北京航空航天大学计算机学院

本科生毕业设计（论文）中期报告

论文题目：基于 CUDA 的遗传算法和神经网络在股票趋势问题中的应用研究

学生姓名：王劭阳

学生学号：12061163

专　　业：计算机科学与技术

指导教师：任健

学院(系)：计算机学院

北京航空航天大学计算机学院

2017 年 03 月 28 日

目录

1.课题简介

1.1 课题背景

金融领域内的优化问题一直以来是让众多投资人、学者、企业感兴趣的问题之一。股票趋势问题则是金融优化问题的重要代表。

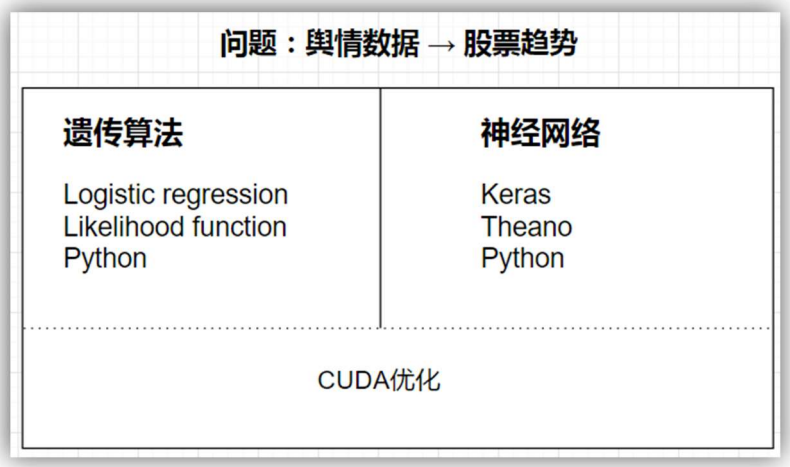
股票的趋势可能与这些事件相关：

- 供求关系：买股票需求上涨导致股票价格上涨；
- 上市公司盈利能力：和股票相关的上市公司业绩增加导致股票价格上涨；
- 大众投资者信心：投资者们倾向于认为股票价格会涨，市场信心足，导致股票价格上涨；
- 周边市场或宏观形势影响：周边国家的股票上涨导致我们的股票随之上涨；
- 未来政策、形势等发展趋势：如果大多数投资者都认为以后一段时间风调雨顺国泰民安、经济持续健康发展，那么股票就会涨；
- 其他投资品种的收益高低：存款、其他投资收益率降低导致股票价格上涨。

本课题旨在通过研究包括但不限于这些事件与股票趋势之间的联系，得出一套通过事件的发生情况来预测股票趋势的模型，投资者可以据此以预测股票市场的动向，并为金融领域内的优化问题提供一种解决方案。

1.2 主要研究内容和目标

本课题计划使用遗传算法和人工神经网络对舆情数据进行分析，得出可以相对准确地预测股票趋势的算法和模型，并使用 CUDA 架构优化算法的性能，最后对比这两种算法的优劣。

