题目：时间序列分析和预测问题的综述

摘要：

关键词：

时间序列，神经网络，预测，多元时间序列

正文：

具体应用场景和意义

时间序列是一组按照时间发生先后顺序进行排列的数据点序列[1]。一般来讲，这些数据点序列的时间间隔为一恒定的值，如1分钟、1小时等。凡是涉及到时间数据测量的学科，如数理统计、信号处理、模式识别、计量经济学、数学金融、天气预报、地震预测、脑电图、控制工程、通信工程等学科，都有时间序列分析的应用。

时间序列预测是基于时间序列的分析而提出的一个概念，它的目标是基于现有的时间序列预测未来的某些时间段的数据值。

在零售业领域，决定订购哪些产品以及他们存储在哪里取决于对不同地区未来需求的预测；在云计算领域，对未来服务和基础设施组件使用情况的预测，将为计算资源和容量的规划提供重要参考；在工业领域，工厂的劳动力安排也很大程度依赖于对于未来工作量的预测。

时间序列的预测精准度，影响到了对于未来数据的判断，如气象预报中对于台风路径和强度的预测，将会决定某地是否采取应急措施，是否安排人员撤离。生活中随处可见的共享单车，它的投放也是参考了城市人口活动热度的时间序列来进行的。

在统计学和计量经济学领域，目前流行的预测方法是从预测单个时间点或小组时间序列的目标下逐步发展而来的，这些时间序列的分析模型是由领域专家设计和调整的。与此同时，数据挖掘和数据库研究人员，也致力于通过在成千上万个时间序列当中寻找潜在的模式，来增加模型的预测精准度。比如在预测超市提供产品的需求度时，可以利用大量相关的时间序列（不同商品的需求度）的过去数据，来对单个时间序列（某一特定商品的需求度）进行预测。使用时间相关的数据不仅可以使得在防止过拟合的情况下让模型变得更加复杂，还可以减少人工成本和传统技术的选择过程。

研究现状分析

时间序列的预测在各行各业的应用是如此频繁，以至于其方法的研究已经较为深入，除了传统的单一时间序列分析预测，还有多元时间序列的预测问题，关于多元时间序列的问题将会在文章的后续部分进行论述。

和其他的机器学习问题类似，时间序列的预测也有传统方法和深度学习的方法。传统方法包括线性回归、线性动力系统和指数平滑化、高斯过程和其他贝叶斯时间序列模型、非线性动力系统。当前流行的方法包括可伸缩张量分析和用于预测的深度学习。

以下是对传统方法和现代方法的详细描述。

**传统方法**

相似性搜索[2]

首先确定距离函数，如欧氏距离

或者是曼哈顿距离（Manhattan）以及其他 L-p 范数。在确定了距离函数之后，要进行时间扭曲（Time Warping）处理，目的是将不同时间变化率的曲线进行压缩变换，使得趋势相同的曲线能够契合的更好，以便于得到更小的距离函数计算值。距离函数还有很多种，比如linear/flat approx，cepstrum等等。

在确定距离函数之后，需要对给定的时间序列，找出序列之中和需要预测或是查询的数据段最相似的区间。比如给定当前一周的天气变化趋势，要找到历史上天气变化和本周最为相似的时间点，这个时候有两种思路，一种是靠专家人工进行筛选和查找，需要的时间成本和人力成本十分巨大。第二种是通过feature extraction的方法，将时间序列区间，比如上文给出的一周的天气变化趋势，提取出多个特征值，然后根据根据这些特征值来描述该区间的数据变化情况。

在提取好特征值之后，利用已经决定好的距离函数来对特征值组成的n维空间进行划分。最后只需要根据查询或者预测的数据段和哪个历史数据段最近，来辅助判断未来的趋势即可。

数字信号处理[3]

