

目录

1	2023-3-23 小论文框架图	2
---	------------------	---

1 2023-3-23 小论文框架图

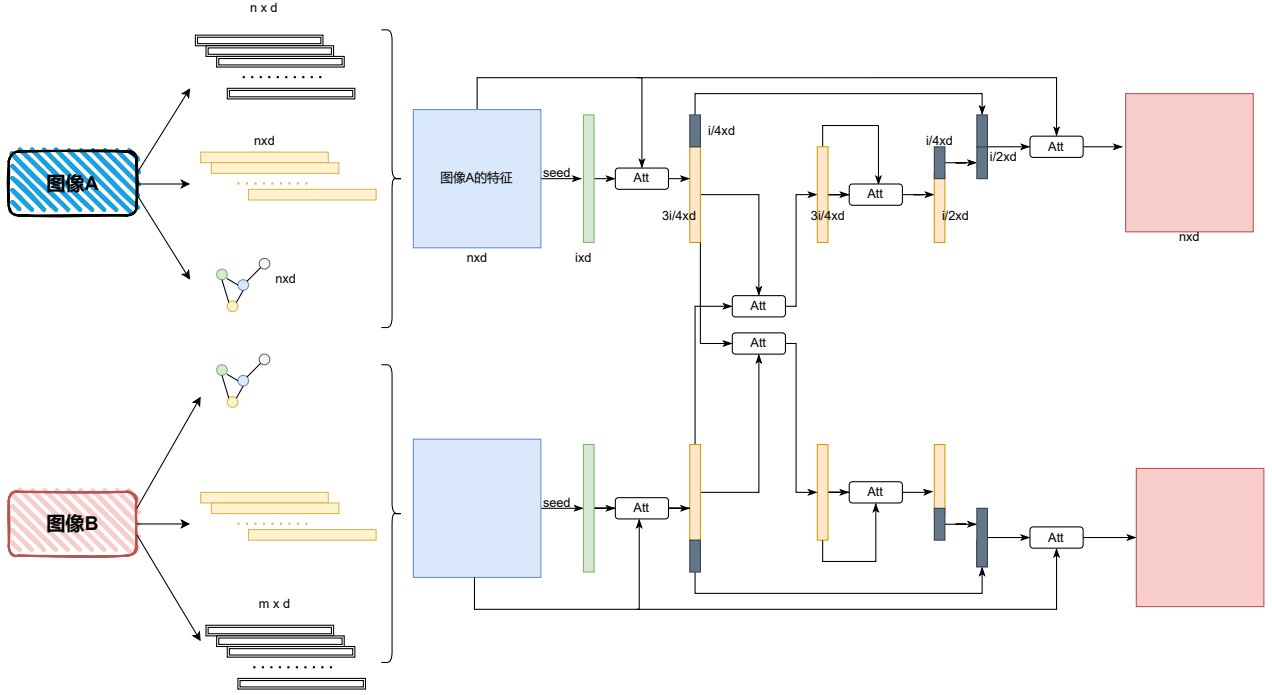


图 1: 论文深度学习网络草图

学习进展：

1、完成图像特征点领域选取和聚合的代码实现。

邻域：

$$\mathcal{N}_i = \{\mathbf{n} \in \text{topk}(\mathbf{D}_i)\} \quad (1.1)$$

聚合方法：也是整个网络采取的一种带权重的注意力机制

$$\mathbf{X}' = \text{Att}(\mathbf{X}, \mathbf{Y}, \omega) \quad (1.2)$$

2、训练网络设计，大致框图如图 1 所示。

3、参照其它特征点匹配文章，选取其中的上下文正则化方法作为训练网络不同层匹配点特征的评价函数。正则化在网络中一般是用来减少网络复杂度，防止梯度消失或爆炸，加速收敛，这个思想也可以用来评价。

4、理解了一下 sinkhorn 算法用于求解图像之间的分配矩阵。因为图之间的匹配可抽象为二次分配问题，通俗来说就是求解两个样本分布或数据之间的最短距离，也可用作图像特征点（像素点）之间最优映射求解方法，通过一个迭代算法使得分配矩阵达到收敛或迭代次数。算法的具体步骤如下：

初始化：给定两个概率分布 μ 和 ν ，初始化一个矩阵 K ，其中 $K_{i,j} = \exp(-c_{i,j})$ ， $c_{i,j}$ 是 μ 中第 i 个元素和 ν 中第 j 个元素之间的距离，可以是欧几里得距离、马氏距离等。

迭代更新：重复执行以下步骤，直至收敛：

- 计算 γ ： $\gamma_{i,j} = \frac{K_{i,j} u_i v_j}{\epsilon + \sum_{i',j'} K_{i',j'} u_{i'} v_{j'}}$ ，其中 u 和 v 是两个向量，分别表示从 μ 和 ν 出发的概率质量。
- 更新 u 和 v ： $u_i = \frac{\mu_i}{\sum_j \gamma_{i,j}}$ ， $v_j = \frac{\nu_j}{\sum_i \gamma_{i,j}}$ 。

输出 γ 。

其中， ϵ 是一个较小的正数，用于避免分母为零。

下周安排：

1、根据框图完成训练网络的代码实现。

2、理解上下文正则化方法，将其改变为匹配点的评价方法，并实现为一种可训练网络层。

3、实现 sinkhorn 算法。

4、损失函数设计，除了最基本对于结果和分配矩阵的损失计算 (参考论文)，需要设计一个对注意力转移的损失函数，约束注意力析出的匹配点为最优匹配。