谷歌地图对自动驾驶系统路线规划强化学习

张仪、赵

机械工程系

美国珀杜大学

西拉斐特，印第安纳州47906

摘要

自动驾驶系统是人工智能的重要应用。我们在路径规划系统中实现了DoubleDQN算法，并介绍了一种学习代理与谷歌地图交互的方法。其结果是，谷歌地图提供的路线能耗减少了10%，但比谷歌地图提供的路线多了两倍。通过应用更复杂的神经网络结构、更高分辨率的导航和对谷歌地图数据库的无限访问，可以改进该结果。

介绍

背景知识

强化学习(RL)是一种涉及到马尔可夫决策过程(MDP)的规划算法。我们可以认为它是在模仿人类的决策。我们称人为代理，会遇到在餐馆看到菜单的情况。**我们把餐厅称为环境，把当前的位置和感觉称为状态。**在餐馆里，我们做一个决定，点一顿饭，点餐就是行动。你很可能吃过很多次的同一顿饭比以往任何时候都太辣了。**我们把这个机会称为状态转移概率。**这意味着，即使我们在相同的状态下做出同样的动作，结果也可能不同于之前的（辛辣的）（正常的味道）。饭后，我们会根据口味或用餐体验对这顿饭或餐厅进行评论。**我们说品味或体验作为奖励。**奖励可以是积极的，这意味着我们真的很享受食物或服务。当我们在菜单上选择几家餐厅或用餐时，这种餐饮体验会影响我们下次做出的决定。这将逐渐形成或改变我们做出决定的例行程序，我们将其描述为政策

整个过程可以在图中进行简化。下面几句话，在一个环境下，代理遇到一个状态并根据策略进行操作。环境会把我们带到下一个状态，也会根据行动给我们一个奖励。我们将采取另一个行动，得到另一个奖励，进入下一个状态，等等。我们将学习根据一系列的国家-行动-奖励程序来制定和修改这个政策。

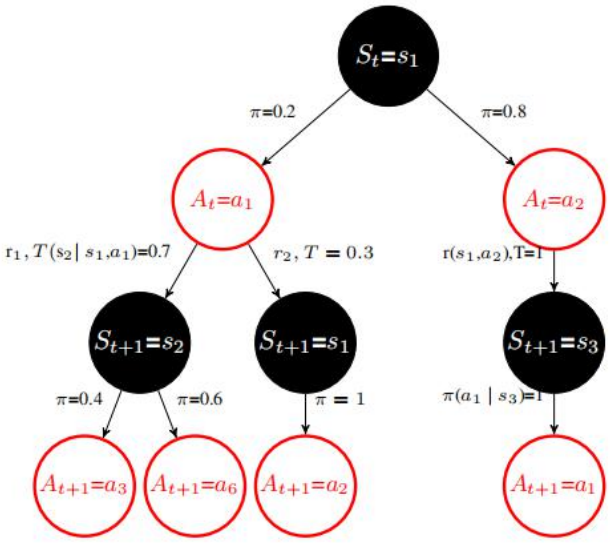

图1：状态、行动和奖励之间的迭代过程。在状态St，根据政策T，我们有0.2个机会选择行动a1和0.8次的机会来选择行动a2.奖励r1和r2可以是不同的。

我们将在图1中使用的一些重要的术语和符号，下面的段落将在这里指定。

.状态集S={s1， s2， s3， }....

.动作设置A={a1， a2， a3， }....

.状态转移概率函数T=T(s\| s,a) = P[St+1=s\| St= s， At= a].这意味着从状态s过渡到状态s的概率\当采取行动。

.奖励函数r=r（=，=）=E[=t+1| St= s， At= a].这意味着给予s和a的奖励的期望值。

.策略T=T(a|)=P[a]t= a | St= s].这意味着选择一个给定的s的概率。

.折扣因子V2[0,1]

