**2018.9.8-2018.9.21汇报内容**

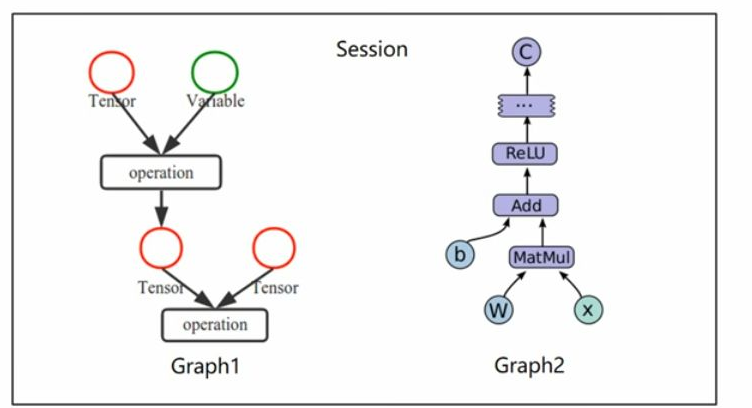
# 1 Tensorflow实战google深度学习框架

## 1.1Tensorflow入门

从8月份一直到9月中旬，一直在看深度学习理论方面的书籍，因此之后便计划刷一遍实战书籍并实现代码加深理解。以下便是我的主要学习内容。本主题代码，我已经上传到Github：<https://github.com/BigBigRadish/tf_cookbook>。它首先要通过编程，构建一个计算图出来，然后启用一个会话来把数据作为输入，通过这个图规定的计算步骤计算，最后得到结果。 普通的命令式编程容易理解和调试，命令语句基本没有优化，按原有逻辑执行。tensorflow这种符号式编程有较多的嵌入和优化，不容易理解和调试，但运行速度会有一定的提升。

### 1.1.1计算模型-计算图

计算图是Tensorflow的一个基本概念，他的所有计算都会被转化为计算图上的节点。Tensor是Tensorflow中一个很重要的概念，它定义了计算的规则而不保存计算的数据，是构建计算图不可或缺的重要组成部分。Tensor是一个代表计算操作的句柄，它不保存数据，只是提供一个计算数据的方法。使用Tensor主要有两个目的，一个是Tensor可以把不同的操作连接起来，从而去构建一个计算图，支持Tensorflow去进行大规模多步的计算，另一个是当Tensor被一个会话Session启用的时候，就可以计算出该操作对应产出的数据。



* 1. 计算图

注:图片来自DATAGURA

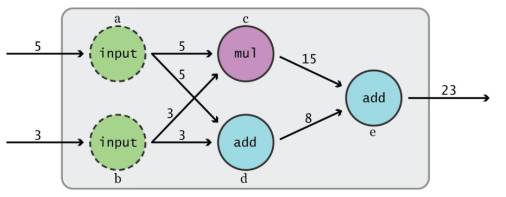


图1-2 计算图实例

### 1.1.2 数据模型-张量

**在tensorflow程序中所有的数据都通过张量的形式来表示。**从功能的角度看，张量可以被理解为多维数组。其中零阶张量表示标量（scalar）也就是一个数；一阶张量为向量，也就是一维数组；n阶张量可以理解为一个n维数组。**但张量的实现并不是直接采用数组的形式，它只是对TensorFlow中运算结果的引用。在张量中并没有保存数字，它保存的是如何得到这些数字的计算过程。** 一个张量主要保存三个属性：**名字（name）、维度（shape）和类型（type）**。

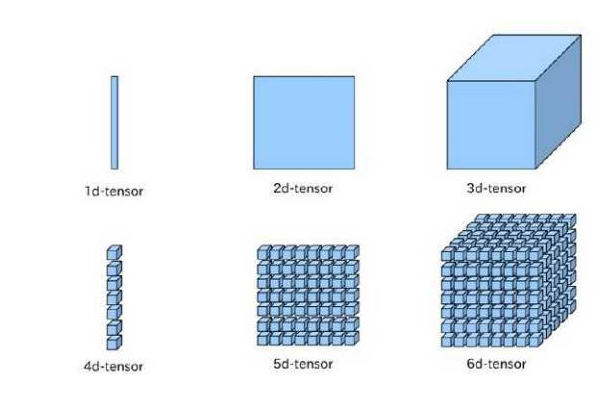


图1-2 张量

### 1.1.3运行模型-会话

Tensorflow中的会话是来执行定义好的运算的。会话拥有并管理Tensorflow程序运行时的所有资源。当计算完成之后需要关闭会话来帮助系统回收资源，否则可能出现资源泄露的问题。TensorFlow中使用会话的模式一般有两种，第一种模式需要明确调用会话生成函数和关闭会话函数。第二种是直接通过上下文管理器进行管理，当上下文管理器退出时，就会自动释放资源。Tensorflow中的会话需要手动指定。

