一个关于基本计算问题的神经算法

相似性搜索（例如，识别数据库中的相似图像或网上的类似文件）是大规模信息检索系统面临的基本计算问题。我们发现果蝇嗅觉通路通过计算机科学算法（称为局部敏感哈希）的变体解决了这个问题。果蝇通路将类似的神经活动模式分配给类似的气味，以便当经历类似的气味时，可以应用从一种气味学习的行为。然而，果蝇算法使用三种不同于传统方法的计算策略。这些策略可以被转换来提高计算相似性搜索的性能。这种观察有助于阐明支持重要感官功能的逻辑，并为解决基本的计算问题提供了一种概念上的新算法。

许多神经回路的基本任务是产生响应输入刺激的神经活动模式，以便可以明确识别不同的输入。 我们研究了用于处理果蝇嗅觉系统中的气味的电路，并揭示了一种计算策略来解决基本机器学习问题：近似相似性（或最近邻居）搜索。

果蝇嗅觉回路为每种气味产生一个“标签”，这是一组神经元，当出现气味时，该神经元就会发出（1）。 这个标签对学习对不同气味的行为反应至关重要（2）。 例如，如果奖励（例如，糖水）或惩罚（例如电击）与气味相关联，则该气味变得有吸引力（苍蝇会接近气味）或排斥性（苍蝇会避免气味）。分配给气味的标签是稀疏的 - 只有一小部分接收气味信息的神经元会响应每种气味（3-5）——并且不重叠：两种随机选择的气味的标签共享极少数（如果有的话）活动神经元，从而不同气味可以很容易区分（1）。

气味标签通过三步程序计算（图1A）。第一步涉及从苍蝇鼻子中的气味受体神经元（ORN）到被称为肾小球结构的投影神经元（PN）的前馈连接。有50种ORN类型，每种类型对不同的气味具有不同的敏感度和选择性。因此，每个输入气味对应有由50个ORN的产生神经冲动的速率（放电速率）确定的50维空间中的位置。对于每种气味，50种ORN类型的ORN放电速率的分布呈指数级分布的，其均值取决于气味浓度（6,7）。对于PNs，这种浓度依赖被删除了（7,8）;也就是说，50种PN类型的放电速率分布呈指数分布，其平均值接近所有气味和所有气味浓度均值的（1）。因此，回路中的第一步基本上是使用称为除法归一化（8）的技术来将平均值居中——在许多计算流水线中的标准预处理步骤。这一步很重要，这样果蝇才不会将气味强度与气味类型混淆。

第二步，主要算法见解开始，涉及神经元数量的40倍扩展：通过稀疏二元随机连接矩阵（9）将50个PN目标投影到2000个Kenyon细胞（KCs）。 每个KC从大约六个随机选择的PN（9）中接收并累加放电速率。 第三步涉及一个胜者全得（WTA）回路，其中的强抑制反馈来自单个抑制性神经元，被称为APL（前侧副神经元）。 结果，除了最高发射率的5％的KCs都被抑制（沉默）了（1,3,4）。 这些剩余5％的放电速率对应于分配给输入气味的标签。

从计算机科学的角度来看，我们把果蝇的回路看作是一个哈希函数，它的输入是一种气味，它的输出是一个对应标签(称为哈希)。虽然标签应该区分气味，但果蝇的优势是将非常相似的气味与类似的标签联系起来(图1B)，这样我们曾经从某一种气味中学习到的条件反射，可以应用到以后遇到的相似的气味上，或者是学习到的气味的噪音版本。这使我们推测，果蝇的电路产生的标签是局部敏感的；即两种气味越相似（由50个ORN放电速率定义），它们分配的标签就越相似。有趣的是，局部敏感哈希[LSH(10 11)]是解决计算机科学中众多相似搜索问题的基础。我们将对果蝇回路的见解转化然后开发了一类LSH算法，用于有效地找到高维点的近似最近邻居

想象一下，你被提供了一张大象的图片，并试图从网上的数十亿张图片中找到100张图片，这些图片看起来和你的大象形象非常相似。这被称为最近邻搜索问题，它在信息检索、数据压缩和机器学习(10)中具有重要的意义。每幅图像通常被表示为一个d维特征向量（果蝇处理的每种气味都相当于一个50维空间的特征向量）。使用距离度量来计算两个图像之间的相似性(特征向量)，目标是有效地找到任何查询图像的最近邻。如果Web只包含很少的图像，那么使用蛮力线性搜索就可以很容易地找到准确的最近邻。如果Web包含许多图像，但是每个图像都由一个低维度的向量表示(例如，10或20个特征)，那么空间分区方法(12)也就差不多足够了。然而，对于具有高维度数据的大型数据库，这两种方法规模都不够(11)。

