**人工智能课程报告**

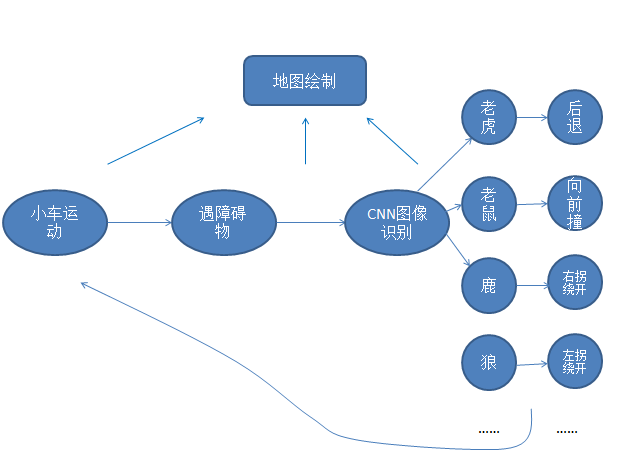
卢淑祺 2015201973

这学期主要学习内容是机器学习，主要工作是组装了能够自动避障同时识别障碍物是什么并对其做出相应反应的人工智能小车。此外，对本课程前半学期的内容，我阅读了周志华的《机器学习》一书，对统计学习的机器学习方式有了一定的了解；对于本课程后半学期的内容，为了加深对深度学习的了解以及加强对tensorflow工具的掌握，我选择了文本自动分类的问题进行研究，同时阅读了大量的论文，建立了对该问题表现良好的循环神经网络模型，有比较高的准确率，同时也加深了对深度学习了解；我们的人工智能小车也应用了cnn图像识别的处理方式，准确率很高，并且在这个过程中我提升了自己组装硬件的能力以及团队合作能力。下面就对关于这门课程这学期的主要工作做一个总结，一是课程大作业——小车实验的主要方法以及主要贡献；二是对课程内容的主要感想与体会；三是课后自己对于本门课程进行的课后阅读以及代码实验。

1. **小车实验**

前半学期我们完成了课程要求的小车蓝牙控制以及自动避障的功能，后半学期我们基于这些功继续优化，实现的功能是：小车在运动过程中遇到障碍物能够进行识别，并根据识别的正确结果做出相应的反应，并在小车的运动过程中使用多线程绘图的方式同时动态展示小车运动的路径，此处我们识别的主要是动物图片。

**实验流程如下**



**原理：**

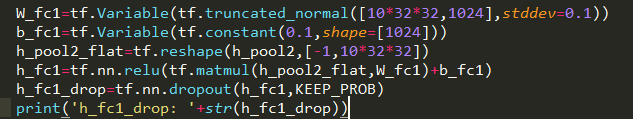
首先先对uno板烧录设计好的程序，我们通过蓝牙控制小车的前行，如果小车蓝牙接收到’r’，则代表右拐信息；手机绑在小车上，通过Android wifi手机摄像头securet liveStream将拍摄到的图片传递到电脑上调用我们训练好的cnn图像识别模型，得出识别结果，小车在运动途中一直以这种方式识别周围环境，在电脑上我们得到了输出结果，然后根据我们设计好的效果，将相应的运动信号通过蓝牙发送给小车，小车作出反应；电脑在判断要给小车发送哪些运动信号的同时通过python多线程将小车的这些运动信号可视化，同步绘制地图，还原小车路径，并将途中识别的动物同步在地图上标注出来。

**主要贡献：**

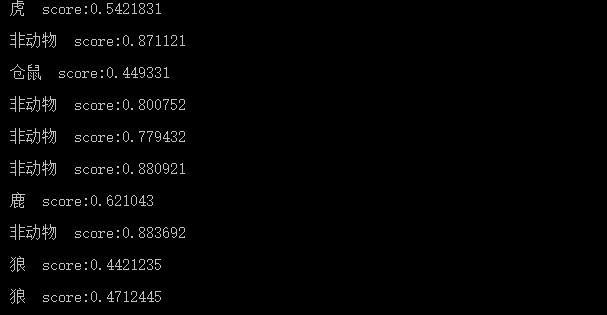
1. 在前期参与小车的组装工作
2. 参与小车uno模块程序调试
3. 主要完成与小车蓝牙模块交互部分代码设计
4. 主要完成地图绘制部分代码
5. 主要参与讨论与完成图像识别算法
6. 参与整个实验流程的设计、实现与完善

