

基于联邦学习的路径优化

Path optimization based on Federated learning

|  |  |
| --- | --- |
| 题 目 | 基于联邦学习的路径优化 |
| 学生姓名 | maybeLocalhost |
| 指导教师 |  |
| 学 院 | 计算机学院 |
| 专业班级 |  |

本科生院制

2020年12月

基于联邦学习的路径优化

摘要

联邦学习一直是在隐私限制下实现不同组织之间机器学习模型的协作训练的热门研究主题。 随着研究人员尝试使用不同的隐私保护方法来支持更多的机器学习模型，开发系统和基础架构的需求就要求简化各种联邦学习算法的开发。 与促进深度学习发展的深度学习系统（例如PyTorch和TensorFlow）相似，联邦学习系统（FLS）同样重要，并且面临着来自有效性，效率和隐私等各个方面的挑战。 在这项调查中，我们对联邦学习系统进行了全面的审查。 为了实现顺畅的流程并指导未来的研究，我们介绍了联邦学习系统的定义并分析了系统组件。 此外，我们根据数据分发，机器学习模型，隐私机制，通信体系结构，联盟规模和联盟动机等六个不同方面对联盟学习系统进行了彻底的分类。 如我们的案例研究所示，分类可以帮助设计联合学习系统。 通过系统地总结现有的联合学习系统，我们介绍了设计因素，案例研究和未来的研究机会。

**关键词：**联邦学习 路径优化 A\*算法 启发式函数 注意力机制

目录

[第1章 引言 1](#_Toc60944881)

[第2章 联邦学习系统概述 1](#_Toc60944882)

[2.1 背景 1](#_Toc60944883)

[2.2 目标 2](#_Toc60944884)

[2.3 定义 2](#_Toc60944885)

[2.4 分类 2](#_Toc60944886)

[2.5 联邦学习框架 4](#_Toc60944887)

[第3章 路径优化 4](#_Toc60944888)

[3.1 作用 4](#_Toc60944889)

[3.2 组成 4](#_Toc60944890)

[3.3 分类 5](#_Toc60944891)

[3.3.1 Dijkstra算法 5](#_Toc60944892)

[3.3.2 A\*算法 7](#_Toc60944893)

[3.3.3 D\*算法 7](#_Toc60944894)

[3.3.4 LPA\*算法 7](#_Toc60944895)

[3.3.5 D\* lite算法 8](#_Toc60944896)

[3.4 A\*算法优化 8](#_Toc60944897)

[第4章 基于联邦学习的路径优化 9](#_Toc60944898)

[4.1 实现框架 9](#_Toc60944899)

[4.2 注意力机制 10](#_Toc60944900)

[结束语 11](#_Toc60944901)

[参考文献 11](#_Toc60944902)

# 第1章 引言

许多机器学习算法都需要大量数据，实际上，在隐私限制的保护下，数据分散在不同的组织中。 由于这些因素，联邦学习（FL）已成为机器学习中的热门研究主题。例如，不同医院的数据是孤立的，成为“数据孤岛”。 由于每个数据岛在大小和实际分布上都有局限性，因此一家医院可能无法训练高质量的模型，该模型对于特定任务具有良好的预测准确性。 理想情况下，如果医院可以结合其数据来共同训练机器学习模型，则可以从中受益更多。 然而，由于各种政策和法规，数据不能简单地在医院之间共享。 在“数据岛”上的这种现象在金融，政府和供应链等许多领域都很普遍。 通用数据保护条例（GDPR）等政策规定了不同组织之间的数据共享规则。 因此，开发一种具有良好的预测准确性同时遵守政策和法规以保护隐私的联邦学习系统具有挑战性。

联邦学习是隐私保护下机器学习的可实现路径以及“数据孤岛问题”的解决方案，允许从跨数据所有者分布的数据中构建集合模型，提供了跨地域的数据使用和模型构建蓝图，可被广泛应用于各种领域。而目前国内外关于联邦学习在智慧物流的路径优化方面的应用几乎空白，因此。研究基于联邦学习的路径优化系统有着十分重要的意义。