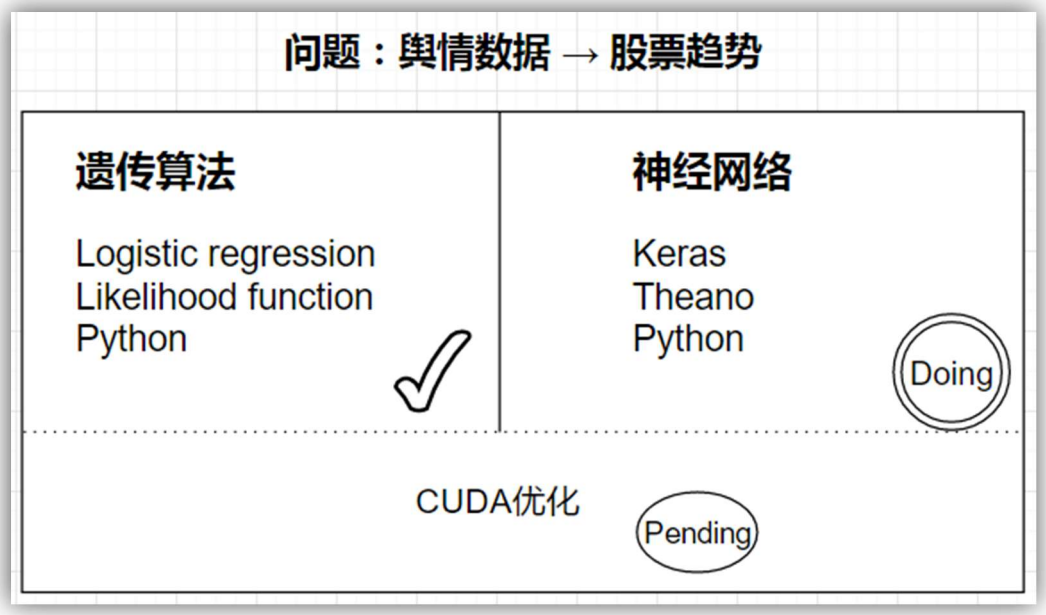


2.论文工作进展情况

按照工作计划，截止至中期答辩，已经完成了大部分的编码工作。包括但不仅限于下面工作：

完成遗传算法部分的设计、编写、调参和模型训练。预测德国 DAX 指数和国际金价成功率均达到 60%左右。

完成神经网络部分的设计、编写。目前预测德国 DAX 指数和国际金价成功率在 52%左右。后续工作将对此模型进一步优化调整。



3.论文工作成果介绍

3.1 问题建模

我们得到的每份数据分为两部分，一部分是 2014~2016 年每日股票的收盘价（价格），一部分是 2014~2016 年每日发生的事件（舆情）。其中每天的舆情数据有四个属性(event_type_id, scope_id, polarity, count)，分别代表：事件类型 id、领域类型 id、极性（积极、中性或消极）、发生次数。

对数据进行预处理。价格数据部分，计算出每日的股票相对于前一天是上涨还是下跌。舆情数据部分，统计所有数据中出现过的所有不同的(event_type_id, scope_id, polarity)，将每一组不同的(event_type_id, scope_id, polarity)给予一个从一开始的正整数 id，称作 feature_id，作为一个特征。这样每天的舆情数据就只剩下两个属性(feature_id, count)，为此舆情的特征 id 和当日发生次数。

设：

m 为样本总数（总天数），n 为特征总数。

对于每日的数据：

$$y^{(j)} = \begin{cases} 1, \ if 第\ j\ 天股票价格上涨或不变 \\ 0, \ if 第\ j\ 天股票价格下跌 \end{cases}$$

$x^{(j)} = (x_0, x_1, x_2, \dots, x_n)$ ，其 x_i 为第 j 天 feature_id = i 的特征发生次数，x_0 = 1。

称一组 $(x^{(j)}, y^{(j)})$ 为一个样本。

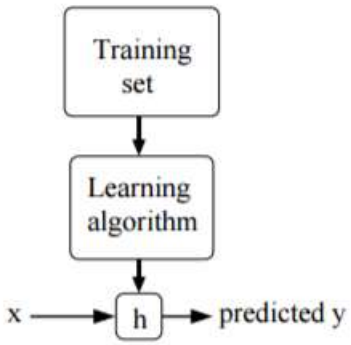
对于全体数据：

$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(m)} \end{pmatrix} \qquad X = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ \dots \\ x^{(m)} \end{pmatrix} = \begin{pmatrix} x_0^{(1)} & \dots & x_n^{(1)} \\ \dots & \dots & \dots \\ x_0^{(m)} & \dots & x_n^{(m)} \end{pmatrix}$$

