖），这是由于梯度消失或者爆炸问题。存在一些方法来解决这些问题，还有些特定的RNNs(如LSTMs)是专门用来解决这类问题。

4.5 RNN扩展

多年来，研究人员开发了更复杂的RNN来处理普通 RNN模型的一些缺点。我们将在后面的文章中更详细地介绍这些内容，而这部分只作为简要概述，以便你熟悉这些模型的分类。

4.5.1 双向RNN

Bidirectional RNN(双向RNN)假设当前t的输出不仅仅和之前的序列有关，并且 还与之后的序列有关，例如：预测一个语句中缺失的词语那么需要根据上下文进 行预测；Bidirectional RNN是一个相对简单的RNNs，由两个RNNs上下叠加在 一起组成。输出由这两个RNNs的隐藏层的状态决定。结构如图6所示。



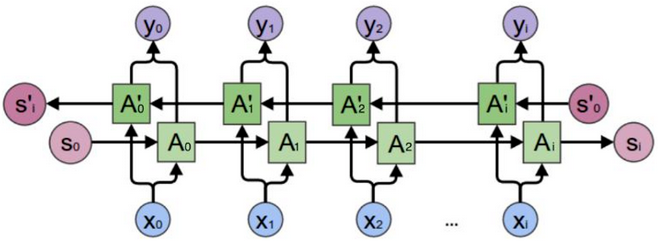


图6 双向RNN结构图

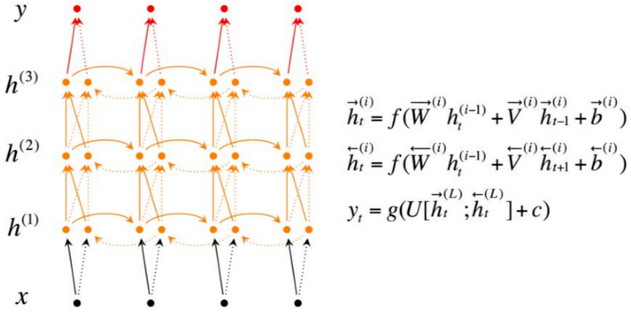


图7 深层双向RNN结构图

4.5.2深层双向RNNs

Deep Bidirectional RNN(深度双向RNN)类似Bidirectional RNN，区别在于每 个每一步的输入有多层网络，这样的话该网络便具有更加强大的表达能力和学习 能力，但是复杂性也提高了，同时需要训练更多的数据。结构如图7所示。

4.5.3 LSTM网络

LSTM算法全称为Long short-term memory，最早由 Sepp Hochreiter和Jürgen Schmidhuber于1997年提出[6]，是一种特定形式的RNN（Recurrent neural network，循环神经网络），而RNN是一系列能够处理序列数据的神经网络的总称。这里要注意循环神经网络和递归神经网络（Recursive neural network）的区别。

一般地，RNN包含如下三个特性：

1）循环神经网络能够在每个时间节点产生一个输出，且隐单元间的连接是循环的；

2）循环神经网络能够在每个时间节点产生一个输出，且该时间节点上的输出仅与下一时间节点的隐单元有循环连接；

3）循环神经网络包含带有循环连接的隐单元，且能够处理序列数据并输出单一的预测。

RNN还有许多变形，例如双向RNN（Bidirectional RNN）等。然而，RNN在处理长期依赖（时间序列上距离较远的节点）时会遇到巨大的困难，因为计算距离较远的节点之间的联系时会涉及雅可比矩阵的多次相乘，这会带来梯度消失（经常发生）或者梯度膨胀（较少发生）的问题，这样的现象被许多学者观察到并独立研究。为了解决该问题，研究人员提出了许多解决办法，例如ESN（Echo State Network），增加有漏单元（Leaky Units）等等。其中最成功应用最广泛的就是门限RNN（Gated RNN），而LSTM就是门限RNN中最著名的一种。有漏单元通过设计连接间的权重系数，从而允许RNN累积距离较远节点间的长期联系；而门限RNN则泛化了这样的思想，允许在不同时刻改变该系数，且允许网络忘记当前已经累积的信息。

LSTM就是这样的门限RNN，其结构如图8所示。LSTM的巧妙之处在于通过增加输入门限，遗忘门限和输出门限，使得自循环的权重是变化的，这样一来在模型参数固定的情况下，不同时刻的积分尺度可以动态改变，从而避免了梯度消失或者梯度膨胀的问题。

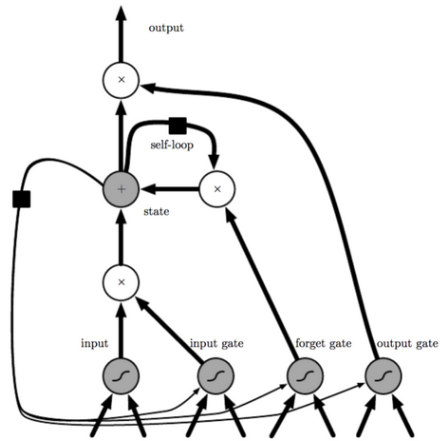
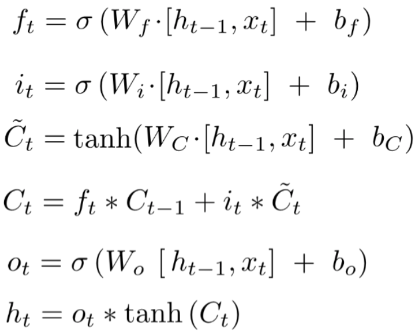


图8 LTSM结构图

根据LSTM网络的结构，每个LSTM单元的计算公式如下图所示，其中Ft表示遗忘门限，It表示输入门限， ̃Ct表示前一时刻cell状态、Ct表示cell状态（这里就是循环发生的地方），Ot表示输出门限，Ht表示当前单元的输出，Ht-1表示前一时刻单元的输出。



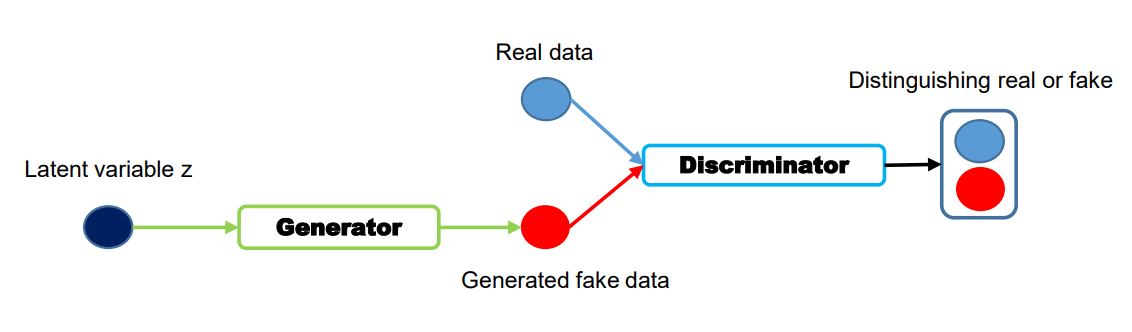
与前馈神经网络类似，LSTM网络的训练同样采用的是误差的反向传播算法（BP），不过因为LSTM处理的是序列数据，所以在使用BP的时候需要将整个时间序列上的误差传播回来。LSTM本身又可以表示为带有循环的图结构，也就是说在这个带有循环的图上使用反向传播时我们称之为BPTT（back-propagation through time）。

**7.生成对抗网络**

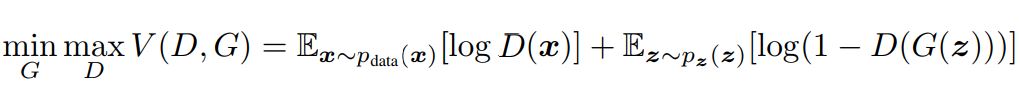
2014年 Goodfellow 等人启发自博弈论中的二人零和博弈 ，开创性地提出了生成对抗网络GAN[1]， 生成对抗网络思想是同时训练两个神经网络。第一个，被称为鉴别器D（Discriminator ）第二个网络被称为生成器G（generator），生成网络G的目标就是最大化判别网络D的错误。而D的目标就是尽量把G生成的数据分布和真实分布区分开**，**这样，G和D构成了一个动态的“博弈过程”。Yann LeCun称GAN是最近十年机器学习领域最酷的想法。GAN的提出引起了学术界轰动，吸引众多学者研究。