学习过程

人类在我们做了一些决定后，从反馈中学习sion和行动。我们希望最终目标的结果尽可能地好。例如，一家五星级的餐厅是

位于山顶上。有坏人和凶猛的动物在我们登顶的路上。我们需要学会一个好的方法来避免它们，并达到顶峰，而不是停下来
在山中的两星级餐厅。

我们的代理人从每个行动的奖励中学习。我们为每个动作分配一个值，并称为这个动作值函数q(s，a)。如果代理处于状态s，并且有4个操作选择，那么我们得到这个状态s的4个q值。q的值最初被设置为0，并且可以通过我们刚刚遇到的奖励进行更新，此外，我们可以考虑到未来的奖励，以防我们可能被当前的奖励所愚弄。这是折扣因子V的功能，它权衡了未来奖励的重要性。如果V接近于1，这意味着未来的奖励几乎和当前的奖励一样重要。我们可以用以下公式来表示当前奖励和未来奖励的组合：

$$q_\pi(s,a) =$$
$$r(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) \sum_{a' \in A} \pi(a'|s') q_\pi(s', a')$$

状态\表示相对于当前状态s的下一个状态。我们不确定是哪个行动\将在代理真正处于状态之前被选择吗\，所以我们需要取q的期望T(s\，a\)，这就是总和和T的作用所在。同样地，我们也不确定是哪个状态的s\代理在状态s下采取动作a后是否会进入，所以我们需要T(s\|，a)来表示可能性，并考虑到所有的机会。在大多数状态中的代理步骤并尝试每个状态中的大部分操作之后，我们可以构造一个指令映射来演示在给定特定状态时执行特定操作的质量。最后，代理可以在每个状态下选择最有可能使我们得到最优结果的最高行动值，我们将其称为贪婪策略。

我们可以通过非策略或策略上更新q(s，a)，其中更新算法将在下一节中指定。这两种策略的区别在于我们如何更新q(s，a)。对于非策略，我们使用maxq(s\，a\）和r（s、a）来更新q（s、a）。对于“政策的，我们使用q(s\，a\）和r（s、a）来更新q（s、a）。

值函数vT(s)是一个函数，通过考虑到该状态下的所有q值来对一个特定的状态进行分级，如下式所示：

vT(s) =对a2AT (a|s)qT(s，a)

论文调查

具有双重性的深度强化学习

Q学习

在这项工作中，作者将神经网络应用于强化学习，并将其命名为Double DQN(VanHasselt，Guez，和Silver2016)。神经网络的功能是将状态s映射到q（s，•）。状态s的输入可以是一个n维向量，如图像。输出是一个m维向量，其中每个元素都可以被解释为q值对应于每个动作。简而言之，神经网络是一个从红外线映射的函数n至IRmθ.我们用来表示神经网络中的参数。

他们建立一个目标或目标来计算计算的q值。θ目标是由另一个被称为目标网络的神经网络生成的θ\它包含了与受它所控制的神经网络相同的体系结构。θ现在我们得到了两个结构相同的神经网络θθ\分别地θ\最初是从。中复制的。θ在一个集的培训中（代理从开始到结束），我们将复制到θθ\每一个N步。换句话说，我们不更新θ\在这些N个步骤中。焦油方程
得到是Yt= rt+1+ Vq (St+1，argmaxq (St+1， a;θt)，).θ′t
a

我们首先输入St+1θ进入网络，选择动作a，动作a对应于输出向量中用argmaxq(S表示的最高值t+1， a;θt).与此同时.

我们输入St+1进入θ\网络，并得到另一个输出向量。我们计算目标Yt通过结合当前的奖励rt+1，其中的q值来自θ\网络q(St+1θ′t， a；).王义宇2016年的论文提供的学习算法见算法1。

算法1：双DQN算法

输入：D空回放缓冲区e初始网络参数e\-电子邮件的副本r-重放缓冲区的最大大小；Nb-训练批量大小；N个目标网络更新频率

为(第e2集{1,2集，...，M}){

初始化帧序列x()；

为(t2{0,1，...}){

设置状态sx，采样动作aT∼B;

下一帧x的样本t从环境e给定（s、a）和接收奖励r，并附加xt到x；

如果|x|>Nf然后删除最老的帧xt分从x端；

集\x，并添加转换元组(s、a、r、s\)到D，替换最古老的元组，如果是|D|N≥r;

取样一小批Nb元组（s、a、r、s\∼) Unif (D);

