课程总结

胡晟 2015201971

在这个学期的学习过程中，我们对人工智能有了较为的了解和认识。并且对神经网络和深度学习这两个之前只听过，从未接触过的两门技术有了一定的了解。

我们最初始的计划是实现小车的自动避障的功能。我们最开始利用安置在小车最前方的超声波传感器来获取前方障碍物的距离。并利用传感器得到的数据来在离障碍物一定距离的时候控制小车进行右转，实现了简单的自动壁障。

之后。我们进而让我们的小车变得更加智能化。

首先，我们利用Arfunio板产生的每秒前进方向的数据。然后再将这些数据做初步的处理。记录下每一次转向之间的时间间隔和转向的时间。并将其存储在某个文件里，并不断地重写以保证数据的事实性。然后利用这些数据做出实时的小车运行轨迹。

然后，我们还将手机安装到了小车上，利用手机摄像头的功能，记录小车行进路上的周遭情况。之后我们又由此想到了，利用手机拍摄的画面来实时的影响小车的移动。我们利用手机拍摄软件，并将画面传输到电脑上。再在电脑上不断地截图、重写来保存下手机拍摄的画面。（我们本想直接获得图片形式的内容，但没有找到相关软件）。然后再利用图片识别方面的代码，来处理手机拍摄到的画面。这样我们就基本实现了小车自动应对周遭环境的功能。

我们在百度图片上搜集了8种动物约100张图，并对他们进行标记。我们首先将这些图片灰度化，再用CNN算法将这些图片训练，最终，我们实现了对这8种动物的图像识别，再结合之前的代码，让我们的小车在面对不同的动物的时候，可以做出不同的反应，同时，还把超声波传感器用手机替代，并收集更多的信息。我们最终利用这8种图片，将小车赋予不同的运行方式。并由此，将我们的地图绘制功能也做出了相应的更新，我们因为是利用图片转向，为了小车即时性，我们利用电脑收集到的手机拍摄画面，直接在电脑上识别图片，利用图片来绘制地图，再用蓝牙将转向信息发送给小车，实现了基于周遭环境的小车控制与实时地图绘制。

除了小车的这些功能，我还对其他的相关算法做了一些调查和学习。

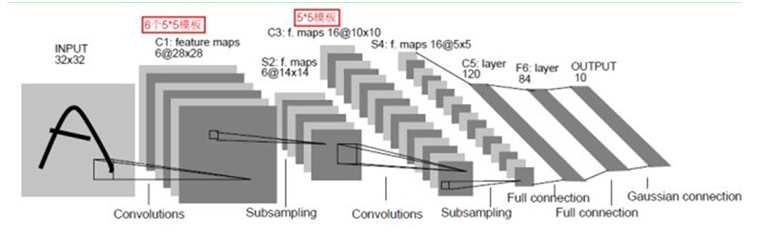
有关深度学习：

深度学习是机器学习中一种基于对数据进行表征学习的方法。深度学习是机器学习研究中的一个新领域，其动机在于建立、模拟人脑进行分析学习的神经网络，它模仿人脑的机制来解释数据。深度学习让层数较多的多层神经网络可以进行训练。深度学习中最著名的就是CNN，它解决了传统较深的网络参数过多难以训练的问题，结合“局部感受视野”和“权值共享”的概念，减少了参数的数量。具体而言就是在原来的全连接的层前面加入了部分连接的卷积层和降维层。相教育多层神经网络而言，深度学习做的步骤是信号->特征->值，并由网络自己选择。

CNN算法(卷积神经网络)：

一般地，CNN的基本结构包括两层，其一为特征提取层，每个神经元的输入与前一层的局部接受域相连，并提取该局部的特征。一旦该局部特征被提取后，它与其它特征间的位置关系也随之确定下来；其二是特征映射层，网络的每个计算层由多个特征映射组成，每个特征映射是一个平面，平面上所有神经元的权值相等。特征映射结构采用影响函数核小的sigmoid函数作为卷积网络的激活函数，使得特征映射具有位移不变性。此外，由于一个映射面上的神经元共享权值，因而减少了网络自由参数的个数。卷积神经网络中的每一个卷积层都紧跟着一个用来求局部平均与二次提取的计算层，这种特有的两次特征提取结构减小了特征分辨率。

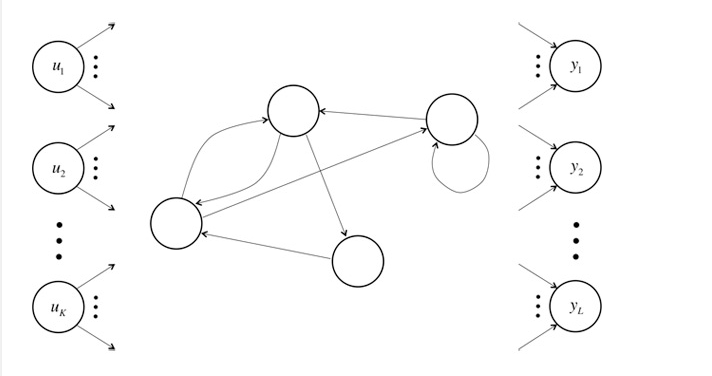
CNN主要用来识别位移、缩放及其他形式扭曲不变性的二维图形。由于CNN的特征检测层通过训练数据进行学习，所以在使用CNN时，避免了显示的特征抽取，而隐式地从训练数据中进行学习；再者由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相对于神经元彼此相连网络的一大优势。卷积神经网络以其局部权值共享的特殊结构在语音识别和图像处理方面有着独特的优越性，其布局更接近于实际的生物神经网络，权值共享降低了网络的复杂性，特别是多维输入向量的图像可以直接输入网络这一特点避免了特征提取和分类过程中数据重建的复杂度。