**7.1生成对抗网络模型介绍**

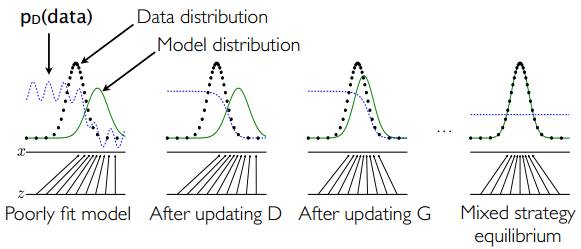
生成对抗网络最早使用在多层感知机上。为了学到从数据 到数据的函数，定义一个多层感知机组成映射（其中Q为参数），来表示噪音分布到数据数据的映射。定义第二个多层感知机,输出为一个标量，来表示X来自数据分布的概率。网络架构如下图所示：



训练D使得最大化D将G产生样本和真实数据分类正确概率。并训练G，以最大化,数学公式为：



一个简单的例子如下图所示：假设在训练开始时，真实样本分布、生成样本分布以及判别模型分别是图中的黑线、绿线和蓝线。可以看出，在训练开始时，判别模型是无法很好地区分真实样本和生成样本的。接下来当我们固定生成模型，而优化判别模型时，优化结果如第二幅图所示，可以看出，这个时候判别模型已经可以较好的区分生成数据和真实数据了。第三步是固定判别模型，改进生成模型，试图让判别模型无法区分生成图片与真实图片，在这个过程中，可以看出由模型生成的图片分布与真实图片分布更加接近，这样的迭代不断进行，直到最终收敛，生成分布和真实分布重合。



GAN训练示意图

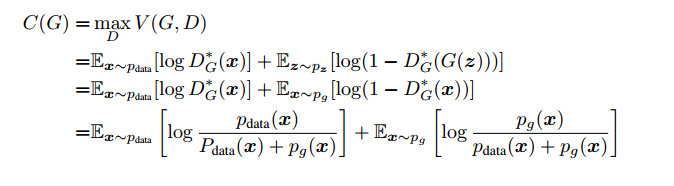
生成对抗网络最优化分析

当网络达到全局最优时生成器G产生的分布与真实数据分布相即：

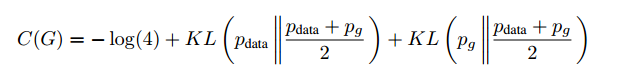
首先讨论给定任意的G优化D，（G固定住），损失函数可以简化为：

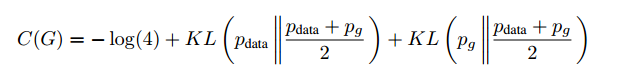
G的目标是最大化V，对V 求导，并令导数为零，取得V最大时：

当D达到时，G最优化损失函数：



将C(G)写成JS散度的形式：





当且仅当Pg=Pdata时，C(G)取得最小值-log(4)，也即是D最优时，G能将损失函数最小化到-log(4)，最小点处Pg=Pdata。即真实数据的分布和生成数据的分布相等。

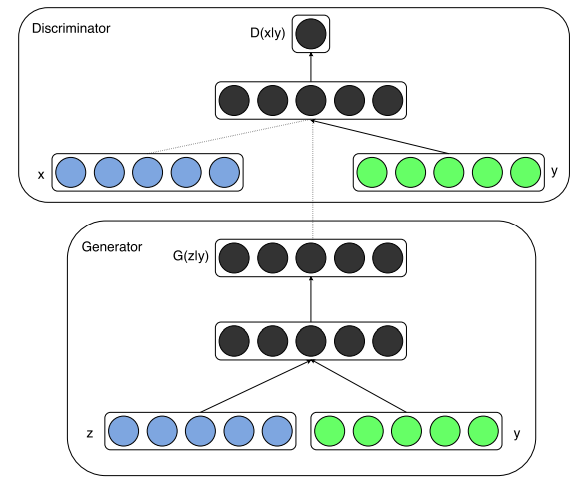
生成对抗网络（GAN）提供了一种不需要大量标注训练数据就能学习深度表征的方式。它们通过反向传播算法分别更新两个网络以执行竞争性学习而达到训练目的。

**7.2 GAN的优化**

GAN难以训练训练困难、梯度消失、模式崩溃,生成结果无法控制的问题。

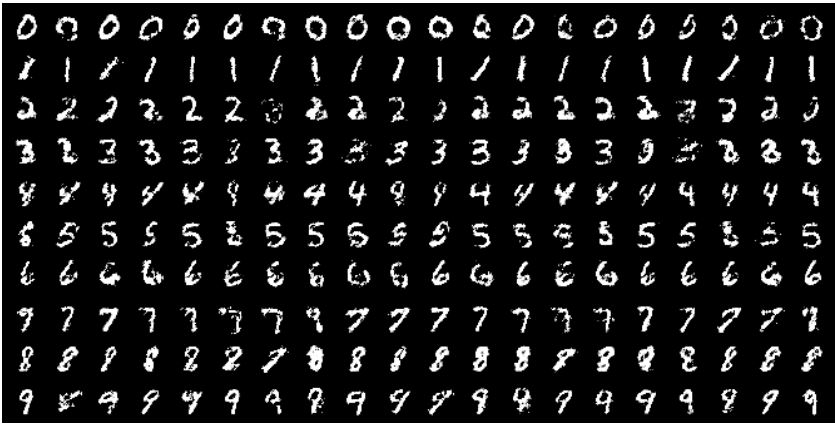
**CGAN**

生成对抗网络（GAN）的基本结构和其训练方式，然而这种方式是基于无监督学习的，没有指定对应的标签，理论上达到可以完全逼近原始数据，因此生成网络G比较自由,这样导致训练好的生成网络G不可控制，为了使得训练的生成网络比较可控，CGAN[2]对原来的无监督学习的对抗网络加一些限制，即将其变为有监督学习的网络，可以很好控制输出结果。架构如下：



CGAN架构

效果



CGAN在MNIST数据集上实验结果

**DCGAN**

因为 CNN 非常适合处理图像数据，所以从全连接到卷积神经网络是一个自然的扩展。早期在 CIFAR-10 上进行的试验表明，用监督学习相同水平和表征能力的 CNN 训练生成器和鉴别器网络会更加困难。DCGAN[3]将CNN运用到原始的GAN中，几乎完全使用了卷积层代替全连接层，判别器几乎是和生成器对称的,DCGAN一共做了一下几点改造:

(1)去掉了G网络和D网络中的pooling layer。

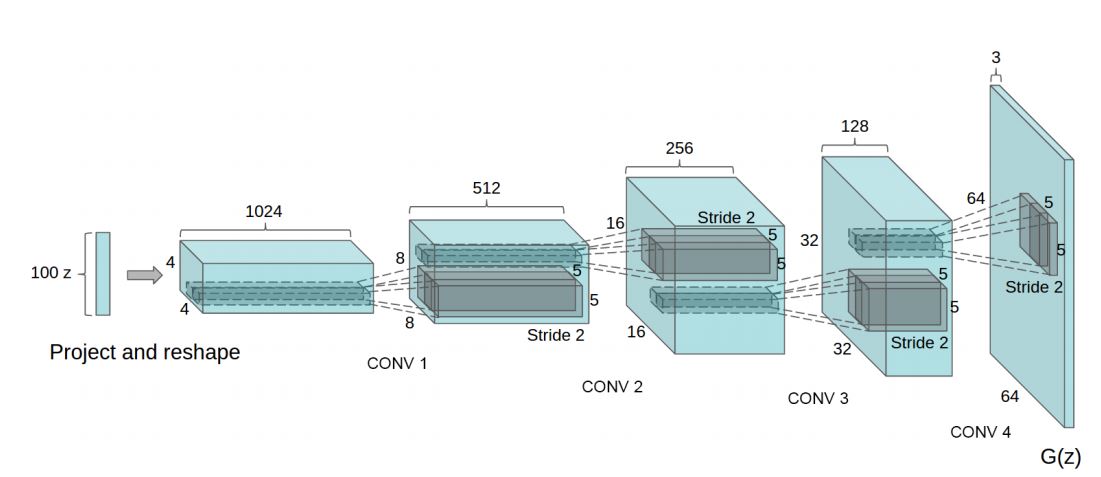
（2）在G网络和D网络中都使用Batch Normalization

（3）去掉全连接的隐藏层

（4）在G网络中除最后一层使用RELU，最后一层使用Tanh

（5）在D网络中每一层使用LeakyRELU。

架构图如下：



DCGAN网络架构图

**WGAN**

与DCGAN不同，WGAN[4]主要从损失函数的角度对GAN做了改进，WGAN相对于原始的GAN采用Wasserstein距离取代JS散度。 而改进后相比原始GAN的算法实现流程却只改了四点：

* 判别器最后一层去掉sigmoid
* 生成器和判别器的loss不取log
* 每次更新判别器的参数之后把它们的绝对值截断到不超过一个固定常数c
* 不要用基于动量的优化算法（包括momentum和Adam）

WGAN理论上给出了GAN训练不稳定的原因，即交叉熵（JS散度）不适合衡量具有不相交部分的分布之间的距离，转而使用wassertein距离去衡量生成数据分布和真实数据分布之间的距离，理论上解决了训练不稳定的问题。解决了模式崩溃的（collapse mode）问题，生成结果多样性更丰富。