**主要算法**：

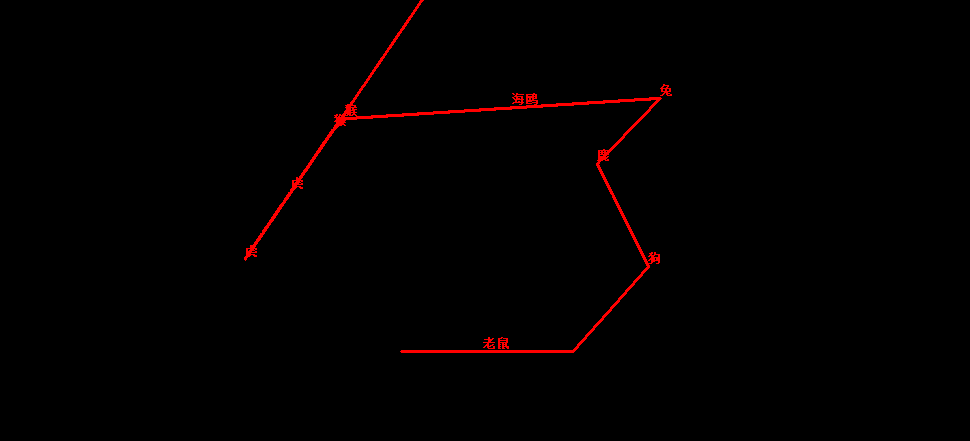
因为在这几年来，卷积神经网络在图像识别方面有了重大突破，课程内容中也有不少篇幅提到了CNN，所以我们考虑使用tensorflow实现卷积神经网络算法。我们先是在网上下载了动物图片然后进行标注作为原始数据集，通过五折验证的方法进行训练。我们将识别动物的问题转化为图像分类的问题，一共分辨八种动物，模拟野外动物出没的场景。参考在minst数据集上的手写数字识别过程，首先我们对图像进行灰度处理，然后我们将分类标注转化为一个8维的向量，属于第几类相应维度为1，其余为0；为模型搭建了三个卷积池化层，并设置卷积层权重和偏置变量，以relu作为卷积后的激活函数再通过池化层，最后将卷积池化层输出摊平通过两个全连接层，输出为一个8维的向量，使用softmax函数作为损失函数，并根据其创建优化器函数；然后不断迭代训练模型，取准确度最高的模型输出。最后模型效果不错，能够基本识别八种动物，准确率达到80%以上，卷积层代码片段如下所示：



**模型的测试情况如下**：



**地图绘制情况如下：**



1. **课程感想与收获**

因为《机器学习》这本书与课程前半部分内容比较契合，所以看了一下，这部分就结合这本书的内容以及课后实验的代码一起来谈谈课程体会。

首先关于模型的训练，我体会比较深的就是过拟合问题，过拟合问题就是模型在训练数据集上面表现非常好但是到了训练集上面表现就很差。在实验中这个问题的产生一般是由于当训练轮次达到一定数量的时候，这个时候训练集上面的loss值依然在降低，准确率依然在升高，但是验证集上的loss值在不断升高而且准确率也开始下降，这就是相当于专门针对训练集训练了一个模型，没有学习出整个数据集上的特征集而是学习了训练集上的特征。针对这个问题我在实验中用五折交叉验证和dropout方法解决。第二个就是模型的评估，针对不同的模型正确的模型评估方式是非常重要的，比如二分类问题，如果验证集中有80%的正例，而模型给出的判断是100%正例，那么准确率也有80%，但这是非常不准确的，换一个数据集如果只有10%的正例准确率就会立即下降，在实验中对于二分类问题计算召回率和查准率。