在许多应用程序中，只要能够很快找到它们，就可以返回接近于查询的最近邻的近似集。这促进了一种方法，通过一种称为局部敏感哈希LSH(10)的概率技术来寻找近似的最近邻。对于果蝇来说，气味的标签(或哈希)对应于该气味的凯尼恩细胞激活率向量。局部敏感特性表明，两种相似(如甲醇和乙醇)的气味将由两个自身相似的标记来表示(图1B)。同样地，对于图像搜索，大象图像的标记将更类似于另一个大象图像的标记，而不是摩天大楼图像的标记。

不像传统的(非LSH)哈希函数，输入点是随机分布的，并且在范围上均匀分布，LSH函数h提供了一个从d维空间到m维空间(后者对应于标记)的点的距离保护。因此，在输入空间中距离较近的点被分配相同或相似的标记的概率要比相隔很远的点的概率高。[(13)给出了一个普遍的定义]

为了设计一个LSH函数，一个常用的技巧是计算输入数据的随机投影(10,11)-将输入特征向量乘以一个随机矩阵。The Johnson-Lindenstrauss lemma(14,15)和它的许多变体(16-18)，为我们使用各种各样随机投影将数据从d维嵌入到m维时，多大程度地保存了局域的相对结构提供了强大的理论边界支持。

果蝇还通过随机投影（图.1A步骤2，50 PNs 2000 KCs)为气味分配标签，这为这部分回路的功能提供了一个关键的线索。然而，果蝇算法与传统的LSH算法有三个不同之处。首先，果蝇使用的是稀疏的、二进制的随机投影，而LSH函数通常使用密集的、高斯随机投影，而这些预测要计算的成本要高得多。第二，果蝇在投影后扩展了输入的维度(d<<m)，而LSH收缩了维度(d>>m)。第三，果蝇用一种用“赢”的机制来稀疏化高维度的表示，而LSH保留了稠密的表示。

图1:

(A)：果蝇嗅觉回路原理图。第一步，果蝇鼻子里的50个气味受体神经元把轴突送到肾小球里的50个投射神经元上；这样投影以后，每一种气味被表示为激活率的指数分布，所有气味和所有气味浓度有相同的均值。第二步，投射神经元进行了维度扩展，即通过连接一个稀疏的、二值的、随机投影矩阵投影到了2000个凯尼恩细胞上；第三步，KCs从前对侧(APL)神经元中获得反馈抑制，这只剩下前5%的KCs会保持对气味的激活刺激，这5%对应于气味的标签(哈希)。   
(B)：气味反应说明。类似的气味(如甲醇和乙醇)的标签比不同的气味更相似。较暗的阴影表示较高的活性。   
(C)：传统LSH和果蝇算法之间的区别。在这个例子中，LSH和苍蝇的计算复杂度是一样的。输入维d=5，LSH计算m=3个随机投影，每一个都需要10次操作(5次乘+5次加)。果蝇计算m=15个随机的投影，每一个都需要2个加法运算。因此，两者都需要30个总操作。

在补充材料(13)中，我们证明了果蝇回路的前两个步骤产生的标签可以在预期中保持输入气味的二范数距离。总之，这些结果证明了果蝇的回路代表了一个新的未知的LSH（局部敏感哈希）家族。

然后，我们根据每个算法识别给定查询点的最近邻点的精确度，凭经验对果蝇算法与传统LSH（10,11）进行评估。 为了进行公平的比较，我们将两种算法的计算复杂度改为相同的（图1C）。 也就是说，对于每个输入，这两种方法被设定为使用相同数量的数学运算来（13）生成长度为k的哈希（即，具有k个非零值的向量）。

我们比较了这两种算法在三个基准数据集上寻找最近邻的效果:SIFT(d=128)、GLOVE(d=300)和MNIST(d=784)(13)。SIFT和MNIST都包含用于图像相似搜索的图像的矢量表示，而GLOVE则包含用于语义相似搜索的词的向量表示。我们收集了每个数据集的一个子集，每个数据集都有10,000个输入，其中每个输入都表示为d维空间中的中的一个特征向量。为了测试性能，我们从10000中选择了1000个随机的查询输入，并将真实的和预期的最近邻进行了比较。也就是说，对于每个查询，我们在输入空间中找到了最接近的2%(200)的最近邻，这取决于特征向量之间的欧氏距离。然后我们在哈希空间中找到了最接近的2%(200)的最近邻，基于欧氏距离标记(哈希)的距离确定。我们对哈希(k)的长度进行改变，并利用中间平均精度(19)计算出了排序表上真值和预测值最近邻之间的重叠。我们在50次试验中平均了中间平均精度，在每一次试验中随机投影矩阵和查询输入都改变了。我们分离了果蝇算法和LSH之间的三个不同之处，以测试它们分别对近邻检索性能的影响。

