基于条件约束的胶囊生成对抗网络

孔 锐 [1, 2] 黄 钢 [2]

摘 要 生成式对抗网络 (Generative adversarial networks, GAN) 是主要的以无监督方式学习深度生成模型的方法之一. 基于可微生成器网络的生成式建模方法, 是目前最热门的研究领域, 但由于真实样本分布的复杂性, 导致 GAN 生成模型在训练过程稳定性、生成质量等方面均存在不少问题. 在生成式建模领域, 对网络结构的探索是重要的一个研究方向, 本文利用胶囊神经网络 (Capsule networks, CapsNets) 重构生成对抗网络模型结构, 在训练过程中使用了 Wasserstein GAN (WGAN) 中提出的基于 Earth-mover 距离的损失函数, 并在此基础上加以条件约束来稳定模型生成过程, 从而建立带条件约束的胶囊生成对抗网络 (Conditional-CapsuleGAN, C-CapsGAN). 通过在 MNIST 和 CIFAR-10 数据集上的多组实验, 结果表明将 CapsNets 应用到生成式建模领域是可行的, 相较于现有类似模型, C-CapsGAN 不仅能在图像生成任务中稳定生成高质量图像, 同时还能更有效地抑制模式坍塌情况的发生.

关键词 生成式对抗网络, 胶囊神经网络, 图像生成, 条件模型

引用格式 孔锐, 黄钢. 基于条件约束的胶囊生成对抗网络. 自动化学报, 2020, **46**(1): 94−107

DOI 10.16383/j.aas.c180590

Conditional Generative Adversarial Capsule Networks

KONG Rui[1, 2] HUANG Gang[2]

Abstract Generative adversarial networks (GAN) is one of the main methods of learning deep generative models in an unsupervised fashion. The generative modeling method based on a network of differential generators is the hottest research field, but due to the complexity of the real sample distribution, the GAN has many problems in the stability of training process and the quality of generation. In the field of generative modeling, the exploration of network structure is an important research direction. And in this paper, we use the capsule networks (CapsNets) to reconstruct the structure of GAN, and the loss function based on Earth-mover distance proposed in Wasserstein GAN (WGAN) is used in the training process, and then we add the conditional constraints to stabilize the model generation process. According to the above, a conditional generative adversarial capsule networks (C-CapsGAN) is established. The results of multiple simulation experiments on the MINIST and CIFAR-10 datasets show that it is feasible to apply the CapsNets to the generative modeling field. Besides, compared with the existing similar model, C-CapsGAN can not only stably produce high-quality images in the image generation task, but also inhibit the occurrence of pattern collapse more effectively.

Key words Generative adversarial networks (GAN), Capsule networks (CapsNets), image generation, conditional model

Citation Kong Rui, Huang Gang. Conditional generative adversarial capsule networks. *Acta Automatica Sinica*, 2020, **46**(1): 94−107

生成式对抗网络 (Generative adversarial networks, GAN)[1] 由 Goodfellow 等在 2014 年提出, 该理论基于博弈论场景, 其中生成器网络通过与对手竞争来学习变换由某些简单的输入分布 (通常是标准多变量正态分布或者均匀分布) 到图像空间的分布 — 即越来越真实的样本 $x = g(z; \theta^{(G)})$. 作为对手, 判别器网络则试图区分从训练数据抽取的样本和从生成器中生成的样本. 训练过程根据判别器网络发出的由 $d(x; \theta^{(D)})$ 给出的概率值 — 指示是真实训练样本而不是从模型中抽取的伪造样本的概率 — 来指导生成器网络不断构造出越来越真的假样本[2]. 通过这种方式, 二者互相竞争, 共同进步, 生成模型产生的数据越来越真, 判别模型的识别能力越来越强, 整体来说, 双方都试图最小化各自的损失, 优化的最终目标是达到 "纳什均衡"[3]. 近些年来, GAN 已经成功地运用到很多问题上, 但训练 GAN 依旧是一件比较困难的事, 稳定 GAN 的学习是一个开放性的问题. 幸运的是当仔细选择模型架构和超参数时, GAN 的学习效果非常好, Radford 等在卷积神经网络 (Convolutional neural network,

收稿日期 2018-09-07 录用日期 2019-01-14
Manuscript received September 7, 2018; accepted January 14, 2019
广东省科技计划 (产学研合作) 项目 (2016B090918098) 资助
Supported by GuangDong Science and Technology Project Fund (2016B090918098)
本文责任编委 金连文
Recommended by Associate Editor JIN Lian-Wen
1. 暨南大学智能科学与工程学院 珠海 519070 2. 暨南大学信息科学技术学院 广州 510632
1. School of Intelligent Systems Science and Engineering, Jinan University (Zhuhai Campus), Zhuhai 519070 2. College of Information Science and Technology, Jinan University, Guangzhou 510632

CNN) 的基础上设计了 DCGAN[4], 它将生成器中的全连接层用反卷积 (Deconvolution) 层代替, 在图像合成的任务中取得了非常好的表现, 并表明其潜在的表示空间能捕获到变化的重要因素. 在 GAN 的学习过程中通过将生成过程分为许多级别的细节能极大地提高 GAN 生成样本的质量, Mirza 等提出通过训练有条件的 GAN (Conditional GAN, CGAN)[5] 来引导 GAN 学习从分布 $p(x|y)$ 中采样, 而不是简单地从边缘分布 $p(x)$ 中采样, 使得 GAN 能着重关注那些能够阐述样本相关的统计特征, 并忽略不太相关的局部特征. 然而, 无论是精心设计的 DCGAN 还是条件驱动的 CGAN, 均是通过丰富的工程手段来优化 GAN 模型, 2017 年, Arjovsky 等[6] 证明了当使用 JS (Jensen-Shannon) 散度作为真实分布与生成分布相近度的度量时, 在真实分布与生成分布的重叠区域可忽略的情况下, JS 散度为一常数, 此时生成器的获得梯度为 0, 网络无法继续优化, 继而提出了使用 Earth-Mover (EM) 距离作为相似度度量的 Wasserstein GAN[7], 为解决 GAN 存在的训练困难、损失函数无法指导训练、生成样本缺乏多样性等问题指明了一个全新的方向.

网络结构的创新和优化是 GAN 得以不断发展的重要原因之一[8]. 在深度学习领域, 从 LeCun 等提出 CNN 模型[9] 开始, 因为计算成本等问题, 此类深度模型一度被掩盖在诸如支持向量机 (Support vector machine, SVM) 等其他机器学习模型的光彩之下. 直到 Krizhevsky 等在 2012 年提出 AlexNet[10] 网络模型, 并藉此取得了当年 ILSVRC 比赛分类项目的冠军之后, 深度卷积神经网络才重新回到了大众的视野, 从这之后, 关于 CNN 的研究和应用层出不穷, 掀起了一个深度学习研究的热潮. CNN 通过稀疏权重、参数共享和池化等技术在完成了对图像像素中的重要特征的检测之外还极大地减少了网络的参数规模. 但是 Hinton 认为 CNN 的内部数据表示并没有考虑到简单和复杂对象之间的重要空间层级, 并依此提出了一种基于动态路由的胶囊神经网络 (CapsNets)[11]. CapsNets 使用 "胶囊" — 封装了多个卷积核能输出包含编码特征之间相对空间关系的网络单元 — 改变了 CNN 的 "神经元" 活动中存在的视角不变性的特点. 也就是说 CapsNets 能够习得输入特征之间的空间结构关系, 意味着该结构下的神经活动将随着物体在图像中的 "外观流形上的移动" 而改变, 与此同时使检测概率保持恒定 — 即让网络具备活动等变性特征[12].

鉴于胶囊网络结构作为有别于传统 CNN 而提出的一种全新的神经网络结构, 本文将使用 CapsNets 来替代标准的 CNN 作为 GAN 中判别器的框架, 并对图像数据进行建模, 藉此通过实验来验证 CapsNets 在 GAN 领域应用的可能性, 同时在网络中加入条件约束, 和使用 WGAN 的改进版本 WGAN-GP[13] 来指导 GAN 的训练过程. 本文提出的基于带条件约束的胶囊生成对抗网络 (C-CapsGAN) 在不同的图像数据集上都能够稳定生成高质量的图像, 同时通过实验与其他常见的几种 GAN 进行了对比, 发现 C-CapsGAN 还能够有效减少模式坍塌问题的发生.