需要提到的一点是，X 和 y 是从总体 X_{total} 和 y_{total}（从古至今每一天的数据）抽样出的一组样本。

现实中存在一个函数 f，满足$y = f(X)$，并且$y_{total} = f(X_{total})$。

我们的目标是找到一个函数 h，使得 h 与 f 尽量相近，使我们可以通过函数 h 来预测股票的趋势。

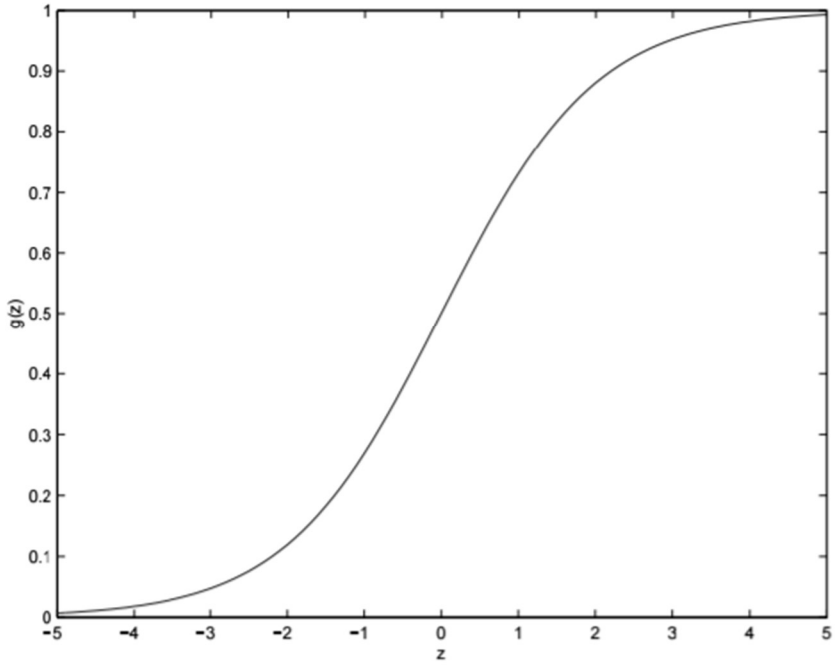


3.2 遗传算法部分的设计

设参数 $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)$ 为一个 n 维向量。

我们选定函数 h 的模型为 $h_\theta(x) = g(\theta^T x) = \frac{1}{1+\ e^{-\theta^T x}}$

其中$g(z) = \frac{1}{1+\ e^{-z}}$ 为 sigmoid 函数，形状如下图所示



由于 x，y 是随机变量，并且$h_\theta(x) \in (0,1)$，所以我们做出如下假设：

$$P(y=1|x,\theta) = h_\theta(x)$$
$$P(y=0|x,\theta) = 1 - h_\theta(x)$$

即是说$P(y|x,\theta)$满足参数为$h_\theta(x)$的伯努利分布。

将两个等式写在一起得到：

$$P(y|x,\theta) = (h_\theta(x))^y(1-h_\theta(x))^{1-y}$$

我们使用极大似然估计的方法来求出θ，假设 m 个样本相互独立：

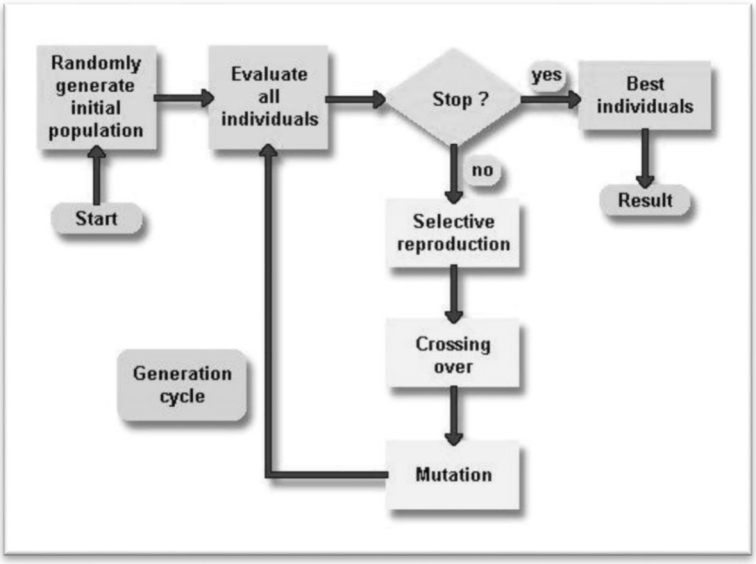
$$\begin{aligned} L(\theta) &= P(y|X,\theta) \\ &= \prod_{j=1}^{m} P\left(y^{(j)}|x^{(j)},\theta\right) \\ &= \prod_{j=1}^{m} (h_\theta(x^{(j)}))^{y^{(j)}}(1-h_\theta(x^{(j)}))^{1-y^{(j)}} \end{aligned}$$

于是θ的值应为使$L(\theta)$取到最大的值。

我们使用遗传算法使$L(\theta)$取到最大值。算法采用实数编码的基因序列，把θ的每个分量当作一个基因，将$L(\theta)$作为个体的适应度函数。为实现方便考虑，设$\theta_i \epsilon [-10000, 10000]$。

