**2020 级《数据挖掘》**

# 课程结业大论文

## 基于双向 GRU 的循环神经网络对电影评论情感分析


**学院：** 信息科学技术学院

**专业：** 软件工程

**班级：**

**学号：** 2020101642

**姓名：** 陈宇

# 目录

# 第1章　　论文引言

随着各种信息技术、新兴媒体技术如同雨后春笋般地出现，还有数据库技术、计算机网络的迅速发展，以及数据库管理系统的广泛应用，人们在各种生产活动中所积累的数据也越来越多。而大量的数据并不意味着信息，尽管我们现代先进的数据库技术可以让我们存储大量的数据而不废吹灰之力，但是缺乏一种成熟的技术帮助我们分析并且理解这些数据的技术。这种现象的背后是数据的迅速增长与数据分析方法的滞后之间的矛盾。如何处理这些基本无序、不同类型的海量数据，并且如何在这些数据中提取出对生产活动有用的信息便成为了一门学问，同时数据挖掘技术也应运而生。

数据挖掘就是在数据中发现潜在的和有用的信息，它的研究领域十分广阔，主要包括：数据库系统、基于知识的系统、人工智能、机器学习、知识获取、统计学、空间数据库、数据可视化等不同的邻域与学科。这种多领域交叉的特征也使得数据挖掘能够提供的技术也十分多样：分类、估计、预测、关联分析、聚类分析等等。

在大三下学期，我修读了这门数据挖掘课程，通过一个学期的对数据挖掘原理的深入探讨、把信息科学、计算科学和统计科学与数据挖掘技术融合在一起，在一定程度上，培养了我具备初步的数据挖掘的科研能力和创新能力。

在这篇结课论文中，我首先对数据挖掘的一些基本技术进行比较全面的概述；再阐述关于数据挖掘的一些过程模型，其中包括知识发现 KDD 模型和一些跨行业的标准流程；还有多种数据挖掘技术模型的评估方法。最后再结合 RNN 循环神经网络和双向 GRU，通过参考一些相关领域的文献综述，对 Kaggle 提供的电影评论情感分析问题进行创新应用实践。

# 第2章　　数据挖掘基本技术概述

## 2.1 聚类分类技术

### 2.1.1 聚类技术概述

#### 基本概述

聚类分析是数据挖掘所采用的起步技术，也是数据挖掘入门的一项关键技术。主要思想是在没有给定的类别划分下，根据数据的相似程度进行样本的分组的方法，具体来说就是将物理或抽象对象的集合分组为由类似的对象组成的多个分类的分析过程。通过对一组对象进行分组的任务，使得同一组的对象与组中的其它对象在某种意义上更相似。有时也会根据数据内在性质会将数据进行特征分类，每一聚合类中的元素都会在某种条件具有相同或者相似的特征。

值得注意的是，聚类技术与分类技术需要使用有标签样本构成的训练数据不同，聚类技术可以建立在无标签的数据上，也就是说，聚类技术是一种无监督的学习方法。聚类虽然与分类相似，但是与分类目的却大不相同，无监督聚类是针对数据的相似性和差异性将数据分为几个类别。属于同一个类别之间的数据相似性很大但是不同类别的相似性很小，跨类的数据关联度很低。其不同之处还在于聚类技术要求划分的类是未知的，在学习训练之前，没有预先定义好分类的实例，数据实例按照某种相似性度量方法，计算实例之间的相似程度，将最为相似的实例聚类在一个组——簇中，最后再通过对每个簇的解释和理解，从中挖掘数据的意义。

#### 基本作用

1. **在数据中发现具有概念形式的有价值的知识**

   聚类的主要作用，对数据自动分组，将相似的数据归为同一组，不相似的数据归为不同的组。

2. **对有指导的学习模型的性能进行评估**

   可以对建立有指导学习模型的训练数据集进行聚类分析，将原来的输出属性进行删除后，按照有指导模型中的输出属性的可能取值的个数设置初始簇个数，进行无指导聚类分析，检查聚类输出以确定来自有指导概念类的实例是否能够自然地聚类在一起。

3. **选择属性，确定有指导学习的最佳输入属性**

   和第三点作用相同，当聚类输出的聚类效果不够明显时，可以断定用于训练的属性不能用于区分概念类，可以重新选择属性，再应用无指导聚类进行前述的属性评估，直到有监督学习选择出一组较为最优的属性。

4. **探测孤立点**

   无指导聚类还可以用于探测数据中出现的非典型实例，非典型事例又被称为孤立点。无指导聚类通过检查那些不能和其它实例自然聚类在一起的那些实例来识别孤立点。识别孤立点对于某些应用来说，是用来判断特异情况的发生。

## 基本分类

1. **基于划分聚类**

   主要思想: 对具有 n 个数据元组成的数据库，划分方法是构建 k 个划分，其中 k≤n。每个划分必须至少包含一个数据元组，没有数据元组必须只属于一个划分，在模糊划分方法中，数据元组不一定只属于一个划分。

2. **基于层次聚类**

   主要思想: 层次分解来创建数据集。根据层次分解的形成方式，又可以分为凝聚和分裂两种方法。凝聚法将每个数据元组形成单独的组，然后逐次合并到相近的数据元组，直到所有的组合并为一个组为止，或者直到满足一定条件。分裂法开始将所有数据元组置于一个簇中，然后分裂为更小的簇，直到最后每个数据都归属与某个簇中。

3. **基于密度聚类**

   主要思想：采用基于密度的方法，只要“邻域”中的数据元组的个数超过某个阈值，就继续分离或聚类。这样可以过滤离群数据，发相关数据组成任意形状的簇。

4. **基于网格聚类**

   主要思想：基于网格的方法对网格结构中的数据空间进行量化，网格数量有限，对网格结构进行聚类。

5. **基于模型聚类**

   主要思想：通过构造密度函数来定位聚类，利用密度函数来反映数据空间分布。此方法根据标准统计数据自动确定集群数量。考虑离群数据的影响，对每个聚类建立一个模型，对给定的模型进行最优数据拟合，从而产生鲁棒聚类方法。

## 经典聚类技术

# K 均值聚类

K 均值聚类，又称为 K-means 聚类技术，是最基础常用的非监督划分聚类算法。该算法的给定条件设置为一个具有 n 个样本的数据集 X；算法目标是将数据集分为 K 个聚类；其具体的聚类分组要求是每个聚类分组中，所有的数据样本，与该分组的中心点的距离之和最小。

K 均值聚类算法是斯图尔特·劳埃德（Stuart Lloyd）于 1982 年提出简单有效的聚类技术。其基本思想是：

1) 随机选择一个 K 值，用以确定簇的总数。
2) 在数据集中任意选择 K 个实例，将它们作为初始的簇中心。
3) 计算这 K 个簇中心与其它剩余实例的简单欧式距离（Euclidean Distance），用这个距离作为实例之间相似性度量，将与某个簇相似度高的实例划分到该簇中，成为其成员之一。
4) 使用每个簇中的实例来计算该簇新的簇中心。
5) 如果计算得到新的簇中心等于上次迭代的簇中心，终止算法过程。否则，用新的簇中心作为簇中心并重复步骤 3-5.

值得注意的是，在随机选择 k 时，需要判断数据可能包含的类；初始簇中心是随机选择的；相似性度量还有曼哈顿距离、切比雪夫距离、编辑距离等。

$$Distance(A-B)=\sqrt{(x_1-x_2)^2-(y_1-y_2)^2}$$

其中，A、B 为两个对象：$x_1$、$y_1$为对象 A 的属性，$x_2$、$y_2$为对象 B 的属性。通过迭代寻找损

失函数最小，其中，损失函数可以定义为各个样本距离所属簇中心点的误差平方和：

$$J(c,\mu)=\sum_{i=1}^{M}\left\|x_i-\mu_{c_i}\right\|^2$$

**K-means 算法的优点：**

1) 高效可伸缩，计算复杂度接近线性
2) 收敛速度快，原理简单，可解释性强

**K-means 算法的缺点：**

1) 受初始值和异常点影响，聚类结果可能不是全局最优而是局部最优
2) K 是超参数，一般需要按经验选择
3) 样本点只能划分到单一的类中

基于 K-means 算法的优缺点，在后续的聚类算法中也产生出了许多优化版本，比如 K-Means++ 、K-Means Ⅱ算法还有 Mini Batch K-means 算法。

## 层次聚类算法

层次聚类算法（Hierarchical Clustering）主要分为两类：自上而下或是自下而上。

自上而下的算法在一开始就将数据点视为一个单一的聚点，然后依次合并，直到所有类合并为一个包含所有数据点的单一聚类。因此，自下而上的参差聚类称为合成聚类或 HAC。合成聚类的层次结构用一颗树（或树状图）表示。树的根是收集所有样本的唯一聚类，而叶子是只有一个样本的聚类。

**算法原理：**

1) 将每个数据点作为一个单独的聚类进行处理。
2) 在每次迭代中，我们将两个聚类合并为一个。将两个聚类合并为具有最小平均连接的组。比如说根据我们选择的距离度量，这两个聚类之间的距离最小，因此是最相似的，应该组合在一起。
3) 重复步骤 2 直到到达树根：只有一个包含所有数据点的聚类。通过这种方法，我们可以选择最终需要多少个聚类，只需选择何时停止合并聚类。

**算法的优点：**

1) 不要求指定聚类的数量，我们甚至可以选择哪个聚类看起来最好。
2) 该算法对距离度量的选择不敏感。
3) 层次聚类方法的一个特别好的用例是，当底层数据具有层次结构，你可以恢复参差结构

**算法的缺点：**

1) 层次聚类的时间复杂度为 O($n^3$)

## DBSCAN 算法

DBSCAN 的全称是 Density-Based Spatial Clustering of Applications with Noise，是基于密度的带有噪声的空间聚类。是比较有代表性的基于密度的聚类算法。

**算法原理：**

1) DBSCAN 以一个从未访问过的任意起始数据点开始。这个点的邻域是用距离 ε （所有在 ε 距离的点都是邻点）来提取的。

2) 如果在这个邻域中有足够数量的点，那么聚类过程就开始了，并且当前的数据点成为新聚类中的第一个点。否则，该点将被标记为噪声（稍后这个噪声点可能会成为聚类的一部分）。在这两种情况下，这一点都被标记为“访问（visited）”。
3) 对于新聚类中的第一个点，其 ε 距离附近的点也会成为同一聚类的一部分。这一过程使在 ε 邻近的所有点都属于同一个聚类，然后重复所有刚刚添加到聚类组的新点。
4) 步骤 2 和步骤 3 的过程将重复，直到聚类中的所有点都被确定，就是说在聚类附近的所有点都已被访问和标记。
5) 一旦完成了当前的聚类，就会检索并处理一个新的未访问点，这将导致进一步的聚类或噪声的发现。这个过程不断地重复，直到所有的点被标记为访问。因为在所有的点都被访问过之后，每一个点都被标记为属于一个聚类或者是噪音。

**算法的优点：**

1) 对原始数据分布规律没有明显要求，能适应任意数据集分布形状的空间聚类。
2) 无需指定聚类数量，对结果的先验要求不高
3) 由于 DBSCAN 可区分核心对象、边界点和噪声，因此对噪声的过滤效果好，能有效应对数据噪点。

**算法缺点：**

1) 对于高维问题，距离$\varepsilon$和密度的定义十分困难。
2) 当簇的密度变化太大时，聚类效果较差。
3) 当数据量增大时，要求较大的内存支持。

## CLIQUE 子空间聚类算法

CLIQUE 算法是基于网格的空间聚类算法，但是它同时也非常好的结合了基于密度的聚类算法，因此既能够发现任意形状的簇，又可以像基于网格的算法一样处理较大的多维数据。它把每个维度划分为不重叠的社区，从而把数据的整个嵌入空间划分成单元，同时使用一个密度阈值来识别稠密单位。

算法需要两个参数：一个是网格的步长，确定空间的划分；第二个是密度的阈值，用来定义密集网络。

**算法思想：**

1) 首先扫描所有网格。当发现第一个密集网格时，便以该网格开始扩展，扩展原则是若一个网格与已知密集区域内的网格邻接并且其其自身也是密集的，则将该网格加入到该秘籍区域中，知道不再有这样的网格被发现为止。（密集网格合并）
2) 算法再继续扫描网格并重复上述过程，知道所有网格被遍历。以自动地发现最高维的子空间，高密度聚类存在于这些子空间中，并且对元组的输入顺序不敏感，无需假设任何规范的数据分布，它随输入数据的大小线性地扩展，当数据的维数增加时具有良好的可伸缩性。

**算法优点：**

1) 给定每个属性的划分，单遍数据扫描就可以确定每个对象的网格单元和网格单元的计数。
2) 尽管潜在的网格单元数量可能很高，但是只需要为非空单元创建网格。
3) 将每个对象指派到一个单元并计算每个单元的密度的时间复杂度和空间复杂度为 O(m)，整个聚类过程是非常高效的

**算法缺点：**

1) 像大多数基于密度的聚类算法一样，基于网格的聚类非常依赖于密度阈值的选择。（太高，簇可能丢失。太低，本应分开的簇可能被合并）
2) 如果存在不同密度的簇和噪声，则也许不可能找到适合于数据空间所有部分的值。
3) 随着维度的增加，网格单元个数迅速增加（指数增长）。即对于高维数据，基于网格的聚类倾

向于效果很差。

### GaussianMixtureModel

高斯混合模型聚类技术的基本思想是任意形状的概率分布都可以用多个高斯分布函数去近似，其中每个高斯分布为一个聚类，这些聚类线性加成在一起就组成了高斯混合模型（GMM）的概率密度函数：

$$p(x) = \sum_{k=1}^{K} \pi_k p(x|k)$$

其中 k 是模型的个数，$\pi_k$是第 k 个高斯的权重，$p(x|k)$是第 k 个高斯概率密度。

**算法思想：**

1) 首先选择聚类的数量（如 K-Means 所做的那样），然后随机初始化每个聚类的高斯分布参数。
2) 给定每个聚类的高斯分布，计算每个数据点属于特定聚类的概率。一个点离高斯中心越近，它就越有可能属于那个聚类。
3) 基于这些概率，为高斯分布计算一组新的参数，这样就能最大程度地利用聚类中的数据点的概率。我们使用数据点位置的加权和来计算这些新参数，权重是属于该特定聚类的数据点的概率。
4) 步骤 2 和 3 被迭代地重复，直到收敛，在那里，分布不会从迭代到迭代这个过程中变化很多。

