1. FaceNet: A Unified Embedding for Face Recognition and Clustering

在FaceNet出现之前，解决人脸识别的思路大多是用一个分类层在训练集上训练，中间层为人脸图像的向量映射，然后以分类层作为输出层。这种思路不直接且效率较低：中间层的向量映射需要在新的人脸上泛化较好，且用于表示每张人脸的尺寸是非常大的，约1000维。因此作者提出了用三元组损失（Triplet Loss）直接训练网络并输出为128维人脸特征的FaceNet。

方法

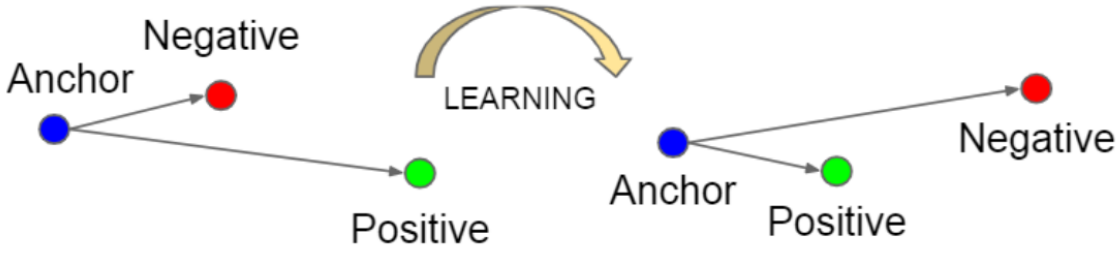
1. 模型设计

图中的Batch即为输入的三元组图像，中间的Deep Architecture实现对三元组图像的特征提取，其结构是CNN网络去掉最后的softmax层，将提取到的特征通过L2范数进行归一化，最后embedding成128维的特征并通过Triplet Loss进行训练。

2. 三元组损失 Triplet Loss

所谓的三元组就是三个样本anchor, positive, negative，其中anchor和positive属于同一类，anchor和negative属于不同类。学习的过程就是对于尽可能多的三元组，使得anchor和positive的距离小于anchor和negative的距离。用数学公式可以表示为：$||x_i^a - x_i^p||_2^2 + \alpha < ||x_i^a - x_i^n||_2^2$.

该过程可视化图如下：



经过等价转换后可以得到Triplet Loss函数为：

$$L = \sum_i^n [||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha]_+$$

x_i^a、x_i^p、x_i^n分别表示目标人脸、正例（与目标同一身份的人脸）、负例（与目标不同身份的人脸），$f(x)$表示图像x对应的人脸特征，α表示margin。$||f(x_i^a) - f(x_i^p)||_2^2$表示目标与正例之间的欧式距离，$||f(x_i^a) - f(x_i^n)||_2^2$表示目标与负例样本之间的欧式距离。

三元组损失的整体含义是尽可能让同一身份的人脸距离与不同身份的人脸距离之间至少为α，因此就能很好地区分不同身份的人脸。

3. 三元组选择 Triplet Selection

三元组的选择关系到网络收敛的速度，过于简单的三元组对网络的贡献很小，因此需要找到那些让三元组损失很大的困难样例来让网络快速收敛，也就是给定一个anchor，找到一个hard positive（难正例）和一个hard negative（难负例），使得anchor和positive之间的距离尽可能大，即$argmax_{x_i^p}||f(x_i^a) - f(x_i^p)||_2^2$，而anchor和anchor和negative之间的距离尽可能小，即$argmin_{x_i^n}||f(x_i^a) - f(x_i^n)||_2^2$。

在实际训练中，直接在全体训练样本上寻找全局最大和最小是不可行的，因为错误标签和质量较差的人脸会主导训练。因此，作者提供了两种难例挖掘思路：

- 离线：每n步离线产生三元组，用最近的网络在数据的一个子集上计算$argmin$和$argmax$
- 在线：从mini-batch中筛选难例

作者采用了在线生成triplet的办法，在每一个mini-batch中选择hard positive和hard negative样例。为了使mini-batch中生成的triplet合理，在生成mini-batch时，保证每个mini-batch中每个人有40张人脸图片作为正样本，并随即筛选其他人脸图片作为负样本。在生成triplet时，需要找出所有的anchor-positive对，并找出其hard negative样本。

选择hardest negative会导致局部最优而让模型崩溃，因此还需要选择一些semi-hard的样例，这些样例不考虑α的因素，保证a-n距离大于a-p距离，虽然二者距离很相似，但仍然是可区分的，其公式表示为：$||f(x_i^a) - f(x_i^p)||_2^2 < ||f(x_i^a) - f(x_i^n)||_2^2$。

4. 深度卷积网络 Deep Convolutional Networks

这部分即是模型设计中的deep architecture模块，论文中探索了两种网络backbone。

- 第一种结构为Zeiler&Fergus网络，包含多个交错的卷积层和非线性激励函数，局部归一化和最大池化层，并额外添加了一些1x1xd的卷积层。
- 第二种结构基于Inception模型，即GoogLeNet，网络利用了一些不同的卷积层和池化层并行和级联响应。