构造目标值，每个N对应一个b元组：定义

amax(s\;e) = argmaxa\q(s\,a\;e)

yj= { r 如果s是终端
r+ Wq (s\, amax(s\;e); e\) 否则

,

做梯度下降步骤与损失|yj−q(s, a; e)|2；替换目标

参数e\e每个N

体验回放是一种受生物启发的技术，以消除数据序列中的相关性。我们随机选择均匀存储在重放缓冲区中的数据来计算学习过程中的损失和更新权重。

价值迭代网络

值迭代网络(VIN)是一种无模型的规划算法(Tamar等。2016).我们可以使用VIN与标准的反向传播和RL算法来处理需要的视觉感知、连续控制和基于自然语言的决策的问题。目标是学习一个策略端到端，它将推广到解决不同的、看不见的领域。

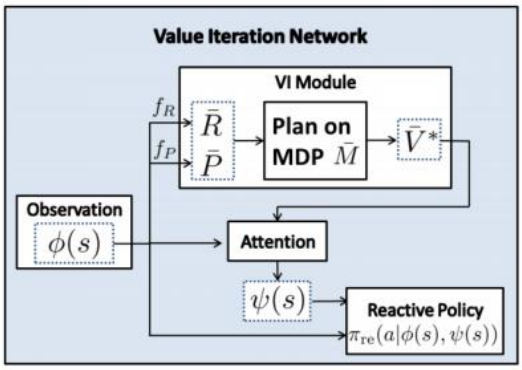

图2：价值迭代网络来自：Tamar，Aviv等。“价值迭代网络。”神经信息处理系统的研究进展。2016

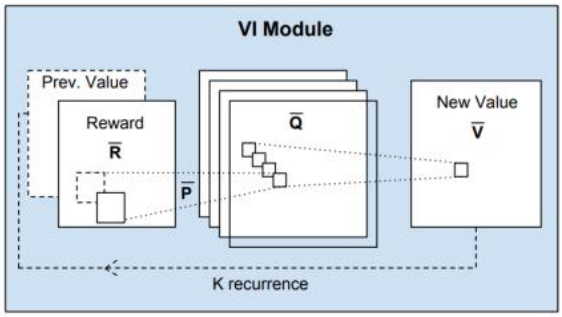

图3：值迭代模块来自：Tamar，Aviv等。“价值迭代网络。”神经信息处理系统的研究进展。2016

ϕ在图2中，输入端是图像（如地形图像）和当前状态。输出$T_{re}\phi$(一个|(s)，ψ(s))是一个动作之上的概率向量。f_R基本上是一个卷积神经网络(CNN)，它将输入的图像转换为一个奖励图像（每个像素都可以代表一个奖励值）。$\bar{R}$fP是一个状态转换函数吗$\bar{P}^{\setminus}$|s，a).$\bar{V}^*$是一个具有相同大小的值函数。$\bar{R}\bar{V}^*$**由于状态s上的最优策略只能依赖于附近的状态，它是状态的一个子集，作者使用注意来表示这个逻辑，并输出一个向量ψ(s)，它表示我们真正关心的状态集的值。**

在图3中，它解释了VIN，VI模块的核心，一个实现方程V(s)=的逻辑的机制

$\max_a$(R(s，a) + V 对$_{s\setminus 2S}$P ($s^{\setminus}$|s，a)V ($s^{\setminus}$)).每次迭代$\bar{R}$的VI模块可以看作是将奖励图像和之前的值函数传递到一个卷积层和最大池化层中，然后输出一个新的值函数$\bar{V}$.卷积层中的每个通道对应于一个特定动作Q(s，a的Q函数1),Q(s,a2),...Q (s,a$_m$)，其中m是动作集的长度。卷积核权值对应于贴现转移概率$^{\setminus}$|s，a).然后，这个层是沿着行动通道的最大池，以产生值函数的下一次迭代$\bar{V}$ $\bar{V}_{i,j}$= $\max_a Q^{-}$(a，i，j).$\bar{V}\bar{R}$然后，我们将它们附加到第二个通道，并将它们输入卷积层和最大池层K次，以执行K次迭代，其中K是将奖励信息从目标传递到状态s的最小值。经过K次迭代后，VI模块将输出给代理来做出决策。$\bar{V}^*$然后我们可以应用DQN或
在图2和图3中训练参数的其他RL方法。

使用样式和结构对抗网络（S2-GAN）的生成图像建模

生成对抗网络(GAN)包含两个模型：生成器G和鉴别器D。生成器试图生成看起来像真实图像的图像，鉴别器试图区分真实图像和生成的图像。

结构（场景的几何形状）和风格（纹理和照明）是图像形成的两个关键成分，而被最近的生成模型忽略了。这篇论文（王和Gupta2016)将生成过程分解为两个过程：(i)使用Ctypuge-GAN生成曲面法线图，(ii)Style-GAN将曲面法线图作为输入，生成二维图像。这两个gan被独立训练，并通过联合学习合并在一起。