**算法优点：**

1) 高斯混合模型在聚类协方差方面比 K-Means 要灵活得多；根据标准差参数，聚类可以采用任何椭圆形状，而不是局限于圆形。
2) 根据高斯混合模型的使用概率，每个数据点可以有多个聚类。

## 2.1.2 分类技术概述

### 基本概述

分类（Classification）是通过有指导的学习训练建立分类模型，使用模型对未知分类的实例进行分类。其目的是根据数据集的特点构造一个分类函数或分类模型（也常称作分类器），该模型能把未知类别的样本映射到给定的类别当中。

构造模型的过程一般分为训练和测试两个阶段。在构造模型前，要求将数据集随机的分为训练集和测试集。在训练阶段，使用训练数据集，通过分析由属性描述的数据库元组来构造模型。在测试阶段，使用测试数据集来评估模型的分类准确率，如果认为可以接受，就可以用该模型对其他数据元组进行分类。一般来说，测试阶段的代价远低于训练阶段。

### 经典分类技术

### K-近邻

K-近邻（K-NearestNeighbor, KNN）算法是一种基于实例的分类方法，最初最初由 Cover 和 Hart 于 1968 年提出，是一种非参数的分类方法。

在 KNN 中，通过计算对象间距离来作为各个对象之间的非相似性指标，避免了对象之间的匹配问题，在这里距离一般使用欧氏距离或曼哈顿距离。
欧氏距离：

$$d(x,y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2}$$

曼哈顿距离:

$$d(x,y)=\sqrt{\sum_{k=1}^{n}|x_k-y_k|}$$

**算法思想：**
1）计算测试数据与各个训练数据之间的距离。
2）按照距离的递增关系进行排序。
3）选取距离最小的 K 个点。
4）确定前 K 个点所在类别的出现频率。
5）返回前 K 个点中出现频率最高的类别作为测试数据的预测分类。
**算法优点：**
1) 在进行类别决策时，只与极少量的相邻样本有关，较好的避免样本不平衡问题。
2) 对于类域的交叉或是重叠较多的待分类样本集较为友好。
**算法缺点：**
1) 计算量大，必须对每个待分类的样本都要计算它到全体已知样本的距离才能求它到 K 个最近邻点。

# 贝叶斯分类

贝叶斯分类是一类分类算法的总称，这类算法均以贝叶斯定理为基础，故统称为贝叶斯分类。贝叶斯定理（Bayes’theorem）是概率论中的一个结果，它与随机变量的条件概率以及边缘概率分布有关。在有些关于概率的解说中，贝叶斯定理能够告知我们如何利用新证据修改已有的看法。通常，时间 A 在事件 B（发生）的条件下的概率，与事件 B 在事件 A 的条件下的概率是不一样的；然而，这两者是有确定的关系，贝叶斯定理就是这种关系的陈述。

贝叶斯的核心公式：

$$P(B|A)=\frac{P(A|B)P(B)}{P(A)}$$

# SVM

SVM (支持向量机)是建立在统计学习理论基础上的机器学习方法，为十大数据挖掘算法之一。通过学习算法，SVM 可以自动寻找出对分类有较好区分能力的支持向量，由此构造出的分类器可以最大化类与类的间隔，因而有较好的适应能力和较高的分准率。SVM 算法的目的在于寻找一个超平面 H，该超平面可以将训练集中的数据分开，且与类域边界的沿垂直于该超平面方向的距离最大，故 SVM 法亦被称为最大边缘算法。
**支持向量机算法优点：**
1) SVM 模型有很高的分准率；

2) SVM 模型有很高的泛化性能；
3) SVM 模型能很好地解决高维问题；
4) SVM 模型对小样本情况下的机器学习问题效果好。

**支持向量机算法的缺点：**

1) SVM 模型对缺失数据敏感
2) 对非线性问题没有通用解决方案，得谨慎选择核函数来处理。

## 决策树

从数据产生决策树的机器学习技术称为决策树学习（Decision Tree）。决策树是数据挖掘中最常用的一种分类和预测技术，使用其可建立分类和预测模型。决策树模型是一个树状结构，树中的每个节点表示经历从根节点到该节点这条路上的对象的值。模型通过树中的各个分支对对象进行分类，叶节点表示的对象的值表达了决策树分类的结果。决策树仅有一个输出，若需要有多个输出，可以建立多棵独立的决策树以处理不同输出。

**以 C4.5 为基础，决策树算法的基本过程如下：**

1) 给定一个“属性-值”格式的数据集 T
2) 选择最能区别 T 中实例的输入属性，C4.5 使用增益率来选择该属性
3) 使用该属性创建树结点，同时创建该节点的分支
4) 使用这些分支对数据实例分类
5) 将子类实例集合设置为 T，对数据集中的剩余属性重复 2-3 步骤，直到满足下列条件之一过程终止，创建叶子节点，沿此类别所表达的分类类别为输出值：
   a) 该子类中的实例满足预定义标准
   b) 没有剩余属性

**决策树关键技术：**

1) 选择最能区别数据集中实例属性的方法
2) 剪枝方法
3) 检验方法

其中，C4.5 使用增益率的概念来选择属性，目的是使得树的层次和节点数最小，数据概化程度最大化，这里会用到信息熵的概念：

$$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2\big(p(x_i)\big)$$

剪枝（Pruning）是为了控制决策树的规模，剪枝基本分为两种：预剪枝（Pre-Pruning）和后剪枝（Post-Pruning）。而后剪枝的计算量代价比预剪枝方法大得多，特别是在大数据集中。而对于小数据集的情况，后剪枝方法优于预剪枝。像其它有指导的学习模型，决策树也需要采取一些检验方法对其分类的正确程度进行评估。

**算法优点：**

1) 不需要任何领域知识和参数设置，适用于探测性的知识发现。
2) 便于理解和解释。树的结构可视化
3) 可以通过数值统计测试来验证该模型。这对解释验证该模型的可靠性成为可能
4) 即使是该模型假设的结果越真实模型所提供的数据有些违反，其表现依旧良好

**算法缺点：**

1) 决策树模型容易产生一个过于复杂的模型，这样的模型对数据的泛化性能会很差。
2) 决策树可能是不稳定的，因为在数据中的微小变化可能会导致完全不同的树生成。这个问题可以通过决策树的集成来得到缓解。

3) 有些概念很难被决策树学习到，因为决策树很难清楚的表述那些概念，例如 XOR，奇偶或者复用器问题；
4) 如果某些类在问题中占主导地位会使得创始的决策树有偏差，因此建议在拟合前先对数据集进行平衡。

## 神经网络

人工神经网络（artificial neural network，ANN），简称神经网络（neural network，NN），是一种模仿生物神经网络的结构和功能的数学模型或计算模型。神经网络由大量的人工神经元联结进行计算。大多数情况下人工神经网络能在外界信息的基础上改变内部结构，是一种自适应系统。现代神经网络是一种非线性统计性数据建模工具，常用来对输入和输出间复杂的关系进行建模，或用来探索数据的模式。该算法就是一组连续的输入/输出单元，其中每个连接都与一个权相关。在学习阶段，通过调整神经网络的权，使得能够预测样本的正确类标号来学习。

**算法优点：**

1) 能处理数值型及分类型的属性。m
2) 分类的准确度高，分布并行处理能力强。
3) 对包含大量噪声数据的数据集有较强的鲁棒性和容错能力。

**算法缺点：**

1) 不能观察之间的学习过程。
2) 学习时间过长，甚至可能达不到学习的目的。
3) 对于非数值型数据需要做大量数据预处理工作。
4) 输出结果难以解释，会影响到结果的可信度和可接受程度。
5) 神经网络需要大量的参数，如网络拓扑结构、权值和阈值的初始值。

# 2. 2 关联规则

## 2.2.1 关联规则概述

关联规则的一般表现为蕴含式规则形式：X→Y。其中 X 和 Y 分别被称为关联规则的前提或先导条件（Antecedent）和结果或后继（Consequent）。

关联规则与传统的用于分类的产生式规则有两点不同。

在某条关联规则中以前提条件出现的属性可以出现在下一条关联规则的结果中。

传统的用于分类的产生式规则的结果中只能有一个属性，而关联规则中允许其结果包含一个或多个属性。

一般情况下，使用置信度（Confidence）来度量每个关联规则在前提条件下结果发生的可能性：

$$conf(X \rightarrow Y) = \frac{sup\ p(X \cup Y)}{sup\ p(X)}$$

可以使用支持度（Support）这个统计量来表示项目集在数据集中出现的频率：

$$sup\ p(X \rightarrow Y) = \frac{|X \cup Y|}{n}$$

## 2.2.2 关联分析

1993 年，阿戈登（Agrawal）等人提出了著名的关联分析算法——Apriori 算法。Apriori 算法基本思想如下：

1) 生成条目集（Item Sets）。条目集是符合一定的支持度要求的“属性-值”的组合。那些不符合支持度要求的“属性-值”组合被丢弃，因此，规则的生成过程可以在合理的时间内完成。
2) 使用生成条目集创建一组关联规则。

## 2. 3 统计学相关技术

## 2.3.1 统计技术概要

统计学是一门数据收集、整理和分析数据，从而得到数据特征和预测对象未来的综合性科学。大多数统计分析方法都具有较强的数学理论基础，在分析数据和预测对象方面有着较高的准确度，从而使其在社会科学和自然科学的各个领域都得到了普遍和成功的应用。统计分析方法和技术也是数据挖掘技术中非常重要和比较成熟的技术，在数据挖掘的过程中，运用延申和拓展了许多统计学方法。常用的分析方法包括回归分析、贝叶斯分析、聚类技术和主成分分析、时间序列分析等。

下面将对几种常见的统计技术进行概述，其中包括回归分析、贝叶斯分析、统计聚类技术、

## 2.3.2 回归分析

回归分析（Regression Analysis）是一种统计分析方法，它可以用来确定两个或者两个以上变量之间定量的依赖关系，并建立一个数学方程作为数学模型，来概化一组数值数据，进而进行数值数据的估值和预测，其应用非常广泛。

回归分析是一种有监督的技术，按照自变量的多少，可以分一元回归分析和多元回归分析；按照自变量和因变量之间的关系类型，可分为线性回归分析（linear Regression Analysis）和非线性回归分析（Nonlinear Regression Analysis）。

### 线性回归分析

根据自变量和因变量相关关系，建立线性回归方程。线性回归方程格式如下：

$$y = a_1x_1 + a_2x_2 + \cdots + a_ix_i + \cdots a_nx_n + c$$

其中，x 是自变量，y 是因变量：a 和 c 是常量。值得注意的是，常用来计算 a 和 c 的统计学方法是最小二乘法（Least-Squares Criterion）。又称为最小平方法，是通过使得因变量预测值与实际值之间的误差的平方和（方差）最小，而给出 a 和 c 的最优解。

### 非线性回归分析

线性和非线性回归分析都是使用最小二乘法进行回归分析，区别只是分析的问题中变量之间的关系呈线性的和非线性的。

**其中常见的非线性回归模型有：**

1) 指数函数：$y = ae^x$

2) 对数函数：$y = a + b\ln x$
3) 幂函数曲线方程：$y = ax^b$
4) 抛物线函数：$y = a + bx + cx^2$
5) 双曲线函数：$y = \frac{x}{a+bx}$
6) S 形曲线函数（Logistic 函数）：$y = \frac{k}{1+ae^{-bx}}$

**非线性回归分析的步骤：**

1) 选择非线性回归方程
2) 通过变量置换，将非线性问题转换为线性回归，利用线性回归方法进行参数估计。
3) 评估线性模型

### 树回归

回归树（Regression Tree），本质上就是一棵决策树，只是其叶节点是数值而不是分类类型值。一个叶结点的值就是经过树到达叶节点的所有实例的输出属性的平均值。其中最著名的就是分类回归树（Classification And Regression Tree, CART），它能够针对复杂的、非线性问题建模。

## 2.3.3 贝叶斯分类

贝叶斯分析（Bayesian Analysis）是一种参数估计方法，在概述分类技术时已经介绍。它将未知参数的先验信息与样本信息相结合，根据贝叶斯公式，得到后验信息，然后根据后验信息去推断未知参数。贝叶斯分析方法在决策支持、分险评估、模式识别等方面都得到了广泛的应用，被用来建立分类模型，就是著名的贝叶斯分类器（Bayes Classifier）。

## 2.3.4 聚类技术

作为数据挖掘重要技术的聚类技术，在前面已经介绍过，其中使用了许多统计分析方法，包括基于划分的聚类方法、基于分层的聚类方法、基于模型的聚类方法。在这里就简要介绍一下三种聚类技术：凝聚聚类、Cobweb 和 EM 算法。

### 凝聚聚类

凝聚聚类（Agglomerative Clustering）是一种很受欢迎的无指导聚类技术。与 K-means 算法需要在聚类之前确定簇的个数不同，凝聚聚类在开始时假定每个数据实例代表它自己的类。

**算法步骤如下：**

1) 开始时，将每个数据实例放在不同的分类中。
2) 直到所有实例都成为某个簇的一部分。
   a) 确定两个最相似簇
   b) 将在 1 中选中的簇合并成为一个簇。
3) 选择一个由步骤 2 迭代形成的簇作为最后的结果。

## Cobweb 分层聚类算法

Cobweb 算法是一种增量式分层聚类算法。Cobweb 使用分类树对实例数据进行分类，分类树的构造过程是一种概念分层的过程，这个过程被称为概念聚类。Cobweb 接受实例格式“属性-值”,并且属性值必须是分类类型的。下面是标准的 Cobweb 概念聚类算法。