总结

- 论文提出了三元组损失Triplet Loss，用于直接反映得到的特征质量如何，将图像映射到特征空间中，图像间的欧氏距离可以直接反映人脸之间的相似度，同一身份的人脸距离小，不同身份的人脸距离大。
- 三元组损失相对于softmax的优势在于softmax不直接，而三元组损失可以直接优化距离，且softmax产生的特征表示向量维度很大。

---

2. In Defense of the Triplet Loss for Person Re-Identification

一些ReID方法使用Triplet Loss的一些变体来训练它们的模型，并取得了一定的效果。但一些方法认为分类损失（classification loss）和验证损失（verification loss）在某些ReID任务上要优于三元组损失。但这两种损失函数都存在问题，随着特征数量的增加，分类损失需要越来越多可学习的参数，但其中的大部分在训练后被舍弃；验证损失只能成对地判断两张图片的相似度，很难应用到目标聚类和检索上去，因为每一张图片需要通过网络和图片库中的所有图像去匹配。

而这篇文章做的就是要捍卫Triplet Loss在ReID中的表现和地位，即Triplet Loss不比classification loss和verification loss要差。作者认为，使用triplet loss的简单CNN网络在许多数据集上优于当前的很多方法，因为它端到端、简单直接，自带聚类属性且特征高度嵌入。

为什么很多研究者不看好Triplet Loss?

困难样本挖掘（hard mining）在Triplet训练中是一个很重要的步骤。在进行原始Triplet训练时，没有困难样本挖掘会使得训练陷入停滞，从而导致最终的收敛结果不佳，而选择过难的样本又会导致训练不稳定、收敛变难。此外，困难样本挖掘比较耗时，当前也没有清楚的定义什么是“Good Hard”。

方法

基于上述问题，以及原始Triplet训练中有大量有效的三元组被浪费，作者建议对Triplet Loss的经典方法进行修改。

1. Batch Hard Loss

作者提出了一种组成batch的方法“PK batch”来提高采样效率，即每个batch随机采样P个类（person ID），再对于每个类随机采样K张图片（person），最终采样到PK张图片组成一个batch。再根据选取hard positive/negative的思想，作者提出了一种triplet loss的改进版本batch hard loss：

$$\mathcal{L}_{\mathrm{BH}}(\theta; X) = \overbrace{\sum_{i=1}^{P}\sum_{a=1}^{K}}^{\text{all anchors}} \Big[m + \overbrace{\max_{p=1\ldots K} D\left(f_\theta(x_a^i), f_\theta(x_p^i)\right)}^{\text{hardest positive}} - \underbrace{\min_{\substack{j=1\ldots P \\ n=1\ldots K \\ j\neq i}} D\left(f_\theta(x_a^i), f_\theta(x_n^j)\right)}_{\text{hardest negative}} \Big]_+, \quad (5)$$

batch hard loss每次对于batch中的一个anchor，选取该batch中的hardest positive/negative，但这在整个数据集中并不一定是hardest的，因此这种办法选择到的positive/negative其实是非常“适中”的，既缓解了没有hard mining造成的训练停滞，又避免了选择过难样本导致的训练不稳定。

这种设定下，一个batch可以生成PK个三元组。

2. Batch All Loss

把Batch Hard Loss中的max和min去掉，即将一个batch中所有的三元组都用来组成loss，就得到了Batch All Loss：

$$\mathcal{L}_{\mathrm{BA}}(\theta; X) = \overbrace{\sum_{i=1}^{P}\sum_{a=1}^{K}}^{\text{all anchors}} \overbrace{\sum_{\substack{p=1 \\ p\neq a}}^{K}}^{\text{all pos.}} \overbrace{\sum_{j=1}^{P}\sum_{\substack{n=1 \\ j\neq i}}^{K}}^{\text{all negatives}} \left[m + d_{j,a,n}^{i,a,p} \right]_+, \quad (6)$$

$$d_{j,a,n}^{i,a,p} = D\left(f_\theta(x_a^i), f_\theta(x_p^i)\right) - D\left(f_\theta(x_a^i), f_\theta(x_n^j)\right).$$

这种设定下，一个batch可以生成$PK(PK-K)(K-1)$个三元组。但由于hinge function $[\quad]_+$的存在，有可能非常多的三元组项对loss的贡献都是0，再进行平均之后会让有效信息变得很少。作者提出了$L_{BA\neq 0}$来解决这一问题，最终只平均那些非0的损失项。

3. Soft Margin

经典的Triplet Loss方法中，如果三元组满足margin关系则该项的loss直接为0，这样的硬截止方式不利于ReID任务，因为ReID需要不断拉近同类样本间的距离。

为此，作者使用了softplus函数来代替hinge函数：$s(x) = ln(1 + e^x)$。softplus函数的行为与hinge函数类似，但它是指数衰减的，而不是硬截止，因此称为soft margin公式。

4. 补充说明

很多相关工作中使用平方欧氏距离作为度量函数，虽然作者没有系统地对比其他度量函数，但在实验中发现平方欧式距离使优化容易崩溃，而非平方欧氏距离表现的更为稳定。此外，平方欧氏距离会降低边距参数margin的可解释性，因为它不再代表特征间的绝对距离。

总结

A well designed triplet loss has a significant impact on the result.