1 生成式对抗网络算法原理

1.1 GAN

GAN 由两部分构成, 生成模型 G 和判别模型 D, 生成模型 G 捕捉真实数据样本的潜在分布, 并生成新的数据样本; 判别模型 D 是一个二分类器, 判别输入是真实数据还是生成的样本. 形象化表示生成式对抗网络中学习的最简单的方式是零和游戏, 其中函数 $V(\theta^{(G)}, \theta^{(D)})$ 定义判别器的收益. 生成器接收 $-V(\theta^{(G)}, \theta^{(D)})$ 作为它自己的收益. 在学习期间, 两位玩家都试图去最大化各自的收益, 因此收敛在:

$$G^* = \arg \min_G \max_D V(G, D) \tag{1}$$

则 V 的默认选择是 (其中, P_{data} 和 P_G 分别表示真实样本概率分布和模型分布):

$$\begin{aligned} V(\theta^{(G)}, \theta^{(D)}) = &\mathrm{E}_{x\sim P_{\text{data}}} [\log\ D(x)] + \\ &\mathrm{E}_{x\sim P_G} [\log\ (1 - D(x))] \end{aligned} \tag{2}$$

这驱使判别器试图学习将样本正确地分类为真或者是假, 同时, 生成器试图欺骗分类器让其相信样本是真实的. 最终在收敛处, 生成器的样本与真实样本不可区分, 并且判别器处处都输出1/2.

GAN 的结构如图 1 所示, 其核心算法描述如下:

生成器与判别器的训练过程是交替进行的, 更新一方的参数时, 另一方的参数固定住不更新. 根据目标公式 $V(\theta^{(G)}, \theta^{(D)})$, 做法是:

1) 首先固定生成器 G, 找到一个 D^* 使得 $V(\theta^{(G)}, \theta^{(D)})$ 最大, 即 $\max_D V(G, D)$, 通过判别器尽可能地将生成图片和真实图片区别开来, 也就是要最大化两者之间的交叉熵;

2) 然后固定判别器 D, 找出使得 $\max_D V(G, D)$ 最小的 G^*, 既代表最好的生成器.

相比其他生成架构, 设计 GAN 的主要动机是学习过程既不需要近似推断, 也不需要配分函数梯度的近似. 当 $\max_D V(G, D)$ 在 $\theta^{(G)}$ 中是凸性的时候, 该学习过程保证收敛且渐近一致. 然而在实践中由神经网络表示的 G 和 D 以及 $\max_D V(G, D)$ 凹

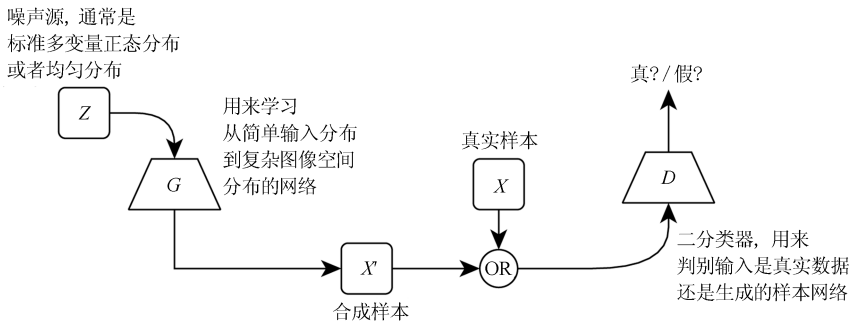


图 1 GAN 架构图

Fig. 1 GAN architecture diagram

凸性往往是不确定的, 这就导致 GAN 的学习过程会变得很困难.

1.2 DCGAN、CGAN 和 WGAN

DCGAN 将卷积网络引入 GAN 的结构, 并通过对网络拓扑结构和超参数的精心设计, 使得 DCGAN 在图像合成任务上表现非常好, DCGAN 相比于原始 GAN 有以下特点[4]:

1) 在判别器网络中使用带步幅的卷积层 (Strided convolutions) 替换传统卷积神经网路中的池化层 (Pooling), 并在生成器网络中使用微步幅卷积层 (Fractionally-strided convolution) 完成从随机噪声到图片的生成过程;

2) 在判别器网络和生成器网络中均使用批量归一化 (Batch Normalization), 此举通过对隐藏层各神经元的输入作标准化处理, 能够提高神经网络训练速度. 同时可以使前面层的权重变化对后面层造成的影响减小, 整体网络更加健壮;

3) 移除全连接层, 此举以牺牲网络收敛性来增加模型稳定性;

4) 判别器网络中的所有层使用 Leaky ReLU 激活函数. 生成器网络中除了输出层以外都使用 ReLU 激活函数, 而输出层则使用 Tanh 激活函数.

CGAN 针对 GAN 本身不可控的缺点, 在生成模型 G 和判别模型 D 的建模中加入条件 C 监督信 (条件 C 可以是任意信息, 比如类别信息或者其他模态信息) 以指导 GAN 网络生成, 相应的 CGAN 的目标函数 $V(\theta^{(G)}, \theta^{(D)})$ 修改成:

$$V(\theta^{(G)}, \theta^{(D)}) = \mathrm{E}_{x\sim P_{\mathrm{data}}}\left[\log D(x|C)\right] + \mathrm{E}_{x\sim P_G}\left[\log(1 - D(x|C))\right] \quad (3)$$

由式 (3) 可知, 除了加入条件监督信息之外, 其他均和原始 GAN 一致, 图 2 是 CGAN 的架构图.

当条件 C 被定义为标签 y 的时候, 则可以认为 CGAN 是将无监督的 GAN 模型变为有监督模型的改进.

WGAN 针对原始 GAN 定义的目标函数的优化过程可以等价为最小化真实分布 P_{data} 和生成分布 P_G 之间的 JS 散度, 而存在当 P_{data} 和 P_G 的支撑集是高维空间中的低维流形时, 两者重叠部分测度为 0 的概率为 1, 使得 JS 散度固定为常数从而无法指导梯度下降的问题, 提出对样本分布进行限制的方法, 即通过假设样本服从某类特殊的函数族以避免梯度消失.

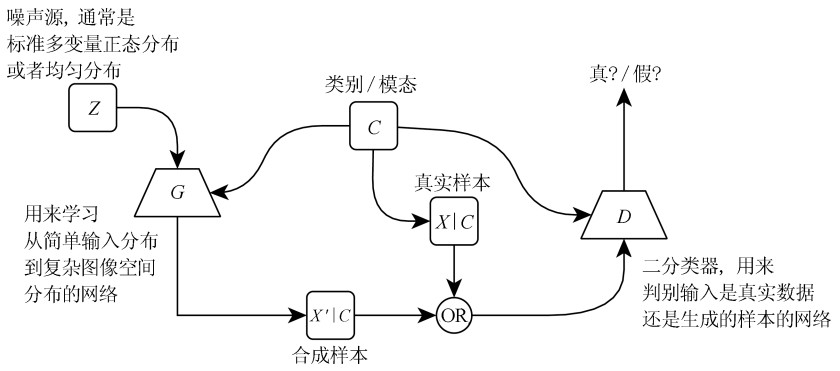


图 2 CGAN 架构图

Fig. 2 CGAN architecture diagram

WGAN 使用 EM 距离替换 JS 散度作为真实分布与生成分布相近度的度量[14], 定义如下 ($\prod$ 是 P_{data} 和 P_G 组合起来的所有可能的联合分布的集合):

$$W(\theta^{(G)},\theta^{(D)}) = \min_{\gamma\in\prod}\sum_{x_G,x_D}\gamma(x_G,x_D)\|x_G-x_D\| \tag{4}$$

由于取最小值的操作无法直接求解, 根据 Kantorovich-Rubinstein 对偶性, EM 距离被转化为如下的形式来近似求解:

$$W(\theta^{(G)},\theta^{(D)}) \approx L = \max_{f\in 1-\text{Lipschitz}}\{\mathrm{E}_{x\sim P_{\text{data}}}[f(x)]-\mathrm{E}_{x\sim P_G}[f(x)]\} \tag{5}$$

上式中 $f(\cdot)$ 是一个满足 Lipschitz 连续条件的函数. 因此可以使用神经网络对 $f(\cdot)$ 进行拟合, 使得 L 尽可能取到最大, 此时 L 就会近似真实分布与生成分布之间的 Wasserstein 距离, 因此 WGAN 的目标函数变为:

$$V(\theta^{(G)},\theta^{(D)}) = \mathrm{E}_{x\sim P_{\text{data}}}[C(x)] - \mathrm{E}_{\widetilde{x}\sim P_G}[C(\widetilde{x})] \tag{6}$$