该方法主要有DFT和DWT两种技术。

DFT是对周期性进行突出强调的一种方法，全称是Discrete Fourier Transform。

给定一个序列，通过以下变换得到它的DFT

其中，。

通过对不同频率的拟合比较，运用amplitude spectrum方法可以找到较好拟合原时间序列的周期，这也是信号处理的标准方法。

DFT受到短时波的影响较大，比如男中音、静音、女高音的切换，这个时候可以用DWT的方法来弥补。

线性预测

通过线性预测来求未来的数据值时，是基于一个假设，即未来时间点的数据，和过去时间点的数据是线性相关的

如上所示， 时刻的数据值，是过去 个数据的线性组合。这就关系到了时间窗口 的取值以及系数的求值。

由于篇幅限制，仅介绍比较经典的最小二乘法。

最小二乘法是一种优化技术，利用数学的方法，通过最小化残差的平方来寻找与数据集最佳匹配的函数。此处有一个较为重要的概念——残差，这是观测值(数据集当中的数据)与拟合值(输入值经过模型得到的值)的差值。通过最小化残差的平方来寻找最优拟合的函数，称之为最小二乘法。

考虑最简单的一种情况，输入的维度只有一维，即训练数据是一组如的序列，其中是输入观测值，是输出观测值。我们希望得到一个线性模型，有如下的表达式

运用残差平方和来衡量模型的拟合程度，即求得一组使得残差平方和最小

记为当系数为时模型的残差平方和，为了使得最小，分别对和求偏导，得到

函数的极值点即为偏导为的点，令式和为零，可以得到使得最小的解

其中，，是的平均值。

上述考虑的情况是一元线性模型的最小二乘法，更为普遍的情况是多元的线性模型，即一开始提到的多维向量，以及对应的。这时候模型为

这时候的回归即为多元线性回归，同样我们可以利用最小二乘法求得最适合的。为了更方便的表示，将组合成向量形式，同样的，我们将数据集表示为一个矩阵，其中为数据的个数，为数据的维度，每行相当于一个观测值，前个数据为属性值，最后一个元素恒为，同理将组合成维度为的向量。效果如下

于是模型就变为了

其中

那么现在就需要求得最优的，类似于之前的式，有

此时的残差平方和，对进行求导，得到

和一元线性回归的方法一样，此时令式等于，可以得到的最优解，但由于上述公式中有矩阵逆的计算，所以比一元变量的情况复杂，此时需要分两种情况讨论。

第一种情况是为满秩矩阵(full-rank matrix)或者是正定矩阵(positive definite matrix)。此时令式等于，得到

其中是矩阵的逆矩阵。

第二种情况则是更为普遍的一般情况，此时不是满秩矩阵。这个时候可以解出多个 使得残差平方和最小，这个时候选择哪个将由具体的算法决定，常见的作法是引入正则化(regularization)[4]。

其他的方法还包括了非线性的预测方法还有张量预测方法，在此不作赘述。

现代预测方法

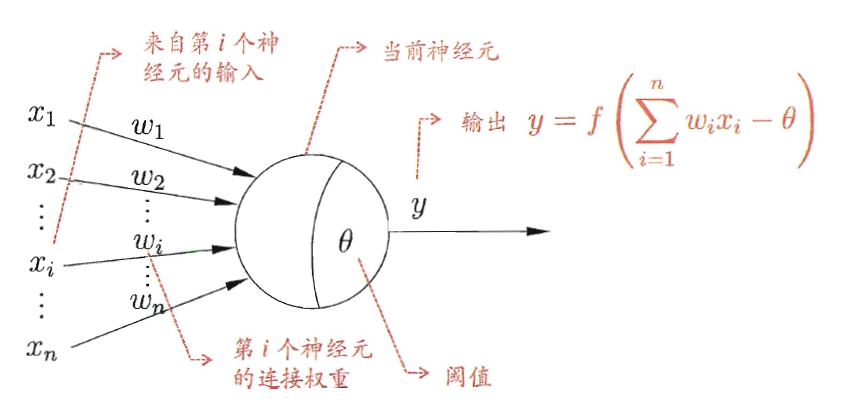
现代的预测方法主要集中在利用深度学习神经网络的思想进行预测，这里简要减少循环神经网络的基本概念。