---

3. Deep Metric Learning with Hierarchical Triplet Loss

传统Triplet Loss的问题?

- 当有N张训练样本时，进行三元组损失训练会生成$O(N^3)$个三元组，而在训练期间遍历这些三元组显然是不可行的，这样会产生大量冗余信息，且在随机采样时，大多数的三元组由于满足margin约束而缺乏有效的训练信息，使网络收敛较慢。
- 传统的Triplet训练将mini-batch作为网络的输入，这样就导致了被训练的三元组只能着眼于当前的batch，而忽视了全局的数据分布。这种未融合全局信息的训练方式很容易使网络陷入局部最优。

方法

层次三元组损失（Hierarchical triplet loss）包括两个主要的部分，构造一个层次类别树（Hierarchical class tree）和使用新的动态损失边界（violate margin）形成层次三元组损失。

层次类别树用于捕获整个数据集上的不同类之间的关系，即全局数据的上下文信息，基于该树进行Anchor-Neighbor三元组采样，配合动态损失边界形成层次三元组损失。

1. Manifold Structure in Hierarchy

在类别层面构建全局层次结构，给定一个使用传统triplet loss预训练的神经网络$\phi_t(\cdot, \theta)$计算类间距离矩阵，其中第p类和第q类之间的距离计算为：

$$d(p, q) = \frac{1}{n_p n_q} \sum_{i \in p, j \in q} ||r_i - r_j||^2$$

其中n_p和n_q分别是第p类和第q类的训练样本数，特征r已经被单位化。

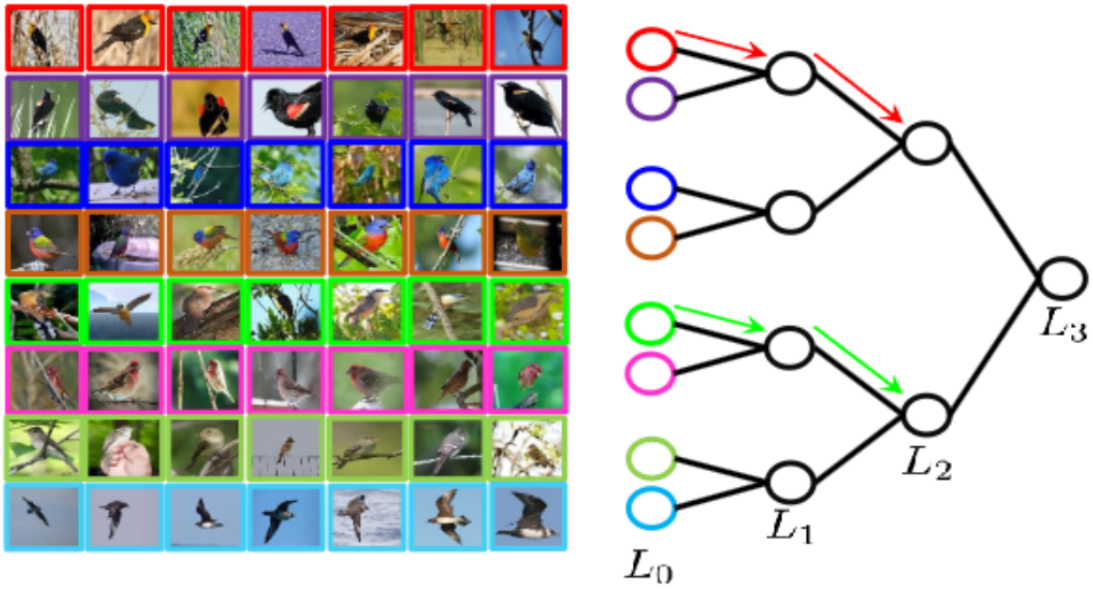
根据计算的类间距离创建层次类别树，第0级的叶子节点是所有原始的图像类，基于计算的距离矩阵递归地合并不同层级的叶子节点，最终完成层次树创建。层次树设置L级，平均类内距离d_0作为合并第0级节点的阈值。

$$d_0 = \frac{1}{C} \sum_{c=1}^{C} (\frac{1}{n_c^2 - n_c} \sum_{i \in c, j \in c} ||r_i - r_j||^2)$$

其中n_c是第c类的样本数，不同的节点使用不同的阈值合并。合并第l级节点的阈值设置为：

$$d_l = \frac{l(4-d_0)}{L} + d_0$$

距离小于d_l的两个类别合并到第l级的节点中，节点从第0级合并到第L级，最终生成一个层次类别树H，它能捕捉整个数据集上的类别的关系，并在训练中不断被更新。



(a) Hierarchical Tree $\mathcal{H}$

2. Hierarchical Triplet Loss

- Anchor Neighbor Sampling

在所构造的层次树的第0级随机选择$l^{'}$个节点，每一个节点代表一个原始的类别，基于计算的类间距离，为$l^{'}$个类的每个类选择$m-1$个在第0级最接近的类，再在每个类中随机选取t张图片，因此一个mini-batch中有$n(n=l^{'}mt)$张图片。

为什么在第0级选择多个类?