在原始 WGAN, 作者采用权重剪枝的方式使得判别函数 C 满足 Lipschitz 连续条件. 而 WGAN 的改进版本 WGAN-GP 则使用梯度惩罚的方式从而进一步提高了网络的稳定性, 使得在多种网络结构上都可实现收敛, 性能优越, WGAN-GP 的目标函数为:

$$V(\theta^{(G)},\theta^{(D)}) = \mathrm{E}_{x\sim P_{\text{data}}}[C(x)] - \mathrm{E}_{\widetilde{x}\sim P_G}[C(\widetilde{x})] + \lambda\mathrm{E}_{\widetilde{x}\sim P_{\widetilde{x}}}\left[\left(\|\bigtriangledown_{\widetilde{x}}D(\widetilde{x})\|_2-1\right)^2\right] \tag{7}$$

2 胶囊神经网络 (CapsNets)

胶囊 (Capsule) 是一组用激活向量来表示一种特定类型实体的实例化参数的神经元, 或者称之为向量神经元. 和 CNN 在层间直接使用重复特征检测器进行卷积运算从而学习得到关于输入分布的有用特征并泛化到其他分布不同, CapsNets 使用向量输出神经元 (激活向量的长度 (限定在 0 到 1 之间) 描述特征检测的概率, 方向表征对应特征的状态 (位置, 颜色, 方向, 形状等)) 和按协议路由的最大池化替代 CNN 的标量输出特征检测器, 使得网络习得对象特征的同时, 还存储了对象特征之间的分层姿态关系.

Sabour 等[9] 在论文中使用了一种动态路由机制. 该机制通过在连续胶囊层中的两两信息传递来实现深层神经网络中胶囊层之间的交互[15]. 对于在第 l 层的每一个胶囊 $h_i^{(l)}$ 和在第 $l+1$ 层的每个胶囊 $h_j^{(l+1)}$ 之间, 耦合系数 C_{ij} 基于 h_i 对 h_j 的输出预测与其实际输出之间的一致性进行迭代调整, 换句话说, 对于每个低层胶囊 $h_i^{(l)}$ 而言, 其权重 C_{ij} 定义了传给每个高层胶囊 $h_j^{(l+1)}$ 的输出的概率分布. 特别的, 对于 CapsNets 结构中的第三层 DigitCaps 层, 胶囊的个数由任务的具体内容而定, 即对于包含 K 类分类的任务而言, 通常设计为具有 K 个胶囊, 其中每个胶囊代表一个类别. 由于胶囊矢量输出的长度代表视觉实体的存在, 因此最后一层中每个胶囊的长度可被视为图像属于特定类 k 的概率.

CapsNets 编码器架构如图 3 所示[9].

第一层, 是一个常规的卷积层, 在 CapsNets 中, 卷积层由 256 个 $(9,9)$ 大小, 步长为 1 的卷积核构成, 使用 ReLU 激活函数, 生成 $(20,20,256)$ 张量. 该卷积层的作用在于先对输入图片作低级特征抽取预处理.

第二层, PrimaryCaps 层, 这一层包含 32 个主胶囊, 接受卷积层检测到的基本特征, 生成特征的组合. 首先, 将 32 个 $(9,9)$ 大小, 步长为 2 的卷积核应用到 $(20,20,256)$ 输入张量, 生成 $(6,6,32)$ 的输出张量; 然后, 将 $(6,6,32)$ 张量维度转换为 $(6,6,1,32)$. 重复使用上述卷积操作最终生成 8 个 $(6,6,1,32)$ 的张量, 因此最终输出为 $(6,6,8,32)$ 张量.

第三层, DigitCaps 层, 在原始论文中将这一层设置为了 10 个数字胶囊, 但其实对于包含 K 类分类的任务而言, 该层可设计为具有 K 个胶囊, 其中每个胶囊代表一个类别. 胶囊接受一个 $(6,6,8,32)$ 张量作为输入, 并通过 $(8,16)$ 权重矩阵将 8 维输入空间映射到 16 维胶囊输出空间, 输出为 $(10,16)$ 矩阵.

3 基于条件约束的胶囊生成对抗网络 (C-CapsGAN)

本文利用 CapsNets 来改进 DCGAN 模型结构的判别器, 然后将条件同时添加到判别器和生成器中负责约束训练, 同时利用 WGAN-GP 损失函数对训练过程进行指导, 从而建立一种带条件约束的胶囊生成对抗网络 (Conditional-CapsuleGAN, C-CapsGAN), 并将通过实验验证该结构在生成图像方面的可行性, 同时探索了胶囊网络对解决训练过程收敛性和模式坍塌等问题的效果, 网络整体的架构如图 4.

仿照 DCGAN 中的生成器结构, C-CapsGAN 的生成模型如图 5 所示 (以 MNIST 手写数字数据集为例):

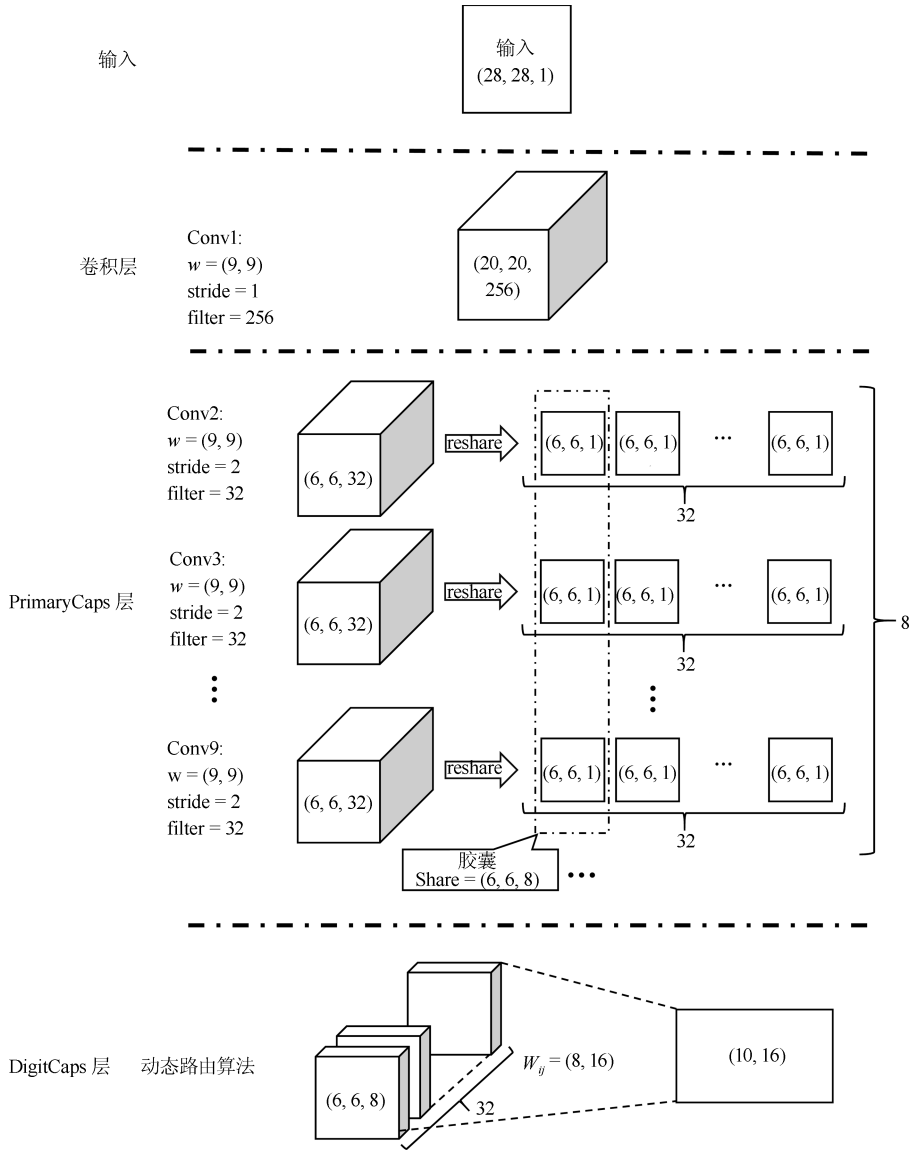


图 3 CapsNets 编码器结构

Fig. 3 CapsNets encoder

1) 将 100 维的随机噪声和 10 维的类别信息拼接 (Concat) 成 110 维的输入数据;

2) 将 110 维的输入数据经过全连接层得到一个 1 024 维输出并与 10 维的类别信息进行拼接得到 1 034 维的数据;

3) 将 1 034 维作为输入通过全连接层得到一个 6 272 维的输出然后维度转换 (Reshape) 成 $(7, 7, 128)$ 的三维张量, 再与 $(1, 1, 10)$ 的三维张量类别信息进行拼接得到 $(7, 7, 138)$ 的张量;

4) 将 $(7, 7, 138)$ 的张量通过卷积核大小为 $(5, 5)$、步长为 2 的转置卷积层, 输出 $(14, 14, 128)$ 的张量, 继续与 $(1, 1, 10)$ 的三维张量类别信息进行拼接得到 $(14, 14, 138)$ 的张量;

5) 将 $(14, 14, 138)$ 的张量重复执行一次上一步操作则输出 $(28, 28, 1)$ 张量, 即为一个生成样本;

C-CapsGAN 的判别模型则使用了 CapsNets 的架构, 如图 6 所示 (以 MNIST 手写数字数据集为例).