# 第2章 联邦学习系统概述

## 2.1 背景

从诞生至今，人工智能的发展经历了两次低谷和三个高峰，目前正处于第三个发展期。然而除了少数几个领域，大多数领域存在数据量有限的问题，不足以支撑人工智能技术的落地实现；同时，不同行业和公司的数据源之间存在壁垒，数据共享面临重重阻力。如何在保证数据隐私安全的前提下进行联合训练，以提升模型性能，成为了全球瞩目的难题。

联邦学习作为一种加密的分布式机器学习技术，是隐私保护下机器学习的可实现路径以及“数据孤岛问题”的解决方案。它允许从跨数据所有者分布的数据中构建集合模型，提供了跨地域的数据使用和模型构建蓝图，可被广泛应用于各种领域。

## 2.2 目标

联邦学习的目标是保障在大数据交换时的信息安全、保护终端数据和个人隐私，保证合法合规的前提下，在多参与方或多计算节点之间开展高效率的机器学习。它的优势在于数据隔离，数据不会泄露到外部，满足用户隐私保护和数据安全的需求；能够保证模型质量无损，不会出现负迁移，保证联邦模型比割裂的独立模型效果好；参与者地位对等，能够实现公平合作；能够保证参与各方在保持独立性的情况下，进行信息与模型参数的加密交换，并同时获得成长。

## 2.3 定义

在上述情况下，联邦学习是一种无需交换用户原始数据的协作学习，如今受到越来越多的关注。尽管机器学习，尤其是深度学习在最近再次引起了很多关注，但联合身份验证和机器学习已成为一个新的热门研究主题。 FL使多方可以共同训练机器学习模型，而无需交换本地数据。它涵盖了来自多个研究领域的技术，例如分布式系统，机器学习和隐私。这里我们给出FLS的正式定义：

我们假设有个不同的参与方，每个参与方用表示，其中i ∈[1， N]。 我们用表示的数据。 对于非联合设置，每个参与方仅使用其本地数据来训练机器学习模型。 的预测精度表示为。 对于联合设置，所有各方共同训练模型，而每一方根据其特定的隐私限制保护其数据。 的预测精度表示为。 然后，对于有效的FLS，存在i ∈[1， N]，使得>。



在以上定义中，我们仅要求存在任何一方可以通过FL获得更高的模型质量。即使某些参与方可能无法从FL获得更好的模式，但他们仍然可以加入联邦并与其他参与方达成协议，要求其他形式的激励措施。目前来说，联邦学习主要用于跨领域企业的数据合作（同领域的比较敏感，一般比较难合作），长远的愿景是基于这种新的联合建模建立新的业态和模式。

## 2.4 分类

联邦学习根据不同的数据类型，主要分为以下三种类型：

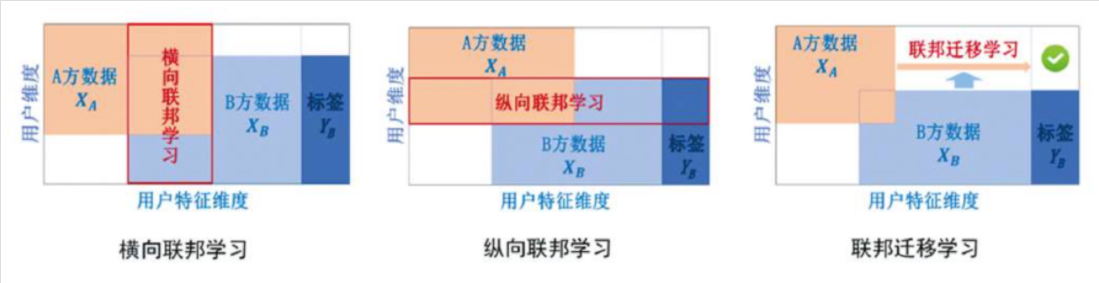


图 2-1 联邦学习模型

横向联邦学习(Horizontal Federated Learning，HFL)也称按样本划分的联邦学习，可以应用于联邦学习的各个参与方的数据集有相同的特征空间和不同的样本空间的场景，类似于在表格视图中对数据进行水平划分。举例，两个地区的城市商业银行可能在各自的地区拥有非常不同的客户群体，所以他们的客户交集非常小，他们的数据集有不同的样本ID。但是他们的业务模型非常相似，因此他们的数据集的特征空间是相同的，这两家银行可以联合起来进行横向联邦学习来构建更好的风控模型。