第一个生成网络的输入是z^(从均匀分布中采样的100d向量)，然后生成表面法线图(G(z^)：72723)。××第一鉴别器网络将图像（72723）作为输入并输出单个值[0,1]，该值告诉表面法线映射是真实（接近于1）或生成（接近于0）。××

$\bar{z}$第二个生成网络的输入为(从均匀分布中采样的100d向量)和地面真实曲面法线，然后网络生成图像(G(Ci,$\bar{z}$i))，具有纹理和照明度。输入到第二个鉴别器网络的是地面真实曲面法线映射，G(Ci,$\bar{z}$i)、真实图像和真实图像的表面法线图。本文还包括了全卷积网络(FCN)，它取G(Ci,$\bar{z}$i)作为输入，并估计它的曲面法线映射，以便使第二个生成网络输出更好的图像与生成的曲面法线映射对齐。

S联合学习[2]在独立地训练了结构化GAN和Style-GAN之后，我们将一起训练这两个网络，如图4所示，但首先我们删除了FCN部分。首先，我们输入z^，得到生成的表面法线G(z^)，并通过将G(z^)输入到结构-gan中的鉴别器网络中来接收第一个损失。$\bar{z}$其次，我们将G(z^)输入到StyleGAN的生成器网络中，得到生成的图像(G(G(z^)，))。$\bar{z}\bar{z}$我们现在通过将G(G(z^)，)和G(z^)输入Style-GAN中的鉴别器网络来获得第二次损失。我们将把第一个损失和第二个损失合并起来(按0计算。1)在结构-gan中训练发电机网络，以产生更好的表面法线图。

我们可以应用这种技术，通过输入电机输出、电池输出、环境温度等参数来生成车辆的热图。如果热图足够真实，我们就可以设计出一个更好的冷却管理系统，并最小化热传感器的数量。该自动驾驶系统可以根据车辆的状况规划出更好的路线。

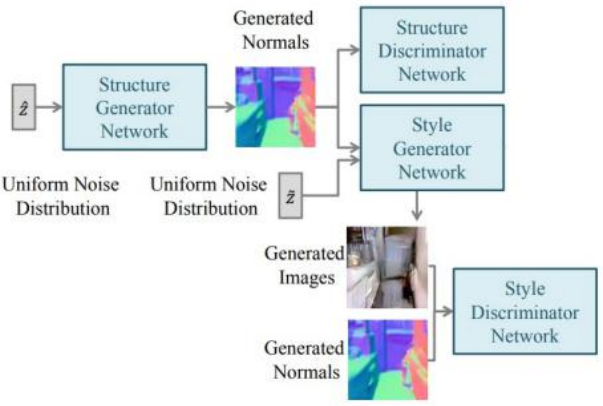

图4：S[2]甘的模特来自：王、小龙和古普塔。“生成的
使用风格和结构的对抗性网络的图像建模。”欧洲计算机视觉会议。施普林格国际出版公司，2016年。

实验

在本节中，我们将首先介绍我们通过张量流库在实验中实现的Double-DQN算法（算法1），然后解释我们如何为学习代理设置环境。最后，我们演示了实验的结果。

学习代理

我们的学习代理是一辆电动汽车，通过选择不同的行动（北、东、南、西）在谷歌地图环境中导航。该动作可以由Double-DQN或随机操作来决定。在学习过程中，代理将首先在地图上随机导航来探索地图，但我们会逐渐减少随机选择动作的部分，但采用Double-DQN模型提供的q值最高的动作。V是0.9和N_b是32。

神经网络架构第一层是一个输入层，它接受当前位置的地理编码，并除以180。输入层之后是两个完全连接的层（分别为10个神经元和6个神经元），其relu激活函数和退出率为0.25。最后一层是一个四维输出，表示每个动作的q值。我们初始化两个相同的网络，第一个是q网络，代理决定它在当前状态下的作用。第二个是目标网络，它作为q网络实现的目标。我们只在q网络中以学习率为0.0001进行反向传播和更新，然后每五步将q网络中的权值复制到目标网络中。