1) 将一张 $(28, 28, 1)$ 的样本和 $(1, 1, 10)$ 的三维张量类别信息拼接作为判别器的输入;

2) $(28, 28, 11)$ 的输入数据通过卷积核大小为 $(9, 9)$、步长为 1 的卷积层, 输出 $(20, 20, 256)$ 的张量, 再与 $(1, 1, 10)$ 的三维张量类别信息进行拼接得到 $(20, 20, 266)$ 的张量;

3) 在 PrimaryCaps 层中, 以 $(20, 20, 266)$ 的张量作为输入, 通过 8 组卷积运算, 每组运算由 32 个 $(9, 9)$ 大小、步长为 2 的卷积核执行, 得到 $(6, 6, 32)$ 的张量再与 $(1, 1, 10)$ 的条件约束拼接得到 $(6, 6, 42)$

张量, 最终得到 $(6,6,8,42)$ 的输出张量, 再维度转换为 $(1\,512,8,1)$ 作为 DiscriCaps 层输入;

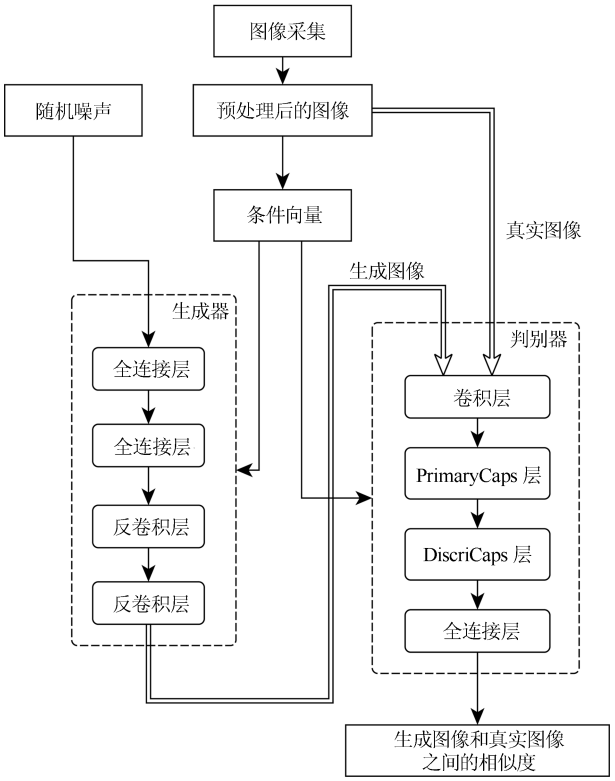


图 4　C-CapsGAN 架构图

Fig. 4　The structure of C-CapsGAN

4) 在 DiscriCaps 层, 该层作为二分类的输出, 因此设置为 1 个胶囊即可, 胶囊接受 $(1\,512,8,1)$ 张量作为输入, 并通过 $(8,16)$ 权重矩阵, 利用动态路由更新算法将 8 维输入空间映射到 16 维胶囊输出空间, 输出为 $(1,16)$ 矩阵, 拼接条件约束之后再通过激活函数得到判别结果.

C-CapsGAN 的模型训练. 判别器的任务不再是尽力区分生成样本与真实样本, 而是尽量拟合出样本间的 Wasserstein 距离, 从分类任务转化成回归任务. 而生成器的任务则是尽力缩短样本间的 Wasserstein 距离.

$$V(\theta^{(G)},\theta^{(D)}) = \mathrm{E}_{x\sim P_{\mathrm{data}}}\left[C(x)\right] - \mathrm{E}_{\widetilde{x}\sim P_G}\left[C(\widetilde{x})\right] + \lambda \mathrm{E}_{\widetilde{x}\sim P_{\widetilde{x}}}\left[\left(\left\|\bigtriangledown_{\widetilde{x}} D(\widetilde{x})\right\|_2 - 1\right)^2\right] \quad (8)$$

由于模型是对每个样本独立地施加梯度惩罚, 所以判别器的模型架构中不使用 Batch normalization, 因为它会引入同一个 Batch 中不同样本的相互依赖关系. 在求得生成器和判别器的损失函数之后, 选择 Adam 作为优化器.

4　实验与分析

本文在 MNIST 数据集和 CIFAR-10 数据集上进行实验分析, 实验环境: 计算机处理器为 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz, 64 GB 运行内存 (RAM), NVIDIA Quadro M1200 GPU, TensorFlow 框架.

4.1　MNIST 实验

MNIST 数据集包含 0～9 共 10 类手写数字灰度图像, 图像尺寸为 $(28,28,1)$, 整个数据集有 60 000 个训练样本, 10 000 个测试样本[16].

在训练过程中, 使用图 5 和图 6 的结构作为 C-CapsGAN 的生成器和判别器对 MNIST 数据集进行训练, 同时为了保持对抗平衡, 设置判别器与生成器的迭代次数为 1 : 2, 其他参数设置如下:

1) Batch 设置为 64;

2) Epoch 设置为 25 轮;

3) Adam 优化器的学习速率设置为 0.0001, 一阶矩估计的指数衰减率为 0.5, 二阶矩估计的指数衰减率为 0.9;

4) 生成器除了最后一层使用 Sigmoid 函数作为激活函数之外, 其他层都使用 ReLU 作为激活函数;

5) 判别器则使用 Leaky ReLU 作为激活函数;

6) 生成器每一层中均使用 Batch normalization 对隐含层的输入进行批量归一化处理[17].

因为每个胶囊产生一个向量输出而不是单纯的标量输出, 同时, 由于每个胶囊都有与它前面层中的所有胶囊相关联的用于对其输出进行预测的附加参数[11]. 所以, 胶囊的个数能够影响判别器的判别性能, 于是在 MNIST 实验过程中, 以 MNIST 数据集为实验对象, 通过逐步减少胶囊的个数从而探究胶囊个数与样本生成质量之间关系, 然后把最优胶囊个数的 C-CapsGAN 从训练过程和生成结果方面与传统 GAN、DCGAN 进行对比.

4.1.1　MNIST 实验结果

实验部分先逐步降低胶囊个数, 然后将效果最好的 C-CapsGAN 和传统 GAN、DCGAN 进行对比.

1) 首先, 将胶囊判别器的结构参数设置为和胶囊网络 CapsNets 相一致, 也就是 PrimaryCaps 层的胶囊个数设置为 32 个. 于是图 7 和图 8 分别表示了 C-CapsGAN 的判别器的损失函数 (d_loss) 和生成器的损失函数 (g_loss) 随训练次数增加而变化的情况.