**WGAN-GP**

WGAN-GP是WGAN之后的改进版，主要还是改进了WGAN中连续性限制的条件，WGAN-GP提出了使用梯度惩罚（gradient penalty）的方式以满足此连续性条件。解决了训练梯度消失梯度爆炸的问题。WGAN-GP贡献如下：

1 提出了一种新的lipschitz连续性限制手法—梯度惩罚，解决了训练梯度消失梯度爆炸的问题。

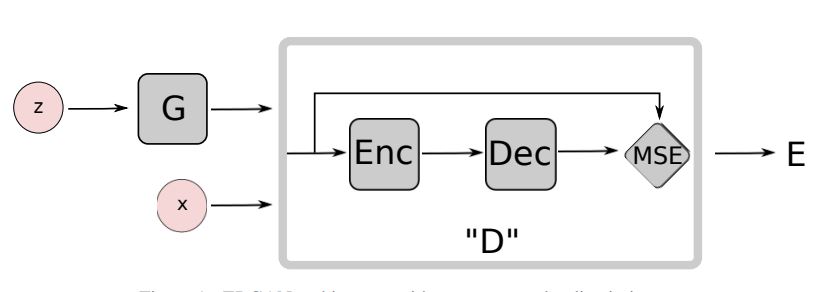
2比标准WGAN拥有更快的收敛速度，并能生成更高质量的样本

3提供稳定的GAN训练方式，几乎不需要怎么调参，成功训练多种针对图片生成和语言模型的GAN架构

**EBGAN**

  EBGAN[6]从能量模型的角度对GAN进行了扩展。EBGAN将判别器看做是一个能量函数，这个能量函数在真实数据域附近的区域中能量值会比较小，而在其他区域（即非真实数据域区域）都拥有较高能量值。因此，EBGAN中给予GAN一种能量模型的解释，即生成器是以产生能量最小的样本为目的，而判别器则以对这些产生的样本赋予较高的能量为目的。

从能量模型的角度来看待判别器和GAN的好处是，我们可以用更多更宽泛的结构和损失函数来训练GAN结构，比如文中就用自编码器(AE) 的结构来作为判别器实现整体的GAN框架，如下图所示：



EBGAN架构图

在训练过程中，EBGAN比GAN展示出了更稳定的性能，也产生出了更加清晰的图像

**Info GAN**

原始的GAN模型通过对抗学习最终可以得到一个能够与真实数据分布一致的模型分布,此时虽然模型相当于已经学到了数据的有效语义特征，但输入信号z中的具体维度与数据的语义特征之间的对应关系并不清楚，比如z中的哪些维度对应于光照变化或哪些维度对应于pose变化是不明确的。而infoGAN不仅能对这些对应关系建模，同时可以通过控制相应维度的变量来达到相应的变化，比如光照的变化或pose的变化。

InfoGAN是生成对抗网络信息理论的扩展，能够以完全非监督的方式得到可分解的特征表示。它可以最大化隐含(latent)变量子集与观测值之间的互信息(mutual information)，并且发现了有效优化互信息目标的下界，info GAN信息论在生成对抗网络的一种非常漂亮的应用。

Info GAN 实现了在SVHN数据集中，得到不同特征可以分解数字在图像中的亮度以及区分图像中不同的数字；在CelebA数据集中，同样的可以通过不同的编码获取一些特征，比如人脸不同的转向角度，是否带了眼镜，发型的不同，情绪的变化。

**BEGAN**

以往的GAN以及其变种都是希望生成器生成的数据分布尽可能的接近真实数据的分布，当生成数据分布等同于真实数据分布时，我们就确定生成器G经过训练可以生成和真实数据分布相同的样本，即获得了生成足以以假乱真数据的能力，所以从这一点出发，研究者们设计了各种损失函数去令G的生成数据分布尽可能接近真实数据分布。 BEGAN[7]代替了这种估计概率分布方法，它不直接去估计生成分布Pg与真实分布Px的差距，而是估计分布的误差的分布之间的差距。EBGAN提出了一种新的简单强大GAN网络结构, 使得网络很快且稳定的收敛

**VAE-GANs**

* VAE-GANS利用学习的表示来更好地测量数据空间中的相似性。 通过将变分**自动编码器(variational autoencoder)**与生成对抗网络相结合，可以使用GAN判别器中的学习特征表示作为VAE重建目标的基础。 因此，用**feature-wise误差**替换**element-wise误差**，以更好地捕获数据分布。将的方法应用于面部图像，并用VAEGANs在CelebA的面部图像上进行训练。 该数据集由202,599个图像组成，注释有40个二进制属性，如眼镜，刘海，苍白皮肤等。

训练后的结果表明，普通的VAE能够清晰地绘制脸部的正面部分，但偏离中心的图像变得模糊；VAEDisl甚至偏离中心产生更清晰的图像，因为重建误差被提升到像素之外。相比之下，VAE / GAN可以产生更清晰的图像。表明它在视觉保真度方面较优，具有元素相似性度量。此外，该方法可以使用简单的算法来修改高级抽象视觉特征（例如，佩戴眼镜）。

**Self-Attention GAN**

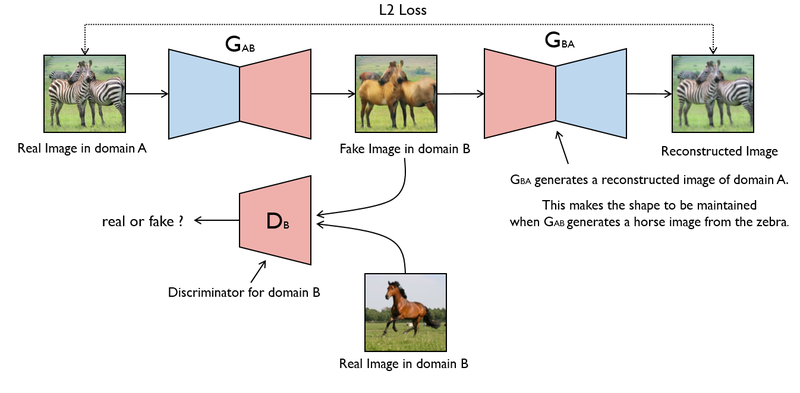
Self-Attention GAN在GAN生成中加入了attention的机制，同时将光谱归一化(spectral normalization)的思想引入到生成器当中。将注意力机制引入生成对抗网络的生成器和鉴别器。它通过对所有深度的特征应用1x1卷积，以一个点表示受关注区域。随后，将多个对原始特征映射的softmax相乘，从而促进注意力区域的特征输出，使得注意力图得到了改善。1x1卷积滤波器能够学习得出最佳结果，相较其他的生成对抗网络Self-Attention GAN需要学习的方式非常少。

SAGAN在极具挑战性的 ImageNet 数据集中将最好的 inception 分数记录从 36.8 提高到 52.52，并将 Frechet Inception 距离从 27.62 减少到 18.65。

**7.3GAN的应用**

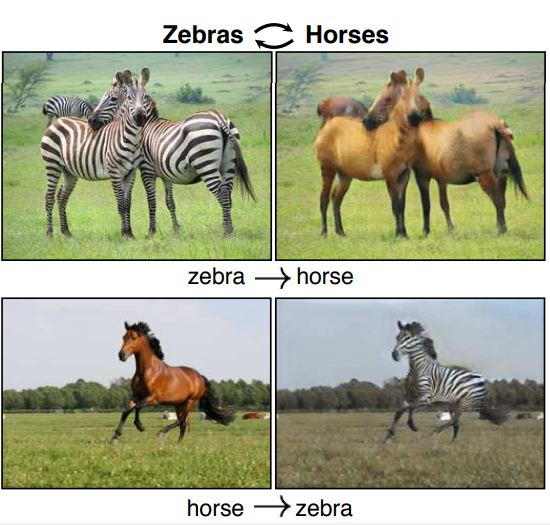
**图片转换**

CycleGAN[8],后续发现DiscoGAN[9]与DualGAN[10]论文是类似的论文。CycleGAN是一个环形的结构，主要由**两个生成器及两个判别器**组成。架构如下：