1) 建立一个簇，使用第一个实例作为它唯一的成员。
2) 对于每个剩余实例，在每个概念分层中，用一个评价函数决定选择以下两个动作之一执行。
   a) 将实例放到一个已存在的簇中。
   b) 创建一个只具有这个新实例的新概念簇。

Cobweb 的评价函数使用一种启发式评价方法——分类效用（Category Utility）来指导分类。CU 定义了聚类的好坏，值越小聚类较差，值越大聚类效果质量越好。

**CU 的计算公式如下：**

$$\boldsymbol{CU = \frac{\sum_{k=1}^{m} P(C_k)\left(\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2\right)}{m}}$$

## EM 算法

EM（Expectation-Maximization）算法是一种采用有限高斯混合模型的统计技术，EM 算法与 K-means 算法相似，都是迭代的进行参数估计直到得到一个期望的收敛值。假设概率分布式正态的，分类簇个数是 2，EM 算法的基本过程如下：

1) 估计 5 个参数的初始值
2) 直到满足度量聚类质量的值不在增大，是用来指聚类所确定的簇的可能值来度量类的质量，值越高表示聚类越理想。

其中，算法使用下式对的正态分布的概率密度函数计算每个实例的分类概率：

$$f(x) = \frac{1}{(\sigma\sqrt{2\pi})e^{\frac{-(x-\mu)^2}{2\sigma^2}}}$$

# 2.4 智能计算技术

智能计算也被称之为“软计算”，是受自然界规律的启迪，根据其原理，模仿求解问题的算法。从自然界得到启迪，模仿其结构进行发明创造，另一方面，我们还可以利用仿生原理进行设计，这就是智能计算的思想。这方面的内容包括人工神经网路、爬山法、遗传算法、模拟退火算法和群体智能算法。

下面将对以上算法一一进行简单的介绍，神经网络算法将在下一节进行概述。

## 2.4.1 爬山法

爬山法（Hill Climbing，HC）是一种局部择优的贪心搜索算法，其本质上是梯度下降法。该算法每次从当前的节点开始，与周围的邻接点进行比较：

1) 若当前节点是最大的，那么返回当前节点，作为最大值。
2) 若当前节点是最小的，就用最高的邻接点替换当前的节点，从而实现向山峰的高出攀爬的目的。

如此往复，直到达到最高点为止。

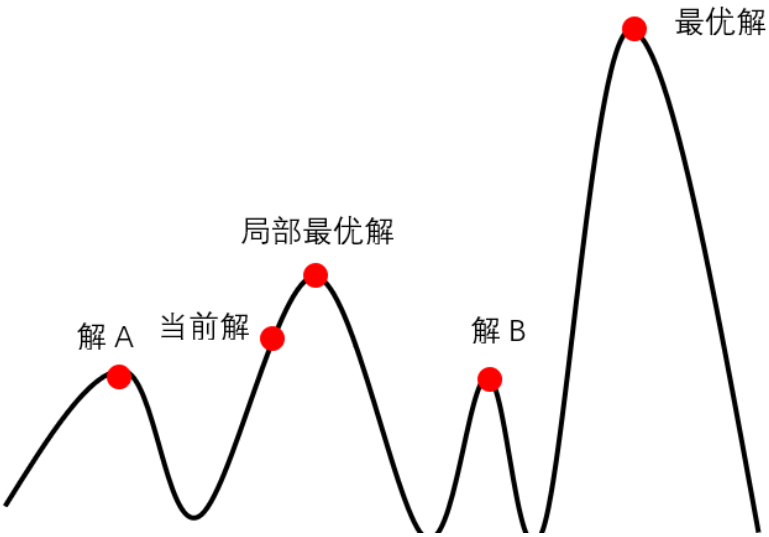


图 1.4.1 爬山法局部最大

此外，如上图 1.4.1，爬山法存在一个主要问题：局部最大，即某个节点会比周围任何一个邻居都高，但只是局部最优解，并非全局最优解。此外，其还存在以下两种问题：

1) 高地问题：搜索一旦到达高地，就无法确定搜索最佳方向，会产生随机走动，使得搜索效率降低
2) 山脊问题：搜索可能会在山脊的两面来回震荡，前进步伐很小

当出现以上问题后，只能随机重启爬山算法来解决。

## 2.4.2 遗传算法

遗传算法（Genetic Algorithms）是基于生物进化理论的原理发展起来的一种广为应用的、高效的随机搜索与优化的方法。其主要特点是群体搜索策略和群体中个体之间的信息交换，搜索不依赖于梯度信息。遗传算法成功的应用包括：作业调度与排序、可靠性设计、车辆路径选择与调度、成组技术、设备布置与分配、交通问题等等。

**遗传算法与其它搜索算法的区别：**

1) 遗传算法从问题解的串集开始嫂索，而不是从单个解开始。
2) 遗传算法同时处理群体中的多个个体，即对搜索空间中的多个解进行评估，减少了陷入局部最优解的风险，同时算法本身易于实现并行化。
3) 遗传算法基本上不用搜索空间的知识或其它辅助信息，而仅用适应度函数值来评估个体，在此基础上进行遗传操作。
4) 遗传算法不是采用确定性规则，而是采用概率的变迁规则来指导他的搜索方向。
5) 具有自组织、自适应和自学习性。

## 2.4.3 模拟退火算法

模拟退火算法(Simulated Annealing, SA)最早的思想是由 N. Metropolis 等人于 1953 年提出。1983 年,S. Kirkpatrick 等成功地将退火思想引入到组合优化领域。它是基于 Monte-Carlo 迭代求解策略的一种随机寻优算法，其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。

模拟退火算法来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部粒子随温度升高变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，

最后在常温时达到基态，内能减为最小。根据 Metropolis 准则，粒子在温度 T 时趋于平衡的概率为 e-ΔE/(kT)，其中 E 为温度 T 时的内能，ΔE 为其改变量，k 为 Boltzmann 常数。

**Metropolis 准则判断新解接受概率：**

$$P=\begin{cases}1, & E_{t+1}<E_t\\ e^{\frac{-(E_{t+1}-E_t)}{kT}}, & E_{t+1}\geq E_t\end{cases}$$

## 2.4.4 群体智能算法

受社会性昆虫行为的启发，计算机工作者通过对社会性昆虫的模拟产生了一系列对于传统问题的新的解决方法，这些研究就是群集智能的研究。群集智能(Swarm Intelligence)中的群体(Swarm)指的是“一组相互之间可以进行直接通信或者间接通信(通过改变局部环境)的主体，这组主体能够合作进行分布问题求解”。而所谓群集智能指的是“无智能的主体通过合作表现出智能行为的特性”。群集智能在没有集中控制并且不提供全局模型的前提下，为寻找复杂的分布式问题的解决方案提供了基础。

在计算智能(Computational Intelligence)领域有两种基于群智能的算法，蚁群算法(Ant Colony Optimization)和粒子群算法(Particle Swarm Optimization)，前者是对蚂蚁群落食物采集过程的模拟，已经成功运用在很多离散优化问题上。

### 蚁群优化算法

受蚂蚁觅食时的通信机制的启发，90 年代 Dorigo 提出了蚁群优化算法(Ant ColonyOptimization，ACO)来解决计算机算法学中经典的“货郎担问题”。如果有 n 个城市，需要对所有 n 个城市进行访问且只访问一次的最短距离。根据“信息素较浓的路线更近"的原则，即可选择出最佳路线。由于这个算法利用了正反馈机制，使得较短的路径能够有较大的机会得到选择，并且由于采用了概率算法，所以它能够不局限于局部最优解。

### 粒子群优化算法

粒子群优化算法(PSO)是一种进化计算技术(Evolutionary Computation)，有 Eberhart 博士和 Kennedy 博士发明。源于对鸟群捕食的行为研究。

PSO 同遗传算法类似，是一种基于叠代的优化工具。系统初始化为一组随机解，通过叠代搜寻最优值。但是并没有遗传算法用的交叉(crossover)以及变异(mutation)。而是粒子在解空间追随最优的粒子进行搜索。

**PSO 算法过程：**

1) 种群随机初始化。
2) 对种群内的每一个个体计算适应值(fitness value)。适应值与最优解的距离直接有关。
3) 种群根据适应值进行复制 。
4) 如果终止条件满足的话，就停止，否则转步骤 2 。

## 2.5 神经网络算法和深度学习算法

有关神经网络算法在之前的分类技术中已经简单地介绍，深度学习（也称为深度结构化学习或

者微分编程）是机器学习算法这个大家庭的一个成员，它是基于人工神经网络和表示学习，并且它的学习可以是有监督的、半监督或者无监督。

和人类的大脑一样，神经网络也包括了很多神经元。每个神经元接受输入的信号，然后乘以对应的权重，并求和然后输入到一个非线性函数。这些神经元相互堆积在一起，并按照层进行组织。如下图 2.5 所示：

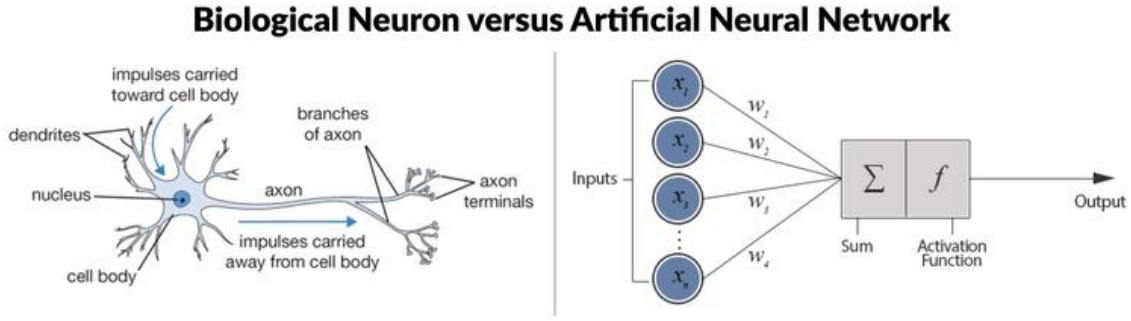


图 2.5 神经网络结构

神经网络通过大量的数据以及反向传播这样一个迭代算法来学习到目标函数。我们将数据传入网络中，然后它输出结果，接着我们将输出的结果和预期结果进行比较（通过一个损失函数），然后根据两者的差异来调整权重。

不断重复这个过程。调整权重的办法是通过一个非线性优化技术--随机梯度下降来实现的。

在训练一段时间后，网络将可以输出非常好的结果，因此，训练到此结束。也就是说我们得到了一个近似的函数，当给网络一个未知结果的输入数据，网络会根据学习到的近似函数输出结果。

## 2.5.1 前向神经网络

前向神经网络（FNN）通常采用的都是全连接层，也就是说每一层的神经元都和下一层的所有神经元连接在一起。这个结构也被叫做多层感知器，最初诞生于 1958 年，如下图 2.5.1 所示。单层的感知器只能学习到线性分离模型，但是一个多层感知器能够学习到数据之间的非线性关系。多层感知器在分类和回归任务上有不错的表现，但相比其他的机器学习算法，多层感知器并不容易收敛。另外，训练数据越多，多层感知器的准确率也越高。

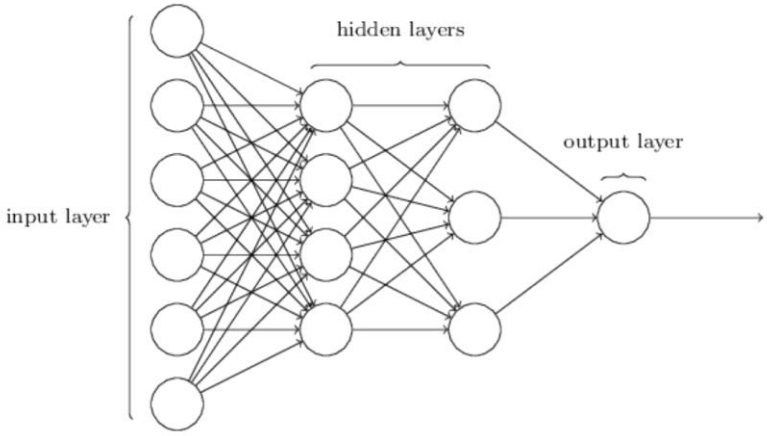


图 2.5.1 FNN 结构

## 2.5.2 卷积神经网络

卷积神经网络采用了一个卷积函数。没有采用层与层之间的神经元都全部进行连接，卷积层只让两层之间部分的神经元进行连接（也就是感受野）。在某种程度上，CNN 是尝试在 FNN 的基础上进行正则化来防止过拟合（也就是训练得到的模型泛化能力差），并且也能很好的识别数据之间的空间关系。一个简单的 CNN 的网络结构如下图 2.5.2 所示。

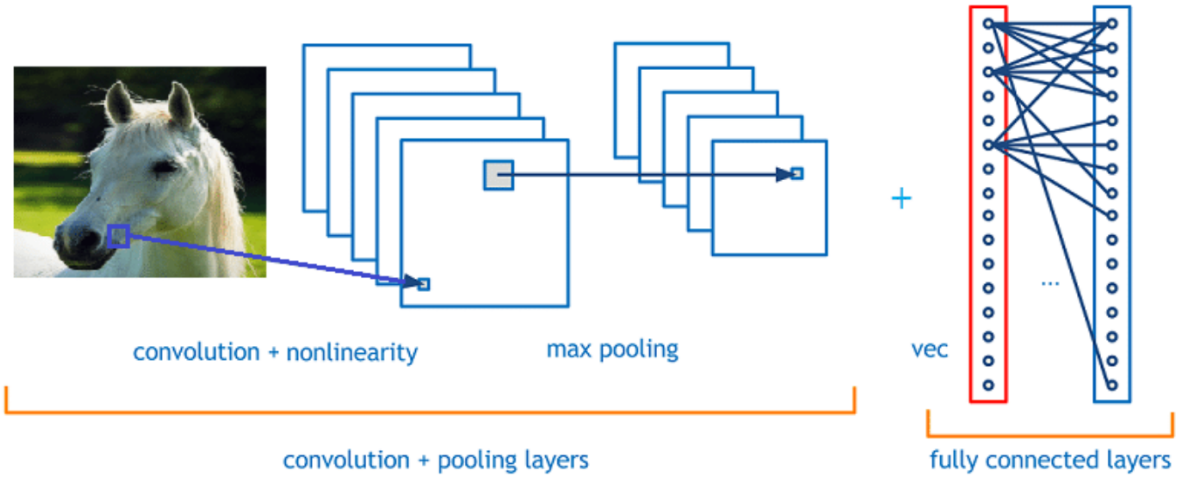


图 2.5.2 CNN 结构

因为能够很好识别数据之间的空间关系，所以 CNN 主要用于计算机视觉方面的应用，比如图像分类、视频识别、医学图像分析以及自动驾驶，在这些领域上都取得超过人类的识别精度。此外，CNN 也可以和其他类型的模型很好的结合在一起使用，比如循环神经网络和自动编码器，其中一个应用例子就是符号语言识别。