本文讲到的神经网络，不涉及生物领域，即非生理构造，而是完全由人类设计的数学模型，成为人工神经网络(Artificial Neural Network，简称ANN)。人工神经网络并非是凭空产生，是人类仿造生物神经网络的构造构建出的数学模型，其中每个运算节点类似于生物的神经元。

本文由于篇幅限制，仅仅介绍神经网络的基本要素和循环神经网络的具体模型。

**1） 神经元**

ANN当中的神经元，也叫做神经节点，类比人类神经元特性，是由多个输入、处理函数、多个输出组成的计算节点。如下图2.2所示

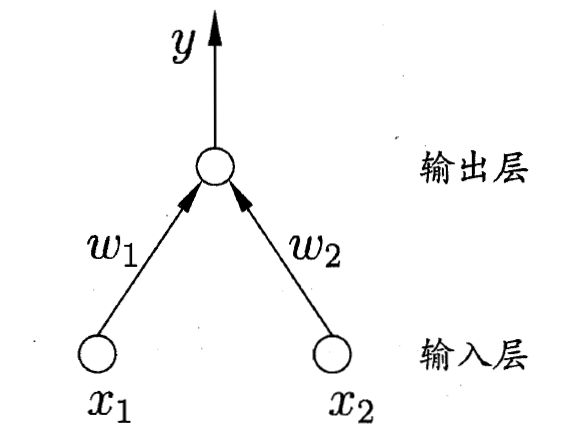
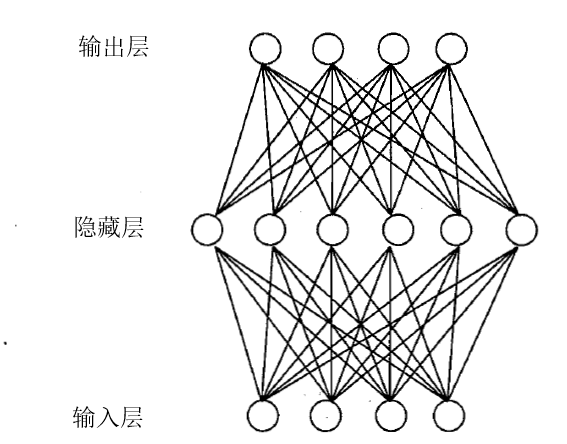


**图 2.2 人工神经元**

该神经节点接收到的多个输入可能是来自之前的神经节点的输出，也有可能是来自数据集的输入，这里的关于的函数 就是上文提到的处理函数，最后得到的可能是一维的，也有可能是多维的，主要看之后连接到该节点的神经节点个数以及该节点在网络当中所处的位置。

**2） 感知机与多层神经网络**

由两层神经节点组成的网络成为感知机(Proceptron)，如下图2.3。输入层接受数据集的输入，传递给输出层的神经节点，经过处理函数之后输出。

**图 2.3 输入神经元为2的感知机 图 2.4 隐藏层数为1的多层神经网络**

感知机的结构相当的简单，且只有输出层的神经节点有处理函数，所以能够解决的问题很有限，基本只能解决线性可分的问题。

更加常见的神经网络大多都含有多个隐藏层(Hidden Layer)， 如上图2.4。每个隐藏层当中含有多个神经节点，这样的神经网络能够较好的进行数据集的训练和拟合，但是相对的由于节点数的增加，训练的收敛速度会减慢。

**3） 前向传播与****反向传播**

前向传播和反向传播是神经网络运作的两个主要的算法，分别对应了由输入计算得到模型的输出和由数据集的数据修正模型的各个参数矩阵。

**（一）前向传播**

上文在介绍神经元的相关信息的时候讲到，神经元的输入和输出的对应关系可以由以下表达式来表示

这里的是单个神经元的输出，由于一层当中会由多个神经元，那么每个神经元都有各自的和，所以对于第层来讲，所有神经元的系数可以合并为矩阵和。有如下的式子

其中为当前层的输入，为这一层在处理函数之后的值，通过激活函数之后成为下一层神经元的输入

以上的运算过程在每一层进行迭代，最后可以得到输出层的结果，即模型接受输入之后计算所得的预测值。

前向传播算法就是上述过程，即利用每一层的系数矩阵和偏移量进行线性计算，再通过激活函数进行激活运算，从输入层一直到输出层，层层计算最后输出结果的过程。

**（二）反向传播**

反向传播算法的目的是通过训练集当中的观测值输出和模型输出来进行参数修正。在建立模型的时候，往往是根据一定的随机算法初始化参数，反向传播算法能够根据训练集当中的数据进行参数修正，最终找到合适的系数。

