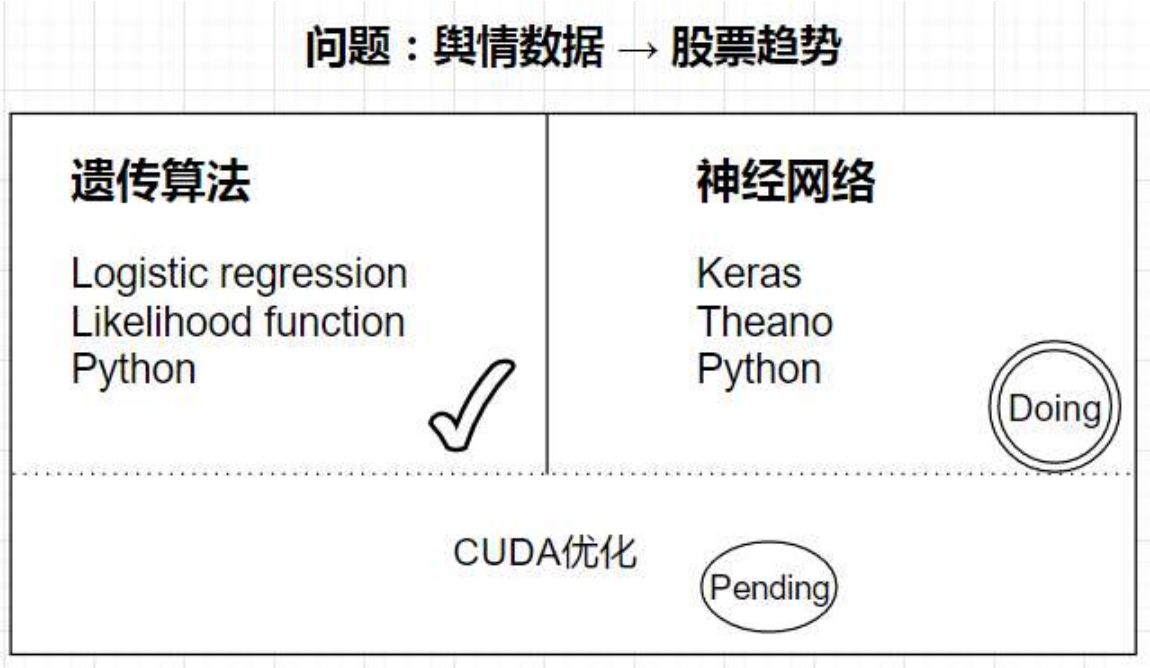
基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究

中期答辩

王劭阳

研究内容和进展

- 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。



成果介绍-问题建模

- 数据
  - 2014~2016
  - 收盘价（价格）
  - 发生的事件（舆情）
    - (event_type_id, scope_id, polarity, count)

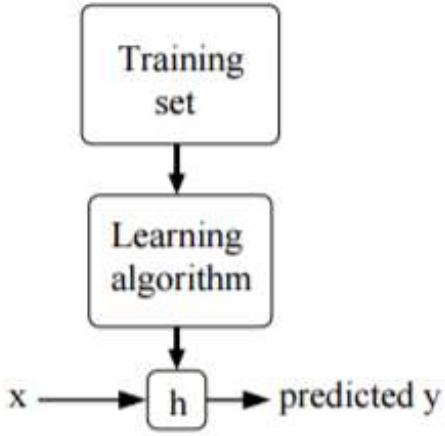
- 预处理
  - 涨跌
  - (event_type_id, scope_id, polarity, count) -> (feature_id, count)