保证了mini-batch中训练样本的多样性，这样在神经网络中使用BN会更稳定和准确。

- Triplet Generation and Dynamic Violate Margin

在mini-batch M上计算的层次三元组损失可以表示为：

$$L_M = \frac{1}{2Z_M} \sum_{T^z \in T^M} [||x_a^z - x_p^z|| - ||x_a^z - x_n^z|| + \alpha_z]_+$$

其中T^M是mini-batch M中的所有三元组，三元组$T^z = (x_a, x_p, x_n)$，在一个mini-batch中得到的三元组数量为$Z_M = A_{l'm}^2 A_t^2 C_t^1$，$A_{l'm}^2$表示从mini-batch中随机抽取两个类（正类和负类），A_t^2表示从正类中选择两个样本（anchor、positive），C_t^1表示从负类中随机选择一个负样本（negative）。

α_z是动态的损失边界，与传统的triplet loss的margin是一个常数不同，它是根据层次类别树上的anchor类y_a和negative类y_n的类间关系计算得到的，表达式如下：

$$\alpha_z = \beta + d_{H(y_a,y_n)} - S_{y_a}$$

其中，$\beta = 0.1$是一个常数，它鼓励图像每一次训练时比之前轮次类间距更远；$H(y_a, y_n)$是层次树上的层级，表示类y_a和类y_n在该级的下一级合并为一个节点，$d_{H(y_a,y_n)}$是在层次树上合并两个类的阈值；S_{y_a}是类y_a样本间的平均距离。

- Implementation Details

Algorithm 1: Training with hierarchical triplet loss

Input: Training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{k=1}^{N}$. Network $\phi(\cdot, \boldsymbol{\theta})$ is initialized with a pretrained ImageNet model. The hierarchical class tree $\mathcal{H}$ is built according to the features of the initialized model. The margin α_z for any pair of classes is set to 0.2 at the beginning.

Output: The learnable parameters θ of the neural network $\phi(\cdot, \boldsymbol{\theta})$.

while *not converge* **do**
- $t \leftarrow t + 1$;
- Sample anchors randomly and their neighborhoods according to $\mathcal{H}$;
- Compute the violate margin for different pairs of image classes by searching through the hierarchical tree $\mathcal{H}$;
- Compute the hierarchical triplet loss in a mini-batch $\mathcal{L}_{\mathcal{M}}$;
- Backpropagate the gradients produced at the loss layer and update the learnable parameters ;
- At each epoch, update the hierarchical tree $\mathcal{H}$ with current model.

总结

Hierarchical Triplet Loss鼓励anchor样本推动与不同类的附近样本远离自己，并且它通过层次类别树融合全部的类信息，计算动态损失边界，从而对距离它非常远的样本点同样贡献梯度。

