MetaCGAN A Novel GAN Model for Generating high quality and diversity images with few training data

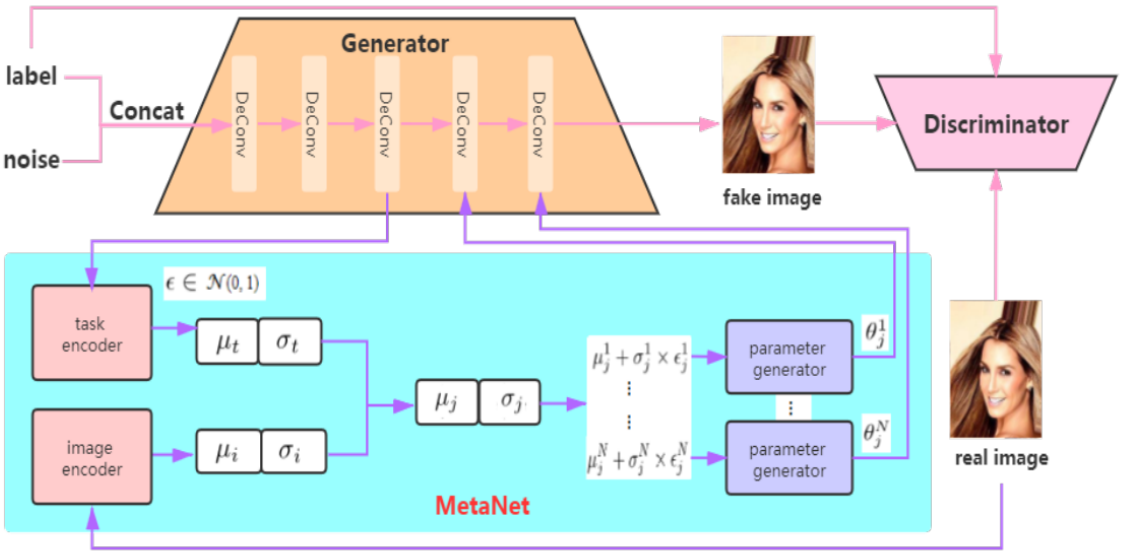

Fig. 1. The architecture of MetaCGAN. The top is the CGAN module, while the below is the MetaNet module.

发表

ICJNN A 会,参考[计算机类顶级会议排名+投稿经验]

核心思想

基于迁移学习，让generator学习到其他类样本的先验信息，迁移到新类上，利用MetaNet生成CGAN反卷积层的权重，以辅助生成质量更高的多样性样本。

难点

代码实现。

对比

- DCGAN 大量数据
- WGAN 大量数据
- CGAN 新类少量，旧类大量

数据集

- MNIST

  0,1,2,3,4作为训练基本类,5,6,7,8,9作为新类,每类只包含10张图片.

- Fashion MNIST

  t-shirt trouser pullover dress coat 作为基本类,sandal shirt sneaker bag ankle boot 作为新类.

- CelebA

  女人作为基本类,男人作为新类,每类50张

训练算法

Algorithm 1 The training scheme of MetaCGAN.

Require: θ_G: G's parameters; θ_j: feature extractor's parameters; θ_d: discriminative layers' parameters; N: batch size; N_c: the learning steps per iteration.

Require: Adam hyperparameters: $\alpha = 0.5$, $\beta = 0.9$, $N_c = 4$.

Initial discriminators parameters θ_d, initial generators parameters θ_G, learning rate$= 2 \times 10^{-4}$.
while not converged **do**
  for $i = 1, 2, ..., epoch$ **do**
    Divide the original data set into base classes and new classes.
    if $number\%N_c = 0$ **then**
      Samples (x_i, z_i, y_i) from the base classes.
    else
      Samples (x_i, z_i, y_i) from the new classes.
    end if
    Employ the task encoder to calculate μ_t and σ_t.
    Employ the image encoder to calculate μ_i and σ_i.
    $\mu_j, \sigma_j = (\mu_t + \mu_i)/2 + (\sigma_t + \sigma_i)/2$.
    Employ reparameterization trick to obtain $z_i = \mu_j^i + \sigma_j^i \times \epsilon_j^i$.
    Employ the parameter generator to obtain deconvolutional parameter θ_j^i.
    Sample and allocate weights θ_j^i for deconvolutional layer of the generator.
    $L_D \leftarrow logD(x_i|y_i) + log(1 - D((G(z_i|y_i))))$
    $L_G \leftarrow logD(G(x_i))$
    $\theta_d, \theta_j = RMSProp(L_D, \theta_d, \theta_j)$
    $\theta_G = Adam(L_D, \theta_G, \alpha, \beta)$
  end for
  Compute batch losses and update weights.
end while

Tricks

- 使用instance Batch normalization,参考:pytorch常用normalization函数

评估

- 定性

  对比生成图片的质量

- 定量

  IS(inception score)=$e^{(E_x[D_{KL}(P(y|x)\|p(y))])}$

- x:生成的图像
- y:训练好的分类器去预测x属于某一个类别的概率所获得的向量
- p(y):边缘分布
- D_{KL}:kl散度
- p(y|x)低意味着决策边界很紧凑,p(y)高意味着图像的高度多样性
- 通常生成图片的多样性可被类之间的决策边界的可区分性所表征.如果两张图片高概率不相似,则应被分为不同的类.

AMT(Amazon Mechanical Turk)

- 评估CelebA上生成的图像.