成果介绍-问题建模

- 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。

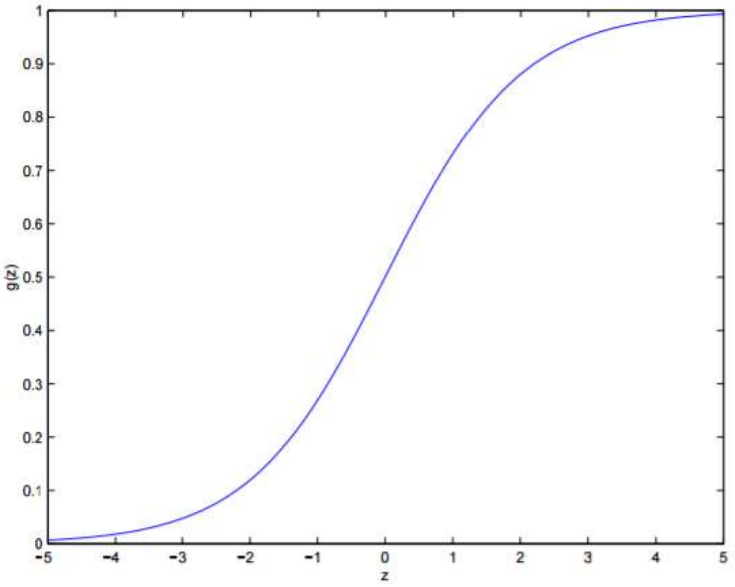
成果介绍-问题建模

- 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。



成果介绍-遗传算法部分的设计

- 设参数 $\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$ 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



成果介绍-遗传算法部分的设计

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

成果介绍-遗传算法部分的设计

- 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：

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

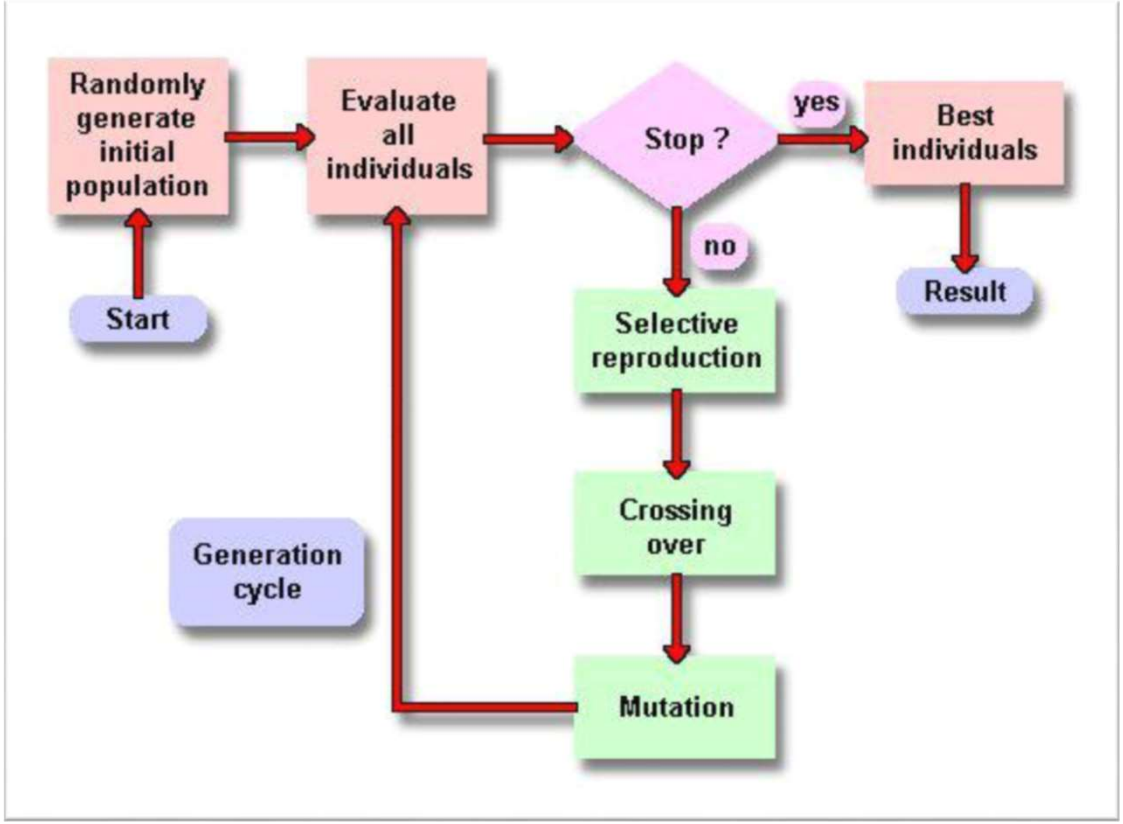
- 于是θ的值应为使$L(\theta)$取到最大的值。

成果介绍-遗传算法部分的设计

遗传算法

- 基因序列实数编码
- 基因 - θ的每个分量
- 适应度函数 - $L(\theta)$
- $\theta_i \epsilon [-10000, 10000]$