关于SVM，SVM是一个效果不错的机器学习方式，尤其是在做实验尝试新的机器学习方式时，SVM一般会被作为一个基础模型与新的模型进行效果比对，但是SVM对于大型的数据集效率会比较低，我在课后也尝试了了一下SVM模型，当时对核函数的用处不太理解，看了一些资料后发现由于SVM的原理是找到一个超平面将数据集分开，但是这样的超平面在原始的样本空间中不一定存在，所以在SVM中常常需要将数据映射到更高维的空间，这就需要使用核函数使映射到高维空间后仍然保持数据的结构，而核函数的主要优点就在于虽然也是把特征进行从低维到高维的转换，但是是事先在低维上进行计算，然后再将实质上的分类效果表现在了高维上。然后在《机器学习》一书中其实还提到了降维方面的核函数思想——主成成分分析与核化线性降维，关键也是找到分割数据的超平面，感觉这部分内容有点难理解。

另外一个比较重要的机器学习门类就是贝叶斯分类，贝叶斯分类器是基于贝叶斯决策论的机器学习模型。对分类任务来说，在所有相关概率都已知的情形下，贝叶斯决策论考虑如何基于这些概率和误判损失来选择最优的类别标记。假设一个东西有n种可能的标记，如果我们已知将某个类别的样本误分为另一个特定类别的损失值，那么根据后验概率就可以得到将这些样本分为这个特定的类别的期望损失，根据贝叶斯判定准则，需要最小化总体风险也就是最小化在每个类别上的损失，就是在每个样本上选择那个能使条件风险最小的类别标记，那么就得到了贝叶斯最优分类器。在最小化风险的过程中，需要获得后验概率。估计后验概率的方式一是直接对后验概率直接进行建模训练比如使用决策树、SVM等方式在样本集上构建模型，从而对后验概率进行估计，这属于判别式模型；二是先对样本和类别的联合分布进行建模然后再得到后验概率，这属于生成式模型。一般是使用生成式模型，基于假设——样本的独立同分布假设，设置分布的参数为a，那么就把任务转换成为了参数的训练问题。。但是数据集可能并不具备独立同分布的性质，模型效果可能不好。

而朴素贝叶斯分类器相比一般贝叶斯分类器做出了改进。不再假设样本都是独立同分布并由一个向量参数唯一确定，而是做了“属性条件独立性假设”，也就是说每个属性都独立地对分类结果产生影响。我觉得这有点类似决策树地方式，也是把特征提取出来作用于分类结果，那么训练过程就是基于整个数据集的频率估计类先验概率，然后对每个属性都计算其条件概率，这个条件概率也可以通过数据集上的频率估计，然后通过概率连乘计算分类的风险值取最小化风险的分类即可。

课程最后讲到了深度学习，这部分内容在后面有提到，还有tensorflow工具，课后我也学习了一下，tensorflow的原理是构建计算图，在深度学习中比较好用也是现在非常常用的的工具，包装了很多函数以及神经网络的基本细胞模型，搭建神经网络非常方便。此外，可视化工具tensorboard也很好用，便于观察构造的神经网络模型的结构以及训练过程中的准确率以及loss值。在最后一部分的课后实验中我的代码主要是用tensorflow实现的。

1. **课后阅读与实现**

**阅读**

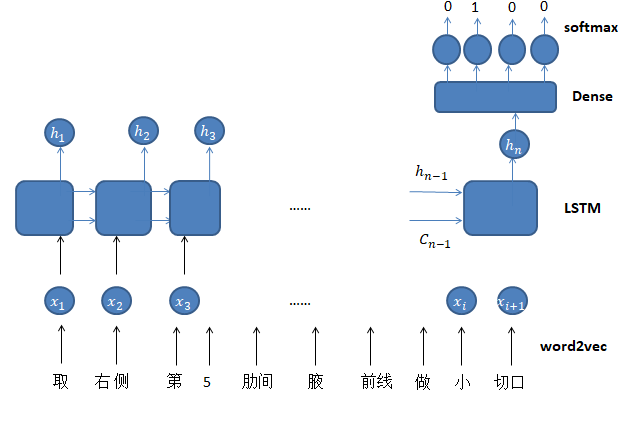
在本学期的前半部分，我在课后阅读了周志华的《机器学习》一书，书中前面部分和课程内容安排比较相似，介绍了模型的基本评估方法、线性模型、决策树模型、SVM、贝叶斯分类器等，关于这些内容的体会与总结上一个部分已经写了，书中其他部分还讲了半监督学习、概率图模型、规则学习以及强化学习等等。