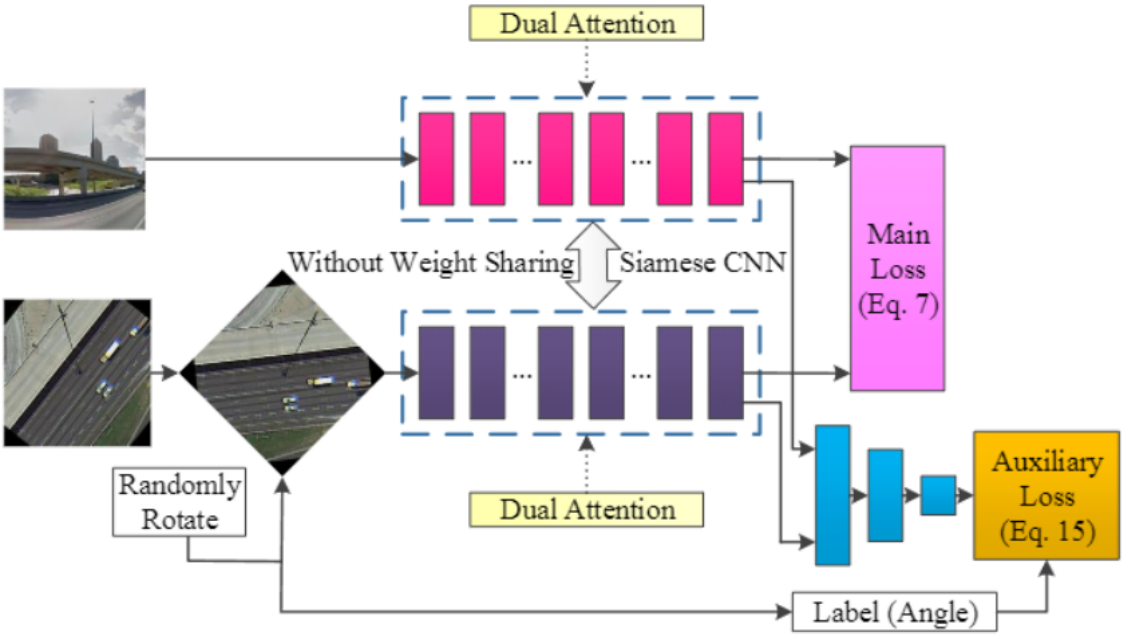
---

4. Ground-to-Aerial Image Geo-Localization With a Hard Exemplar Reweighting Triplet Loss

目前跨视角图像匹配及难样本挖掘存在的问题？

- 跨视角的图像匹配的挑战源自巨大的观察位置差异、光线变化和地-空图像的方位不确定性。
- 难样本挖掘还不到位，因为需要确定每个样本的难度等级。

因此，作者从网络结构的角度出发，提出了一种基于Dual Attention的模块FCAM，集成到残差网络中搭建一个孪生网络，用以提取更好的特征表示；从loss的角度出发，提出了一种可以根据难度等级来给三元组分配权重的损失HER，这样训练就可以聚焦于带有大量信息的难样本。整体框架如下：

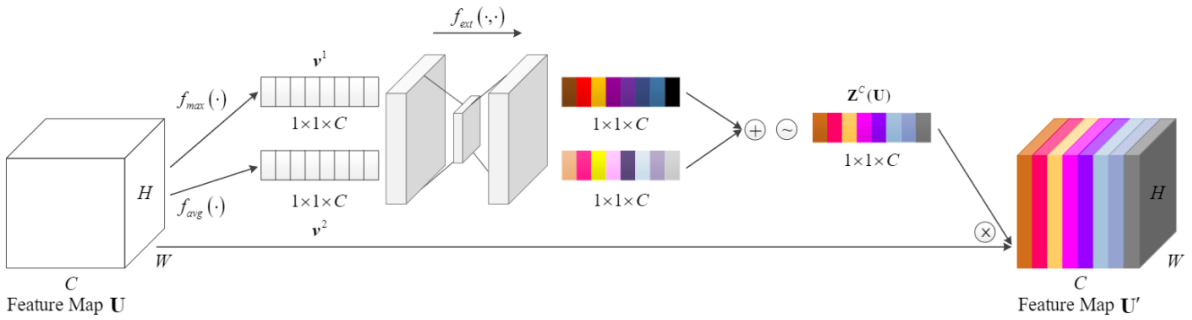


方法

1. Feature Context-Based Attention Module

- Channel attention submodule

通道注意力被用于关注那些能提供更多信息的通道，此处采用通道注意力子模块去利用CNN特征的通道之间的相互依赖性，示意图如下：



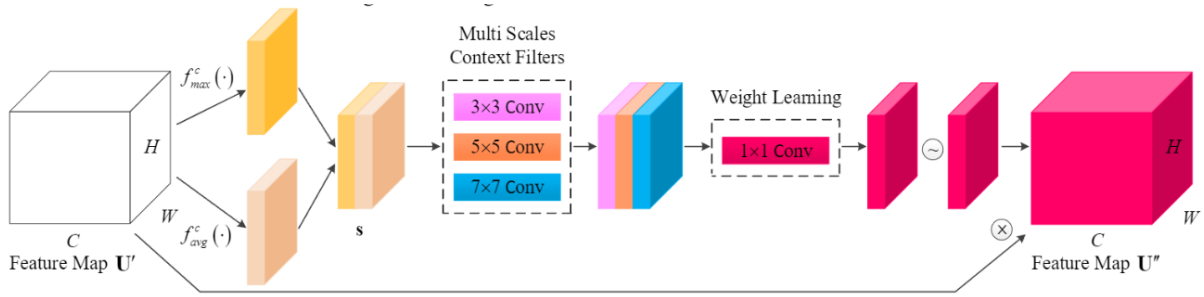
对于输入的特征图U，通过进行最大池化操作f_{max}和平均池化操作f_{avg}生成两个一维的全局通道描述向量v^1和v^2，并将它们通过一个MLP来分析通道之间的相互依赖性，最终经过相加和sigmoid激活函数之后得到通道注意力描述向量$Z^C(U)$，最终输出的通道注意力图U'由通道注意力描述向量和输入特征图进行对应元素相乘得到。