纵向联邦学习（Vertical Federated Learning，VFL）也可以理解为 按特征划分的联邦学习，可应用于数据集上具有 相同的样本空间、 不同的特征空间的参与方所组成的联邦学习。 出于不同的商业目的，不同组织拥有的数据集通常具有不同的特征空间，但这些组织可能共享一个巨大的用户群体，如上图。 通过使用VFL，我们可以利用分布于这些组织的异构数据，搭建更好的机器学习模型，并且不需要交换和泄露隐私数据。

HFL和VFL要求所有的参与方具有相同的特征空间或样本空间，从而建立起一个有效的共享机器学习模型。但是，在更多的实际情况下，各个参与方拥有的数据集可能存在高度的差异：

* 参与方的数据集之间可能只有少量的重叠样本和特征；
* 这些数据集的分布情况可能差别很大；
* 这些数据集的规模可能差异巨大；
* 某些参与方可能只有数据，没有或只有很少的标注数据

为解决上述问题，联邦学习可以结合迁移学习技术，使其可以应用于更广的业务范围，同时可以帮助只有少量数据（较少重叠的样本和特征）和监督（较少标记）的应用建立有效且精确的机器学习模型，并且遵守数据隐私和安全条例的规定。将这种组合称为联邦迁移学习。

## 2.5 联邦学习框架

这里主要列举一些开源的联邦学习框架项目（根据开源时间排序）：

* PySyft 是一个深度学习安全隐私计算框架，支持 Tensorflow/Keras/Pytorch，主要用于学术研究和原型开发；
* Asylo 是 Google 发布的机密运算框架，在可信执行环境(Trusted Execution Environment, TEE)中执行；
* Apache Teaclave 原来是百度的 MesaTEE 项目，目前在 Apache 孵化中，也是 TEE 方案；
* CrypTen 是 Facebook 发布的基于 PyTorch 的隐私保护机器学习框架；
* FATE 是微众银行开源的联邦学习平台化方案，大而全；
* Tensorflow Federated 是谷歌开源的联邦学习框架；
* PaddleFL 是百度开源的基于PaddlePaddle的开源联邦学习框架。

目前网络上对于 PySyft 和 FATE 的介绍相对较多，大部分的文章中也会主要以这两个框架/平台进行试验和研究。

# 第3章 路径优化

## 3.1 作用

路径规划（path planning）目的是为了寻找成本最小的路径, 需要给出的条件包括: 地图, 起始地点, 目标地点, 成本函数. 路径规划模块性能的高低直接关系车辆行驶路径选择的优劣和行驶的流畅度，而路径规划算法的性能优劣很大程度上取决于规划算法的优劣，如何在各种场景下迅速、准确的规划出一条高效路径且使其具备应对场景动态变化的能力是路径规划算法应当解决的问题。

## 3.2 组成

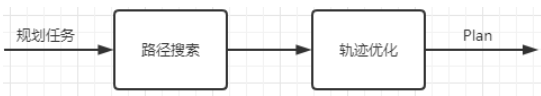


图 3-1 路径规划

路径规划主要由两部分组成，前端是路径搜索，后端是轨迹规划，最后生成一条机器人可执行的路径。

* 路径搜索——在地图中，搜索出一条避开障碍物的轨迹；
* 轨迹规划（优化）——对搜索到的轨迹进行优化，从而符合机器人的运动学和动力学约束。

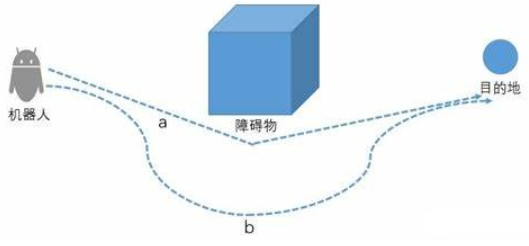


图 3-2 轨迹优化

如上图所示，a路径为前端路径搜索得出的，为一条无碰撞的路径，b为a路径经过轨迹规划过的路径，该路径满足运动学和动力学约束，为一条机器人可执行的路径。

## 3.3 分类

路径规划（搜索）的算法主要可分成这三类：

1. 基于搜索的算法：DIjkstra、A\*、JPS；
2. 基于采样的算法：PRM、RRT、RRT\*；
3. 满足动力学约束的搜索算法：Hybrid A\*。