在趋势上, 判别器的损失函数处于缓慢下降状态, 生成器的损失函数处于缓慢上升状态. 然而从 d_loss 趋势图发现, 判别器的损失总是很高, 意味着

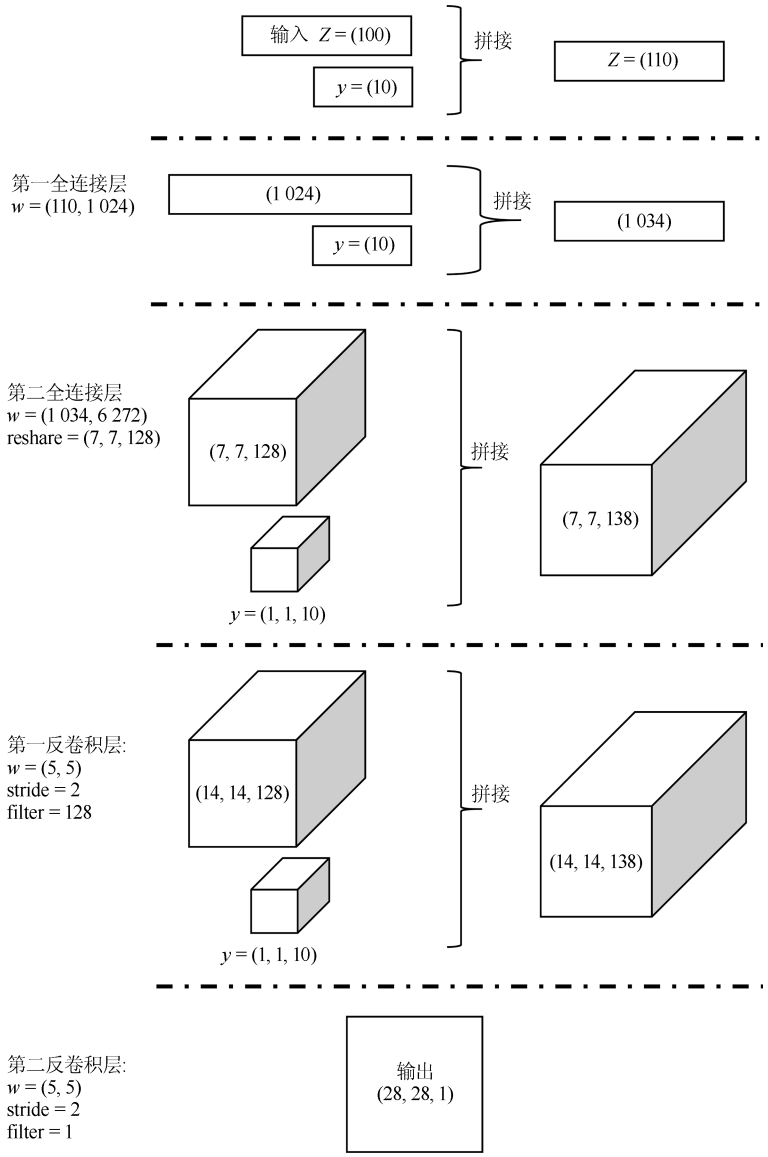


图 5 C-CapsGAN 生成器结构

Fig. 5 The structure of C-CapsGAN generator

它总是能将生成样本判定为假样本, 这对生成器而言, 意味着无法找到能够欺骗判别器的参数集, 使得最终呈现的效果就是: 生成样本无法往更优的方向发展. 图 9 为数据集的其中一个 Batch, 即 64 个生成样本随着 Epoch 次数的增加, 生成样本的进化情况, 可以看出收敛效果不是很好, 当训练结束之后, 生成样本的质量还有很大的提升空间.

2) 然后, 降低胶囊判别器的中 PrimaryCaps 层的胶囊个数, 此次设置为 24 个. 图 10 和图 11 分别表示了 C-CapsGAN 的判别器的损失函数 (d_loss) 和生成器的损失函数 (g_loss) 随训练次数增加而变化的情况.

此时在趋势上发现, 在迭代 10 000 次之前, 判别器的损失函数和生成器的损失函数的状态和当 PrimaryCaps 层的胶囊个数为 32 的时候类似, 训练前期, 双方均处于成长阶段, 呈现出图中大幅震荡状态. 当迭代超过 10 000 次之后, 判别器损失出现明显下降, 并在 0 ∼ 1 之间小幅震荡, 意味着通过减少胶囊个数, 成功降低了判别器的判别能力, 同时由于动态路由算法, 即使存在有梯度消失风险, 也无需担心判别器出现失效情况. 图 12 为数据集的其中一个 Batch, 即 64 个生成样本随着 Epoch 次数的增加, 生成样本的进化情况, 可以看出当胶囊个数变少之后, 生成器的收敛速度明显加快, 所以降低胶囊个数有助于生成器提高生成质量.

3) 实验继续降低胶囊判别器的中 PrimaryCaps 层的胶囊个数, 此次设置为 16 个. 图 13 和图 14 分别表示了 C-CapsGAN 的判别器的损失函数 (d_loss)

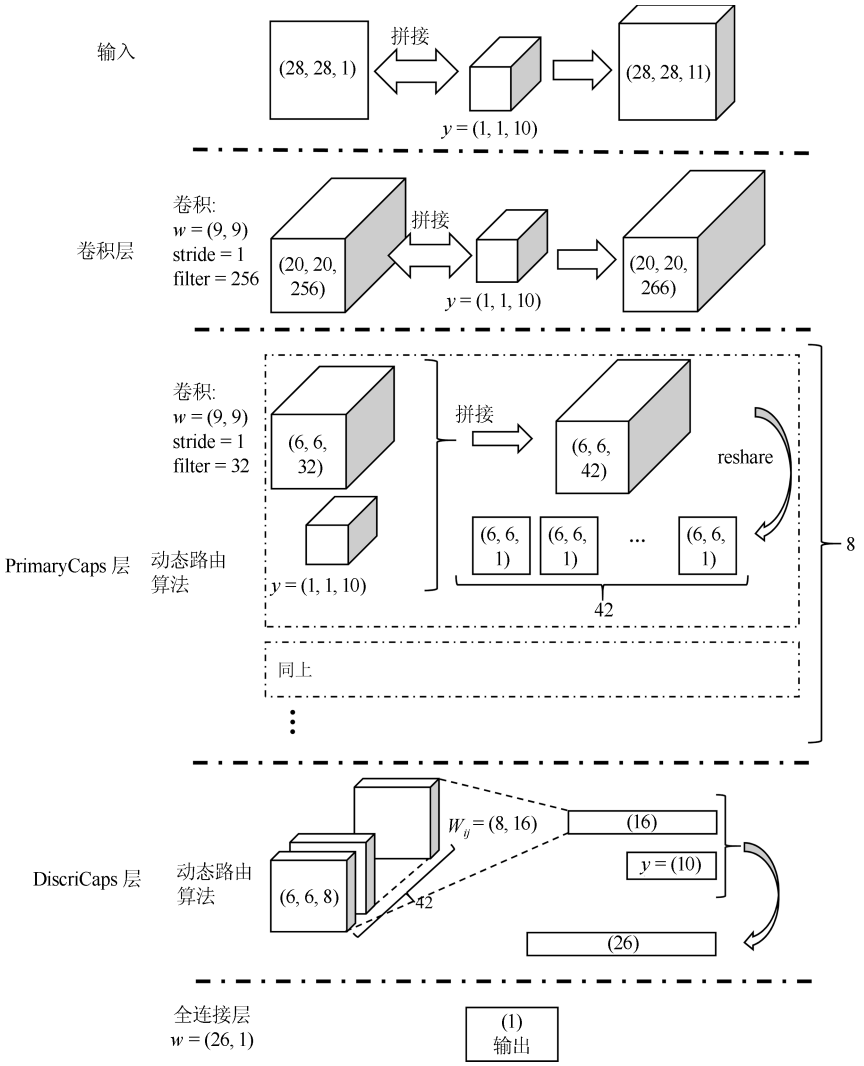


图 6　C-CapsGAN 判别器结构

Fig. 6　The structure of C-CapsGAN discriminator

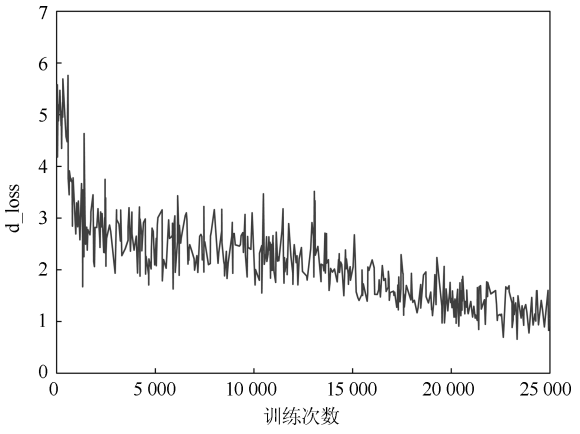


图 7　MNIST 上 d_loss 变化趋势 (PrimaryCaps 层胶囊个数为 32)