遗传算法的流程图如下。首先随机生成一个种群，种群中包括随机生成的初始个体，每个个体有自己的基因序列，然后不断迭代产生新的种群，直到种群中的某个个体适应度足够高。每次迭代产生新种群的过程包括两个步骤：交叉、变异。交叉过程随机选出两个较优秀的原种群个体交叉产生新的个体；变异过程扫

描新种群内的所有个体，按一定概率对其中的基因做突变。



交叉过程的逻辑为：从原种群中随机选出小部分个体，选出这部分中适应度最高的个体 A，再用同样的方式选出另一个个体 B；遍历生成新个体的基因，以 50%的概率从 A 或 B 相同位置的基因序列中选出。

变异过程的逻辑为：遍历新个体的基因，通过生成$[0,1)$的随机数并与设定好的变异率作比较的方式决定每个位置的基因是否要做突变。如果要做突变，那么新的基因值 $\theta_i' = \theta_i + \gamma \cdot d(\gamma,\theta_i) \cdot (1 - r^{(1-\frac{t}{T})^b})$ ，其中γ为随机的 1 或-1，$d(\gamma,\theta_i)$为θ_i到上界或下界的距离即$d(\gamma,\theta_i) = \begin{cases} 10000 - \theta_i \ , \ \gamma = 1 \\ \theta_i - (-10000) \ , \ \gamma = -1 \end{cases}$，t 为当前的代次，T 为总迭代数，b 位设定好的控制学习速度的参数，r 为$[0,1)$的随机数。

精英保留策略：每次生成新一代种群时，上一代适应度最高的个体总是得以原样保留。

3.3 遗传算法部分的实验

分别用德国 DAX 股票指数和国际金价两组数据对算法进行测试。两组数据的样本容量分别为 504 和 517。将样本分为三个部分，300 个样本作训练集，100 个样本作验证集，剩余样本作测试集。

DAX 股指的测试结果如下：

```
Total: 104, Correct: 65, Accuracy: 0.625000
```

金价的测试结果如下：

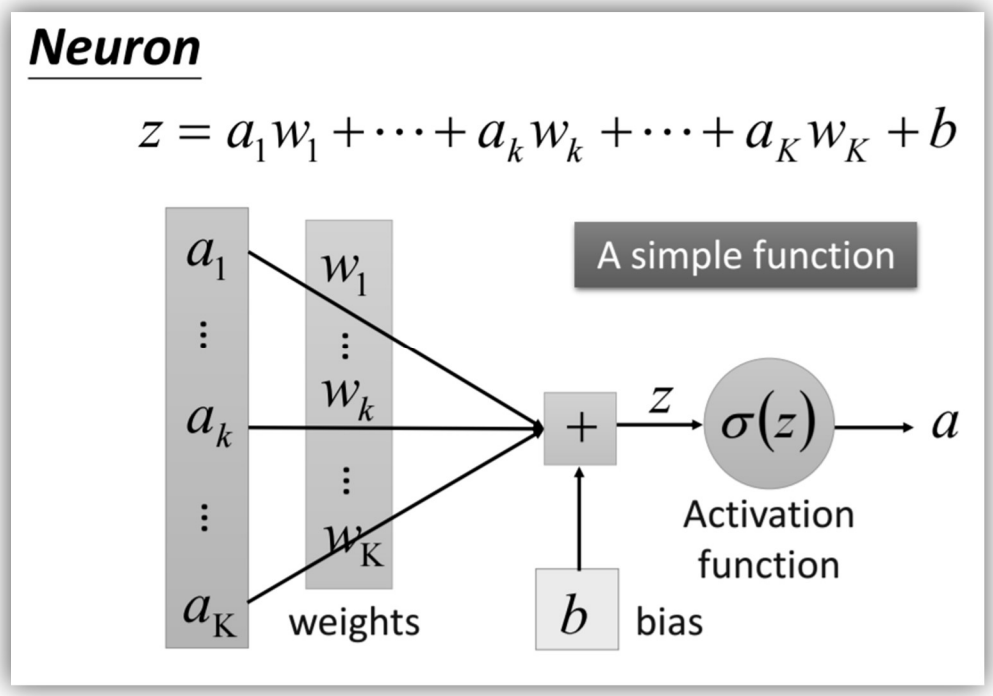
```
Total: 117, Correct: 79, Accuracy: 0.675214
```