## 2.5.3 循环神经网络

循环神经网络非常适合时间相关的数据，并且应用于时间序列的预测。该网络模型会采用反馈的形式，也就是将输出返回到输入中。你可以把它看成是一个循环，从输出回到输入，将信息传递回网络，因此，网络模型具有记住历史数据并应用到预测中的能力。

为了提高模型的性能，研究者修改了原始的神经元，创造了更复杂的结构，比如 GRU 单元和 LSTM 单元，分别如下图 2.5.3 所示。LSTM 在自然语言处理的任务中应用得非常广泛，包括翻译、语音生成、从文本生成语音等。

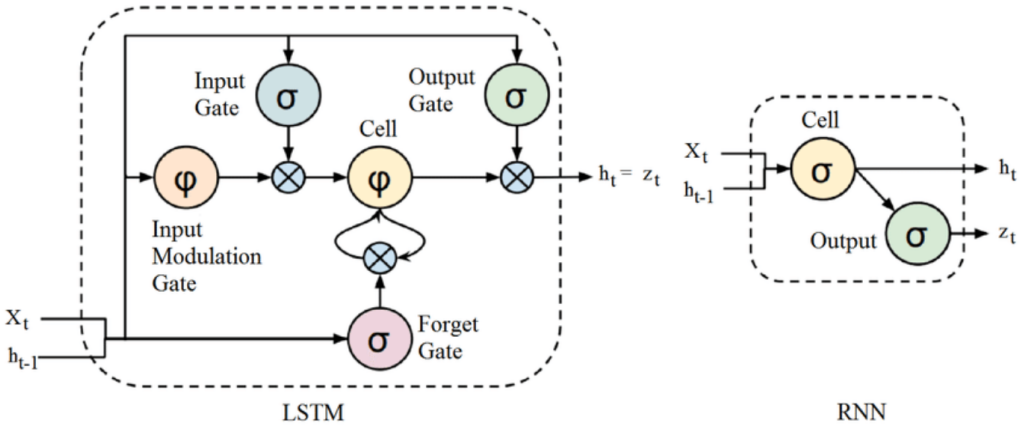


图 2.5.3 RNN 结构

## 2.6 目前数据挖掘技术和方法在相关领域的应用

在大数据时代下，数据挖掘作为最常用的数据分析手段得到了各个领域的认可，目前国内外学者主要研究数据挖掘中的分类、优化、识别、预测等技术在众多领域中的应用。下面将在这四个主要领域对数据挖掘技术的应用进行阐述。

### 2.6.1 分类应用

大数据时代下数据挖掘技术的运用更多体现在数据分析层面，通常数据挖掘在市场营销领域内运用最为广泛。借助数据挖掘技术，能够搜集和掌握大量的市场用户信息资源，通过数据分析的形式来获取用户们的真实需求。特别是通过分类技术分析用户群像，商户也可在后台的数据管理系统当中，为客户选择性推送一些与商品相关的衍生产品，从而由此让用户们获得更多选择，极大的提升用户们的产品使用感受。

### 2.6.2 优化应用

道路的交通状况与人们的出行关系密切，随着城市的快速发展、生活水平的改善，机动车的规模也逐渐扩大，带来了交通拥堵等问题。数据挖掘技术可以有效解决交通道路和物流网络之间的优化问题，在人工智能的优化时代，使用无人机探测道路状况反馈的数据，采用数据挖掘技术精准计算物流网络运输所需要的参数，可以轻松高效地缓解物流运输瘫痪的问题。

### 2.6.3 识别应用

自从 20 世纪 50 年代数字图像出现以来，数字图像成为人类社会中必不可少的“数据”。在计算机应用中，数据挖掘在图像识别的应用越来越普遍，有代表性应用为人脸识别和指纹识别。人脸识别通过对获得的信息库进行数据挖掘，进一步分析和处理可靠的、潜在的数据，充分准备资料的分析工作和未来的开发工作。

### 2.6.4 预测应用

预测问题是各领域中研究最多的问题，其目的是通过历史数据预测出未来的数据值或发展趋势。大部分历史数据是时间序列数据，即指按照时间的顺序排列，得到了一系列观测值。由于信息技术的不断进步，时间序列的数据也日益剧增，如气象预报、石油勘探、金融等。时间序列数据挖掘的最终目标就是通过分析时间序列的历史数据，预测未来一段时间的变化趋势及其带来的影响。

在大数据时代下，银行、证券公司、保险公司等每天的业务都将生成海量数据，采用当前的数据库系统可以高效地实现数据的录入、查询和统计等功能，目前，从简单的查询提升到利用数据挖掘技术挖掘知识、提供决策支持的层次显得格外重要。数据挖掘技术在金融行业应用具有可行性，将理论基础应用到相关的实例包括预测股票指数、发现金融时间序列中的隐含模式、信用风险管理及汇率预测等。

# 第3章　　数据挖掘的过程模型概述

## 3.1 经典 KDD 处理模型

经典 KDD 处理模型又称阶梯处理模型，是 Fayyad 等人提出的具有九个步骤的阶梯递进的 KDD 处理模型（如图 3.1）。其中九个步骤分别如下：

1) 数据准备：了解应用领域相关情况，熟悉相关背景知识。
2) 数据选择：根据用户的要求从数据库中提取与 KDD 相关的数据，KDD 将主要从这些数据中进行知识提取。
3) 数据预处理：对从数据库中提取的数据进行加工，对其中噪声数据、缺失数据进行处理。
4) 数据缩减：对经过预处理的数据，根据知识发现的任务对数据进行再处理，主要通过投影或数据库中的其它操作减少数据量。
5) 确定 KDD 目标：根据用户要求，根据知识发现的任务对数据进行再处理。
6) 确定知识发现算法：选取合适的模型和参数，使得知识发现算法与整个 KDD 的评价标准相一致。
7) 数据挖掘：运用选定的知识发现算法，从数据中提取用户所需要的知识。
8) 模式解释：对发现的模式进行解释。
9) 知识评价：将发现的知识以用户能理解的方式呈现给用户，同时对所发现的知识进行检验和评估。

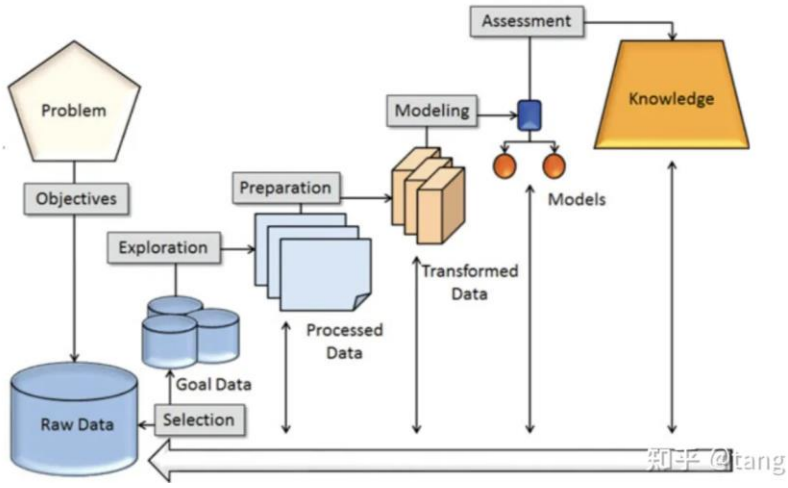


图 3.1 KDD 处理结构

KDD 是一个交互、迭代、多步骤处理过程.一次 KDD 并不一定得到理想的结果，因此 KDD 是一个目标和数据不断优化的过程。可以在当前选择的知识发现算法不变的情况下，对学习参数进行调整，并重新训练和评价，直到达到满意的结果为止。

## 3.2 跨行业数据挖掘标准流程

另一种在应用中已经得到公认的处理模型是 CRISP-DM（Cross Industry Standard Process for Data Mining，跨行业数据挖掘标准流程）。该模型包括以下六个过程,模型如下图 3.2。

1) 商业理解（Business Understanding）：关注的焦点是项目目标和商业前景的需求。给出了数据挖掘问题的定义和最初的计划。

2) 数据理解(Data Understanding)：重点是数据的收集和假设构造。
3) 数据准备(Data Preparation)：选择表、记录和属性，为所选的模型工具清洗数据。
4) 建模(Modeling)：重点是选择和应用一个或多个数据挖掘技术。
5) 评估(Evaluation)：通过对发现的结果进行分析，判断模型是否达到目标，同时确定未来的使用价值。
6) 部署(Deployment)：制定行动计划应用模型。

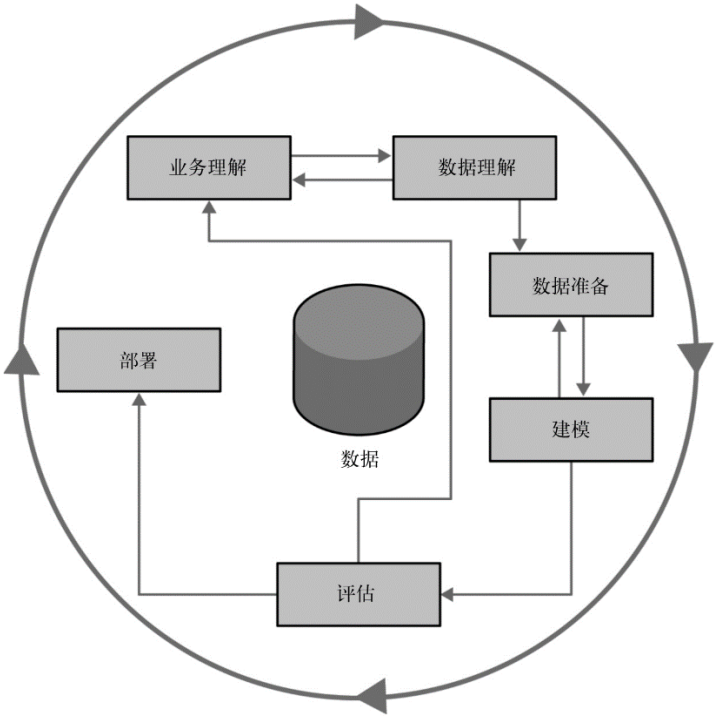


图 3.1 CRISP-DM 处理结构

## 3.3 数据仓库

支持数据挖掘项目执行的一个重要基础是大量的、高质量的数据。数据的采集和手机是数据挖掘过程中基础且重要的一个步骤。所以需要一种有组织、高效的数据存取结构，集成存储，而数据仓库正式具备这样功能的数据存储架构。

数据仓库（Data Warehouse）,可简写为 DW 或 DWH,数据仓库，是为了企业所有级别的决策制定计划过程，提供所有数据类型的战略集合。它出于分析性报告和决策支持的目的而创建。为需要业务智能的企业 ,为需要指导业务流程改进、监视时间，成本，质量以及控制等。 数据仓库是依照分析需求、分析维度、分析指标进行设计的。

根据恩门（Inmon）的数据仓库定义，数据库应该要体现以下几个特点：

**1) 数据仓库是面向主题的：**

数据仓库中的数据是按照一定的主题域进行组织。 主题是一个抽象的概念,是指用户使用数据仓库进行决策时所关心的重点方面,一个主题通常与多个操作型信息系统相关。

**2) 数据仓库是集成的：**

数据仓库中存储的数据是来源于多个数据源的集成，原始数据来自不同的数据源，存储方式各不相同。要整合成为最终的数据集合，需要从数据源经过一系列抽取、清洗、转换的过程。