成果介绍-遗传算法部分的设计



成果介绍-遗传算法部分的设计

交叉过程

- 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
- 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B
- 遍历生成新个体的基因（50% A or B）

成果介绍-遗传算法部分的设计

变异过程

- 遍历新个体的基因，以设定好的突变率决定该基因是否突变，不突变则continue
- 新基因 $\theta_i' = \theta_i + \gamma \cdot d(\gamma, \theta_i) \cdot (1 - r^{(1-\frac{t}{T})^b})$
- $d(\gamma, \theta_i) = \begin{cases} 10000 - \theta_i\,, & \gamma = 1 \\ \theta_i - (-10000)\,, & \gamma = -1 \end{cases}$

成果介绍-遗传算法部分的设计

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

成果介绍-遗传算法部分的实验

- 样本划分
  - 训练集 – 300
  - 验证集 – 100
  - 测试集 – 100左右

- 德国DAX股票指数

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

- 国际金价

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

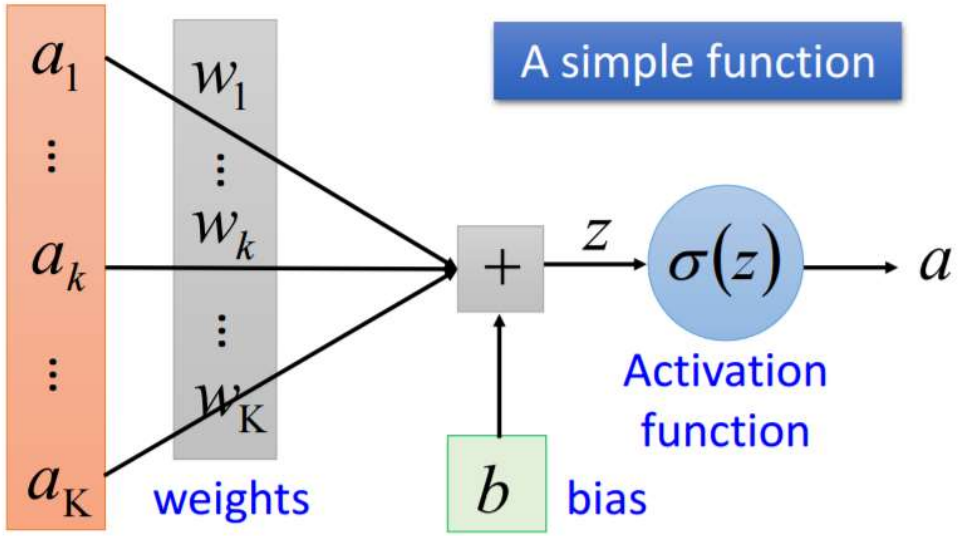
成果介绍-神经网络部分的设计

- $h \in H$
- 输入层 – x
- 隐层
  - 共2层
  - 每层8个神经元
  - 全连接
- 输出层 – h(x)
- 神经元 – 整个遗传算法部分使用的模型