环境

为了了解我们的代理能否与谷歌地图API(谷歌地图API，地理编码API、方向API和高程API)提供的路线相比，在Double DQN算法下找到具有最小能量成本和可接受持续时间的最佳路线，我们设置的实验如下。

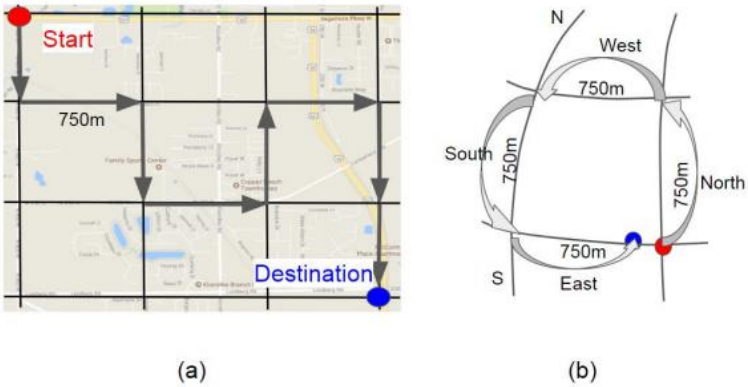

图5：代理所导航的环境的(a)网格贴图。每个箭头的长度和方向表示代理所采取的步骤和动作的一定位移长度。(b)在现实世界中，如果代理采取一个圆形的步骤序列，它将不会回到相同的位置。这是由球体的几何形状引起的。

我们首先指定起始位置（可以是地名、地址或地理编码）和目的地位置，并输入到地理编码API，以检索这两个位置的地理编码。然后，我们使用开始地理代码和目的地地理代码作为两个相对的角来构建网格地图的矩形边界，其中我们的代理只能在上面导航。网格图和符号如图5(a).所示严格地说，网格图中的每个网格都不是一个矩形。这种现象是由球体的几何形状和我们对步幅长度的限制造成的，如图5(b).所示

代理商有四个方向选择（北、东、南、西）。每个箭头表示从当前位置导航到下一个位置的代理），在网格图上的位移为750米。但是，代理的实际导航距离将大于或等于750米，这取决于方向API提供的路线。例如，在图6中，假设代理处于用A表示的当前位置，并向南进入下一个用B表示的位置。显然，由方向API提供的路线是395号公路，距离超过750米。我们可以得到导航指令列表的形式：{地理编码，持续时间从A到1，距离A到1，地理代码1}，{地理代码1，持续时间从1到2，距离1到2,2}，地理编码，1、2、3、B在图6所示后我们输入的地理编码和地理编码B的方向API。....num
指令是基于指令API的
是图6中从A到B的四个指令。我们使用导航指令列表中的每个地理代码从高程API得到每个位置的高度和计算每个指令内的海拔如海拔（图7中位置1）和1（图7中位置2），1和2之间的海拔，2和3之间的海拔，3和B之间的海拔。在每个指令中，我们忽略了道路中间两个位置之间的高度，以简化复杂性

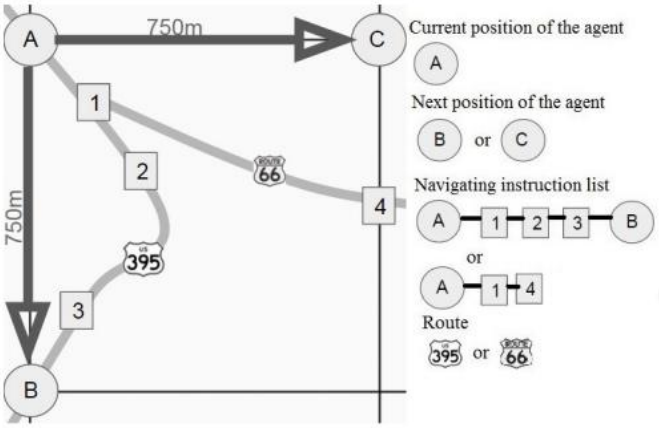

图6：该图只显示了网格地图的一部分。代理只能从一个点导航到另一个附近的点，而不越过边界。如果代理从当前位置A移动到下一个位置B，我们将分析道路信息，如道路395。