### 3.3.1 Dijkstra算法

Dijkstra算法是由E.W.Dijkstra于1959年提出，又叫迪杰斯特拉算法。该算法采用了一种贪心模式，其解决的是有向图中单个节点到另一节点的最短路径问题，其主要特点是每次迭代时选择的下一个节点是当前节点最近的子节点，也就是说每一次迭代行进的路程是最短的。而为了保证最终搜寻到的路径最短，在每一次迭代过程中，都要对起始节点到所有遍历到的点之间的 最短路径 进行更新，具体看一下过程。

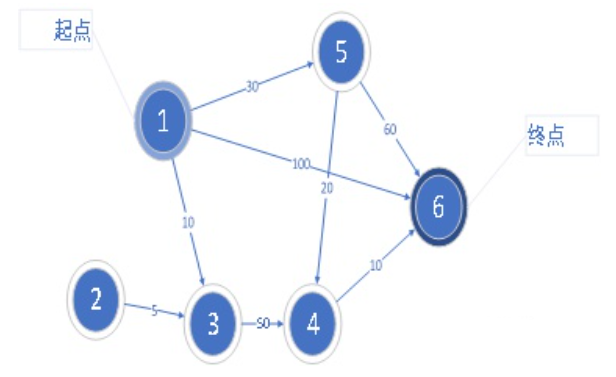


图 3-3 节点图

初始化：建立distance[](起点到其他所有点的距离信息)、Top\_node[]（最短路径信息）两个列表存放信息。其中，distance[]的维度为节点的个数，每一个数值为到达对应索引节点的最短路径距离，比如distance[2]的值代表当前迭代时刻到达3号节点的最短距离。初始状态distance[0 inf 10 inf 30 100]，其中0代表自身，inf代表无法到达；Top\_node[num1]，其中num1代表一号节点并以此类推。

搜索最小点：找到当前节点到下一点的最小值，即从num1开始搜索到1->5/1->3/1->6三条路，并找到距离最小的路1->3。则此时到达num3点的最短路径确定为10，将num3存入Top\_node[]。

松弛：确定num3找到最短路径，然后num3开始搜寻其弧尾，找到3->4路径，此时1->3->4路径距离为10+50=60，小于inf，故将列表更新为distance[0 inf 10 60 30 100]。注意这里通过3->4这条路径缩短1->4这条路径的过程叫做“松弛”，该算法证实通过这样的方法进行路径寻优。

重复迭代：除去num1和num3，从剩余点搜寻距离最小，找到num5，故将num5加入Top\_node[]。找到弧尾路径5->4/5->6，进行松弛，其中1->5->4距离为30+20=50<60，1->5->6距离为30+60=90<100，所以列表更新为distance[0 inf 10 50 30 90]。

重复迭代：除去num1、num3和num5，其余点寻最小，找到num4，将其加入Top\_node[]。然后找到弧尾4->6，进行松弛，1->5->4->6距离30+20+10=60<90，1->3->4->6距离10+50+10=70>60，进行列表更新distance[0 inf 10 50 30 60]。

重复迭代：除去num1、num3、num4和num5，其余点寻最小，找到num6，将其加入Top\_node[]，然后没有找到弧尾，则此时到达num6的最优路径找到。

以上就是Dijkstra算法的简单实例，可见其主要是通过贪心原则逐个遍历最小子节点，然后利用松弛方法去优化路径选择，最终将最优路径存放到可读列表当中，以此来解决最优路径规划问题。

### 3.3.2 A\*算法

A\*算法是启发式搜索算法，启发式搜索即在搜索过程中建立启发式搜索规则，以此来衡量实时搜索位置和目标位置的距离关系，使搜索方向优先朝向目标点所处位置的方向，最终达到提高搜索效率的效果。