CycleGAN架构

CycleGAN本质上是两个镜像对称的GAN，构成了一个环形网络。两个GAN共享两个生成器，并各自带一个判别器，即共有两个判别器和两个生成器。效果如下：



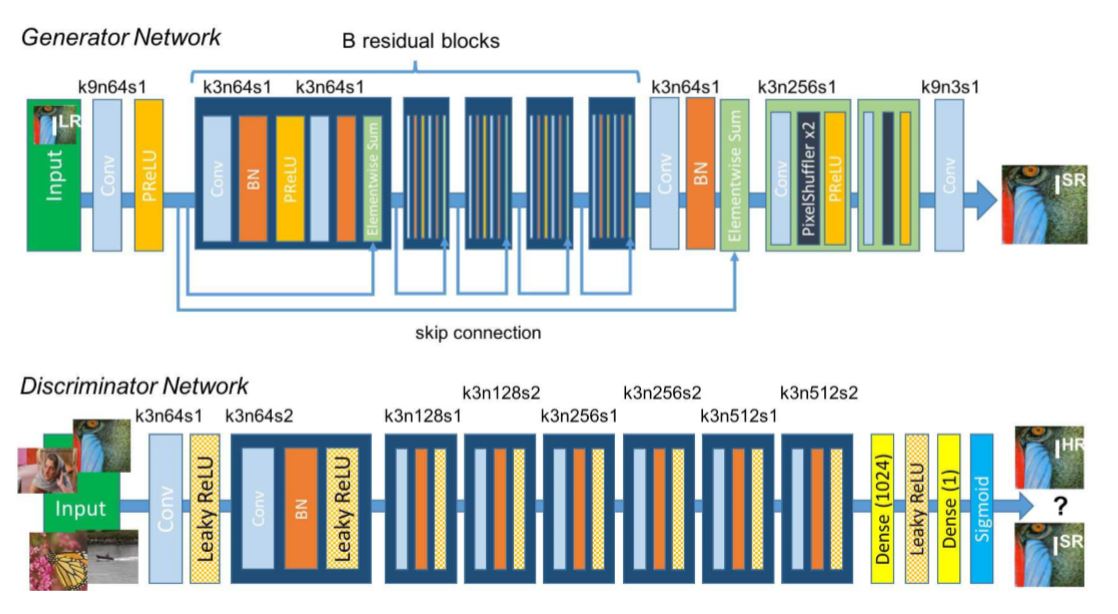
CycleGAN效果展示

效果展示**多域转换**

CycleGAN，都是解决了一对一的问题，即一个领域到另一个领域的转换，无法实现多个领域的转换。StarGAN[11]，顾名思义，就是星形网络结构，在StarGAN中，生成网络G被实现成星形。能够实现多个领域之间的互换。

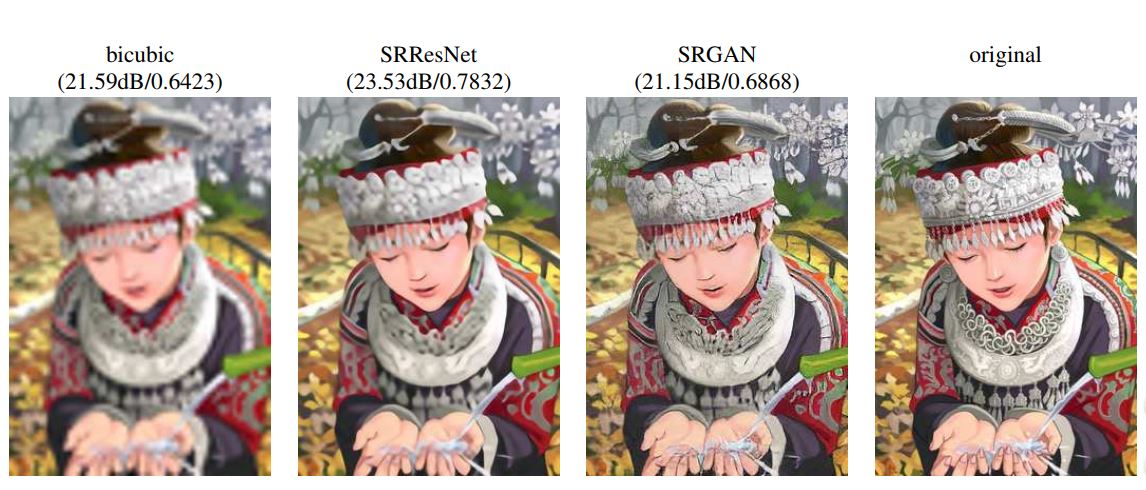
**图片超分辨率**

SRGAN[12]将GAN运用到图片超分辨率当中。SRGAN使用了很深的卷积神经网络构建生成网络和判别网络，并使用残差块以增加网络深度。为了弥补MSE（均方误差）造成细节缺失，提出了perceptual loss这样一个损失函数（论文中基于VGG19预训练模型）。网络架构如下



SRGAN网络架构

和其他超分辨率方法对比如下：



多种超分辨率算法对比

**小脸检测**

Cvpr2018一篇Finding Tiny Faces in the Wild with Generative Adversarial Network[12]是GAN在目标检测领域的力作。GAN在图片合成中一个突破是成功应用是超分辨率重建（super-resolution, SR）**，**发表于CVPR2017[11]**。** GAN模块是本文的贡献所在，包括生成网络和判别网络，其中生成网络又由两个子网络组成：上采样网络（up-sample sub-network）和提纯网络 （refinement sub-network），其目的是将低分辨率的目标候选框生成为高分辨率的人脸patch。判别网络最后包含两个分支，一个负责判别是否是高分辨率，另一个负责判别是否是人脸。

**字体风格迁移**

Multi-Content GAN for Few-Shot Font Style Transfer[14]首次提出用端到端的方案来解决从少量相同风格的字体中合成其他艺术字体，例如 A-Z 26 个相同风格的艺术字母，已知其中 A-D 的艺术字母，生成剩余 E-Z 的艺术字母。通过对传统 Condition GAN 做扩展，提出了 Stack GAN 的两段式架构，首先通过 Conditional GAN根据已知的字体生成出所有 A-Z 的字体，之后通过 Conditional GAN加上颜色和艺术点缀。

**Photo Enhancement**

CVPR2018来自南京理工大学的论文Single Image Dehazing via Conditional Generative Adversarial Network基于CGAN进行图片去雾[18]，与基本cGAN中的生成网络不同，这篇论文提出了一种编码器和解码器架构，以便它可以产生更好的结果。为了生成逼真的清晰图像，通过引入VGG特征和L1正则化梯度先验来进一步修改基本cGAN公式。

近年来，随着计算机软件和硬件技术的不断发展，计算能力不断增强，对雨天图像进行去雨处理已经成为可能，这反过来又对去雨图像的清晰度和真实感提出了新的要求。在雨天情况下，由于场景的能见度低和背景场景被遮挡，图像中目标的对比度和颜色等特征都会出现不同程度的衰减，导致背景信息（即目标图像）表达不明确，这使得一些视频或图像系统不能正常工作，因此需要消除雨天对图像场景的影。队论文提出了使用注意力生成对抗网络(Attentive GAN)[22]。主要思想是模拟人的视觉注意力(Visual attention),将注意力进行量化，然后同时应用于生成网络(Generative network)和判别网络(Discriminative network)进行训练。在训练过程中，量化的视觉注意力可以学习到更多的雨水区域及其周围的信息。因此，将视觉注意力应用于生成网络和判别网络，可以使得生成网络能更好的聚焦于雨水区域及其周围的结构信息，也可以使得判别网络获得图像恢复区域的局部一致性信息。

CVPR2018论文Deep Photo Enhancer: Unpaired Learning for Image Enhancement from Photographs with GANs [19]采用不对称的图像增强学习方法。给定一组具有所需特征的照片，所提出的方法学习照片增强器，其将输入图像变换为具有这些特征的增强图像。该方法基于双向生成对抗网络（GAN）的框架。

**3D-GAN**

3D-GAN[17], 它利用体积卷积网络和生成对抗网络的最新进展，使用GAN从二维图片生成三维模型。使用对抗性标准而不是传统的启发式标准，使得生成器能够隐式捕获对象结构并合成高质量的3D对象。

**DA-GAN**

**微软亚洲研究院研发的DA-GAN技术，则将对今后的艺术创造模式产生重大影响。**而关于DA-GAN的论文也已被CVPR 2018接收。

**DA-GAN研究团队的技术突破得益于特征表达技术的发展。**以往的特征表达工作，多是让机器理解图片并抽取特征，然后再对图片进行分类；而DA-GAN则有些反向思维的意味，**在提取图片特征后，将特征在人类的视觉空间中还原出来。DA-gan首先要能够根据现实世界的鸟，总结出鸟的结构和特征，然后再根据用户的需求，输出他们所需要的鸟**。之所以选择鸟类作为研究对象，是因为鸟类的特征非常丰富，仅头部就有几十种特征，鸟类专家就是利用这些细微的差别来判断鸟的种类，而特征丰富就意味着可以更好地去验证模型的生成能力。

**voice conversion**

语音转换（VC）是一种在保留语言信息的同时转换指定话语的语言信息的技术。VC可以在很多地方得到应用，如文本到语音（TTS）系统的说话人身份（男女，老少等）修改， 口语辅助，语音增强和发音转换。VC的发展也是一直在进步，应用较为广泛的就是基于高斯混合模型（GMM）的发展。这几年随着深度学习的铺开，RNN， GAN以及基于非负矩阵分解（NMF）在VC上也到了应用和发展。