**3) 数据仓库是相对稳定的：**

数据仓库中保存的数据是一系列历史快照，不允许被修改。用户只能通过分析工具进行查询和

分析。这里说明一点，数据仓库基本上是不许允许用户进行修改，删除操作的。大多数的场景是用来查询分析数据。

**4) 数据仓库是反应历史变化的：**

数据仓库会定期接收新的集成数据，反应出最新的数据变化。这和稳定特点并不矛盾。

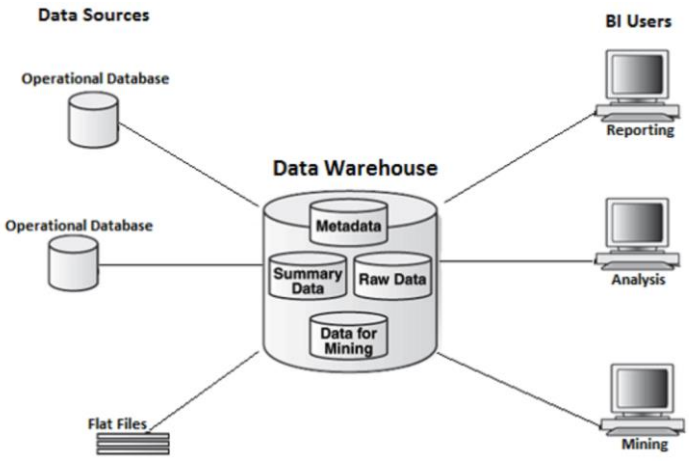


图 3.3 数据仓库

# 第4章　　数据挖掘评估方法

## 4.1 评估方法概述

模型的性能评估是数据挖掘过程中非常重要的步骤，是模型是否能够最终投入实际应用的一个重要环节。下图给出建立模型的过程中可能对模型性能产生影响的因素。

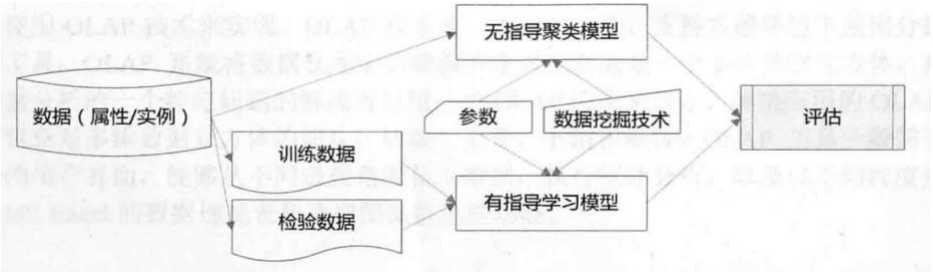


图 4 数据挖掘过程涉及内容环节

从中可以看出数据挖掘的评估内容包括：

1) 数据：高质量的数据大程度影响模型质量
2) 参数：在数据挖掘的模型建立过程中，需要设置多个参数，其对模型的影响很大。
3) 数据挖掘技术：用不同数据挖掘技术建立的有指导学习模型显示出检验集正确率往往相差无几。
4) 模型：对于有指导模型，通常会在检验数据上进行评估。
5) 检验集：数据集一般分为训练集和检验集，在校验集上的评估被称为检验集评估。

评估工具包括混淆矩阵、统计学方法、有指导学习和无指导聚类技术，下面将做简要介绍。

### 4.1.1 混淆矩阵

混淆矩阵（Confusion Matrix）是评估有指导学习模型的基本工具，它能够直观地给出模型检验及分类正确或错误情况。一个混淆矩阵如表 4.1.1 所示：

表 4.1.1 混淆矩阵

| | C1 | C2 | C3 |
|---|---|---|---|
| C1 | C11 | C12 | C13 |
| C2 | C21 | C22 | C23 |
| C3 | C31 | C32 | C33 |

可以使用混淆矩阵中的数值来计算模型的准确度。将主对角线上的值之和除以检验集实例总数，即得到模型的检验集分类正确率。由于模型准确率经常表示为错误率，可以使用 1.0 减去模型正确率来计算模型的错误率。

$$\text{模型检验集正确率}=\frac{\sum_{i=1}^{3}\sum_{j=i}^{i}C_{ij}}{\sum_{i=1}^{3}\sum_{j=1}^{3}C_{ij}}$$

### 4.1.2 统计学方法

统计学中经常会使用以下基本概念，这些概念是模型评估的统计方法的基础。

## 均值和标准差:

数值数据的一个总体可以用均值、标准差和数据中出现的值的频率或概率分布来唯一定义。

1) 均值（Mean）就是平均值，用$\mu$来表示，是所有数据的平均数。
2) 方差（Variance）度量了每个数据与均值的离差量，用$\sigma^2$表示，是所有数据与均值之差的平方和的平均值。
3) 标准差是一组数据距离其均值的分散程度的一种度量。标准差越大，表示大部分数据值距离其均值的差异越大，标准差越小，表示这些数据值越接近均值。

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

## 总体分布

总体分布（Population Distribution）可能是正态分布、指数分布、Gamma 分布等，其中正态分布（Normal Distribution）是一种容易理解、很重要的数据分布。

## 假设检验和 Z 检验

假设检验，也称为显著性检验，通过样本的统计量来判断与总体参数之间是否存在差异（差异是否显著）。即我们对总体参数进行一定的假设，然后通过收集到的数据，来验证我们之前作出的假设（总体参数）是否合理。在假设检验中，我们会建立两个完全对立的假设，分别为原假设 H0 与备择假设 H1。然后根据样本信息进行分析判断，是选择接受原假设还是拒绝原假设。

Z 检验用来判断样本均值是否与总体均值具有显著性差异。Z 检验是通过正态分布的理论来推断差异发生的概率，从而比较两个均值的差异是否显著。Z 检验适用于：

1) 总体呈正态分布。
2) 总体方差已知。
3) 样本容量较大（≥30）。

Z 检验计算公式：

$$Z = \frac{\bar{x} - \mu_0}{S_{\bar{x}}} = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}}$$

# 4.1.3 有指导学习和无指导聚类技术

有指导学习技术和无指导聚类技术互为补充，有指导学习模型能够分类和预测具有定义明确的分类，能够弥补无指导聚类没有明确目标和缺乏对聚类结果进行解释的局限；反之，无指导的聚类技术利用某种相似度度量方法对实例进行自然聚类，能够从中发现类的自然属性，对于有指导学习前的属性和实例有所帮助。

所以可以使用每种技术去评估对方或作为对方的方法补充。

## 4. 2 有指导学习模型的评估方法

在有指导模型中，分为分类模型与回归模型，所以评估方法也有所不同。对于分类模型评估指标中，常见方法有：混淆矩阵、ROC 曲线、AUC 值；对于回归模型评估指标，常见方法有：MSE、RMSE、MAE、$R^2$。除了混淆矩阵在概述中已经提到，下面将对它们做简要介绍。

## 4.2.1ROC 曲线

ROC 曲线是一种用于评价和比较二分类器的工具。它和精确率/召回率曲线有着很多的相似之处，当然它们也有所不同。它将真正类率（true positive rate，即 recall）和假正类率（被错误分类的负实例的比例）对应着绘制在一张图中，而非使用精确率和召回率。

ROC 关注两个指标：

$$truepostivaterate(recall):TPR=\frac{TP}{TP+FN}$$

$$falsepositiverate:FPR=\frac{FP}{PP+TN}$$

## 4.2.2 AUC 值

AUC（Area Under Curve）被定义为 ROC 曲线下的面积，显然这个面积的数值不会大于 1。

简单说：AUC 值越大的分类器，正确率越高。

AUC=1：完美分类器，采用这个预测模型时，不管设定什么阈值都能得出完美预测。绝大多数预测的场合不存在完美分类器。

AUC>0.5：这个分类器（模型）妥善设定阈值的话，能有预测价值。

AUC=0.5：跟随机猜测一样（例：丢铜板），模型没有预测价值。

AUC<0.5：比随机猜测还差；但只要总是反预测而行，就优于随机猜测，因此不存在这种情况。

## 4.2.3 MSE、RMSE、MAE、$R^2$

对于它们的定义如下所示：

MSE(Mean Squared Error) 均方误差：$P=\frac{1}{m}\Sigma\left(y_i-f(x_i)\right)^2$

RMSE(Root Mean Squared Error) 均方根误差：$p=\sqrt{\frac{1}{m}\Sigma\left(y_i-f(x_i)\right)^2}$

MAE(Mean Absolute Error) 平均绝对误差：$P=\frac{1}{m}\Sigma|y_i-f(x_i)|$

$R^2$，决定系数：

$$\mathrm{R}^2=1-\frac{\sum_{\mathrm{i}=1}^{\mathrm{n}}\left(\mathrm{y_i}-\hat{\mathrm{y}}_{\mathrm{i}}\right)^2/\mathrm{n}}{\sum_{\mathrm{i}=1}^{\mathrm{n}}\left(\mathrm{y_i}-\bar{\mathrm{y}}\right)^2/\mathrm{n}}=1-\frac{\mathrm{MSE}}{\mathrm{V\,ar}}$$

MSE 表示均方误差，Var 表示方差。$R^2$可以通俗地理解为使用均值作为误差基准，观察预测误

差是否大于或者小于均值基准误差

MSE、RMSE、MAE、$R^2$主要用于回归模型。它们的优缺点在于 MSE 和 RMSE 可以很好的反应回归模型预测值和真实值的偏离成都，但如果存在个别离群点的偏离程度非常大时，即使其数量非常少也会使得 RMSE 指标变差（因为用了平方）。解决这种问题主要有三个方案：

1) 如果认为是异常点时，在数据预处理的时候就把它过滤掉；
2) 如果不是异常点的话，就提高模型的预测能力，将离群点产生的原因建模进去；
3) 此外也可以找鲁棒性更好的评价指标，如：MAE；

## 4. 3 无指导学习模型的评估方法

有指导的学习模型有明确的输入和输出，其建立的目的是用于分类和预测，模型的应用目标明确。而无指导的聚类模型则不同，通常在聚类之前目标并不明确，所以也造成了对无指导聚类模型的性能评估比有指导模型更为困难。

因为聚类的结果是形成一些依据相似度而聚集的实例簇，所以对于这些簇的质量的度量是评估无指导聚类模型性能的最一般考虑。度量簇的质量常用的方法是计算每个簇中的实例与其簇中心之间的误差平方和。误差平方和越小，簇的质量就越高。

第二种评估无指导聚类的方法是使用有指导学习技术。因为有指导学习的输出是定义明确的类，可以利用这点来解释和评估不能明确表达聚类结果的无指导模型。步骤如下：

(1）建立无指导聚类模型之后，将形成的每个簇作为一个类。如通过无指导聚类形成了 3 个簇，则将它们作为 3 个类。

(2）从这每个类中随机选择 1 个实例样本集，随机选取的目的是保证每个类表示在随机样本中的比率与表示在整个数据集中的比率相同。选取的所有实例数最好占整个数据集的 2/3。

(3）将随机选取的实例作为训练数据，创建以这些类为输出属性的有指导学习模型，并使用剩余的实例作为检验集实例检验有指导模型的分类正确率。

(4）观察这样建立的有指导模型的分类正确率，若分类性能较好说明无指导聚类模型所形成的簇的定义良好；若分类正确率较低，说明聚类所形成的簇没有明确的定义。本章小

## 4. 4 其它评估方法

### 4.4.1 比较有指导学习模型

Lift（提升度或提升指数）度量了一个偏差样本内的类$C_i$的期望集中度相对于总体内的$C_i$的集中度的百分比的变化，可以使用条件概率来表示，如下式所示。

$$Lift = \frac{P(C_i|Sample)}{P(C_i|Population)}$$

其中：$E_1$为模型$M_1$的检验集分类错误率；$E_2$为模型$M_2$的检验集分类错误率；q 为两个模型的分类错误率的平均值，即$q = (E_1 + E_2)/2$；$n_1$和$n_2$分别为检验集 A 和检验集 B 中的实例个数；q(1-q)是用$E_1$和$E_2$计算出来的方差值。

还可以在使用训练数据建立模型之后，先对模型进行比较，选择分类正确率最高的模型，再进行检验集上的检验，获得模型对未知实例预测的性能。对模型的比较可以使用验证数据（Validation Data），它是训练数据和检验数据的补充，可以帮助从多个用同样训练集建立的模型中选择一个。验证数据还可以用于优化有指导模型的参数设置，以获得最高的分类正确率。

## 4.4.2 属性评估

影响模型性能的一个重要因素是数据，包括数据的质量，以及数据集的属性和势力的选择。可以使用属性相关性检查和散点图找出属性冗余，同时，可以使用假设性检验找出对分类预测能力较小的数值属性，将它们从训练集中删除，以提高模型的质量。

### 数值型属性的冗余检查

相关系数（ Correlation Coefficient ）度量了两个数值型属性之间的线性相关程度，对于样本用 r 或 p 表示，对于总体则用希腊字母 rho 表示。相关系数的值介于［-1,1］之间。两个属性正相关（ Positive Correlation ）是指两个属性具有同时增加或减少的特性， r 接近于 1。如身高和体重就是两个正相关性较强的属性。两个属性负相关（ Negative Correlation ）是指一个属性增加而同时另一个属性减少的特性， r 接近于 - 1。如年龄和奔跑速度就是两个负相关性较强的属性。如果 r 接近于 0，则表示两个属性不具有线性相关性。对于属性之间的相关性的判定，除了使用相关系数之外，还需要使用显著性检验，来排除两个属性之间的相关性联系偶然出现的可能。

如果两个输入属性正向或负向高度相关，则只能选择其中的一个用于数据挖掘。正确的选择是选择具有较大重要性值的属性。

### 数值型属性显著性假设检验

使用假设检验来确定属性的显著性分数，过程如下。

(1）设数值型属性 A 具有 n 个类$C_1$, $C_2$,…, $C_n$,，各类中该属性的均值为 $\overline{X_1}$ , $\overline{X_2}$,…, $\overline{X_n}$。

(2）对每一对类$C_i$和$C_j$，使用下式计算显著性分数 Z：

$$z_{ij} = \frac{|\bar{x}_i - \bar{x}_j|}{\sqrt{\left({}^{v_i}/_{n_i} + {}^{v_j}/_{n_j}\right)}}$$

其中：$\overline{X_1}$是类$C_i$的均值；$\overline{X_j}$是类$C_j$的均值；$v_i$是属性 A 的$C_i$的方差；$v_j$ 是$C_j$的方差，$n_i$是类$C_i$中的实例数，$n_j$是类$C_j$中的实例数。

(3）如果 Z 〗的任意一个值≥1.96，则该属性是重要的。即对于属性 A ，在任何一对类的比较中都表现出显著的差异，则该属性应被认为对于分类是重要的。

# 第5章　　基于双向 GRU 的循环神经网络对电影评论情感分析

**[摘要]** 文本情感分析是用算法来分析文本中表达的情感。例如分析一个文本中的情感，判断高兴、悲伤、愤怒等情绪。其也是自然语言处理的一个重要研究方向，如果能将这种文字转为情感的操作让计算机自动完成，就节省了大量的人工时间，这对于目前的大数据时代的数据挖掘工作具有重要意义。本章通过对循环神经网络的深入学习，使用双向 GRU 对电影评论文本情感数据做分类应用。