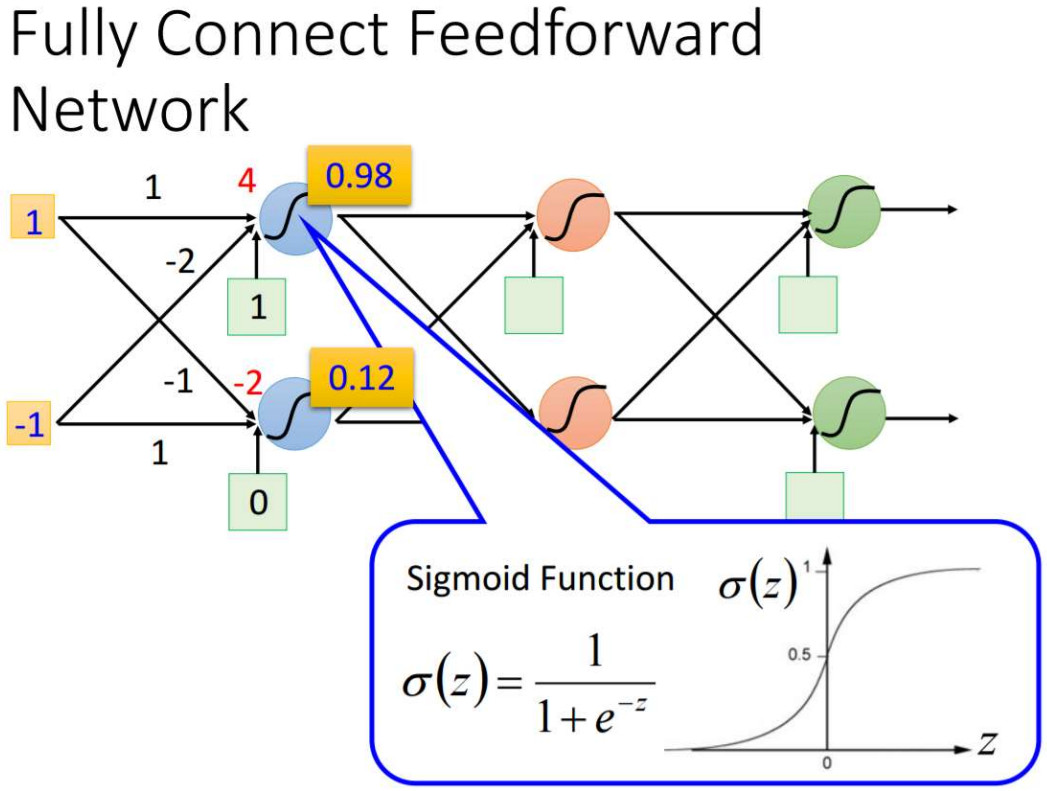
成果介绍-神经网络部分的设计

Neuron

$$z = a_1 w_1 + \cdots + a_k w_k + \cdots + a_K w_K + b$$



成果介绍-神经网络部分的设计



成果介绍-神经网络部分的设计

每层在做什么？

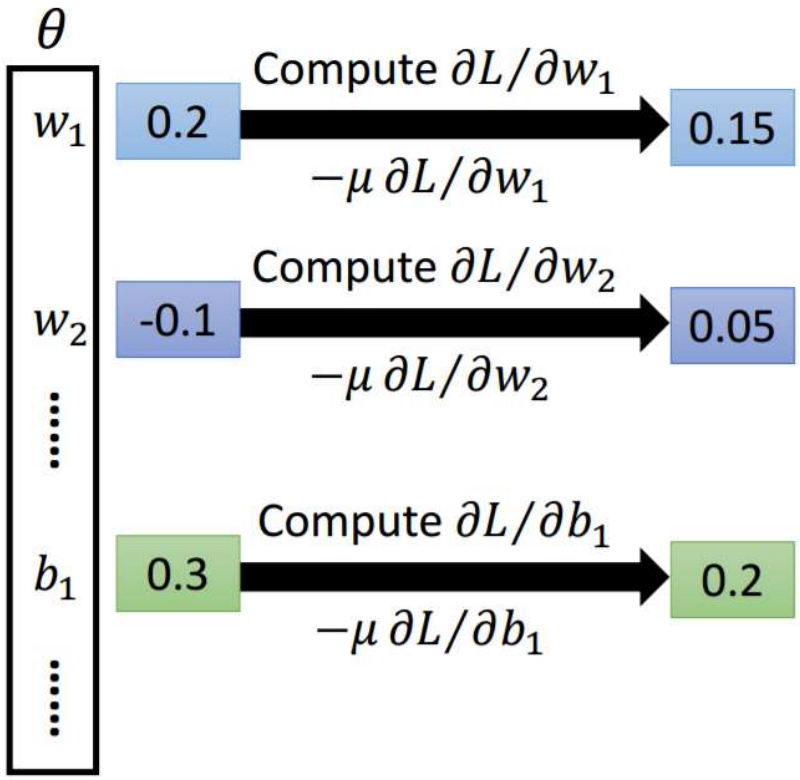
- x'和y'是该层的输入输出向量
- Θ是该层的参数矩阵
- A是矩阵为Θ的线性变换
- $y' = \sigma(A(x') + b) = \sigma(\Theta x' + b)$
  - A – 升降维、放大缩小、旋转
  - $+b$ – 平移
  - σ – 将输入空间投向另一个空间（弯曲）

- 函数模型复杂 -> H很大

成果介绍-神经网络部分的设计

- 后向传播
- 初始化所有Θ
- 从后向前操作
  - 计算残差（使用均方误差）
  - 梯度下降

成果介绍-神经网络部分的设计



$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \vdots \\ \frac{\partial L}{\partial b_1} \\ \vdots \end{bmatrix}$$

gradient

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

谢谢