Fig. 7　Trends of d_loss on MNIST (32 capslue in PrimaryCaps layer)

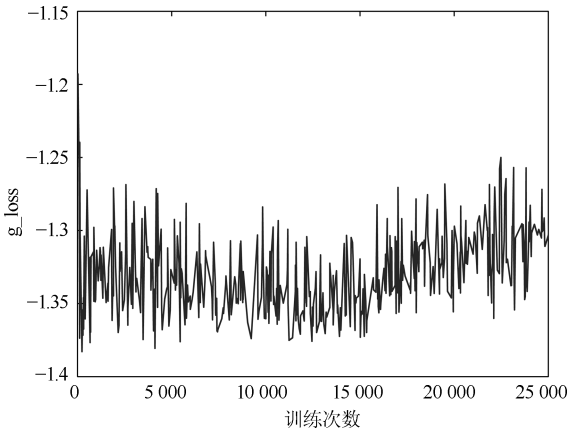


图 8　MNIST 上 g_loss 变化趋势 (PrimaryCaps 层胶囊个数为 32)

Fig. 8　Trends of g_loss on MNIST (32 capslue in PrimaryCaps layer)

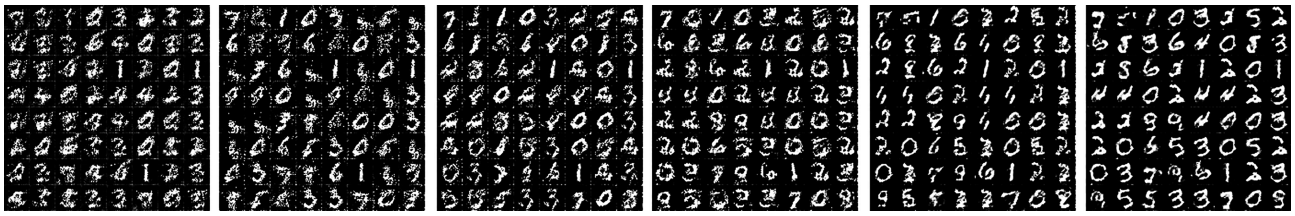


图 9 C-CapsGAN-32 在 MNIST 数据集训练生成结果 (从左到右分别从 Epoch1、5、10、15、20、24 采样得到)

Fig. 9 Sample images generated by C-CapsGAN-32 in MNIST dataset (sampled from Epoch1, 5, 10, 15, 20, 24 from left to right)

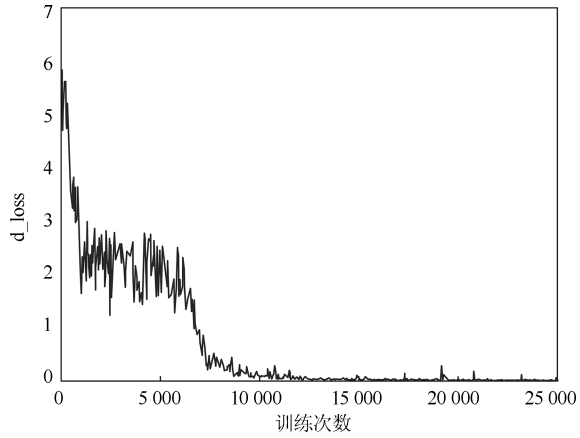


图 10 MNIST 上 d_loss 变化趋势 (PrimaryCaps 层胶囊个数为 24)

Fig. 10 Trends of d_loss on MNIST (24 capslue in PrimaryCaps layer)

和生成器的损失函数 (g_loss) 随训练次数增加而变化的情况.

当实验继续降低胶囊个数的时候, 从图 13 我们发现, 在训练初期判别器的损失就急剧下降, 这意味着判别器的判别能力由于胶囊个数太少而过低了. 图 15 为数据集的其中一个 Batch, 即 64 个生成样本随着 Epoch 次数的增加, 生成样本的进化情况, 可以看出当胶囊个数继续变少, 导致判别器的判别能力继续下降. 当训练结束的时候, 生成样本的质量则不如胶囊个数为 24 的网络生成的样本高.

综上所述, 对于本文提出的 C-CapsGAN 方法, 胶囊的个数对生成样本的收敛速度和生成样本的质量至关重要, 胶囊个数存在最优值, 当胶囊个数过多, 判别器过于强大, 则很容易在训练过程中始终对生成器进行严厉惩罚, 从而使得生成器总是无法产生能够 “迷惑” 判别器的样本, 导致生成效率低下; 而当胶囊个数过少 — 判别器太弱, 则生成器很容易产生出判别器无法分辨的 “真样本”, 虽然生成效率很高, 但是生成质量会有所下降. 对 MNIST 数据集实验来说, 设置 PrimaryCaps 层的胶囊个数在 16 ∼ 24 之间较好.

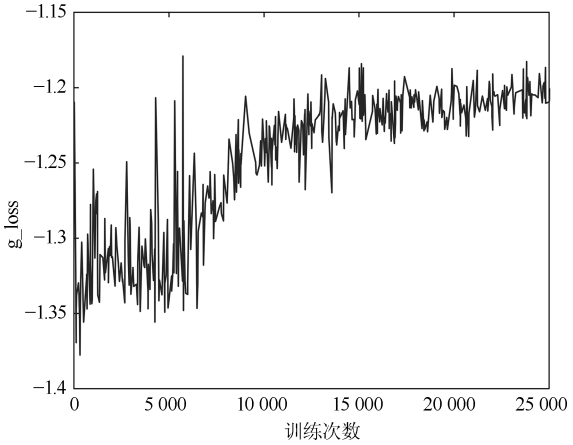


图 11 MNIST 上 g_loss 变化趋势 (PrimaryCaps 层胶囊个数为 24)

Fig. 11 Trends of g_loss on MNIST (24 capslue in PrimaryCaps layer)

图 12 C-CapsGAN-24 在 MNIST 数据集训练生成结果 (从左到右分别从 Epoch1、5、10、15、20、24 采样得到)

Fig. 12 Sample images generated by C-CapsGAN-24 in MNIST dataset (sampled from Epoch1, 5, 10, 15, 20, 24 from left to right)

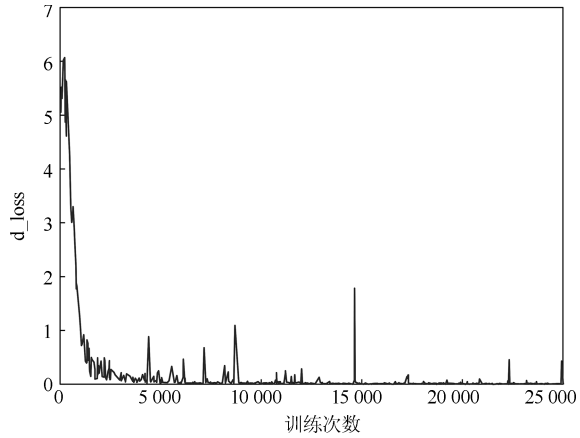


图 13　MNIST 上 d_loss 变化趋势 (PrimaryCaps 层胶囊个数为 16)

Fig. 13　Trends of d_loss on MNIST (16 capslue in PrimaryCaps layer)

传统 GAN 架构代表了全连接网络在生成式对抗领域的应用, 图 16 是传统 GAN 在 MNIST 数据集下的训练过程和最优结果; 而 DCGAN 代表了卷积神经网络在生成式对抗领域的应用, 图 17 是 DCGAN 在 MNIST 数据集下的训练过程和最优结果. 通过将传统 GAN、DCGAN 和 C-CapsGAN (24 个胶囊) 的生成样本进行对比从而直观展示了在生成对抗网络领域, 网络结构的升级能够提高生成样本质量.

实验发现: 本文提出的方法在收敛性和生成图片的质量上均优于传统 GAN 结构上的表现, 表明了 C-CapsGAN 对比传统 GAN 的优越性; 在和 DCGAN 训练过程的对比表明, DCGAN 的收敛性在前期 (10 轮之前) 较 C-CapsGAN 好, 在 DCGAN 中, 当训练持续进行到第 10 轮时, 生成样本质量较真实样本质量虽有差距但能够接受, 然而随着训练的继续进行, DCGAN 的优化效果不再显著, 意味着生成器并没有随着训练的继续而持续优化. 反观在 C-CapsGAN 中, 生成样本则随着训练轮数的增加持续变好, C-CapsGAN 对生成器的优化是持续有效的, 在 24 次 Epoch 之后, 网络生成的图片质量能够和 DCGAN 媲美. 因此, 本文提出的方法较之 DCGAN 而言, 不容易陷入优化停滞, 其收敛的速度更加稳定和持续, 只要增加训练 Epoch 的大小, C-CapsGAN 最终能够生成和真实样本无异的图像.