3.4 神经网络部分的设计

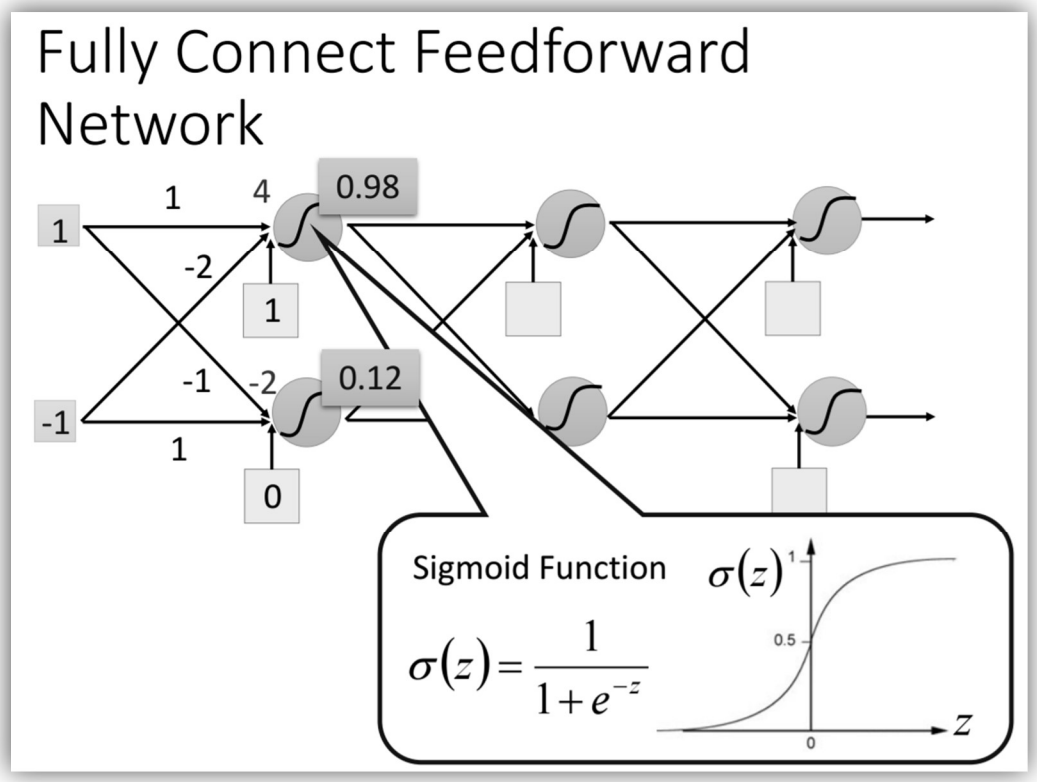
设h ∈ H，神经网络算法找到的函数 h 将会比上几节描述的遗传算法找到的 h'更加复杂。

整个神经网络分为输入层、隐层、输出层。输入层为 x，输出层为 h(x)，把在遗传算法部分描述的模型当作神经网络中的一个神经元，隐层包含 2 层，每层有 8 个神经元，隐层间全连接。

下图是神经元的一个示例：



下图是神经网络的一个示例：



对于每一层而言，设x'和y'是该层的输入输出向量，Θ是该层的参数矩阵，A是矩阵为Θ的线性变换，则$y' = \sigma(A(x') + b) = \sigma(\Theta x' + b)$。可以看出神经网络的每一层对输入向量先进行一个线性变换（升降维、放大缩小、旋转），然后平移，然后再进行一个非线性变化（弯曲），将输入空间投向另一个空间。多层神经网络组合，可以产生复杂的函数模型，使空间 H 很大。

神经网络采用后向传播的方式进行训练。先随机初始化每层的Θ，然后从输出层向前开始计算残差（使用均方误差），使用梯度下降的办法从后向前更新每一个参数。

4.后期进度安排

日期	进度
2017.03.30-2017.04.20	完成神经网络部分的优化
2017.04.20-2017.05.01	使用 CUDA 对代码性能进行优化
2017.05.01-2017.05.20	撰写毕设论文，并根据论文情况补充或调整代码
2017.05.20-之后	准备毕设答辩