Generative Latent Implicit Conditional Optimization when Learning from Small Sample

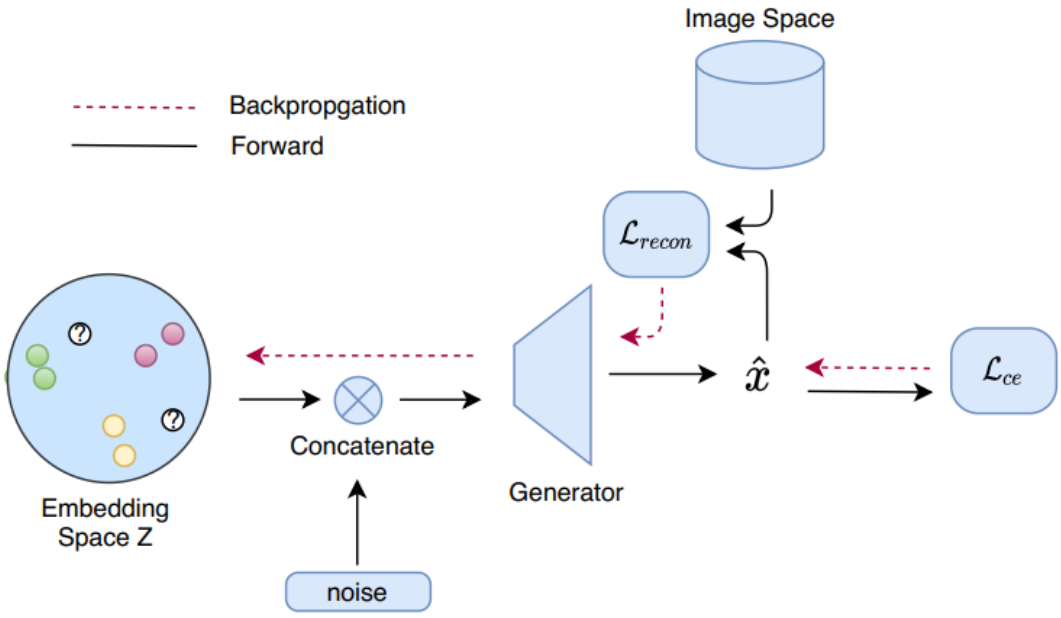

Fig. 2. Schematic illustration of our method. Every $\mathbf{z}_i \in Z$, where Z denotes the unit sphere, is mapped to a specific image x_i in the image space. In this illustration, the color of each image frame marks the class label of the image, while black frames mark unlabeled images. The classifier is used to propagate error only when a label is given.

发表

ICPR A 会,参考计算机类顶级会议排名+投稿经验

核心思想

提出GLICO(Generative Latent Implicit Conditional Optimization),学习一个训练空间到潜在空间的隐射,生成器从潜在空间向量中生成图片.不依赖于大量未标记图片,GLICO只需要小的带标签数据点集.实际上,合成新的图片每类只需要5到10张在总共10类样本中,不需要任何先验信息.最后用提出的方法从潜在空间中利用球面插值采样用生成器生成新的图片,生成的图片可以提升分类精度.

问题提出

1 严格的小样本设置,学习者每一类只能访问到小数量样本(通常5~10),类别数也不多.

- 半监督

  学习者学习到大量无标签数据.

- 小样本场景

  学习者可以接触到大量带标签的各个类别数据,只是这些类别的数据不参与当前分类任务.

大多数小样本算法依赖于迁移学习,从大量带标签训练样本中,大多半监督学习算法从无标签数据中迁移知识.

2 大多数针对小数据样本场景设计的方法不能利用外围数据来进行迁移学习,可通过施加强的先验到模型中来解决这个问题,但是在许多场景下由于未知域以及这样的先验知识不可得.

数据增强可能会有用,利用

- 有关数据的弱先验,如半监督学习,传导学习.
- 自训练方法
- 生成模型

3 GAN为何不能用?

- 需要大量数据来进行有效训练.
- 小样本场景下表现差.

方法解决

在潜在空间中进行优化.为每一个数据点分别学习一个潜在空间码.在潜在空间中,让内类样本尽量相互靠拢,类间样本尽量远离.通过一个由已知多标签数据训练的分类器.

损失函数

$L_{percep}(x_i, z_i; \theta) = \sum_j \lambda_j \parallel \xi_j(G_\theta([z_i, \varepsilon])) - \xi_j(x_i)\|_1$

感知损失函数

- $z_i \epsilon Z$,Z:单位球面,在R^d 空间中,表示隐层空间.
- $x_i \epsilon X \mid X_i \epsilon R^{3*H*W}$,表示训练集中的样本.
- G_θ:生成器.
- $\xi_j(x_i)$:卷积网络的输出层.

Here we use the perceptual loss [21] to measure the reconstruction loss. More specifically, in order to compute the perceptual loss we extract the activation vectors in layers '$conv_{1\_2}$', '$conv_{2\_2}$', '$conv_{3\_2}$', '$conv_{4\_2}$' and '$conv_{5\_2}$' of a VGG-16 network. Denoting the output tensor of layer '$conv_{j\_2}$' for input image x by $\xi_j(x)$, we compute the difference between the original image and its reconstructed version by:

$$\mathcal{L}_{percep}(x_i, \mathbf{z}_i; \theta) = \sum_j \lambda_j ||\xi_j(G_\theta([\mathbf{z}_i, \varepsilon])) - \xi_j(x_i))||_1 \quad (1)$$

Above θ denotes the parameters of the generator G_θ, and λ_j the weight of layer j (usually the weighted average).

优化目标

$min_\theta \sum_{i=1}^{n} [min_{z_i \epsilon Z} L_{percep}(x_i, z_i; \theta)]$

分类损失