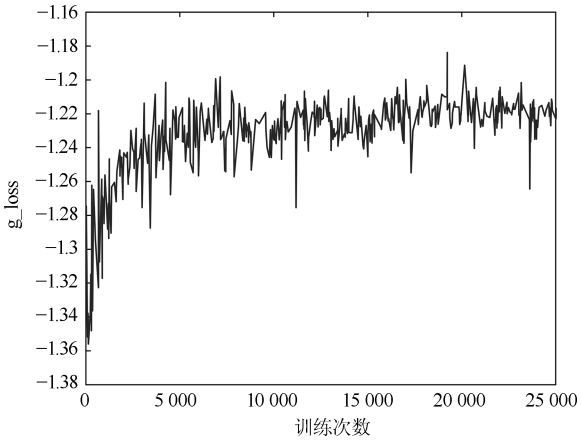


图 14　MNIST 上 g_loss 变化趋势 (PrimaryCaps 层胶囊个数为 16)

Fig. 14　Trends of g_loss on MNIST (16 capslue in PrimaryCaps layer)

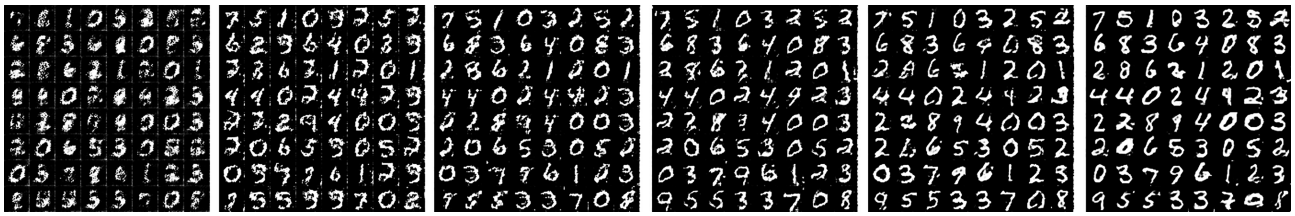


图 15　C-CapsGAN-16 在 MNIST 数据集训练生成结果 (从左到右分别从 Epoch1、5、10、15、20、24 采样得到)

Fig. 15　Sample images generated by C-CapsGAN-16 in MNIST dataset (sampled from Epoch1, 5, 10, 15, 20, 24 from left to right)

图 16　传统 GAN 在 MNIST 数据集训练的生成结果 (从左到右分别从 Epoch1、5、10、15、20、24 采样得到)

Fig. 16　Sample images generated by GAN in MNIST dataset (sampled from Epoch1, 5, 10, 15, 20, 24 from left to right)

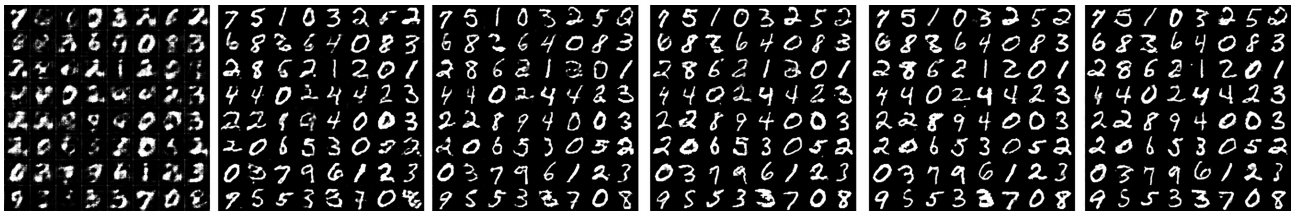


图 17 DCGAN 在 MNIST 数据集训练生成结果 (从左到右分别从 Epoch1、5、10、15、20、24 采样得到)

Fig. 17 Sample images generated by DCGAN in MNIST dataset (sampled from Epoch1, 5, 10, 15, 20, 24 from left to right)

4.2 CIFAR-10 实验

CIFAR-10 数据集包含 10 类 60 000 个图像尺寸为 $(32, 32, 3)$ 的彩色图像, 其中有 50 000 个训练样本和 10 000 个测试样本[18]. 在 GAN 领域, CIFAR-10 数据集作为一个有着 3 通道的彩色多类别数据集, 在保证样本数据精巧的同时不失复杂度, 一直是验证新型 GAN 有说服力的实验数据集之一.

在训练过程中, 与 MNIST 数据集的实验相似, 使用类似图 5 和图 6 的结构作为 C-CapsGAN 的生成器和判别器对 CIFAR-10 数据集进行训练, 生成器的输出和判别器和分类器的输入为 $(32, 32, 3)$ 大小的样本矩阵, 设置判别器与生成器的迭代次数为 2:1, 其他参数设置如下:

1) Batch 设置为 32;

2) Epoch 设置为 75 轮;

3) Adam 优化器的学习速率设置为 0.0001, 一阶矩估计的指数衰减率为 0.5, 二阶矩估计的指数衰减率为 0.9;

4) 生成器除了最后一层使用 Tanh 函数作为激活函数之外, 其他层都使用 ReLU 作为激活函数;

5) 判别器则使用 Leaky ReLU 作为激活函数;

本实验将胶囊判别器的网络参数设置为和原始 CapsNets 网络参数相同, 为了使实验结果更具客观性, 同时体现本文提出的 C-CapsGAN 方法良好的学习性能和对抗能力, 将会把 C-CapsGAN 采样结果与传统 GAN、DCGAN 在 CIFAR-10 数据集上的采样结果进行比较.

4.2.1 CIFAR-10 实验结果

图 18 和图 19 分别表示了 C-CapsGAN 的判别器的损失函数 (g_loss) 和生成器的损失函数 (d_loss) 随训练次数增加而变化的情况. 从损失变化图可以看到, 由于使用的 CIFAR-10 数据集的复杂性, 使得模型在轮数迭代期间, 判别器和生成器的训练过程一直呈现出此消彼长的震荡状态.

图 20 是设置为 32 个胶囊 C-CapsGAN 在执行完指定 Epoch 之后随机采样生成的 CIFAR-10 数据样本, 可以看出, 经过充分的学习之后, C-CapsGAN 生成的样本多样性好, 图像清晰锐利, 同时样本数据分布与真实的 CIFAR-10 数据集样本非常接近.

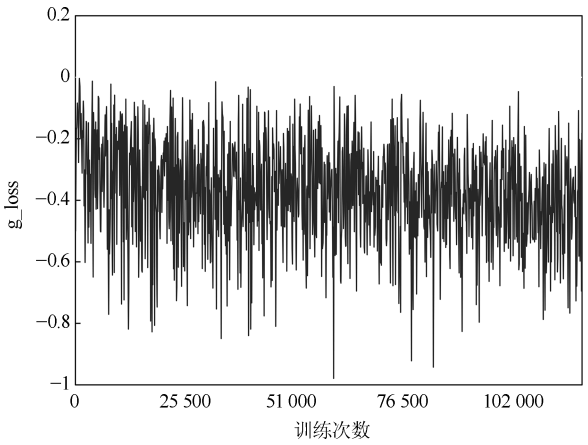


图 18 CIFAR-10 上 g_loss 变化趋势

Fig. 18 Trends of g_loss on CIFAR-10

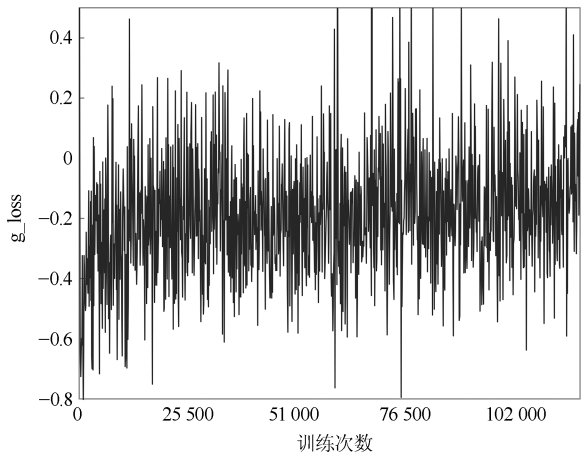


图 19 CIFAR-10 上 d_loss 变化趋势

Fig. 19 Trends of d_loss on CIFAR-10

为了更为客观地反映 C-CapsGAN 优越的学习性能和生成能力, 本文还将 C-CapsGAN 与传统 GAN 和 DCGAN 同时在 CIFAR-10 数据集上训练的采样结果进行比较, 三者除了网络模型架构不同, 训练超参数、损失函数等其他非网络结构参数均保

持一致. 如下图 21 是 DCGAN 结构在 CIFAR-10 数据集训练了 Epoch 次之后的随机采样结果; 图 22 是传统 GAN 结构在 CIFAR-10 数据集训练了 Epoch 次之后的随机采样结果. 可以直观地看出无论是多样性还是样本质量, 本文提出的方法均显著优于传统 GAN 和 DCGAN.