反向传播需要一个合适的损失函数，损失函数的输入为训练样本的输出和模型的输出，通过对该函数进行优化求最小化的极值，对应的系数矩阵和偏移量就是我们需要的最终结果。

对神经网络的损失函数运用“梯度下降法”进行迭代求极小值的过程就是反向传播算法。

以均方误差(Mean Squared Error)为损失函数举例。此时损失函数的表达式为

其中为输出层的输出，||S||2为S的L2范数，L2范数是指对向量各元素的平方和求平方根。

有了损失函数之后，就可以用梯度下降法(Gradient descent)来求解每一层的系数矩阵和偏移量。通过每一层的损失函数，分别对系数矩阵和偏移量求偏导，具体步骤和线性回归的方法类似。

之所以称之为反向传播算法，是因为算法是从最后一层的系数开始，逐渐往前计算损失函数的偏导，从后往前进行参数的修正。在进行修正的时候，需要设置一个学习率，每次更新参数的时候都要乘以该学习率，合适的能够较好的提升模型的训练质量。

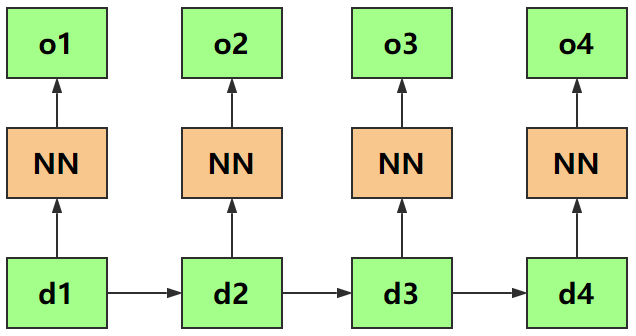
反向传播算法就是从最后一层开始计算每层的损失函数，通过对系数矩阵和偏移量求偏导，令偏导为0来最小化损失函数，从而更新系数矩阵的过程。

**2.2.2 SimpleRNN**

RNN是循环神经网络(Recurrent Neural Network)[17]的缩写，RNN的精髓在于“循环”两个字。之所以称之为循环，是因为RNN的预测是和数据的时间先后有关系的，和传统的神经网络不同，RNN能够对输入序列产生一定的记忆性，因此RNN的输出和输入的先后有潜在的联系。