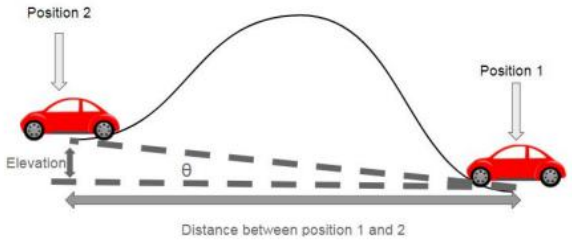

图7：我们只计算了位置1和位置2之间的高程，而忽略了途中的高程

如图7所示。但是，代理从A旅行到C的情况将取决于方向API返回的信息。如果C在一个湖中或某个无法到达的地方，方向API将返回false，代理将把点C视为一个块。否则，位置C被认为是可到达的，由方向API返回的导航指令列表是路线66，然后我们将分析位置A、位置1和位置4之间的高程，即使位置4与位置C并不相同。我们允许方向API返回的路由位于网格地图边界之外，而代理应该始终在网格地图边界内的点上导航。

我们根据网格地图上的750米位移（实际距离超过750米）移动所需的能量来评估每个动作。例如，为了计算图6中位置A和位置1之间所需的能量，我们使用位置A和位置1之间的持续时间和距离来计算平均速度V。θ结合V与高程，我们可以得到道路的角度，并考虑道路的高度为线性递增或递减，如图7所示。因为我们在实验中没有考虑再生制动，所以我们把下坡路处理平坦。在图8中，我们演示了我们如何计算所需的功率(加西亚-瓦勒和Lopes

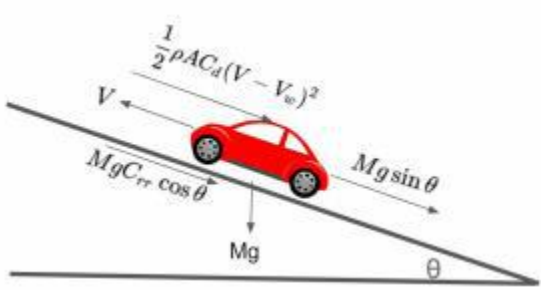

图8：施加在车辆上的力的自由体图

θ质量为M(kg)的汽车在速度V（m/s）下以角度（度）行驶，如下所示

$$P = f_m M \alpha V + MgC_{rr} V \cos() + \rho AC\theta \frac{1}{2} d(V - VW)^2 + MgV罪()\theta$$

其中，P(W)为功率，f_m是质量因子，M(kg)是总体质量，g（m/s^2)是重力的加速度，C_{rr}为轮胎与路面之间的滚动阻力系数，ρ为空气密度(kg/m^3)，A (m^2)是车辆的正面区域，Cd是空气阻力系数和V吗W是风速。我们不考虑加速的早期阶段，所以α是零。这些参数如表1所示。能量消耗应该通过将功率P乘以持续时间来计算。

表1：功率计算参数

f_m	1.05
α	0 m/s^2
质量	2000 kg
C_{rr}	0.02
ρ	1.225 kg/m^3
A	2 $m^2$0
Cd	.5
VW	0 m/s

奖励安排定义奖励的基本概念是基于从当前位置到下一个位置的一步步的能量消耗，例如，如图6所示，从A到B。能量是由前一节提供的方法计算出来的。然后我们把能量除以10000，再乘以-1。为了最小化训练期间的总步数，我们添加了-0。如果下一个位置可到达，则每个过渡都为1。换句话说，采取任何可到达步骤的奖励r将是r=-0。1-（能源消耗量/100000）。如果下一个位置无法到达，如湖泊或河流，r=-1和代理保持在相同的当前位置，并采取其他行动。如果下一个位置和目的地位置的距离小于预定的位移的长度（图6中为750米），采取此行动的奖励r将成为r=+1-（从当前位置到下一个位置的能量消耗/10000）-（从下一个位置到目的地位置的能量消耗/10000）。这里注意到+1出现在奖励中，因为这个动作的成功导致代理到达目的地。