目前VC的方法大部分对于并行数据中涉及的对齐处理不是很理想，因此可能需要仔细的预筛选和手动校正才能使这些框架可靠地工作。为了绕过这些限制，StarGAN-VC[25]使用对抗性损失进行生成器训练， 以鼓励生成器输出与真实语音无法区分，并确保每对属性域之间的映射将保留语言信息，StarGAN-VC在测试时不需要任何有关输入语音属性的信息。并且仅仅几分钟就可以生成逼真的语音。

Voice Impersonation using Generative Adversarial Networks[24]提出了一种新的基于神经网络的语音质量和样式模拟框架，用于合成模拟语音。该框架建立在快速准确的生成对抗网络模型之上。给定源和目标说话者的声音的光谱表示，模型学习模仿目标说话者的语音质量和风格，而不管任一语音的语言内容。实验表明，该模型可以生成效果非常好的模拟语音样本。它甚至能够有效地冒充不同性别的声音。使用用于评估合成语音的标准程序定性地评估结果。

**SeqGAN**

SeqGAN[23]的提出是用来解决过去的GAN方法面对**离散表征(discrete token)**的生成序列时，具有一定的局限性的问题。将数据生成器建模为**强化学习（RL）**中的随机策略，SeqGAN通过直接执行梯度策略更新来绕过生成器区分问题。 RL的**reward信号**来自在完整序列上判断的GAN判别器，并且使用**蒙特卡罗搜索(Monte Carlo search)**传递回中间状态。

SeqGAN案例如下文本生成：使用了16,394个中文绝句的语料库(每个绝句包含四行，共20个字符)，创作诗歌；使用了一个奥巴马政治演说的语料库(11,092段)，来生成政治演讲。音乐创作：使用**诺丁汉(Nottingham)数据集**作为训练数据(695个midi文件格式的民间音乐集合)。使用88个数字来表示88个音高(对应于钢琴上的88个音符)。 通过每0.4s的音高采样，我们将midi文件转换为1到88的数字序列，长度为32。

**7.5 GAN训练技术**

PG-GAN

ICLR2018论文Progressive Growing of GANs for Improved Quality, Stability, and Variation提出一种新的GAN训练技术，称为PG-GAN。它的关键思想是循序渐进、层层发展。渐进式GAN要求生成器和鉴别器先从低分辨率的图像开始，随着训练的进行逐步增加生成器中的层数和鉴别器输出图像的分辨率，以此模拟日益精细的细节。实验证明，这种方法既加快了训练速度，又大大稳定了训练过程，且生成图像的质量非常高。

SN-GAN

SN-GAN[16]来自日本研究者的 ICLR 2018 论文，提出了一种叫做 “谱归一化”（spectral normalization）的新的权重归一化（weight normalization）技术，来稳定判别器的训练。这种新归一化技术计算轻巧，易于并入现有的部署当中。在 CIFAR10，STL-10 和 ILSVRC2012 数据集上测试了谱归一化的功效，通过实验证实了相对于那些使用此前提出的训练稳定技术训练的 GAN，谱归一化 GAN（SN-GAN）能够生成质量相同乃至更好的图像。

BigGAN

在一篇提交到 ICLR 2019 的论文中，研究者尝试在最大规模的数据集中训练生成对抗网络，并研究在这种规模的训练下的不稳定性。研究者发现应用垂直正则化（orthogonal regularization）到生成器可以使其服从简单的「截断技巧」（truncation trick），从而允许通过截断隐空间来精调样本保真度和多样性的权衡。这种修改方法可以让模型在类条件的图像合成中达到当前最佳性能。当在 128x128 分辨率的 ImageNet 上训练时，本文提出的模型—BigGAN—可以达到 166.3 的 Inception 分数（IS），以及 9.6 的 Frechet Inception 距离（FID），而之前的最佳 IS 和 FID 仅为 52.52 和 18.65。

GAN Lab

深度学习领域的最新进展吸引了众多从业者及学生的目光，激发他们当中的很多人去学习这项新技术。尽管视觉和交互的方法已经让深度学习变得更易掌握，但多数现有的工具聚焦于较简单的模型。本文提出了 GAN Lab[20]，这是第一个为非专业人士学习、试验生成对抗网络（一种流行的复杂深度学习模型）而设计的交互式视觉工具。用户可以利用 GAN Lab 交互地训练生成模型，并可视化动态训练过程的中间结果。GAN Lab 紧密集成了总结 GAN 结构的模型概述图（model overview graph）和帮助用户解释子模型之间相互作用的分层分布视图（layered distributions view）。GAN Lab 引入了新的交互试验特征，用于学习复杂的深度学习模型，如在多个抽象层次上进行逐步训练以理解复杂的训练动态。GAN Lab 使用 TensorFlow.js 实现，任何人都可以通过网页浏览器进行访问，无需安装，也无需专门硬件，克服了部署深度学习交互式工具的一个重大实际挑战。

**8神经网络训练技术**

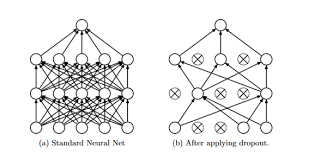
神经网络难以训练，促进了训练技术的发展。

**8.1Dropout**

Dropout[1]是指在深度学习网络的训练过程中，对于神经网络单元，按照一定的概率将其暂时从网络中丢弃。对于随机梯度下降来说，由于是随机丢弃，故而每一个mini-batch都在训练不同的网络。Dropout是防止过拟合的有效方法，并且可以和任何一种网络结合。相对于ensemble，Droupout更容易训练。

**8.2 Batch Normalization**

Batch Normalization[2]于2015年提出，已经广泛运用于神经网络的训练。核心思想是在深度网络的中间层内添加正态标准化处理（作为 BN 层出现），同时约束网络在训练过程中自动调整该标准化的强度，从而加快训练速度并降低权值初始化的成本。 在神经网络训练时遇到收敛速度很慢，或梯度爆炸等无法训练的状况时可以尝试BN来解决。另外，在一般使用情况下也可以加入BN来加快训练速度，提高模型精度。



Dropout

**8.3 Layer Normalization**

对于RNN这种没法用mini-batch的网络，没办法用Batch Normalization，所以提出了Layer Normalization[3]，**对每一层的进行归一化。**

**Distilling**

Distilling[4]通过设计新的网络结构和目标函数将这种知识“传授给”规模较小的深度学习网络，实现对网络模型的压缩，以显著缩小的网络规模和计算需求展现出尽量高的模型表现。最近，为了将这种思想部署在分布式环境中，用以突破如今常常被使用的分布式 SGD（同步和异步形式）的瓶颈，Hinton 团队又发布了名为「LARGE SCALE DISTRIBUTED NEURAL NETWORK TRAINING THROUGH ONLINE DISTILLATION」（通过在线蒸馏的神经网络大规模分布式训练）[5]的论文。

**8.4 Network Pruning**

神经网络层数越深、参数越多，所得出的结果就越精细。但与此同时，问题也来了：越精细，意味着所消耗的计算资源也就越多。在不影响效果前提下对网络进行修减可以减少文件空间，加快计算速度。在 1998年 Lecun 分类手写数字网络少于1M 参数[6]。Song Han提出的Deep Compression[7]，实现AlextNet压缩了35倍，从 240MB，减小到 6.9MB；把VGG-16压缩了49倍，从 552MB 减小到 11.3MB。主要思想是保留重要连接，删除不重要连接，重新训练网络，并采用权值共享，霍夫丁编码方式进一步减少压缩网络。Channel Pruning[8]针对CNN,作者思路是找到神经元之间关系，选出具有代表性的神经元。Network Slimming[9]在卷积层的每个通道引入一个尺度因子γ,具体操作为将尺度因子γ与该通道的输出相乘. 在训练的过程中,对这些尺度因子采用sparsity regularization操作,以自动鉴定不重要的通道。NPIS2018上一篇论文来自微软亚洲研究院的论文**Frequency-Domain Dynamic Pruning for Convolutional Neural Networks，文中**考虑到卷积神经网络中，卷积滤波器会有很大的空间冗余，我们提出在频率域进行网络系数的动态裁剪的方法，针对每次训练迭代和不同的频带，用动态的阈值来指导裁剪。实验结果表明，频域动态裁剪显著优于传统的空域裁剪方法。特别是对于ResNet-110，在不牺牲网络性能甚至有所提高的情况下，我们的方法可以达到8倍的系数压缩和8.9倍的计算加速。