$$Z^C(U) = \delta(f_{ext}(f_{max}(U)) + f_{ext}(f_{avg}(U)))$$

$$U' = Z^C(U) \otimes U$$

- Spatial attention submodule

空间注意力被用于关注那些有意义的特征单元，示意图如下：

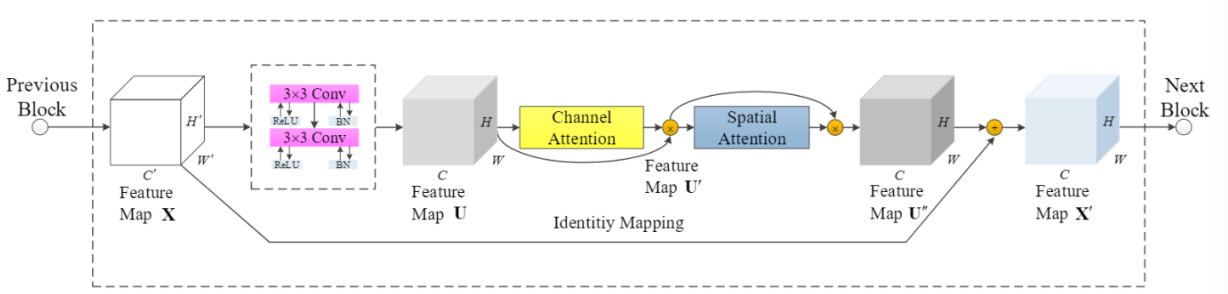


对于输入的特征图U'，使用通道维度的最大池化操作f^c_{max}和平均池化操作f^c_{min}来得到压缩的特征掩码，并将它们拼接成一个$W \times H \times 2$的向量s，并通过三种不同尺寸的context filter来利用特征单元的上下文信息，将结果拼接得到$W \times H \times 3$的特征掩码，接着使用1×1的卷积去学习权重，最终经过sigmoid激活函数得到空间注意力掩码$Z^S(U')$。空间注意力图由空间注意力掩码和通道注意力图进行对应元素相乘得到，它可以被看作是特征掩码的加权和。

$$Z^S(U') = \delta(f^{1\times1}(f^{3\times3}(s); f^{5\times5}(s); f^{7\times7}(s)))$$

$$U'' = Z^S(U') \otimes U'$$

总体的注意力模块是通过通道注意力子模块和空间注意力子模块的顺序组合形成的，示意图如下：



2. Hard Exemplar Reweighting Triplet Loss

受到soft-margin三元组损失的启发，作者提出了基于三元组重赋权重的在线难样本挖掘策略，给每个三元组分配的权重由它们的困难等级决定，loss的定义式如下：

$$L_{hard}(A_i, P_i, N_{i,k}) = w_{hard}(A_i, P_i, N_{i,k}) * log(1 + exp(d_p(i) - d_n(i,k)))$$

作者在此引入了$gap(i,k) = d_n(i,k) - d_p(i)$的概念，满足$gap(i,k) \leq 0$的样本被视为极端难样本，满足$gap(i,k) \geq m$的样本被视为带有很少有效信息的简单样本，同时考虑上述两种情况，为每个anchor定义一个参考负样本距离$D_{ref} = dp(i) + \frac{m}{2}$，即满足$gap(i,k) = \frac{m}{2}$的样本被称为参考负样本。

基于$gap(i,k)$，作者提出了权重的计算方法：

$$w(A_i, P_i, N_{i,k}) = -log_2(p_{match}(A_i, P_i, N_{i,k}))$$

$$p_{match}(A_i, P_i, N_{i,k}) = \frac{1}{1 + exp(-gap(i,k) + \beta)}$$

其中$\beta = \frac{m}{2}$表示距离校正因子，p_{match}表示三元组匹配正确的概率，当网络将anchor与positive和negative匹配的概率相等时，$p_{match} = 0.5$，此时该三元组的权重w为1。其它$gap(i,k)$更小的三元组会导致p_{match}减小，从而权重w会大于1；$gap(i,k)$更大的三元组会导致p_{match}增大，从而权重w会小于1。

此外，作者为了避免权重过大或者过小，还加入了权重阈值：

$$w_{high} = -log_2(\frac{1}{1+exp(\beta)})$$

$$w_{low} = -log_2(\frac{1}{1+exp(-m+\beta)})$$

权重大于w_{high}则被设置成w_{high}，权重小于w_{low}就被设置成一个很小的值$\frac{\varepsilon}{\beta}$，最终的三元组权重公式为：

$$w_{hard}(A_i, P_i, N_{i,k}) = \begin{cases} \frac{\varepsilon}{\beta}, & gap(i,k) \geq m \\ w_{high}, & gap(i,k) \leq 0 \\ w(A_i, P_i, N_{i,k}), & otherwise \end{cases}$$

3. Orientation Regression

在现有的跨视角地理定位问题中，anchor和它对应的positive样本的角度在训练集中是固定的，但在测试集中是被打乱的。因此，由随机旋转产生的角度可以作为训练的标签。为了解决方位不确定性的问题，作者添加了一个额外的方向回归辅助损失如下：

$$L_{OR}(A_i, P_i, N_{i,k}) = w_{hard}(A_i, P_i, N_{i,k}) * (d_R^1(i) + d_R^2(i))$$

其中的$d_R^1(i)$和$d_R^2(i)$分别表示由旋转角度带来的sin和cos值的回归损失。最终的HER loss由主损失和辅助损失的组合得到：

$$L_{HER}(A_i, P_i, N_{i,k}) = \lambda_1 * L_{hard}(A_i, P_i, N_{i,k}) + \lambda_2 * L_{OR}(A_i, P_i, N_{i,k})$$

总结

- attention的思想贯穿了网络结构
- 在获取每个样本的权重上很有创新，抛弃了已有的如使用小型网络获取权重的方法，而利用逻辑回归来计算每个样本的权重

---

5. LoOp: Looking for Optimal Hard Negative Embeddings for Deep Metric Learning

难样本挖掘和难样本生成的劣势？

- 在难样本挖掘中，已经满足判别条件的样本对损失的贡献较少，这会导致训练收敛变慢，且最终模型在难样本的一个子集上过拟合，不能充分利用其他样本所提供的信息。
- 在难样本生成中，尽管能利用到所有的训练样本，但仍需要一个额外的网络作为生成器，这将增加训练时间和计算负载，最终导致优化困难。