**[关键词]** 循环神经网路；GRU；LSTM；文本情感分析；数据挖掘

## 5.1 背景与思想

### 5.1.1 设计背景

当今社交媒体和在线评论平台上的评论数量呈爆炸式增长，这些评论中包含了大量的情感信息。情感分析是一种自然语言处理技术，旨在自动识别和提取文本中的情感信息。

本章旨在利用神经网络构建一种电影评论情感分析模型，该模型可以自动地对电影评论进行分类，判断评论者的情感倾向是积极的、消极的还是中立的。通过这种方式，我们可以更好地了解观众对电影的反应，为电影制片人和营销人员提供有用的反馈信息。

### 5.1.2 设计思想

文本情感分析主要有三大任务：文本情感特征提取、文本情感特征分类以及文本情感特征检索与归纳。而文本情感分析的方法主要分为两类：一种是基于情感词典的方法，包括人工构建情感词典和自动构建情感词典；另一种是基于机器学习的方法，包括朴素贝叶斯、SVM 支持向量机等。

随着深度学习在图像处理的领域不断发展创新，深度学习技术也开始涉及文本情感分析领域。随着深度学习的快速发展，词向量模型等的提出恰好为相关研究提供了契机。

同时结合知识库和语料库的优点，并借助深度学习的方法将词语转换成为词向量，同时利用神经网络构造情感极性分类器，判断词语的情感极性，从而避免分类不准确情况的产生。在深度学习种，可以应用于情感分析的技术有很多，比如：前馈神经网络（FNN）、Word2Vec 词嵌入技术、卷积神经网络（CNN）、循环神经网络（RNN）、LSTM 网络。

经过查阅不同资料后，我将尝试使用深度学习的方法对文本进行情感分析，主要工作如下：

1) 研究循环神经网络的变种，学习比较双向 GRU 与其他模型的优点。
2) 对模型进行实践，由于硬件设备的限制等原因，使用的数据集是基于 kaggle 竞赛网站的赛题——Sentiment Analysis on Movie Reviews，是一个五分类任务。
3) 在初步实现模型后，并对神经网络模型方面尝试进行优化。

在本章的实践前，我在模型选择过程中，为了构建比较好的模型，我也收集了很多资料，对比了 RNN,单向 GRU 以及双向 GRU 这三个在自然语言处理方面表现比较好的模型：

1) RNN：最基本的循环神经网络，通过将前一个时间步的隐藏状态作为当前时间步的输入，可以捕捉到时间序列中的相关信息。然而，RNN 在处理长序列时容易出现梯度消失或梯度爆炸的问题，限制了其在长期依赖关系建模方面的能力。
2) GRU（单向）：GRU 是一种单向的循环神经网络，它按顺序处理输入序列。它具有较少的参数和计算复杂度，适用于较简单的序列建模任务。

3) 双向 GRU：双向 GRU 在处理序列数据时引入了正向和反向两个方向的隐藏状态，分别从两个方向上对输入序列进行处理。这意味着双向 GRU 可以同时捕捉到过去和未来的上下文信息，从而更好地建模序列中的依赖关系。

经过上诉对比后，对于数据集相对比较庞大的影评情感 5 分类问题（仅相对于我的笔记本电脑），双向 GRU 可以更好地理解评论中的语境和语义，能更好地捕捉上下文，从而更准确地判断情感极性，所以我选择了双向 GRU 作为我实践的模型隐含层。

## 5.2 数据挖掘技术和方法

### 5.2.1 RNN 技术

RNN 循环神经网络技术在前面章节中也有过比较全面的概述，这里就不再花费篇幅再概述一遍。这里主要概述一下 RNN 技术在文本情感上的原理与应用。

循环神经网络与普通的前馈神经网络大致相同，除了依然拥有输入层、隐含层和输出层以外，RNN 还考虑了时间信息。在 RNN 模型上，句子中的每一个单词都会被加上时间的属性。实际上，时间步长的数量在语言处理上等价于最长序列长度。


图 5.2.1-1 英文文本时间属性

循环神经网络的来源就是为了刻画一个序列当前的输出与之前信息的关系。从网络结果上来说，RNN 会记忆之前的信息，并利用之前的信息影响后面的输出。也就是说，RNN 的隐藏层之间的结点是有连接的，隐藏层的输入不仅包括输入层的输出，还包含上一时刻隐藏层的输出。

一个典型的 RNN 结构，还有循环展开结构如下图所示：

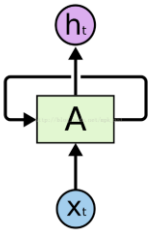


图 5.2.1-2 RNN 单元

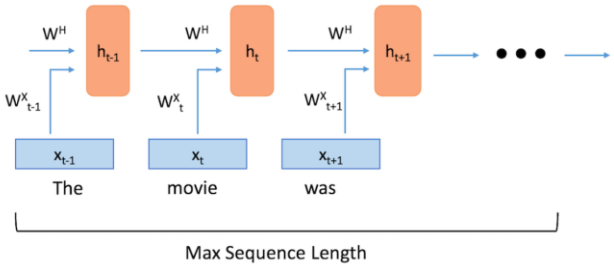


图 5.2.1-3 RNN 结构

在这之中，与每个时间步骤相关联的中间状态也被作为一个新的组件，称为隐藏状态向量 h(t) 。从抽象的角度来看，这个向量是用来封装和汇总前面时间步骤中所看到的所有信息。就像 x(t) 表示一个向量，它封装了一个特定单词的所有信息。隐藏状态是当前单词向量和前一步的隐藏状态向量的函数。并且这两项之和需要通过激活函数来进行激活，计算如下式所示：

$$h_t = \sigma(W^H h_{t-1} + W^X x_t)$$

### 5.2.2 LSTM 长短时记忆网络

在介绍 LSTM 之前，需要先对普通 RNN 的缺点进行介绍，也就是长期依赖问题（Long-Term Dependencies）问题。具体来说就是对于一个典型的 RNN 网络，隐藏状态向量对于现在的单词的存储信息量可能比前一单词的信息量会大很多。但是在一个文本的情感问题上，信息量的决定并不是

一个这样简单规律的递增。所以引入了 LSTM——长短时记忆网络。

是一种特殊的 RNN 类型，可以学习长期依赖信息。LSTM 由 Hochreiter&Schmidhuber(1997)提出，并在近期被 Alex Graves 进行了改良和推广。在很多问题，LSTM 都取得相当巨大的成功，并得到了广泛的使用。

对于 LSTM 单元，该单元根据输入数据 x(t)，隐藏层输出 h(t)。在这些单元中，h(t) 的表达形式比经典的 RNN 网络会复杂很多。这些复杂组件分为四个部分：输入门，输出门，遗忘门和一个记忆控制器。如下图所示：

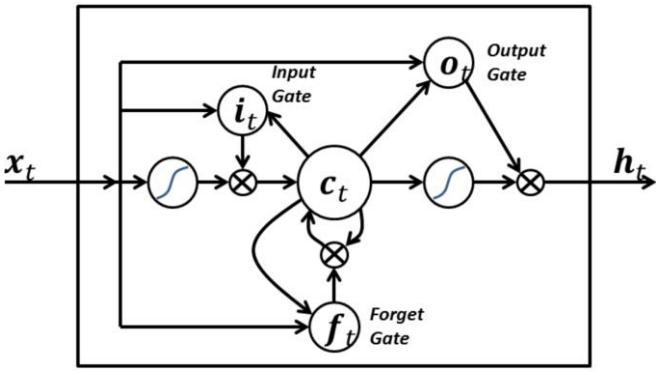


图 5.2.2-1 LSTM 单元

每个门都将 x(t)和 h(t-1)作为输入（没有在图中显示出来），并且利用这些输入来计算一些中间状态。每个中间状态都会被送入不同的管道，并且这些信息最终会汇集到 h(t)。

## 5.2.3 GRU 与双向 GRU

GRU 是 Cho 等人在 LSTM 上提出的简化版本，也是 RNN 的一种扩展，也是 LSTM 的变种，GRU 把 LSTM 中的遗忘门和输入们用更新门来替代。把 cell state 和隐状态 h(t)进行合并，在计算当前时刻新信息的方法和 LSTM 有所不同。

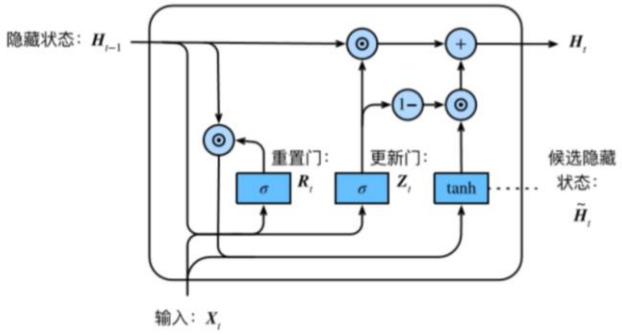


图 5.2.3-1 单向 GRU 单元

重置门（reset gate）：如果重置门关闭，会忽略掉历史信息，即历史不相干的信息不会影响未来的输出。更新门（update gate）：将 LSTM 的输入门和遗忘门合并，用于控制历史信息对当前时刻隐层输出的影响。如果更新门接近 1，会把历史信息传递下去。

简而言之，重置门有助于捕捉时间序列里短期依赖关系；更新门有助于捕捉时间序列里长期依赖关系。

双向 GRU 在处理序列数据时引入了正向和反向两个方向的隐藏状态，分别从两个方向上对输入序列进行处理。这意味着双向 GRU 可以同时捕捉到过去和未来的上下文信息，从而更好地建模序列中的依赖关系。

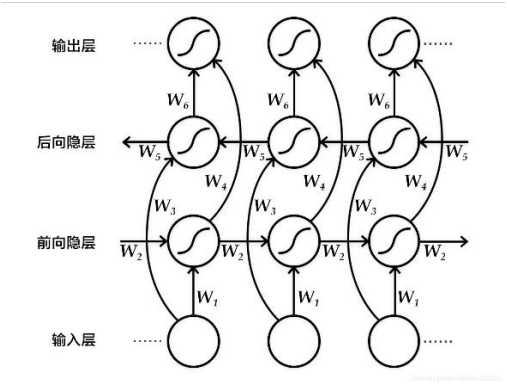


图 5.2.3-2 双向 RNN

## 5.3 主要内容和创新点

在使用双向 GRU 模型对 Kaggle 电影评论数据集进行训练的过程中，主要内容包括：数据集初步分析、数据预处理、模型搭建、模型训练与校验、模型优化调参过程。具体内容，如下流程图所示：

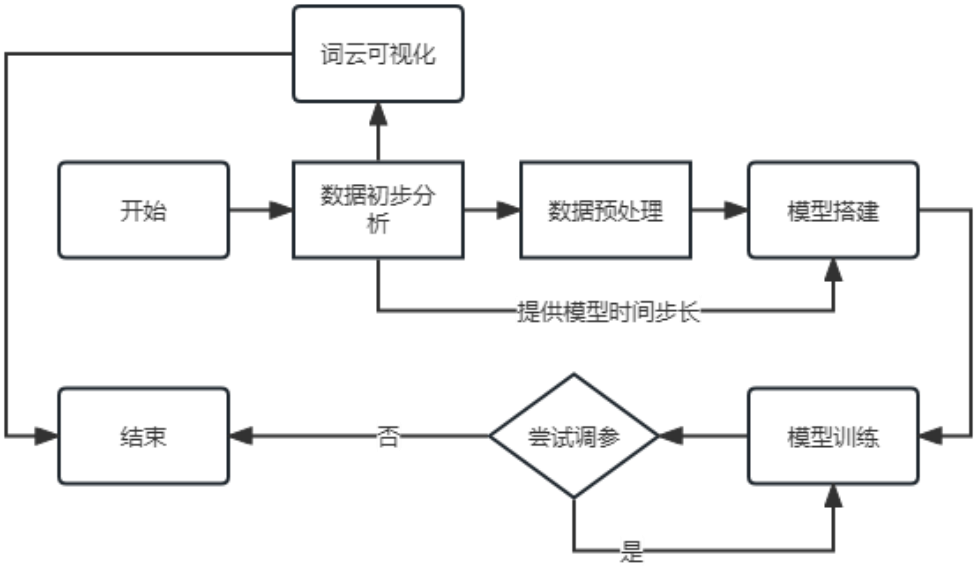


图 5.3-1 实验流程图

在模型训练过程中，我设计了如下的训练流程：

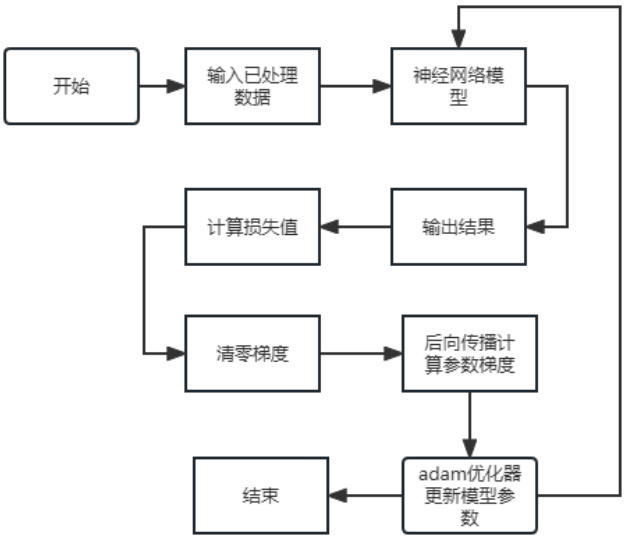


图 5.3-2 模型训练流程图

本项目的主要创新点是实现使用双向 GRU 单元的 RNN 神经网络模型的数据挖掘技术对 kaggle 竞赛数据集中的电影评论情感分析的应用创新；同时，为了更加直观的表现出模型的性能，我保存了我本次项目最后训练好的模型，设计了一个简单的可视化窗口，可以在窗口的文本输入框内输入英文文本电影评论，点击预测可以得到训练好的模型的评论分析的五分类结果，最后实现结果如下图所示：