综合起来说，CNN通过卷积来模拟特征区分，并且通过卷积的权值共享及池化，来降低网络参数的数量级，最后通过传统神经网络完成分类等任务。

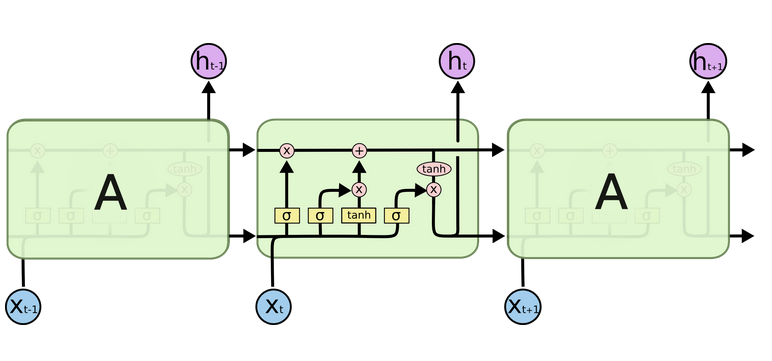
RNN（循环神经网络）：

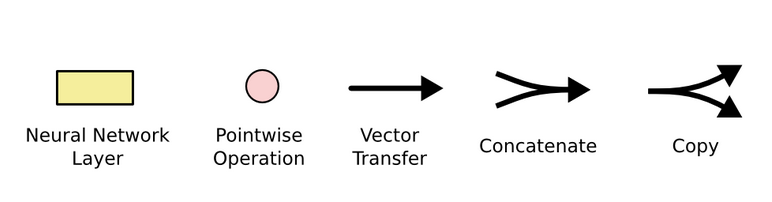
RNN相较于传统的FNN（前向反馈神经网络）RNNs引入了定向循环，能够处理那些输入之间前后关联的问题。



 RNNs的目的使用来处理序列数据。在传统的神经网络模型中，是从输入层到隐含层再到输出层，层与层之间是全连接的，每层之间的节点是无连接的。但是这种普通的神经网络对于很多问题却无能无力。例如，你要预测句子的下一个单词是什么，一般需要用到前面的单词，因为一个句子中前后单词并不是独立的。RNNs之所以称为循环神经网路，即一个序列当前的输出与前面的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。理论上，RNNs能够对任何长度的序列数据进行处理。但大多情况都只和前几个状态有关。我们选择用LSTM来解决长时间倚赖的问题。

RNN在这次的智能小车中，可以用于语音识别。我们做过这样的设想：我们可以利用RNN和LSTM算法来实现语音识别功能。





LSTM中第一步是决定哪些信息需要从单元状态中抛弃。这项决策是由一个称为“遗忘门限层”的sigmoid层决定的。它接收和，然后为单元状态中的每个数字计算一个0到1之间的数字。1表示“完全保留”，而0则表示“完全抛弃”。

我们来回顾一下那个语言模型的例子，试图根据前面所有的词语来预测下一个词。在这种问题中，单元状态可能包含当前主语的性别，所以可以使用正确的代词。当碰到一个新的主语时，我们希望它能够忘记旧主语的性别。

接下来我们需要决定在单元状态中需要存储哪些新信息。这分为两个部分。首先，一个叫做“输入门限层”的sigmoid层决定哪些值需要更新。接下来，一个tanh层创建一个向量，包含新候选值，这些值可以添加到这个状态中。下一步我们将会结合这两者来创建一个状态更新。

在语言模型的例子中，我们希望在单元状态中添加新主语的性别，来替换我们忘记的旧主语性别

现在来更新旧单元状态了，输入到新单元状态。之前的步骤已经决定了需要做哪些事情，我们只需要实现这些事情就行了。

我们在旧状态上乘以，忘记之前决定需要忘记的。然后我们加上，这就是新的候选值，它的规模取决于我们决定每个状态值需要更新多少。

在语言模型的例子中，这里就是我们实际丢弃旧主语性别信息，根据之前步骤添加新信息的地方。

最后，我们需要决定需要输出什么。这个输出将会建立在单元状态的基础上，但是个过滤版本。首先，我们运行一个sigmoid层来决定单元状态中哪些部分需要输出。然后我们将单元状态输入到tanh函数（将值转换成-1到1之间）中，然后乘以输出的sigmoid门限值，所以我们只输出了我们想要输出的那部分。

对于语言模型例子来说，因为它只看到了一个主语，它可能想输出与动词相关的信息，为接下来出现的词做准备。比如，它可能输出主语是单数还是复数，那么我们知道接下来修饰动词的应该成对。

运用LSTM技术，对RNN算法有了极大的提升空间。而且也让语音识别等和上下文有关的深度学习有了比较妥当的解法。