A\*算法的基本思想如下：引入当前节点x的估计函数f(x),当前节点x的估计函数定义为：

f (x)= g(x)+h(x)

其中g(x)是从起点到当前节点x的实际距离量度（代码中可以用两点之间距离代替）；h(x)是从节点x到终点的最小距离估计，h(x)的形式可以从欧几里得距离或者曼哈顿距离中选取。算法基本实现过程为：从起始点开始计算其每一个子节点的f值，从中选择f值最小的子节点作为搜索的下一点，往复迭代，直到下一子节点为目标点。

### 3.3.3 D\*算法

基于A\*算法，Anthony Stentz在1994年提出了Dynamic A\*算法，也就是D\*算法。D\*算法是一种反向增量式搜索算法，反向即算法从目标点开始向起点逐步搜索；增量式搜索，即算法在搜索过程中会计算每一个节点的距离度量信息H(x)，在动态环境中若出现障碍物无法继续沿预先路径搜索，算法会根据原先已经得到的每个点的距离度量信息在当前状态点进行路径再规划，无需从目标点进行重新规划。

其中，距离度量信息H(x)=H(y)+C(y, x)，H(y)代表x点到目标点的距离度量，C(y，x)代表y点到x点的距离度量，在算法中均可用两点间实际距离代替。

### 3.3.4 LPA\*算法

搜索起始点为所设起点（正向搜索），按照Key值的大小作为搜索前进的原则，迭代到目标点为下一搜索点时完成规划；Key值中包含启发式函数h项作为启发原则来影响搜索方向；处于动态环境时，LPA\*可以适应环境中障碍物的变化而无需重新计算整个环境，方法是在当前搜索期间二次利用先前搜索得到的g值，以便重新规划路径。

其中，Key[]为一个二维数组：



图 3-4 key结构

g(n)代表起点到当前点的距离度量。rhs(n)为min(g(n’)+c(n’, n))，n’为n的父节点，h(n, goal)为启发项；搜索原则为：优先判断k1大小，若k1小则优先遍历，若k1=k2，则选择k2较小的点。

### 3.3.5 D\* lite算法

D\* lite算法是Koenig S和Likhachev M基于LPA\_star 算法基础上提出的路径规划算法。D\* lite与LPA\*的主要区别在于搜索方向的不同，这就将Key[]定义中涉及到的目标点goal替换为起始点start的相应信息。

D\* lite算法是先在给定的地图集中逆向搜索并找到一条最优路径。在其接近目标点的过程中，通过在局部范围的搜索去应对动态障碍点的出现。增量式算法的优势在于，各个点的路径搜索已经完成，在遇到障碍点无法继续按照原路径进行逼近时，通过增量搜索的数据再利用直接在受阻碍的当前位置重新规划出一条最优路径，然后继续前进。

## 3.4 A\*算法优化

h(n) 对A星算法的影响如下：

* h(n) 等于0，那么意味着 f(n) = g(n) , A星算法变成了Dijkstra算法；
* 如果h(n) 小于 实际 cost(n) 代价，A星算法可以找到最短路径，但是搜索效率略低，h(n)越小，意味着搜索节点越多，效率上越低，但是精度上越准确，越接近最优解，因为趋近于Dijkstra算法。（cost(n) 表示实际从n节点到目标节点的代价 ）；
* h(n) 等于 cost(n)，是A星最优状态。
* h(n) 远远大于 g(n) ,那么 f(n)的值就主要取决于 h(n),A星就演变成了BFS算法。

在A星算法的优化中，动态衡量启发式：f(n) = g(n) + w(n) \* h(n) , 其中 w(n) >= 1 。与传统的A星不同的是，原本的h(n) ，变成了 w(n) \* h(n) , w(n) 可以影响评估值。在搜索过程中，我们可以通过改变w(n) 来影响搜索过程如上面提到的 h(n)对A星算法的影响。可以推理出，w(n) 越大，越趋近于BFS算法，而 w(n) 相对越小，则相对于趋近于Dijkstra算法。