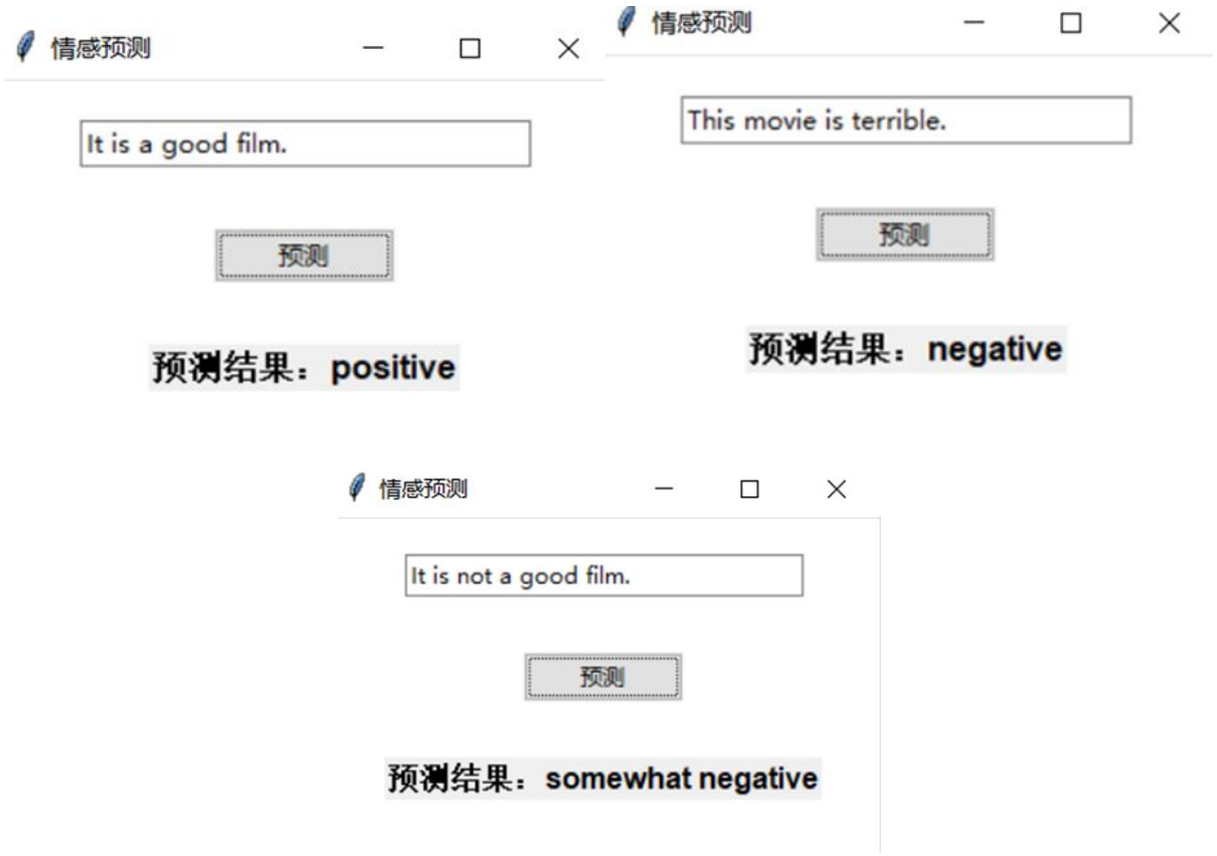


图 5.3-3 可视化窗口应用

## 5.4 实验步骤

### 5.4.1 数据集初步分析

数据集有 156060 条数据，每条数据有 4 个字段：PhraseId ,SentenceId Phrase, Sentiment, 分别代表短语 ID,评论 ID,评论内容和情感标签。情感标签中 0 表示负面，1 表示有点消极，2 表示中性，3 表示有点积极，4 表示积极。我对数据集进行了探索性数据分析：文本长度分布，不同情感类别的平均句子长度以及词云可视化。

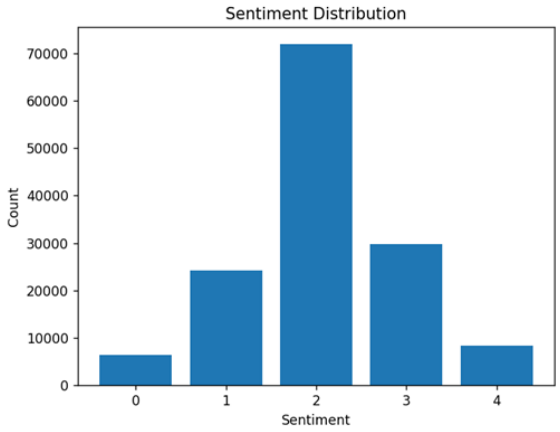


图 5.4.1-1 各情感类别样本数量柱状图

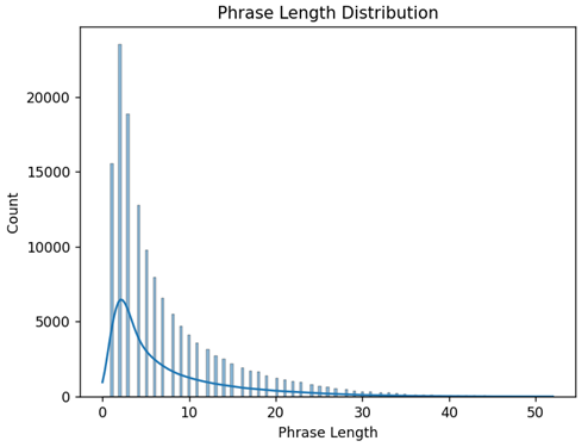


图 5.4.1-2 文本长度分类曲线及柱状图

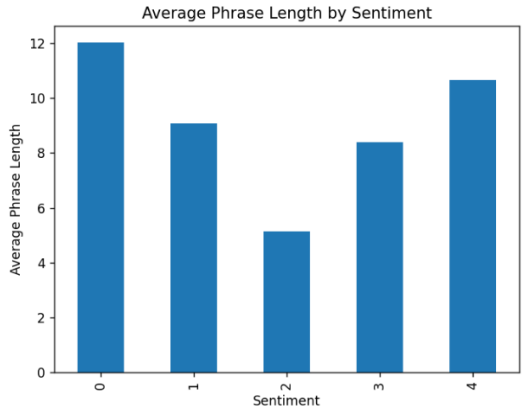


图 5.4.1-3 不同情感类别的平均句子长度

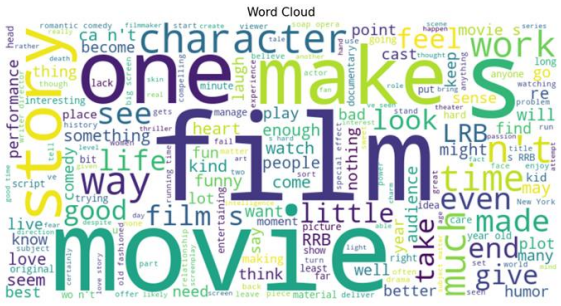


图 5.4.1-4 词云可视化

### 5.4.2 数据预处理

根据上述对数据集的分析，我将数据集按 8:1:1 的比例划分为训练集、验证集和测试集。首先，需要将文本字符信息转换为数据，输入到神经网络中进行训练。因此，我将评论内容中的所有字符转换成相应的 ASCII 码，然后将对应的字符转换成 torch 的张量使其输入到词嵌入层。由于每个数据样本的评论内容的长度不一致，我在进行转换时就将长度小于样本中最大序列长度的文本做填充 0 的处理。实现代码功能表和部分代码截图如下：(数据集中，每一个 phrase 包含多条评论)

| 函数名 | 功能 |
| --- | --- |
| **InitDataset** | 构建数据样本、分割训练集、测试集、验证集 |
| **phrase_sentiment_to_tensor** | 对数据集中的电影评论和情感标签转换成 |

| | |
|---|---|
| | PyTorch 能够处理的张量 |
| **phrase_to_list** | 由于电影评论是 string 字符串，需要将其转化成 ASCII 值表 |
| **Phrase_sentiment_to_tensor_test** | 对测试集中的电影评论转换成 PyTorch 能够处理的张量 |

数据预处理部分代码如下（phrase_sentiment_to_tensor）：

```
# 对数据集中的’电影评论‘和‘情感标签’ 转换成 PyTorch 能够处理的张量（tensor）
def phrase_sentiment_to_tensor(phrase,sentiment):
      # 电影评论字符串 -> 字符数组 -> 字符数组 对应 ASCII 码数组
      sequences_and_lengths = [phrase_to_list(item) for item in phrase]
      # 电影评论序列:每个元素就代表一个电影评论的字符数组
      phrase_sequences = [item[0] for item in sequences_and_lengths]
      # 序列长度:LongTensor 是 PyTorch 中的一种张量类型，用于存储整数数据
      sequences_length = torch.LongTensor([item[1] for item in sequences_and_lengths])
      sentiment = sentiment.long()
      # 电影评论张量 -> 张量大小 = 电影评论序列数量 * 最大序列长度 ; 张量类型 -> 整数类型
      # 填充零
      sequences_tensor =
torch.zeros(len(phrase_sequences),sequences_length.max()).long()
      # index 当前迭代的索引，sequences 是字符序列，length 是该字符序列的长度
      for index, (sequences, length) in enumerate(zip(phrase_sequences,
sequences_length)):
            # 创建一个 torch.LongTensor，其中的元素是 sequences 中的整数值（对应 ASCII 码）。
            # 在 sequences_tensor 中的当前索引行中，从列索引 0 到 length-1 的位置，将对应的字符序列赋值给该张量的切片。
            # 每次迭代都会将字符序列存储到 sequences_tensor 中相应位置的切片中。如果字符序列的长度小于 length，则会自动在后续位置补零
            sequences_tensor[index, : length] = torch.LongTensor(sequences)
      # 对 sequences_length 进行降序排序，并返回排序后的结果和排序后的索引
      sequences_length,sequences_index = sequences_length.sort(dim=0,descending=True)
      # 使用 sequences_index 对其进行重新排序，以使其与 sequences_length 的排序顺序相匹配
      sequences_tensor = sequences_tensor[sequences_index]
      sentiment = sentiment[sequences_index]
      return
initial_tensor(sequences_tensor),initial_tensor(sequences_length),initial_tensor(senti
ment)
```

## 5.4.3 模型搭建

模型是基于 Python 的 Pytorch 第三方库搭建的，我使用 torch.nn 模块中的 Embedding()方法搭建了具有 128 个输入节点且能够将输入的张量转换为 256 维的词嵌入层，GRU()方法搭建了 3 层 256 维的双向 GRU 层作为模型的隐藏层，并使用 Linear()方法构建线性层将 GRU 层的输出转换为 5 维向量作为模型的输出结果，在这之中，我设置双向 GRU 单元的激活函数为 tanh 进行实验。
具体模型图如下：

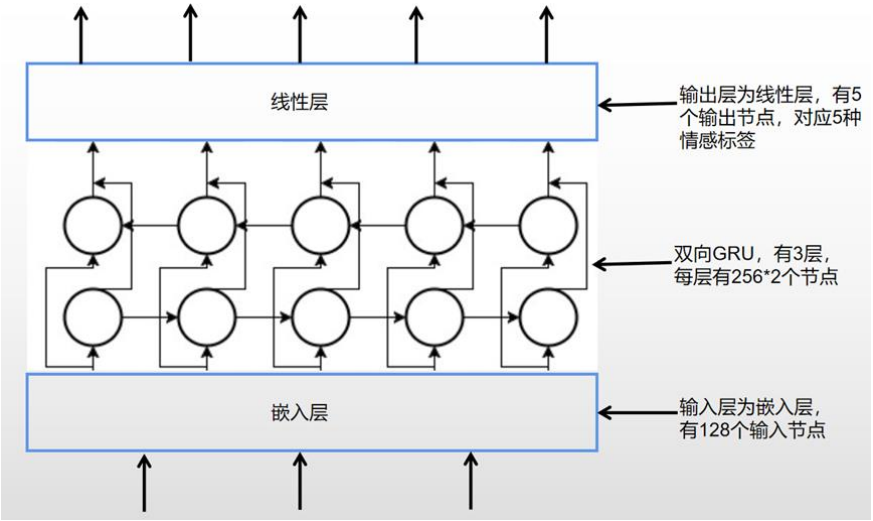


图 5.4.3-1 神经网络模型图

具体模型搭建代码部分如下：

```
class RNNClassifier(torch.nn.Module):
      # 初始化方法
      def __init__(self, input_size, hidden_size, output_size, n_layers=1,
bidirection=True):
            super(RNNClassifier, self).__init__()
            self.hidden_size = hidden_size
            self.n_layers = n_layers
            self.n_directions = 2 if bidirection else 1

            self.embedding = torch.nn.Embedding(input_size, hidden_size)
            self.gru = torch.nn.GRU(hidden_size, hidden_size, n_layers,
bidirectional=bidirection)
            self.fc = torch.nn.Linear(hidden_size * self.n_directions, output_size)

      # 初始化隐藏状态
      def _init_hidden(self, batch_size):
            hidden = torch.zeros(self.n_layers * self.n_directions, batch_size,
self.hidden_size)
            return initial_tensor(hidden)
```

```
    # 模型首先对输入进行转置，以便将时间步作为序列的第一个维度
    def forward(self, input, seq_lengths):
        # 转置 B x S -> S x B
        input = input.t()
        batch_size = input.size(1)
        hidden = self._init_hidden(batch_size)
        embedding = self.embedding(input)
        gru_input = pack_padded_sequence(embedding, seq_lengths.to('cpu'))
        output, hidden = self.gru(gru_input, hidden)
        if self.n_directions == 2:
            hidden_cat = torch.cat([hidden[-1], hidden[-2]], dim=1)
        else:
            hidden_cat = hidden[-1]
        fc_output = self.fc(hidden_cat)
        return fc_output
```

## 5.4.4 模型训练与校验

对于模型的训练，我选择了 Cross Entropy Loss 交叉熵作为损失函数，采用 Adam 优化器依据损失值和初始学习率来更新网络的权重。Adam 融合了 Momentum 优化方法和 RMSProp 优化方法，可以帮助优化算法提高精度。它还可以自动调整学习率，因此不需要太多参数调整。