基于以上挑战，作者阐述了一个名为“寻找两条有界曲线之间的最小距离”的通用问题，并给出了解法。基于该解法，作者提出了一种寻找最优难负样本的方法，既不像难样本挖掘那样会忽略部分样本，也不像难样本生成那样会增加训练复杂性和优化难度，并且能够轻松地泛化到不同的损失函数上。

方法

1. 问题阐述

给出特征空间中属于两个类的两对点x_1, x_2和y_1, y_2，它们已经被l_2归一化，且分布在单位超球面上。多数基于度量学习的损失函数尝试优化$d(x_1, x_2)$和$d(x_1, y_1)$之间的差距，其中$d(\cdot, \cdot)$表示欧氏距离。考虑对损失函数具有较大贡献值的样本，我们应该试着增大上述差距。如果考虑生成样本x_3使得$d_(x_1, x_3) > d(x_1, x_2)$，在缺少其他假设的前提下很难保证它的类别归属。因此，可以利用剩余的样本(x_2, y_2)来最小化$d(x_1, y_1)$表示的负样本距离。

寻找两个点p_1和p_2，它们分别位于连接x_1和x_2、y_1和y_2的曲线上，最小化p_1和p_2之间的距离，最终用它作为anchor和negative之间的距离，即：

$$d(p_1, p_2) = ||p_1 - p_2||_2 = \sqrt{2(1 - p_1 \cdot p_2)} \tag{1}$$

作者提出一个论点，给出特征空间中属于两个类的两对点x_1, x_2和y_1, y_2，那么在曲线$\widehat{x_1x_2}$和$\widehat{y_1y_2}$上的点由很大的概率和x_1、x_2和y_1、y_2属于同一类别。

基于以上论点，点p_1由点x_1沿曲线$\widehat{x_1x_2}$向点x_2旋转角度α得到，α位于0和$\alpha_0 = cos^{-1}(x_1 \cdot x_2)$之间；同理，点$p_2$由点$y_1$沿曲线$\widehat{y_1y_2}$向点$y_2$旋转角度$\beta$得到，$\beta$位于0和$\beta_0 = cos^{-1}(y_1 \cdot y_2)$之间。

使用Gram-Schmidt正交化得到基向量为：

$$\mathbf{n_1} = \mathbf{x_1}\ ;\ \mathbf{n_2} = \frac{\mathbf{x_2} - (\mathbf{x_1} \cdot \mathbf{x_2})\mathbf{x_1}}{||\mathbf{x_2} - (\mathbf{x_1} \cdot \mathbf{x_2})\mathbf{x_1}||_2}. \tag{2}$$

使用Rodriguez theorem计算旋转矩阵为：

$$\begin{aligned}\mathbf{R} = \mathbf{I} &+ \sin\alpha(\mathbf{n_2}\mathbf{n_1}^T - \mathbf{n_1}\mathbf{n_2}^T) \\ &- (1 - \cos\alpha)(\mathbf{n_1}\mathbf{n_1}^T + \mathbf{n_2}\mathbf{n_2}^T),\end{aligned}$$

由$p_1 = Rx_1$和$p_2 = Rx_2$，化简后得到：

$$\mathbf{p_1} = \mathbf{n_1}\cos\alpha + \mathbf{n_2}\sin\alpha. \tag{3}$$

$$\mathbf{p_2} = \mathbf{n_3}\cos\beta + \mathbf{n_4}\sin\beta, \tag{4}$$

对（1）式进行优化，得到目标函数f形式如下：

$$\begin{aligned}f(\alpha, \beta) = -\mathbf{p_1} \cdot \mathbf{p_2} &= a\sin\alpha\sin\beta \\ &+ b\cos\alpha\sin\beta + c\sin\alpha\cos\beta + d\cos\alpha\cos\beta,\end{aligned} \tag{5}$$

应当将f优化到尽可能小，其中$a = -n_2 \cdot n_4$,$b = -n_1 \cdot n_4$,$c = -n_2 \cdot n_3$,$d = -n_1 \cdot n_3$。再添加两个约束如下：

$$g_1 = -\alpha \le 0\ ;\ g_2 = \alpha - \alpha_0 \le 0. \tag{6}$$

$$g_3 = -\beta \le 0\ ;\ g_4 = \beta - \beta_0 \le 0. \tag{7}$$

最终的约束优化拉格朗日函数如下：

$$L(\alpha, \beta, \lambda_1, \lambda_2, \lambda_3, \lambda_4) = f(\alpha, \beta) - \sum_{i=1}^{4} \lambda_i g_i, \qquad (8)$$