**8.5 高性能网络架构**

谷歌的”Xception“网络[10]引入了”深度可分离卷积”的概念，将普通的卷积运算拆分成逐通道卷积（depthwise convolution）和逐点卷积（pointwise convolution）两部进行，有效地减少了计算量和参数量；而 Facebook 的“ResNeXt”[11]网络则首先使用逐点卷积减少输入特征的通道数，再利用计算量较小的分组卷积（group convolution）结构取代原有的卷积运算，同样可以减少整体的计算复杂度。ShuffleNet[12] 网络结构同样沿袭了稀疏连接的设计理念。作者通过分析 Xception 和 ResNeXt 模型，发现这两种结构通过卷积核拆分虽然计算复杂度均较原始卷积运算有所下降，然而拆分所产生的逐点卷积计算量却相当可观，成为了新的瓶颈。例如对于 ResNeXt 模型逐点卷积占据了 93.4% 的运算复杂度。可见，为了进一步提升模型的速度，就必须寻求更为高效的结构来取代逐点卷积。很多重要的需要语义信息的计算机视觉任务，如目标检测、物体识别等，都需要一个好的“基础模型”作为特征提取器。在移动设备越发重要的今天，在其之上运行的视觉算法模型会越多、准确率要求会越来越高。

**8.6 分布式训练**

随着大型神经网络和大型数据集的出现而蓬勃发展。然而，大型神经网络和大型数据集往往需要更长的训练时间，而这正好阻碍研究和开发进程。研究分布式训练很有必要，主流方法采用数据并行，模型并行等方式。 Facebook论training imagenet in 1 hour[14]，作者使用较大批量大小（达 8192 张图片）进行训练时，实验几乎没有精度损失。为了实现这一结果，我们采用了线性缩放规则（linear scaling rule）作为批量大小函数来调整学习率，同时我们开发了一种新的预热方案（warmup scheme），该方案会在训练前期克服优化的困难。通过这些简单的技术，我们基于 Caffe2 的系统可以使用批量大小为 8192 进行训练 ResNet-50 网络，该训练不仅在 256 块 GPU 上只花费一小时就能完成，同时还有匹配小批量的精度。在使用标准硬件从 8 到 256 块 GPU 调整时，我们的实现达到了 90% 以上的缩放效率（scaling efficiency）。该系统能使我们针对大型互联网数据高效地执行视觉识别任务。Facebook的大规模训练经验很有参考意义。同步SGD算法中，网络通信成为约束分布式系统的新瓶颈。杜克大学NIPS 2017 Oral论文[15] 提出TernGrad使用三元渐变来加速数据并行中的分布式深度学习。需要三个数值级别{-1,0,1}，这可以极大地减少通信时间。 同时在数学上证明了TernGrad在梯度约束下的收敛性。

**结论**

尽管深度学习（DL）比以往任何时候都更快地推进了世界的发展，但仍有许多方面值得我们去研究。我们仍然无法完全地理解深度学习，我们如何让机器变得更聪明，更接近或比人类更聪明，或者像人类一样学习。DL 一直在解决许多问题，例如难以训练，鲁棒性差，需要很大训练数据和较长的训练时间。未来神经网络鲁棒性会进一步提高，需要更高效网络以及将深度学习运用到各个方面，尽快落地。