对于模型的校验，训练时我保存下使得验证集准确率最高的模型。根据最初搭建的模型得出的结果，我发现在使用十折交叉校验的方法后，验证集的损失值随 Epoch 的增加一直在下降，无法反映出模型的过拟合问题(如下图所示)：

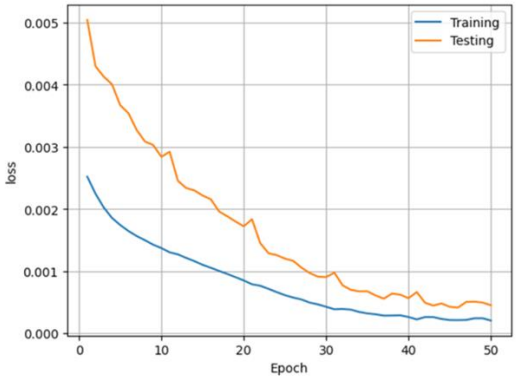


图 5.4.4-1 十折交叉验证下的训练集和验证集的损失值

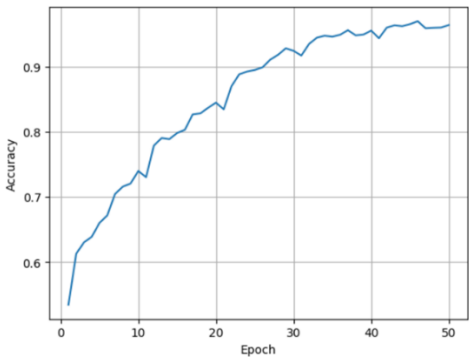


图 5.4.4-2 十折交叉验证法下的验证集准确率

验证集的准确率一直在提升，最高时达到98%，而用测试集测试得到的模型，准确率只有54.2%，与验证集的准确率相差甚远。

## 5.4.3 模型调优

为了解决上述训练出现的问题，我使用了留出法验证，每次用固定的验证集验证模型。这是验

证集的损失值和准确率变化很好地体现出了模型的过拟合问题，使我能保存下最优的模型。最终用测试集测试所得模型的准确率为 60.3%，代码上传 kaggle 运行后得分为 0.61103。

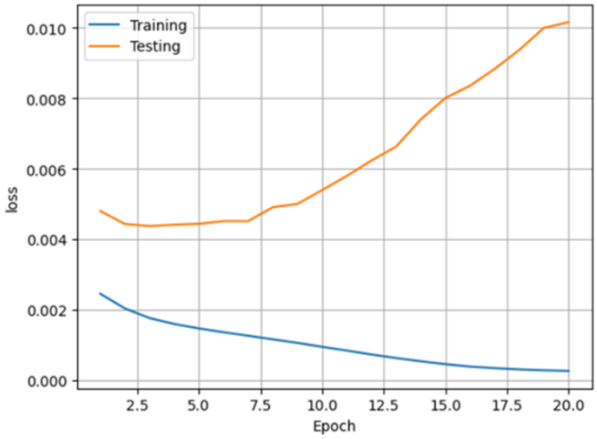


图 5.4.3-1-4 留出法验证下的训练集和验证集损失值

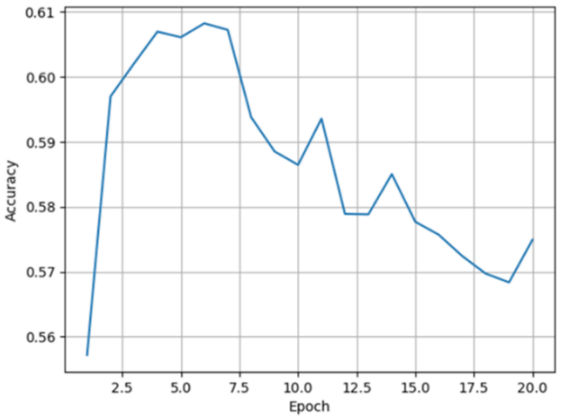


图 5.4.3-2 留出法验证下的验证集准确率

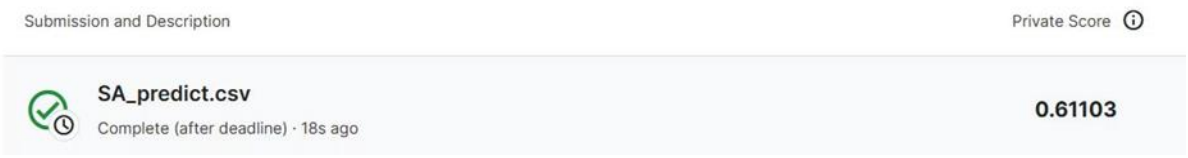


图 5.4.3-3 kaggle 评分

| # | Team | Members | Score | Entries | Last |
|---|---|---|---|---|---|
| 1 | Mark Archer | | 0.76526 | 22 | 8y |
| 2 | Armineh Nourbakhsh | | 0.76096 | 7 | 9y |
| 3 | Merlion | | 0.70936 | 9 | 9y |
| 4 | Puneet Singh | | 0.70789 | 14 | 8y |
| 5 | Yoon | | 0.68765 | 4 | 9y |
| 6 | DrStrangelove | | 0.67931 | 42 | 8y |
| 7 | akqwerty | | 0.67931 | 55 | 8y |
| 8 | MDAKMlab | | 0.67854 | 200 | 8y |
| 9 | st_sopov | | 0.67590 | 55 | 8y |

图 5.4.3-3 该竞赛排名前列参赛者得分

我除了上述调优之外，还使用其他方法来针对模型过拟合问题。主要是依据测试集的准确率来衡量模型的泛化性能，同时依据训练集和验证集的损失值曲线图以及收集到的信息和先知经验来缩小参数空间的范围。在调参过程中，我发现模型在较早的时候就出现了过拟合现象，尝试了以下的方法：

1) 采取 dropout 方法
2) 降低网络的复杂度
3) 适当减少 epoch 的次数

4) 降低学习率

但最后可能由于硬件设备限制或者其他原因，经过上述调参后的模型，在测试集准确率上，并没有明显提高。最后，我保存了最后设计的模型参数，经过一些简单的可视化，达到了最终的效果。

## 5.5 模型改善思考与未来工作

对于模型的改善思考，可以使用常见的英文文本停用词（stop word）来对文本做一个预处理，去掉自然语言文本中的噪声，不仅可以节约存储空间提高搜素效率，同时还能降低英文文本中的词语噪声对文本情感分析的干扰。在最后，由于硬件设备和时间关系，就没有继续实现。

信息技术的快速发展带来了电子商务的发展热潮和网络平台的急剧增加，对这些平台上的大量言论信息进行情感分析所得到的结果可以用于网络平台评论分类、产品分析推荐、消费预测等方面，具有极高的商业价值。

传统的文本情感分析方法耗费大量人力资源，然而人工提取的特征覆盖面有限且人工的非理性行为会影响结果的正确性，因此传统方法不具有普适性。

随着深度学习能自动提取特征、学习修正输出、可以处理非线性复杂数据等优势凸显，深度学习的方法在自然语言处理方面正受到众多学者的关注，可以预测深度学习的方法将成为文本情感分析研究的重要趋势。

在经典的循环神经网络中，状态的传输是从前往后单向的。然而，在有些问题中，当前时刻的输出不仅和之前的状态有关系，也和之后的状态相关。这时就需要双向 RNN（BiRNN）来解决这类问题。例如本项目中的评论情感分析，一个英文文本中，有可能拥有多重否定的句式，单纯的使用单向 RNN 网络可能效果没有双向 RNN 好，在双向 RNN 模型中，加入 GRU 单元不仅解决经典 RNN 的长期依赖问题，还能使用更新门、重置门去解决 RNN 的梯度消失问题；最后相比于 LSTM 长短时记忆单元，GRU 在较小的数据集中性能更好，参数更少更容易收敛。所以，在处理类似于 Kaggle 提供的这个数据集上，使用基于 RGU 的双向 RNN 循环神经网络在一定程度上具有优势，而基于 RGU 的双向 RNN 循环神经网络模型不仅在文本情感分析上具有杰出的性能，在文本预测领域也可以发光发热，相信这个模型可以应用的领域还有很多，仍待发掘。

# 第6章　　课程学习体会

互联网信息技术的快速发展，大数据时代悄然到来。大数据是是一种时代背景及数字平台，内容包含数据优化、分配与管理。数据本身潜在的信息很难实现准确的查询，这就需要采取深入挖掘或者优化数据挖掘技术的方式来应对，数据挖掘技术随之产生。各行业随着发展，都开始渗透大数据技术，大数据分析成为行业发展主流，更是当前企业打破发展瓶颈的重要手段，因此以往单一数据信息分析系统开始逐步淘汰，全新的数据挖掘技术成为往后发展的主要趋势。

通过对数据挖掘一个学期的入门学习，我也对数据挖掘有了一个全新的认识，数据挖掘是一门交叉学科，需要综合运用概率统计、机器学习、人工智能、数据可视化、算法等多门计算机领域核心知识；也是一门实践性很强的学科，需要通过大量的案例分析和实验验证来提高数据分析和解决问题的能力；还是一门紧跟技术前沿的学科，需要关注最新的业界技术和前沿研究，如大数据架构、大数据精准语义搜索、大数据语义分析挖掘、知识图谱等。

在这个课程中，我也理解并掌握了数据挖掘课程的基本思想和基本方法，包括有概念学习、归纳学习、监督学习、无监督学习、它们的基本特点、基本分类；还在韩老师的带领下，了解了与数据挖掘有关的数据查询方法、专家系统。

我在这门课程中，最觉得是意外收获的是学习了如何使用 Weka 软件，同时能够通过这个数据挖掘软件可以对许多机器学习模型、数据挖掘方法对课程带有的数据集进行实践操作。通过不断地实践，也逐步学习了基本数据挖掘技术，比如说分类中的决策树算法，并能够对决策树可视化；在聚类方法中，重点学习了 K-means 聚类，并且能够在基本的 KDD 知识发现数据挖掘流程中，使用无监督学习对有监督学习进行属性评估，从而对分类模型的属性进行降维操作，这个知识我也能在别的课程中使用，比如神经网络课程最后的模型优化上，我就使用了这个技术去优化我的模型。此外，在数学建模比赛中，感觉这个软件也可以帮助更快地对数据进行预处理与预训练。

同时，也了解到了除了数据库之外，还有数据仓库这个概念，与数据库最大的区别就在于数据仓库是具有大量的高质量数据，同时具有一定的主题域去组织数据，最特别的一点一定是数据仓库是不允许修改的。在各种企业级的决策制定计划过程中起到重要的作用，并能够充分发挥数据挖掘技术的作用，能够提供所有类型数据的战略集合。

此外，学习到如何对相关技术、算法评估也十分重要。当在处理一个问题的时候，较好模型的选择会显得十分重要，选择一个优秀的模型就离不开对模型的评估。而模型的评估方法多种多样，包括有监督学习模型评估和无监督学习模型评估。对一个特定的模型选择一项适合的评估方法这也是一个值得探究和学习的方面。

对于数据挖掘中的统计技术，涉及了不少概率论的知识点，对于我来说即熟悉又陌生。一系列的数据的均值、方法、标准差还有各种分布都是对这系列数据所挖掘出来的“新数据”。如何运用好统计学中的相关知识对处理数据进行挖掘，也是一个值得考究的知识点，不仅可以对数据进行增值，发现其规律和数据知识，还能以一个更快的效率提供对数据的一个基本描述。

而神经网络、深度神经技术则是我一直想要学习的一项知识，在这个学期中我也同样选修了神经网络这门课程，这对我深入学习这项技术提供了巨大的帮助，特别是最近 Chatgpt 的横空出世，我也对这项技术越发的好奇。这种无明显目的的编程技术对我来说有种暴力的美，通过对人类大脑的仿生，利用现代硬件的处理算力以及大量数据的训练，如同魔法般的得到一个在一些方面能够超过人类其本身能力的“大脑”。

除了有关数据挖掘的知识学习以外，这门课程也为我提供了一个能够培养自己数据科学思维的机会，能够在一些数据中发现规律与价值；大数据、人工智能也是未来世界发展的一项热门技术，在这门课程中掌握数据挖掘的相关知识，也能提高我在大数据、人工智能行业中的就业竞争力；同时，在拓展知识视野方面，为未来深入研究数据挖掘或相关邻域打下坚实的基础。

# 参考文献

[1]碧谭飞雪.大数据时代下数据挖掘技术的应用.知乎.2020.7. [https://zhuanlan.zhihu.com/p/164866738]
[2]袁柯.基于 BERT 和多粒度卷积胶囊网络的文本情感分析研究[D].华东交通大学,2021.DOI:10.27147/d.cnki.ghdju.2021.000373.
[3]李海生.融合 GRU 的循环神经网络模型在网络入侵识别中的应用[J].信息与电脑(理论版),2022,34(11):46-48.
[4]刘砚博. 基于深度学习的情感分析研究与应用[D].电子科技大学.2019.
[5]LeonG. 文本情感分析小结.知乎.2020.[ https://zhuanlan.zhihu.com/p/106588589]
[6]程艳,孙欢,陈豪迈,李猛,蔡盈盈,蔡壮. 融合卷积神经网络与双向 GRU 的文本情感分析胶囊模型. 中文信息学报. 2021, 35 (5): 118-129. [http://jcip.cipsc.org.cn/CN/Y2021/V35/I5/118]
[7]袁和金,张旭,牛为华,崔克彬. 融合注意力机制的多通道卷积与双向 GRU 模型的文本情感分析研究. 中文信息学报. 2019, 33 (10): 109-118. [http://jcip.cipsc.org.cn/CN/Y2019/V33/I10/109]
[8]梁军,柴玉梅,原慧斌,等.基于深度学习的微博情感分析. 中文信息学报,2014,28 (5):155-161. [http://jcip.cipsc.org.cn/CN/Y2014/V28/I5/155]