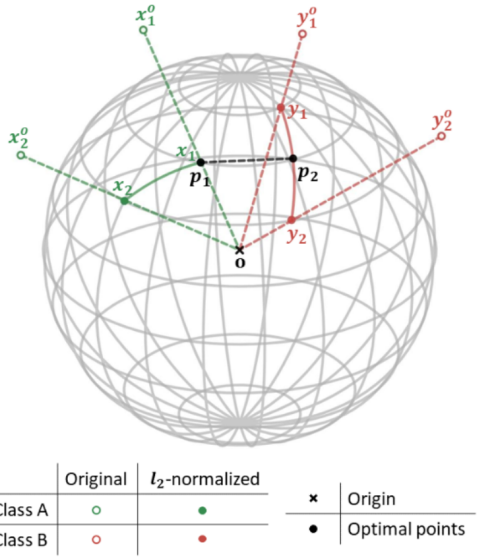
2. Finding Optimal Distance

根据KKT条件对（8）式进行优化，得到以下式子：

$$\begin{aligned} \frac{\partial L}{\partial \alpha} &= a\cos\alpha\sin\beta - b\sin\alpha\sin\beta + c\cos\alpha\cos\beta \\ &\quad - d\sin\alpha\cos\beta + \lambda_1 - \lambda_2 = 0, \qquad (9) \\ \frac{\partial L}{\partial \beta} &= a\sin\alpha\cos\beta + b\cos\alpha\cos\beta - c\sin\alpha\sin\beta \\ &\quad - d\cos\alpha\sin\beta + \lambda_3 - \lambda_4 = 0, \qquad (10) \end{aligned}$$

$$\lambda_i g_i = 0\,;\, i = 1,2,3,4, \qquad (11)$$
$$\lambda_i \le 0\,;\, i = 1,2,3,4, \qquad (12)$$
$$g_i \le 0\,;\, i = 1,2,3,4. \qquad (13)$$

最终根据以上约束条件解得一个最优解α和β，代入（3）和（4）式可以解得点p_1和p_2，再根据（1）式计算出anchor与negative之间的距离d，至此已经完成了通用问题的解答。



3. Optimal Hard Negative Embeddings for Deep Metric Learning

令$d_{i,j}$表示样本$X[i]$和$X[j]$之间的距离，$d_{i,j,k,l}$表示连接$X[i]$和$X[j]$、$X[k]$和$X[l]$的曲线之间的最优距离，其中有$c[i] = c[j] \neq c[k] = c[l]$。下面将该距离用于已有的损失函数上。

- Triplet Loss

原始的表达形式为：

$$\mathcal{L}_{Tri} = \frac{1}{|\mathcal{P}|} \sum_{\substack{(i,j)\in\mathcal{P} \\ k:\mathbf{c}[i]\neq\mathbf{c}[k]}} [d_{i,j} - d_{i,k} + m]_+ ,$$

使用LoOp修改后的表达形式为：

$$\mathcal{L}'_{Tri} = \frac{1}{|\mathcal{P}|} \sum_{\substack{(i,j)\in\mathcal{P} \\ k,l:\mathbf{c}[i]\neq\mathbf{c}[k]=\mathbf{c}[l]}} [d_{i,j} - d_{i,j,k,l} + m]_+ .$$

- HPHN Loss

Hard positive and hard negative(HPHN)用于在一个batch中利用对训练帮助最大的样本，原始的表达形式为：

$$\mathcal{L}_{HPHNtri} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left[\max\left(\max_{\mathbf{c}[i]=\mathbf{c}[k]} d_{i,k}, \max_{\mathbf{c}[j]=\mathbf{c}[l]} d_{j,l} \right) + m - \min\left(\min_{\mathbf{c}[i]\neq\mathbf{c}[k]} d_{i,k}, \min_{\mathbf{c}[j]\neq\mathbf{c}[l]} d_{j,l} \right) \right]_+ .$$

使用LoOp修改后的表达形式为：

$$\mathcal{L}'_{HPHNtri} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left[\max\left(\max_{\mathbf{c}[i]=\mathbf{c}[k]} d_{i,k}, \max_{\mathbf{c}[j]=\mathbf{c}[l]} d_{j,l} \right) + m - \min_{\mathbf{c}[i]\neq\mathbf{c}[k]=\mathbf{c}[l]} d_{i,j,k,l} \right]_+ .$$

- Lifted Structure Loss

Lifted Structure Loss的目的是使得同类得一对样本远离其他所有不同类的样本，原始的表达形式为：

$$\mathcal{L}_{LS} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left[d_{i,j} + m - \min\left(\min_{\mathbf{c}[i]\neq\mathbf{c}[k]} d_{i,k}, \min_{\mathbf{c}[j]\neq\mathbf{c}[l]} d_{j,l} \right) \right]_+ . \qquad (16)$$

使用LoOp修改后的表达形式为：

$$\mathcal{L}'_{LS} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left[d_{i,j} + m - \min_{\mathbf{c}[i]\neq\mathbf{c}[k]=\mathbf{c}[l]} d_{i,j,k,l} \right]_+ .$$

总结

文章提出了一种基于最小化两条类曲线之间的距离来寻找最优难负样本的方法，这种设计结合了难样本挖掘和难样本生成的优点，又基于全部样本来计算难负样本距离，而非直接生成，巧妙地同时避免了二者的缺点。