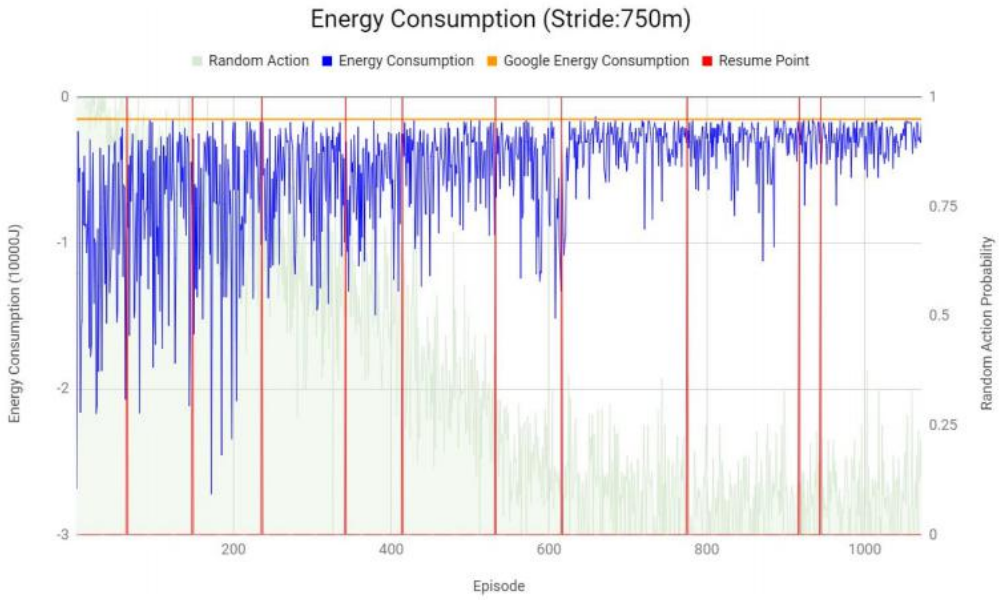

图9：学习主体的能量消耗。蓝线是所消耗的能量，橙色线是走谷歌地图推荐的路线所需的能量。绿线表示随机采取行动的概率，红线表示我们从之前的训练模型中恢复训练

电动车辆电池在实验中，电池的性能不会影响训练过程。该电池能够携带总计5万瓦时的能量，这是电动汽车制造商特斯拉的标准产品。在电化学中，建议使用从一个电池的90%到20%的充电状态(SOC)来提高其使用寿命，我们也实现了这一点。~SOC是由电流能量与总能量的比值计算出来的。我们不会把电池的退化纳入实验。进一步的工作可以考虑到电池性能的真实因素，作为训练过程的一部分。在这个实验中，我们只演示了在理想条件下需要消耗多少能量和充电多少次。

结果和讨论

代理正在从开始位置开始接受培训(地理代码：40。4682572，-86.9803475)和目的地(geocode：40.445283，-86.948429)，步长为750米。一集内超过64个步骤将被视为失败。注意到在训练过程中，谷歌地图api经常阻塞我们的服务器，我们被迫结束训练过程，从中断的事件中恢复模型（红线）。这个问题将导致空的重放缓冲区，我们在学习过程中均匀随机选择样本来计算损失和更新权重。因此，我们需要恢复模型，并开始随机选择动作来重新填充重放缓冲区，并逐渐减少随机动作的部分。

图9中的蓝线显示了该药剂所消耗的能量。代理能够找到一种在600集后减少能量消耗的方法。前600集的振荡是由高度随机的ac-值（绿线）和Q网络提供的不准确的Q值。当不准确Q值的损失最小时，振荡减轻，能量消耗变得更少和稳定。在随机作用下，agent所能达到的最小能量为1327(J)，相应的时间为659秒，其中谷歌地图提供的路线的能量和时间分别为1489(J)和315秒。

结论

Double-DQN能够从奖励和经验中学习。将步幅减少到20米或50米可以提高准确性，也可能会减少能量，但仍需要更多的实验来确认。注意到您能获得的精度越高，您需要的计算资源和时间就越多。源代码：https://github.com/Dungyichao

参考文献

加西亚-瓦莱和洛佩斯。A.P.2012.*电动汽车融入现代电力网络*。施普林格科学和商业媒体。

塔玛尔，一个。；吴，Y.；托马斯，g.；莱文，美国；和Abbeel，P。2016.数值迭代网络。*在神经信息处理系统的研究进展中，2146-2154*。

范哈塞尔特，H.；Guez， A.；和银的， D.2016.基于双q学习的深度强化学习。在AAAI，2094-2100.

王十和古普塔。2016.使用风格和结构的对抗性网络的生成图像建模。*在欧洲计算机视觉会议上，318-335*。蹦跳的人