除此外，还有以下优化方法：

* 简化搜索空间：传统的A星算法是效率并不高的算法，常常会因为搜索空间的太大，而影响搜索效率；
* 分级寻径：把搜索过程拆分开了，如查找空间A中的p1点到空间B中的p2点最短路径，那么可以分为两部分，先查找p1点到空间B的路径，再搜索到p2的路径，整个过程分为了两步，甚至是将计算一次的消耗，拆分成了两次，计算压力也变小了；
* 优化Open表：传统的Open表，因为语言的不同，可能选择的是：Array、List、Queue等等结构来存储。而在A星搜索过程中，随着搜索深度越深，意味着可能要查找的节点越多，后期Open表中存储的节点越点。在数据结构的选择上，我们希望所选择的节点带有“可参照性”，而不是单单的直接从Open表中取。如果Open表在放入数据之前，我们就让它按照一定顺序，那么我们从Open表中取数据时，这些数据就是具备顺序的，是有“可参照性的”。可以选择优先队列的方式。

# 第4章 基于联邦学习的路径优化

智慧物流是指通过智能软硬件、物联网、大数据等智慧化技术手段，实现物流各环节精细化、动态化、可视化管理，提高物流系统智能化分析决策和自动化操作执行能力，提升物流运作效率的现代化物流模式。最后一公里指从物流分拣中心到客户手中这一段距离，通过运输工具，将货物送至客户手中的过程。为最后一公里配送决策提供理论支撑，降低配送产生的综合成本，提升快递员工作效率和客户满意度。

## 4.1 实现框架

基于联邦学习的路径优化模型的底层模型可采用A\*算法与神经网络优化相结合，上层采用联邦学习框架搭建，使用Fedavg（联邦平均算法）与联邦学习持续优化神经网络。

该框架中，启发式函数中的g(n)作为衡量已知路径的费用，通过输入历史信息、个性化信息、道路网络、起点终点和出发时间等信息，采用注意力机制并结合历史轨迹，生成状态信息，然后h(n)作为衡量未来的费用，使用道路网络探测当前点信息、距离信息、终点信息并生成MLP预测估计成本，最后综合二者输出推荐轨迹。

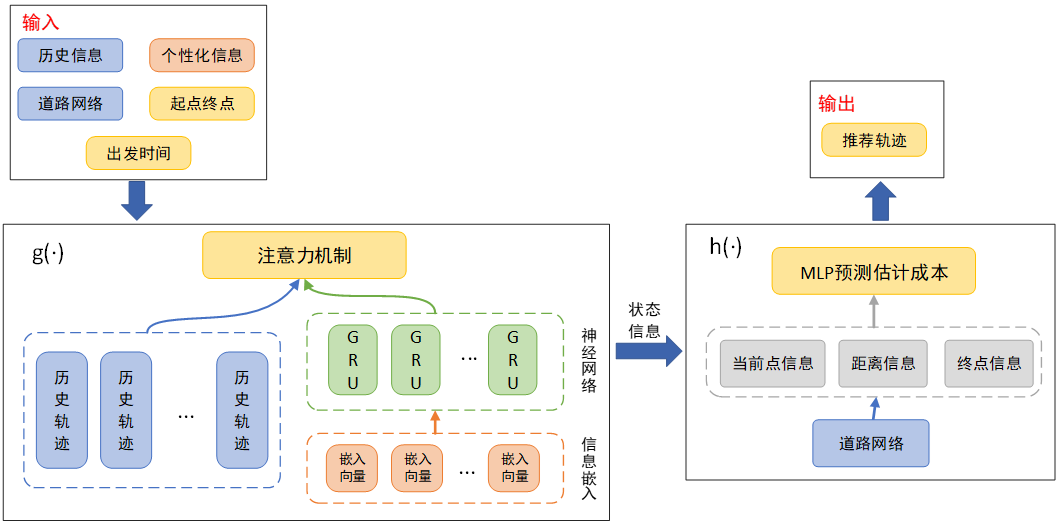


图 4-1 实现框架

## 4.2 注意力机制

注意力机制（Attention Mechanism）源于对人类视觉的研究。在认知科学中，由于信息处理的瓶颈，人类会选择性地关注所有信息的一部分，同时忽略其他可见的信息。上述机制通常被称为注意力机制。人类视网膜不同的部位具有不同程度的信息处理能力，即敏锐度（Acuity），只有视网膜中央凹部位具有最强的敏锐度。为了合理利用有限的视觉信息处理资源，人类需要选择视觉区域中的特定部分，然后集中关注它。例如，人们在阅读时，通常只有少量要被读取的词会被关注和处理。综上，注意力机制主要有两个方面