图 20　C-CapsGAN 生成的样本图像

Fig. 20　Sample images generated by C-CapsGAN in CIFAR-10 dataset

同时, 在进行对比实验中发现, 无论是传统 GAN 还是 DCGAN, 在训练后期, 随机采样的样本总是出现类似图 24 所展示的情况, 样本之间的颜色呈现一致性. 忽略样本内容分别观察 DCGAN 在 Epoch 为 55、65、75 的随机采样结果发现: Epoch 为 55 时, DCGAN 采样结果的颜色均偏亮黄色、Epoch 为 65 时, DCGAN 采样结果的颜色均偏暗黄色、Epoch 为 75 时, DCGAN 采样结果的颜色均偏灰色, 这和 CIFAR-10 数据集的真实分布是不相符合的. 然而, 此类情况在本文提出的 C-CapsGAN 中并未出现, 正如图 23 所示, 在训练后期通过对 C-CapsGAN 生成样本进行随机采样发现, 本文方法所生成的样本颜色多彩丰富. 即意味着, 本文提出的方法相比传统 GAN 和 DCGAN,

图 21　DCGAN 生成的样本图像

Fig. 21　Sample images generated by DCGAN in CIFAR-10 dataset

图 22　传统 GAN 生成的样本图像

Fig. 22　Sample images generated by GAN in CIFAR-10 dataset

图 23　C-CapsGAN 在 Epoch 分别为 55、65、75 随机采样的样本

Fig. 23　Sample images generated by C-CapsGAN in CIFAR-10 dataset (sampled from Epoch55, 65, 75 from left to right)

图 24 DCGAN 在 Epoch 分别为 55、65、75 随机采样的样本

Fig. 24 Sample images generated by DCGAN in CIFAR-10 dataset (sampled from Epoch55, 65, 75 from left to right)

能够有效抑制模式坍塌的出现, 换言之, 正因为 C-CapsGAN 网络内部胶囊的存在, 才使得网络在习得对象特征的同时还存储了对象特征之间的分层姿态关系 (位置、颜色、方向、形状等), 从而胶囊判别器能够指导生成器朝着真实样本分布的方向持续优化.

5 结论

本文利用 WGAN、CGAN 的优点, 来探索最新的神经网络架构 CapsNets 在生成领域的应用, 新方法 C-CapsGAN 在 MNIST 数据集和 CIFAR-10 数据集均取得优异的生成结果. 特别地, 在 MINST 数据集上着重探索了 C-CapsGAN 中胶囊个数对生成对抗网络的训练收敛性和生成质量的影响, 得出胶囊个数对于生成结果来说存在最优值的结论; 在 CIFAR-10 数据集上则探索了 C-CapsGAN 在彩色复杂数据集上的应用, 并通过对比现有的生成对抗网络架构的训练过程和样本结果发现, 本文的 C-CapsGAN 方法在胶囊结构的支持下能够更加有效地抑制模式坍塌问题. 总的来说, 通过将胶囊个数调节合适的 CapsNets 架构运用到生成式对抗网络的判别器中, 使得在生成式对抗网络的内部同时存在反向传播和动态路由两种优化算法, 在生成高质量图像的同时能更好地应对可能出现的模式坍塌等常见问题. 本文仍有不足之处, 由于 CapsNets 网络的引入, 胶囊层之间特征的聚类算法是通过迭代的方式进行的, 因此训练速度较慢, 在今后的工作中, 如何从网络结构的角度提高训练速度将是重要的研究命题. 同时具有位姿特征的三维模型是当下研究的难点, 而胶囊的特性使得它能够比传统的卷积神经网络更适合三维结构模型, 未来的工作也会着重思考如何利用 CapsNets 在 GAN 的应用来更好地寻找刻画三维模型特征的方法.

References

1 Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal, Canada: MIT Press, 2014. 2672−2680

2 Kurach K, Lucic M, Zhai X, et al. The GAN Landscape: Losses, Architectures, Regularization, and Normalization. arXiv preprint, arXiv: 1807.04720, 2018.

3 Wang Kun-Feng, Gou Chao, Duan Yan-Jie, Lin Yi-Lun, Zheng Xin-Hu, Wang Fei-Yue. Generative adversarial networks: the state of the art and beyond. *Acta Electronica Sinica*, 2017, **43**(3): 321−332
(王坤峰, 苟超, 段艳杰, 林懿伦, 郑心湖, 王飞跃. 生成式对抗网络 GAN 的研究进展与展望. 自动化学报, 2017, **43**(3): 321−332)

4 Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint, arXiv: 1511.06434, 2015.

5 Mirza M, Osindero S. Conditional generative adversarial nets. arXiv preprint, arXiv: 1411.1784, 2014.

6 Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks. arXiv preprint, arXiv: 1701.04862, 2017.

7 Arjovsky M, Chintala S, Bottou L. Wasserstein gan. arXiv preprint, arXiv: 1701.07875, 2017.

8 Lin Yi-Lun, Dai Xing-Yuan, Li Li, Wang Xiao, Wang Fei-Yue. The new frontier of AI research: generative adversarial networks. *Acta Electronica Sinica*, 2018, **44**(5): 775−792
(林懿伦, 戴星原, 李力, 王晓, 王飞跃. 人工智能研究的新前线: 生成式对抗网络. 自动化学报, 2018, **44**(5): 775−792)

9 LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, **86**(11): 2278−2324

10 Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe, USA: Curran Associates, Inc., 2012. 1097−1105

11 Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. Long Beach, USA: Springer, 2017. 3856−3866

12 Hinton G E, Krizhevsky A, Wang S D. Transforming auto-encoders. In: Proceedings of the 21st International Conference on Artificial Neural Networks. Espoo, Finland: Springer, 2011. 44−51

13 Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved training of wasserstein GANs. arXiv preprint, arXiv: 1704.00028, 2017.

14 Wang Kun-Feng, Zuo Wang-Meng, Tan Ying, Qin Tao, Li Li, Wang Fei-Yue. Generative adversarial networks: from generating data to creating intelligence. *Acta Electronica Sinica*, 2018, **44**(5): 769−774
(王坤峰, 左旺孟, 谭营, 秦涛, 李力, 王飞跃. 生成式对抗网络: 从生成数据到创造智能. 自动化学报, 2018, **44**(5): 769−774)

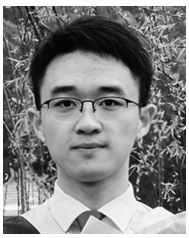
15 Jaiswal A, AbdAlmageed W, Natarajan P. CapsuleGAN: Generative Adversarial Capsule Network. arXiv preprint, arXiv: 1802.06167, 2018.

16 Kussul E, Baidyk T. Improved method of handwritten digit recognition tested on MNIST database. *Image and Vision Computing*, 2004, **22**(12): 971−981

17 Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint, arXiv: 1502.03167, 2015.

18 Hinton G E, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R R. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv: 1207.0580, 2012

孔 锐 暨南大学智能科学与工程学院教授. 主要研究方向为图像识别. 本文通信作者. E-mail: tkongrui@jnu.edu.cn
(**KONG Rui** Professor at the School of Intelligent Systems Science and Engineering, Jinan University (Zhuhai Campus). His main research interest is image recognition. Corresponding author of this paper.)



黄 钢 暨南大学信息科学技术学院硕士研究生. 主要研究方向为生成对抗网络, 模式识别.
E-mail: hhhgggpps@gmail.com
(**HUANG Gang** Master student at the College of Information Science and Technology, Jinan University. His research interest covers generative adversarial networks and pattern recognition.)