用稀疏的二进制随机投影替代LSH的稠密的高斯随机投影，并不会影响确定最近邻的精确度。我们发现，这两个随机投影类型在所有三个数据集和所有哈希长度(图2A)中产生了几乎相同的检索性能。这些结果支持我们的理论计算，即果蝇的随机投射确实是对局部敏感的。此外，稀疏的二元随机投影在计算上节省了20倍于稠密的高斯随机投影的计算量(图S1)。

在扩展维度时，使用WTA来稀疏标签会比使用随机标，签导致更好的性能（图2B）。 与随机标签会随机选择k个KC不同，WTA会选择前k个最高放电速率的KC作为标签。 对于这两种情况，我们对果蝇使用了20k的随机投影，将果蝇算法和LSH两个算法所用的数学运算的数量等同了起来。

图2: 不同随机投影类型和标签选择方法的实验比较。在所有的图中，x轴是哈希的长度，而y轴是中间平均精度(越高越好)。A）：稀疏的二元随机投影提供了几乎相同的性能，就像稠密的高斯随机投影一样，但是前者节约了大量的计算。B）：扩展的维度(来自k到20k)加上“赢者通吃”的稀疏化，进一步提高了性能，相比没有扩展的来说。所有三个基准数据集的结果是一致的。误差条表示标准偏差超过50次试验。 

图3：果蝇算法和局部敏感哈希算法的总体比较。在所有的图中，x轴是哈希的长度，而y轴是中间平均精度(越高越好)。10d的扩展用于果蝇。在所有这三个数据集中，果蝇算法方法优于LSH，最显著的是短代码。误差条表示标准偏差超过50次试验。

例如，在使用hash长度k=4的SIFT数据集上，随机选择产生了17.7%的平均精度，而使用WTA(32.4%)的平均值是大约两倍。因此，选择激活性能靠前的神经元最好地保持了输入之间的相对距离;增加的维度也使得隔离不同的输入变得更加容易。对于随机标签选择，我们选择了k个随机(但所有输入都是固定的)KCs为标签;因此，它的性能与仅仅只是做k个随机投影是完全相同的，就像在LSH中一样。

为了更接近于果蝇回路，我们实现了完整的果蝇算法，但是从20k到10d肯杨细胞的维度的进一步扩展。总的来说，与LSH在所有三个数据集(图3)相比，我们发现了显著的增长。在非常短的代码中，获得的收益最高，我们看到平均平均精度提高了将近3倍(例如，对于MNIST来说，k=4，LSH是16.0%，而果蝇则是44.8%)。 

在对更高维度和二进制LSH测试果蝇算法时，我们收获了相似的结果。 因此，果蝇算法是可扩展的，并可能在其他LSH家族(种类)中有效。

我们的工作认识到了大脑中的相似匹配策略和大型信息检索系统中进行最近邻搜索的算法之间的协同作用。这个发现可能也可以应用到重复检测、聚类和高效的深度学习中。/ 有许多局部敏感哈希的扩展,包括使用多个哈希表(11)来提高精度(过去我们两个算法仅仅使用一个哈希表),使用多探测器——这样类似的标签就可以组合在一起到相同的容器(因为标签是稀疏的，可能对于果蝇来说更容易实现),各种用于离散化哈希的量化技巧 (25)，和学习[叫做数据相关哈希]。还有一些方法可以加速随机投影乘法，对于LSH，使用快速的Johnson-Lindenstrauss变换(26,27)，对于果蝇，使用快速稀疏矩阵乘法。我们的目标是公平地比较用于解决最近邻搜索问题的在概念上不同的两个方法;在实际应用中，所有这些扩展都需要移植到果蝇算法中。

一些果蝇算法的策略在以前就被使用过。例如，MinHash(28)和winner-take-all哈希(29)都使用了与wta相似的组件，尽管它们都不支持扩展维度;类似地，在许多LSH家族中，随机投影也被使用，但据我们所知，没有一个是使用稀疏的、二进制的投影。因此，这些计算成分的结合似乎是新奇的，而且我们认为进化论已经发现了它们应用于果蝇嗅觉。 果蝇嗅觉回路似乎已经演变为使用这些计算成分的独特组合。 果蝇的嗅觉主题（基序）主题的三个特征也可能出现在其他脑区和物种中（表1）。因此，局部敏感哈希可能是大脑中进行计算的普遍原理。