* 决定需要关注输入的哪部分。
* 分配有限的信息处理资源给重要的部分。

注意力机制（attention）的基本思想就是想让系统学会注意力——能够忽略无关信息而关注重点信息。注意力是一种机制，或者方法论，并没有严格的数学定义。比如，传统的局部图像特征提取、显著性检测、滑动窗口方法等都可以看作一种注意力机制。在神经网络中，注意力模块通常是一个额外的神经网络，能够硬性选择输入的某些部分，或者给输入的不同部分分配不同的权重。本文的注意力机制主要指代神经网络中的注意力机制。

注意力机制可以分为四类：基于输入项的柔性注意力（Item-wise Soft Attention）、基于输入项的硬性注意力（Item-wise Hard Attention）、基于位置的柔性注意力（Location-wise Soft Attention）、基于位置的硬性注意力（Location-wise Hard Attention）。对于基于项的注意力和基于位置的注意力，它们的输入形式是不同的。基于项的注意力的输入需要是包含明确的项的序列，或者需要额外的预处理步骤来生成包含明确的项的序列（这里的项可以是一个向量、矩阵，甚至一个特征图）。而基于位置的注意力则是针对输入为一个单独的特征图设计的，所有的目标可以通过位置指定。这种思想，进而演化成两种不同类型的注意力，一种是软注意力(soft attention)，另一种则是硬注意力(hard attention)。

软注意力的关键点在于，这种注意力更关注区域或者通道，而且软注意力是确定性的注意力，学习完成后直接可以通过网络生成，最关键的地方是软注意力是可微的，这是一个非常重要的地方。可以微分的注意力就可以通过神经网络算出梯度并且前向传播和后向传播来学习得到注意力的权重。

强注意力与软注意力不同点在于，首先强注意力是更加关注点，也就是图像中的每个点都有可能延伸出注意力，同时强注意力是一个随机的预测过程，更强调动态变化。当然，最关键是硬注意力是一个不可微的注意力，训练过程往往是通过增强学习(reinforcement learning)来完成的。

结束语

在本次“基于联邦学习的路径优化”的课外调研学习中，我加深了对A\*搜索和联邦学习的认识，通过查阅文献资料了解了不同的路径优化算法，并对A\*算法的优化方法有了更深的了解。并结合学姐在课上的展示分享，对基于联邦学习的路径优化框架有了更全面的认识，理解了其中启发式函数在该优化方案中的重要作用。同时，在学习的过程中，我发现并补全了一些自己之前在学习启发式函数时的没有认识到的地方，增强了我对人工智能学习的热情。

参考文献

1. Chen D, Ong C S, Xie L. Learning points and routes to recommend trajectories[C]//Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. 2016: 2227-2232.
2. Ling Yin Wei, Yu Zheng, and Wen Chih Peng. 2012. Constructing popular routes from uncertain trajectories. In SIGKDD. 195–203.
3. Cheng Yang, Maosong Sun,Wayne Xin Zhao, Zhiyuan Liu, and Edward Y. Chang. 2017. A Neural Network Approach to Jointly Modeling Social Networks and Mobile Trajectories. ACM T-IS 35, 4 (2017), 36:1–36:28.
4. Liu Y, James J Q, Kang J, et al. Privacy-preserving Traffic Flow Prediction: A Federated Learning Approach[J]. IEEE Internet of Things Journal, 2020.
5. Li Q, Wen Z, He B. Federated learning systems: Vision, hype and reality for data privacy and protection[J]. arXiv preprint arXiv:1907.09693, 2019.
6. Li X, Huang K, Yang W, et al. On the convergence of fedavg on non-iid data[J]. arXiv preprint arXiv:1907.02189, 2019.