**代码实验**

在本学期的后半部分，学习了tensorflow与卷积神经网络与循环神经网络之后，我将它运用到了实验室学习中，以lstm神经网络为基础构建了文本分类模型，尝试了多种优化，包括用attention机制进行优化，有不错的准确。下面简单介绍一下这项工作。

工作的主要目的是为了提取一段手术报告做出的切口数目，因为文本中的切口最多只有三个，我通过构建神经网络模型对手术报告进行文本分类，分为三类，从而判断文本中所描述的手术切口数目。我的神经网络模型的基础是lstm。LSTM是RNN的一种特殊类型，解决了传统的RNN不能有效保留长期记忆的问题，在自然语言处理等序列化输入的情况中有较为广泛的应用。相比于标准的RNN神经网络中重复神经网络模块只具有一个简单tanh层的结构，LSTM也具有这样的重复神经网络模块链式结构，但是重复模块中有四个交互的层的结构。这四个层分别是LSTM中的遗忘门、输入门以及输出门和tanh层。LSTM通过三个门的结构控制和更新记忆的单元也就是cell的状态，并把RNN的输入层和隐藏层移入记忆单元。通过这四个层的交互，LSTM实现了保留长期记忆的功能。

在这个文本分类的实验中采用jieba分词系统先对文本进行分词，在此分词基础上，采用word2vec模型中的CBOW模型和Negative Sampling解法，以分词后的文本作为输入，训练词向量模型。Word2vec模型在词向量训练中的应用也十分广泛，它能够保证单词转换为词向量之后在向量空间中还能够维持词语之间的语义相似度，也就是语义越相近的词在向量空间中距离也越近。并且解决了词向量维数维度爆炸的问题。本文将文本单词映射到向量空间后，将此文本在向量空间中的矩阵作为输入，训练LSTM模型。模型结构如下图：



**整个实验的原理是：**

首先将文本中的词转换为向量后，那么整个文本转换为矩阵：

input =

将矩阵元素逐步输入LSTM层，对每一个t时刻的输入有：

Wf是遗忘门权重，bf是遗忘门偏置，σ是sigmoid函数

在这一步，产生了一个遗忘向量，确定了前面内容中不需要保留的无用信息位置，、、的初始值通过正态分布随机化处理。

相应地，由于当前时刻又有新的向量输入，所以这个时刻，综合前文所传入的所有信息产生了一个蕴含新信息的向量：

但是保留了遗忘层确定要遗忘的信息，也就是说中含有一部分无用信息，所以还必须有一个记忆向量确定需要保留的信息位置，否则遗忘向量的设置就非常多余了。

遗忘向量与记忆向量同时作用使得我们既能够通过前面的经验积累判断哪些是无用信息，又能够根据后文的输入保留一部分与后文联系紧密的“无用”信息。所以综合的信息向量如下，并传递给下一个时刻：