## 1.2案例实战

### 1.2.1 CNN实战案例

Mnist数字识别问题

这是深度学习入门的第一个案例。这个案例，我尝试用深度学习来解决这个问题。用的神经网络是CNN。CNN具有局部平移不变性，相当于按窗口对每个像素点采样特征。然后池化层，选取的最大池化，相当于提取权重最大的特征，全连接用于输出预测维度的向量，最后一层softmax层输出概率最大的那一个分类。神经网络之所以叫表示学习，其一，在传统机器学习中，有特征提取这一步骤，这一步骤经常基于我们认为提取。而在深度学习中，我们称这一课程为表示性学习，由机器基于某一分布由机器自动编码，这样往往能保留输入数据的最大信息量。

在这个案例中我尝试了不同的网络架构，第一种是所有的层次都采用全连接层来构建网络架构，第二种隐藏层采用CNN串联的方式的Resnet网络，第三种尝试并联架构。准确率以此提升。通过这一案例，我学会了搭建深度网络架构并了解了较多TensorFlow Api。

### 1.2.2 RNN实战案例

在自然语言处理领域中，我们的数据经常是非结构化文本的序列数据，而基于序列数据的上下文关系，CNN这种网络结构很难对其上下文的信息进行关联。RNN便是解决这种问题的网络结构，并由此衍生出lstm，gru，以及attention机制，较好的应用于这些文本数据。

这个案例的主要内容是nlp的一个子领域，机器翻译。他的核心思想是基于标注数据，建立翻译文本和被翻译文本的词对齐，被采集每个词儿周围的数据，对其加入关注，然后将这些词通过自编码或者词袋的形式固定到一个维度，由深度学习模型进行驯练。我在看网络博客的时候，有的人又称这叫结构化学习。

本次案例采用的是Seq2Seq网络，基于lstm的双层rnn架构，对数据进行了，清洗，处理然后训练，然后保存模型，最后用保存的模型对数据进行了预测，有较好的翻译效果。

## 1.3 Tensorflow封装

虽然原生态的TF 的api 可以很灵活的支持多种网络结构，但是其代码量相当大，TF的高层封装便是解决这个问题的方法。这里主要学的是Keras，因为他的使用范围较广，并有多人进行维护和更新。

### Keras的使用

Keras API 对模型定义，损失函数，训练过程等进行了封装。他分为数据处理，模型定义，模型训练三个部分。可以使用api先定义一个Sequential类，然后在实例中通过函数添加网络层。Keras把卷积层，池化层，RNN结构，全连接层等常用的神经网络结构做了封装，可以很方便实现神经网络。后面会把keras的详细文档看一看，目前还不是很熟。

## 1.4总结

花了接近半个月，把这本书走了一遍，并实现了书上的例子，对深度学习在实践程度上，有了较多的收获，后面会在此基础上，看一下论文，回顾理论书籍上的知识，那本书书上的知识确实太深了，还需多看。

# 2自然语言处理应用

在建模之前我就有个想法，希望自己参加一次nlp领域的比赛，刚好最近收到一个推送，关于文本摘要的比赛，由此，在进行深度学习的同时，也在看自然语言处理综论那本书。Nlp的应用主要分四个子领域，信息抽取，问答，摘要，机器翻译。这四个领域是互相关联的。信息抽取用于构建结构化或半结构化数据，方便对数据进行分析和管理。问答是目前在市场应用比较火的领域，比如客服机器人。摘要，主要是对文章进行压缩，可用于构建笔记之类的。机器翻译，已经有很多年了，国内国外都很火，目前的效果已经非常不错了。自然语言处理才是al领域的皇冠，目前很多领域的正确率都不是太高，还具有非常长远的研究价值。

## 2.1信息抽取

信息抽取又叫文本分析。他的主要方法包括有限状态机，概率模型，句法组块分析，这些方法都是传统的抽取方法。目前有较多论文·是通过深度学习进行抽取任务，但是还没学到这一块儿。

信息抽取的第一步是NER，命名实体识别，用于抽取特殊的名词。然后对这些名词通过聚类，然后使每一个集合指向其真实实体的意义，这个任务便是指代消解。然后对实体间的关系进行识别和分类，这对应于现实中的事件识别和分类。然后找出这些事件间的时序信息，由此引出时间表达式识别和时序分析问题。时序信息对于上面的几个领域都是有作用，因为上面几个领域大部份建立在序列数据的基础上。对于文本摘要领域，模板填充是一个较好的方法，因为许多文本都聚焦于某一个领域，然后找出适用于这个领域的模板进行填充具有较好的效果。