参考文献

2. Young, Tom, et al. "Recent trends in deep learning based natural language processing." *ieee Computational intelligenCe magazine* 13.3 (2018): 55-75.
3. LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436.
4. Zhang Z, Geiger J, Pohjalainen J, et al. Deep learning for environmentally robust speech recognition: An overview of recent developments[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2018, 9(5): 49.
5. Zhu X X, Tuia D, Mou L, et al. Deep learning in remote sensing: a comprehensive review and list of resources[J]. IEEE Geoscience and Remote Sensing Magazine, 2017, 5(4): 8-36.
6. Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. arXiv preprint arXiv:1512.07108, 2015.
7. Wang H, Raj B. On the origin of deep learning[J]. arXiv preprint arXiv:1702.07800, 2017.
8. Goodfellow I. NIPS 2016 tutorial: Generative adversarial networks[J]. arXiv preprint arXiv:1701.00160, 2016.
9. Schmidhuber J. Deep learning in neural networks: An overview[J]. Neural networks, 2015, 61: 85-117.
10. Nielsen M A. Neural networks and deep learning[M]. USA: Determination press, 2015.
11. McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5(4): 115-133.
12. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological review, 1958, 65(6): 386.
13. Marvin Minsky and Seymour Papert, 1972 (2nd edition with corrections, first edition1969) *Perceptrons: An Introduction to Computational Geometry*, The MIT Press, Cambridge MA, ISBN 0-262-63022-2
14. Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533.
15. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. Neural networks, 1989, 2(5): 359-366.
16. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological review, 1958, 65(6): 386.
17. Nowlan S J，Hinton G E．Simplifying neural networks by soft weight-sharing[J]．Neural Computation，1992，4(4)： 473-493．
18. Srivastava N，Hinton G E，Krizhevsky A，et al．Dropout： a simple way to prevent neural networks from overfitting [J]．Journal of Machine Learning Research，2014，15： 1929-1958．
19. Nair V, Hinton G E, Farabet C. Rectified linear units improve restricted Boltzmann machines//Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel, 2010: 807-814.
20. Hagan M T, Demuth H B, Beale M H. Neural Network Design. Translated by Dai Kui, Beijing: China Machine Press, 2002(in Chinese).(Hagan M T,Demuth H B,Beale M H. 神经网络设计. 戴葵, 译. 北京: 机械工业出版社, 2002.)
21. Kiranyaz S, Ince T, Gabbouj M. Real-time patient-specific ECG classification by 1D convolutional neural networks. IEEE Transactions on Biomedical Engineering, 2016, 63(3): 664-675.
22. Hubel D H, Wiesel T N. Receptive fields binocular interaction, and functional architecture in the cat's visual cortex. Journal of Physiology, 1962, 160(1): 106-154.
23. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 1980, 36(4): 193-202.
24. Yoo H-J. Deep convolution neural networks in computer vision: A review. IEIE Transactions on Smart Processing and Computing, 2015, 4(1): 35-43.
25. LeCunY, Bottou L, BengioY, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
26. GaoLi-Gang, ChenPai-Yu, YuShi-Meng. Demonstration of convolution kernel operation on resistive cross-point array. IEEE Electron Device Letters, 2016, 37(7): 870-873.
27. Lecun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436-444.
28. Jin Lin-Peng. Study on Approach of Electrocardiogram Classification for Clinical Application[Ph.D. dissertation]. Suzhou Institute of Nano-tech and Nano-bionics, Chinese Academy of Sciences, Suzhou, 2016(in Chinese).(金林鹏. 面向临床应用的心电图分类方法研究[博士学位论文]. 中国科学院苏州纳米技术与纳米仿生研究所, 苏州, 2016.)
29. Xu Bing, Wang Nai-Yan, Chen Tian-Qi, et al. Empirical evaluation of rectified activations in convolution network. arXiv: 1505. 00853v2, 2015.
30. Jarrett K, Kavukcuoglu K, Marc'Aurelio Ranzato, et al. What is the best multi-stage architecture for object recognition?//Proceedings of the 2009 IEEE 12th International Conference on Computer Vision. Kyoto, Japan, 2009: 2146-2153.
31. Krizhevsky A, Sutskever II, Hinton G. Imagenet classification with deep convolutional neural networks//Proceedings of the Advances in Neural Information Processing Systems. Lake Tahoe, USA, 2012: 1097-1105.
32. Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828.
33. He Kai-Ming, Sun Jian. Convolutional neural networks at Constrained time cost//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, USA, 2015: 5353-5360.
34. GuJiu-Xiang, Wang Zhen-Hua, Jason Kuen, et al. Recent advances in convolutional neural networks. arXiv: 1512. 07108v5, 2017.
35. Boureau Y-L, Roux N L, Bach F, et al. Ask the locals: Multi-way local pooling for image recognition//Proceedings of the 2011 International Conference on Computer Vision. Barcelona, Spain, 2011: 2651-2658.
36. Zeiler M D, Fergus R. Stochastic pooling for regularization of deep convolutional neural networks. arXiv: 1301. 3557v1, 2013.
37. Boureau Y-L, Ponce J, LeCun Y. A theoretical analysis of feature pooling in visual recognition. International Conference on Machine Learning, 2010, 32(4): 111-118.
38. BoureauY-L, Bach F, LeCun Y, et al. Learning mid-level features for recognition//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. San Francisco, USA, 2010: 2559-2566.
39. Sainath T N, Mohamed A, Kingsbury B, et al .Deep convolutional neural networks for LVCSR//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver, Canada, 2013: 8614-8618.
40. O'Shea K, Nash R. An introduction to convolutional neural networks. arXiv: 1511. 08458v2, 2015.
41. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 2014, 15(6): 1929-1958.
42. Sainatha T N, Kingsburya B, Saona G, et al. Deep convolutional neural networks for large-scale speech tasks. Neural Networks, 2015, 64(SpecialIssue): 39-48.
43. Chu J L, Krzyzak A. Analysis of feature maps selection in supervised learning using convolutional neural networks//Proceedings of the 27th Canadian Conference on Artificial Intelligence. Montreal, Canada, 2014: 59-70.
44. Cao K, Jain A K. Latent orientation field estimation via convolutional neural network//Proceedings of the 2015 International Conference on Biometrics. Phuket, Thailand, 2015: 349-356.
45. Jin Lin-Peng, Dong Jun. Deep learning research on clinical electrocardiogram analysis. Science China: Information Sciences, 2015, 45(3): 398-416(in Chinese)(金林鹏, 董军. 面向临床心电图分析的深层学习算法研究. 中国科学: 信息科学, 2015, 45(3): 398-416)
46. Huang Jui-Ting, Li Jin-Yu, Gong Yi-Fan. An analysis of convolutional neural networks for speech recognition//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). South Brisbane, Australia, 2015: 4989-4993.
47. Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994.
48. C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.
49. W. L. Briggs, S. F. McCormick, et al. A Multigrid Tutorial. Siam, 2000.
50. K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC, 2011.
51. M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. IJCV, pages 303–338, 2010.
52. R. Girshick. Fast R-CNN. In ICCV, 2015.
53. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.
54. X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010.
55. I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. arXiv:1302.4389, 2013.
56. K. He and J. Sun. Convolutional neural networks at constrained time cost. In CVPR, 2015.
57. K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014.
58. K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015. [13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing coadaptation of feature detectors. arXiv:1207.0580, 2012.
59. S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Diploma thesis, TU Munich, 1991.
60. S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
61. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.
62. H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. TPAMI, 33, 2011.
63. H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. TPAMI, 2012.
64. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093, 2014.
65. A. Krizhevsky. Learning multiple layers of features from tiny images. Tech Report, 2009.
66. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012
67. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1989.
68. Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Muller. Efficient backprop. ¨ In Neural Networks: Tricks of the Trade, pages 9–50. Springer, 1998.
69. C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeplysupervised nets. arXiv:1409.5185, 2014.
70. M. Lin, Q. Chen, and S. Yan. Network in network. arXiv:1312.4400, 2013. [26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: Common objects in ´ context. In ECCV. 2014.
71. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
72. G. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of ´ linear regions of deep neural networks. In NIPS, 2014.
73. V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.
74. F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In CVPR, 2007.
75. T. Raiko, H. Valpola, and Y. LeCun. Deep learning made easier by linear transformations in perceptrons. In AISTATS, 2012.
76. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.
77. B. D. Ripley. Pattern recognition and neural networks. Cambridge university press, 1996.
78. A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In ICLR, 2015.
79. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. arXiv:1409.0575, 2014.
80. A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120, 2013.
81. N. N. Schraudolph. Accelerated gradient descent by factor-centering decomposition. Technical report, 1998. [38] N. N. Schraudolph. Centering neural network gradient factors. In Neural Networks: Tricks of the Trade, pages 207–226. Springer, 1998.
82. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.
83. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
84. R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. arXiv:1505.00387, 2015. [42] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. 1507.06228, 2015.
85. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.
86. R. Szeliski. Fast surface interpolation using hierarchical basis functions. TPAMI, 1990.
87. R. Szeliski. Locally adapted hierarchical basis preconditioning. In SIGGRAPH, 2006.
88. T. Vatanen, T. Raiko, H. Valpola, and Y. LeCun. Pushing stochastic gradient towards second-order methods–backpropagation learning with transformations in nonlinearities. In Neural Information Processing, 2013.
89. A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.
90. W. Venables and B. Ripley. Modern applied statistics with s-plus. 1999.
91. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In ECCV, 2014.
92. Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. In EMNLP, pages 1044–1054, 2013.
93. Alan Baddeley, Sergio Della Sala, and T.W. Robbins. Working memory and executive control [and discussion]. Philosophical Transactions of the Royal Society B: Biological Sciences, 351(1346):1397–1404, 1996.
94. Pierre Baldi and Gianluca Pollastri. The principled design of large-scale recursive neural network architectures–DAG-RNNs and the protein structure prediction problem. The Journal of Machine Learning Research, 4:575–602, 2003.
95. Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, 2005.
96. Justin Bayer, Daan Wierstra, Julian Togelius, and J¨urgen Schmidhuber. Evolving memory cell structures for sequence learning. In Artificial Neural Networks–ICANN 2009, pages 755–764. Springer, 2009.
97. Richard K. Belew, John McInerney, and Nicol N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. In In. Citeseer, 1990.
98. Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. Neural Networks, IEEE Transactions on, 5(2):157–166, 1994
99. . Yoshua Bengio, R´ejean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. The Journal of Machine Learning Research, 3:1137–1155, 2003.
100. Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. Advances in optimizing recurrent networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 8624– 8628. IEEE, 2013.
101. James Bergstra, Olivier Breuleux, Fr´ed´eric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy), volume 4, page 3. Austin, TX, 2010.
102. Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is NPcomplete. In Machine Learning: From Theory to Applications, pages 9–28. Springer, 1993.
103. Bob Carpenter. Lazy sparse stochastic gradient descent for regularized multinomial logistic regression. Alias-i, Inc., Tech. Rep, pages 1–20, 2008.
104. Ronan Collobert, Koray Kavukcuoglu, and Cl´ement Farabet. Torch7: A matlablike environment for machine learning. In BigLearn, NIPS Workshop, 2011
105. . Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Advances in Neural Information Processing Systems, pages 2933–2941, 2014.
106. Wim De Mulder, Steven Bethard, and Marie-Francine Moens. A survey on the application of recurrent neural networks to statistical language modeling. Computer Speech & Language, 30(1):61–98, 2015.
107. John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research, 12:2121–2159, 2011.
108. Charles Elkan. Learning meanings for sentences. http://cseweb.ucsd.edu/ ~elkan/250B/learningmeaning.pdf, 2015.
109. Accessed: 2015-05-18. Jeffrey L. Elman. Finding structure in time. Cognitive science, 14(2):179–211, 1990.
110. Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8):1915–1929, 2013.
111. Felix A. Gers. Long short-term memory in recurrent neural networks. Unpublished PhD dissertation, Ecole Polytechnique F´ed´erale de Lausanne, Lau- ´ sanne, Switzerland, 2001.
112. Felix A. Gers and J¨urgen Schmidhuber. Recurrent nets that time and count. In Neural Networks, 2000. IJCNN 2000,
113. Proceedings of the IEEE-INNS-ENNS International Joint Conference on, volume 3, pages 189–194. IEEE, 2000.
114. Felix A. Gers, J¨urgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. Neural computation, 12(10):2451–2471, 2000.
115. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume, volume 15, pages 315–323, 2011.
116. Yoav Goldberg and Omer Levy. word2vec explained: deriving Mikolov et al.’s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722, 2014.
117. Alex Graves. Supervised sequence labelling with recurrent neural networks, volume 385. Springer, 2012.
118. Alex Graves and J¨urgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5):602–610, 2005.
119. Alex Graves, Marcus Liwicki, Santiago Fern´andez, Roman Bertolami, Horst Bunke, and J¨urgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31(5):855–868, 2009.
120. Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing machines. arXiv preprint arXiv:1410.5401, 2014.
121. Frdric Gruau, L’universite Claude Bernard lyon I, Of A Diplome De Doctorat, M. Jacques Demongeot, Examinators M. Michel Cosnard, M. Jacques Mazoyer, M. Pierre Peretto, and M. Darell Whitley. Neural network synthesis using cellular encoding and the genetic algorithm., 1994.
122. Steven A. Harp and Tariq Samad. Optimizing neural networks with genetic algorithms. In Proceedings of the 54th American Power Conference, Chicago, volume 2, 2013.
123. Geoffrey E. Hinton. Learning distributed representations of concepts, 1986.
124. Sepp Hochreiter and Jurgen Schmidhuber. Bridging long time lags by weight guessing and “long short-term memory”. Spatiotemporal Models in Biological and Artificial Systems, 37:65–72, 1996
125. . Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.
126. Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and J¨urgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks, 2001.
127. John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79 (8):2554–2558, 1982.
128. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.
129. Michael I. Jordan. Serial order: A parallel distributed processing approach. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986.
130. Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/, 2015.
131. Accessed: 2015-08-13. Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. arXiv preprint arXiv:1412.2306, 2014.
132. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, pages 1097–1105, 2012
133. . John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. In Advances in Neural Information Processing Systems, pages 905– 912, 2009.
134. Yann Le Cun, B. Boser, John S. Denker, D. Henderson, Richard E. Howard, W. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. In Advances in Neural Information Processing Systems. Citeseer, 1990
135. . Zachary C. Lipton and Charles Elkan. Efficient elastic net regularization for sparse linear models. CoRR, abs/1505.06449, 2015.
136. URL http://arxiv. org/abs/1505.06449. Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize F1 measure. In Machine Learning and Knowledge Discovery in Databases, pages 225–239. Springer, 2014.
137. Marcus Liwicki, Alex Graves, Horst Bunke, and J¨urgen Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long shortterm memory networks. In Proc. 9th Int. Conf. on Document Analysis and Recognition, volume 1, pages 367–371, 2007
138. . Andrew L. Maas, Quoc V. Le, Tyler M. O’Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y. Ng. Recurrent neural networks for noise reduction in robust ASR. In INTERSPEECH. Citeseer, 2012.
139. Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-RNN). arXiv preprint arXiv:1412.6632, 2014.
140. James Martens and Ilya Sutskever. Learning recurrent neural networks with Hessian-free optimization. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 1033–1040, 2011.
141. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
142. Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 807–814, 2010. 35
143. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318.
144. Association for Computational Linguistics, 2002. Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1211.5063, 2012
145. . Barak A. Pearlmutter. Gradient calculations for dynamic recurrent neural networks: A survey. Neural Networks, IEEE Transactions on, 6(5):1212–1228, 1995.
146. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014), 12, 2014.
147. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.
148. Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on, 45(11):2673–2681, 1997.
149. Hava T. Siegelmann and Eduardo D. Sontag. Turing computability with neural nets. Applied Mathematics Letters, 4(6):77–80, 1991.
150. Yoram Singer and John C. Duchi. Efficient learning using forward-backward splitting. In Advances in Neural Information Processing Systems, pages 495– 503, 2009.
151. Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop, pages 1–9, 2010.
152. Richard Socher, Eric H. Huang, Jeffrey Pennin, Christopher D. Manning, and Andrew Y. Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In Advances in Neural Information Processing Systems, pages 801–809, 2011a.
153. Richard Socher, Cliff C. Lin, Chris Manning, and Andrew Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 129–136, 2011b.
154. Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 151–161.
155. Association for Computational Linguistics, 2011c. 36
156. Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. Transactions of the Association for Computational Linguistics, 2:207–218, 2014.
157. Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. arXiv preprint arXiv:1502.04681, 2015.
158. Ruslan L. Stratonovich. Conditional markov processes. Theory of Probability & Its Applications, 5(2):156–178, 1960.
159. Ilya Sutskever, James Martens, and Geoffrey E. Hinton. Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 1017–1024, 2011.
160. Ilya Sutskever, James Martens, George Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning (ICML-13), pages 1139–1147, 2013.
161. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 3104–3112, 2014.
162. Tijmen Tieleman and Geoffrey E. Hinton. Lecture 6.5- RMSprop: Divide the gradient by a running average of its recent magnitude. https://www. youtube.com/watch?v=LGA-gRkLEsI, 2012.
163. Alan M. Turing. Computing machinery and intelligence. Mind, pages 433–460, 1950.
164. Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence–video to text. arXiv preprint arXiv:1505.00487, 2015.
165. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3156–3164, 2015.
166. Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. Information Theory, IEEE Transactions on, 13(2):260–269, 1967. Paul J. Werbos. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78(10):1550–1560, 1990.
167. Wikipedia. Backpropagation — Wikipedia, the free encyclopedia, 2015.
168. URL http://en.wikipedia.org/wiki/Backpropagation. [Online; accessed 18- May-2015]. 37 Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. Neural Computation, 1(2):270–280, 1989.
169. Wojciech Zaremba and Ilya Sutskever. Learning to execute. arXiv preprint arXiv:1410.4615, 2014.
170. Matthew D. Zeiler. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012.
171. Matthew D. Zeiler, M. Ranzato, Rajat Monga, M. Mao, K. Yang, Quoc V. Le, Patrick Nguyen, A. Senior, Vincent Vanhoucke, Jeffrey Dean, et al. On rectified linear units for speech processing. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 3517– 3521. IEEE, 2013.
172. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in neural information processing systems. 2014: 2672-2680.
173. Mirza M, Osindero S. Conditional Generative Adversarial Nets[J]. Computer Science, 2014:2672-2680.
174. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.
175. Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks[C]//International Conference on Machine Learning. 2017: 214-223.
176. Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans[C]//Advances in Neural Information Processing Systems. 2017: 5767-5777.
177. Zhao J, Mathieu M, LeCun Y. Energy-based generative adversarial network[J]. arXiv preprint arXiv:1609.03126, 2016.
178. Berthelot D, Schumm T, Metz L. BEGAN: boundary equilibrium generative adversarial networks[J]. arXiv preprint arXiv:1703.10717, 2017.
179. Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[J]. arXiv preprint, 2017.
180. Kim T, Cha M, Kim H, et al. Learning to discover cross-domain relations with generative adversarial networks[J]. arXiv preprint arXiv:1703.05192, 2017.
181. Yi Z, Hao (Richard) Zhang, Tan P, et al. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation[C]//ICCV. 2017: 2868-2876.
182. Choi Y, Choi M, Kim M, et al. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation[J]. arXiv preprint, 2017, 1711.
183. Ledig C, Theis L, Huszár F, et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network[C]//CVPR. 2017, 2(3): 4.
184. Bai Y, Zhang Y, Ding M, et al. Finding tiny faces in the wild with generative adversarial network[J]. CVPR. IEEE, 2018.
185. Azadi S, Fisher M, Kim V, et al. Multi-content gan for few-shot font style transfer[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, 11: 13.
186. Karras T, Aila T, Laine S, et al. Progressive growing of gans for improved quality, stability, and variation[J]. arXiv preprint arXiv:1710.10196, 2017.
187. Miyato T, Kataoka T, Koyama M, et al. Spectral normalization for generative adversarial networks[J]. arXiv preprint arXiv:1802.05957, 2018.
188. Wu J, Zhang C, Xue T, et al. Learning a probabilistic latent space of object shapes via 3dgenerative-adversarial modeling[C]//Advances in Neural Information Processing Systems. 2016: 82-90.
189. Li R, Pan J, Li Z, et al. Single Image Dehazing via Conditional Generative Adversarial Network[J]. methods, 2018, 3: 24.
190. Chen Y S, Wang Y C, Kao M H, et al. Deep Photo Enhancer: Unpaired Learning for Image Enhancement From Photographs With GANs[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 6306-6314.
191. Kahng M, Thorat N, Chau D H P, et al. GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation[J]. IEEE transactions on visualization and computer graphics, 2018.
192. Ma S, Fu J, Chen C W, et al. DA-GAN: Instance-level Image Translation by Deep Attention Generative Adversarial Networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 5657-5666.
193. Qian R, Tan R T, Yang W, et al. Attentive Generative Adversarial Network for Raindrop Removal from A Single Image[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2482-2491.
194. Yu L, Zhang W, Wang J, et al. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient[C]//AAAI. 2017: 2852-2858.
195. Gao Y, Singh R, Raj B. Voice Impersonation using Generative Adversarial Networks[J]. arXiv preprint arXiv:1802.06840, 2018.
196. Kameoka H, Kaneko T, Tanaka K, et al. StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks[J]. arXiv preprint arXiv:1806.02169, 2018.
197. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
198. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.
199. Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.
200. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015.
201. Anil R, Pereyra G, Passos A, et al. Large scale distributed neural network training through online distillation[J]. arXiv preprint arXiv:1804.03235, 2018.
202. Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to
203. Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv preprint arXiv:1510.00149, 2015.
204. He Y, Zhang X, Sun J. Channel pruning for accelerating very deep neural networks[C]//International Conference on Computer Vision (ICCV). 2017, 2(6).
205. Liu Z, Li J, Shen Z, et al. Learning efficient convolutional networks through network slimming[C]//Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017: 2755-2763.
206. Chollet F. Xception: Deep learning with depthwise separable convolutions[J]. arXiv preprint, 2017: 1610.02357.
207. Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks[C]//Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, 2017: 5987-5995.
208. Hluchyj M G, Karol M J. Shuffle Net: An application of generalized perfect shuffles to multihop lightwave networks[J]. Journal of Lightwave Technology, 1991, 9(10): 1386-1397.
209. Zhenhua Liu, Jizheng Xu, Xiulian Peng, Ruiqin Xiong . **Frequency-Domain Dynamic Pruning for Convolutional Neural Networks** [J]. Neonatal network: NN, 1993, 12(6): 59-66.
210. Goyal P, Dollár P, Girshick R, et al. Accurate, large minibatch SGD: training imagenet in 1 hour[J]. arXiv preprint arXiv:1706.02677, 2017.
211. Wen W, Xu C, Yan F, et al. Terngrad: Ternary gradients to reduce communication in distributed deep learning[C]//Advances in neural information processing systems. 2017: 1509-1519.