假定有一组输入数据，对应的输出为。

将这组输入输出带进神经网络的应用场景，如下图2.5所示。



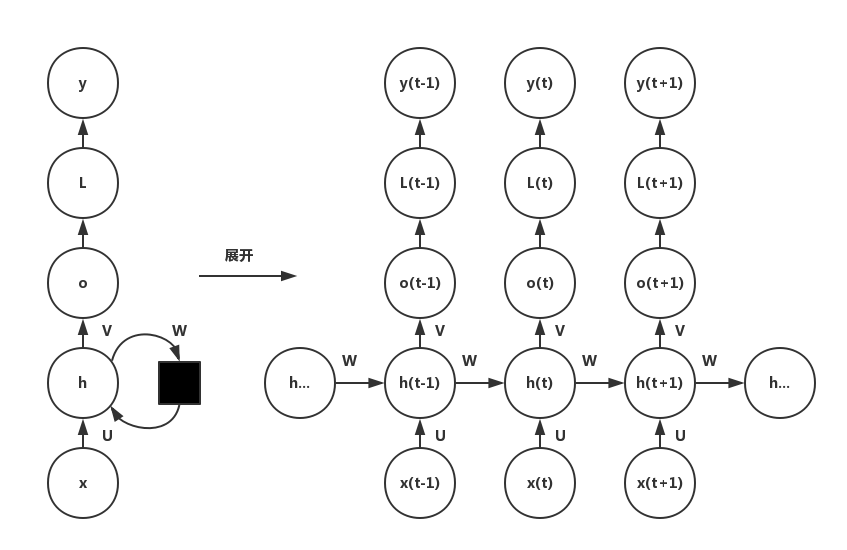
**图 2.5 普通神经网络应用示例**

可以看到，在进行预测的时候，普通神经网络模型仅仅基于当前的对应的数据，在不同数据之间毫无关系的时候，比如温度和气压的关系这种问题上面，能够较好的进行预测，但是在数据集是有顺序的时候，比如烧饭做菜，掌控调料和火候的顺序是固定的，此时普通的神经网络并不能较好的理解隐含在数据集当中的这种关联。这个时候就需要RNN的帮助。

RNN的基本思路就是记住之前发生事情，比如在预测的时候，不仅仅参考还参考了上一次的数据集和 。因而利用这种思想，设计出了如下图2.6的RNN模型。

图2.6中左边是RNN模型没有展开的状态，右侧是RNN模型在时刻展开的状态。其中为训练样本的输入，为隐藏状态，分别代表了模型输出、损失函数和训练样本的真实输出。为线性关系的参数，在整个网络当中是共享的。

这里比较重要的是RNN引入了隐藏状态的概念，隐藏状态是由隐藏状态和训练样本共同决定的，这样一来RNN就能够对潜在的时间先后关系进行记忆。同时，共享的线性关系参数也体现了RNN模型“循环”的思想。



**图 2.6 主流RNN模型**

在理论上，RNN可以解决很多问题，类似分类、回归，只要由先后时序关系，就可以进行模拟和预测，然而不可避免的是简单的RNN只能记住较为近的信息，无法进行中长期的信息记忆，所以后续引入了LSTM和GRU。

技术挑战分析和解决方案

由于单变量的传统预测模型和随机森林等机器学习方法不能够灵活调整、扩张和添加外部变量，因此有paper通过引入RNN来解决Uber数据的预测问题[5]。

RNN的长时间特性提取不好，针对这个问题，时间关注模型被提出来[6]用来提高RNN在长时间依赖上的表现能力。

除了预测问题，RNN还被用来进行异常值检测[7]，目的是为了克服多变量时间序列当中存在的随机性和时间依赖复杂性。

RNN还被用来做缺失值（missing data）的填补[8]。

除了RNN以外，也有开发新型神经网络来进行时间序列当中频率信息的提取[9]，还有对各类方法在时间序列分类上的比较[10]。

在笔者本科阶段，曾参加过一个涉及时间序列分析的项目。

系统设计与实现

参考文献：

1. Time series, Wikipedia, [*https://en.wikipedia.org/wiki/Time\_series*](https://en.wikipedia.org/wiki/Time_series).
2. Lin, Rake& Agrawal King-lp, and Harpreet S. Sawhney Kyuseok Shim. "Fast similarity search in the presence of noise, scaling, and translation in time-series databases." *Proceeding of the 21th International Conference on Very Large Data Bases*. 1995.
3. Faloutsos, Christos, et al. "Forecasting Big Time Series: Theory and Practice." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019.
4. Scholkopf, Bernhard, and Alexander J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
5. Laptev, Nikolay, et al. "Time-series extreme event forecasting with neural networks at uber." *International Conference on Machine Learning*. Vol. 34. 2017.
6. Shih, Shun-Yao, Fan-Keng Sun, and Hung-yi Lee. "Temporal pattern attention for multivariate time series forecasting." *Machine Learning* 108.8-9 (2019): 1421-1441.
7. Su, Ya, et al. "Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019.
8. Lipton, Zachary C., David C. Kale, and Randall Wetzel. "Modeling missing data in clinical time series with rnns." *arXiv preprint arXiv:1606.04130* (2016).
9. Wang, Jingyuan, et al. "Multilevel wavelet decomposition network for interpretable time series analysis." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018.
10. Lines, Jason, Sarah Taylor, and Anthony Bagnall. "Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification." *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016.