在初学nlp的时候，很难找到这些知识与实际应用领域找到对应关系。在信息抽取这一部分，我较好的发现了应用领域与研究领域的关系。这对于我后面的学习过程有较好的导向性。

## 2.2总结

目前正在看自然语言处理这本书，老师说过，要少看书，多看论文，我觉得多看书才能更好的理解这些知识，前沿的论文。但是目前从我读过的论文来看，都是在强调应用的新颖性。都是建立在这些基础上，但是研究者都很盲目，目前关键还是找到感兴趣的点，

# 3数学建模

这次数学建模，我选的是一个多目标优化的题目。目标是合理规划每天的飞机的停机位安排，开始有想着用强化学习来做，倒腾了两天，没弄出来。然后实现了网上的一篇论文，关于智能算法的，只能找到局部最优。Github： <https://github.com/BigBigRadish/Huawei_math_model>

## 算法及思路

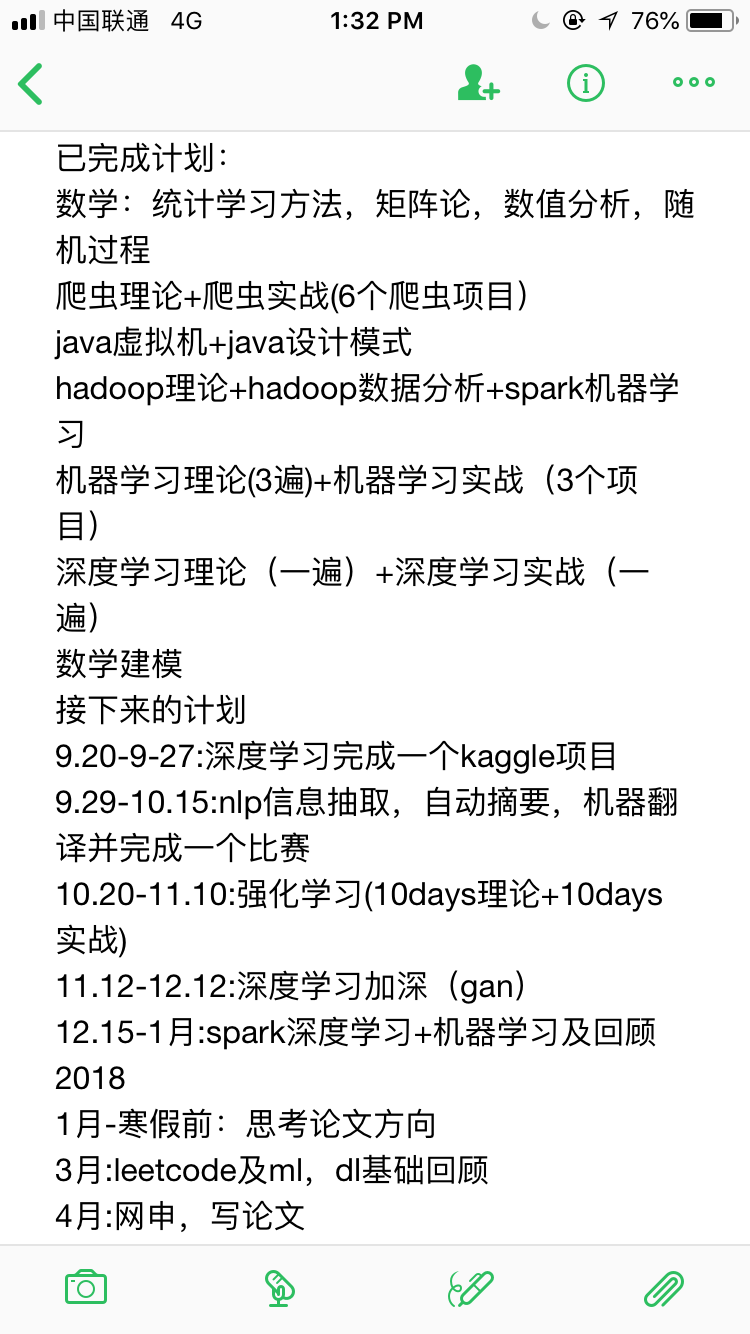
第一步，按先到先服务贪心的寻找一个可行的解，然后寻找50个，用于构建初始种群

第二步，使用遗传算法，构造大量的可行解

第三步，定义适应度函数，就是数学模型，我构造的一个类似数学挖掘中F1-score的类似数学模型。

第四步，通过适应度函数，通过大量迭代，选出较优的个体

# 4学习安排



# 5 附录

[1] DanielJurafsky, JamesH.Martin, 朱夫斯凯, 马丁, 冯志伟, & 孙乐. (2005). *自然语言处理综论*. 电子工业出版社.

[2] Lecun Y, Bengio Y, Hinton G. Deep learning.[J]. Nature, 2015, 521(7553):436.