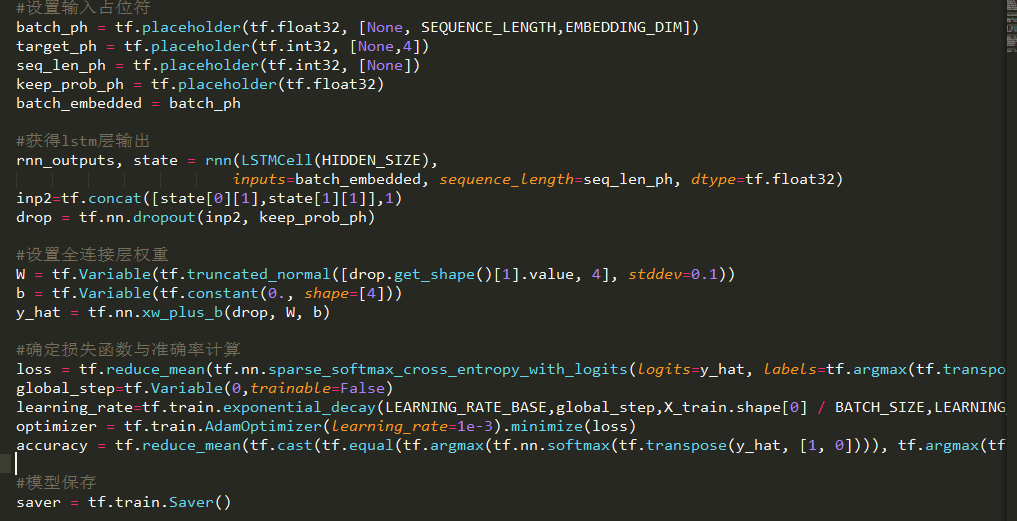
但是当前的输出必须确定输出当前信息的哪些部分，这通过上一个时刻的输出和当前时刻的

输入来确定，因为下一个时刻的输出往往于上一个时刻联系紧密：

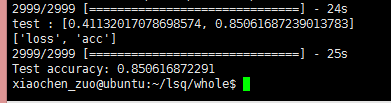
最后当得到最后一个时刻的输出，将其输入全连接层进行降维处理为了获得分类结果再进行softmax处理：

为了提高处理效率，在实验过程中采用批量处理文本的方式，batc\_size大小设置为32，将lstm隐含层大小设置为128，为了防止过拟合现象设置0.5的dropout概率，最后LSTM层的输出通过四维的全连接层，以softmax作为激活函数，输出得到最后的分类结果。

实际上代码实现通过tensorflow实现，模型构建的代码片段如下：

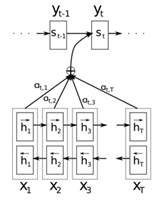


实验效果准确率能够达到85%，（在服务器上面跑的）还有优化的空间，实验截图如：



**进一步研究优化**

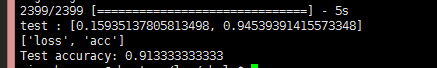
此外我看了**Bahdanau D, Cho K, Bengio Y, Neural machine translation by jointly to align and translate. Eprint Arxiv:1409.0473.2014**这篇论文，讲的是在利用神经网络模型进行机器翻译，里面提到了一个attention优化机制，就是在循环神经网络中，rnn的输出不再只取输入序列最后一个时步的输出，而是将每个时步的输出都保留，认为它们对最后的输出都有贡献只不过每个输出贡献的权重不一样，再设置一个神经网络对这个权重向量的分配进行训练，原理图如下：



在机器翻译模型中，预测下一个词只是根据上下文以及神经元的输出计算词出现的概率。上下文向量蕴含了到这个词为止所有的输入信息，在传统的神经网络模型中，上下文向量就是神经网络最后一个时刻的输出，在引入attention机制之后，计算公式变为

就是需要训练的权重向量，也就是说最终得到的向量不再是简单的神经网络 最后一个时刻的输出，而是每个输入时刻的输出的线性加和，权重代表着模型对这个部分的关注度。

根据这两个公式可以求出最后的 。我基于这个理论构造注意力机制，将 lstm 层的所有时刻输出作为 attention 层的 输入 ，对于函数f，初始化了一个矩阵操作和状态向量，然后利用神经网络反向传播梯度下降的办法不断训练矩阵参数与。根据训练出的矩阵、输入的 与状态向量，计算输出的权重向量，将权重向量与的乘积作为 attention 层的输出,之后输入四维的全连接层，以 softmax 作为激活函数，得出分类结果，实验结果如下所示，准确率升高到90%，表现良好。



1. **总结**

通过这门课的学习我对机器学习有了更深的了解，希望以后能够在这个基础上继续不断研究探索，学习到更多的东西。