分类号 C8/255　　密级
UDC　　编号 10741

兰州财经大学
LANZHOU UNIVERSITY OF FINANCE AND ECONOMICS

硕士学位论文

论文题目 GLM与神经网络的集成模型及其应用

研究生姓名：李仁祥

指导教师姓名、职称：孟生旺、教授

学科、专业名称：统计学、数理统计学

研究方向：保险与精算

提交日期：2021年6月6日

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：李仁祥 签字日期：2021.6.1

导师签名：[signature] 签字日期：2021.6.1

关于论文使用授权的说明

本人完全了解学校关于保留、使用学位论文的各项规定，同意（选择“同意”/“不同意”）以下事项：

1.学校有权保留本论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文；

2.学校有权将本人的学位论文提交至清华大学“中国学术期刊（光盘版）电子杂志社”用于出版和编入CNKI《中国知识资源总库》或其他同类数据库，传播本学位论文的全部或部分内容。

学位论文作者签名：李仁祥 签字日期：2021.6.1

导师签名：[signature] 签字日期：2021.6.1

Integrated Model of GLM and Neural Network and Its Application

Candidate : Li Ren Xiang

Supervisor: Meng Sheng Wang

摘要

广义线性模型是非寿险精算的标准模型，广泛应用于非寿险费率厘定和准备金评估。在索赔频率预测中，常用的广义线性模型有泊松回归模型和负二项回归。虽然广义线性模型有着很好的模型解释力，但是其并不能很好的反映数据之间的复杂关系。而神经网络模型有比较好的模型效果，可以很好的揭示数据之间的复杂关系。本文的主要目标是将 *GLM*(*Generalize Linear Model*)与神经网络模型结合，获得更好预测效果的同时也使模型具有解释力。

本文先介绍多个索赔频率模型并针对模型变量选择问题建立 *Lasso-GAMLSS*(*Generalized Additive Models For Location,Scale And Shape*)模型，然后将多个深度学习神经网络引入到非寿险精算中，基于实际车险数据的分析结果表明，神经网络的模型效果要优于传统索赔频率的模型效果。

基于上述结果，将广义线性模型与神经网络模型结合，引入 *CANN*（*Combined actuarial neural network*）模型。因为 *GAM*(*Generalized Additive Models*)比广义线性模型有着更好的拟合效果，所以将其扩展为广义可加模型的 *CANN*。又因为 *CNN*(*Convolution Neural Network*)模型能够自动选择变量，在变量多的情形下有好的模型效果，所以将其扩展为 *CNN* 的 *CANN* 模型。

为了提高 *CANN* 模型的解释力，本文提出了基于深度学习神经网络修正 *GLM* 残差的集成模型。先从理论的角度进行研究，说明相关集成模型的优点，接着建立集成模型。首先利用广义线性模型对损失数据进行初步预测，计算预测值与观测值的残差。将多个神经网络模型对损失数据进行预测，选择结果相对较好的几个模型以及模型的最优参数。以损失影响因素为自变量以及残差为因变量，建立改进 *BP*(*Back Propagation*)、*DNN*(*Deep Neural Network*)和 *CNN* 模型进行残差拟合，接着使用深度神经网络的残差预测值对广义线性模型的预测值进行修正就得到了预测结果。基于集成的思想，从残差修正模型中选择合适的模型建立两种集成模型。第一种集成方法是对残差修正模型使用 DNN 模型进行集成，第二种集成方法是对预测模型使用线性回归模型进行集成。通过实证研究发现，残差修正模型比广义线性模型的效果好，同时与神经网络模型的结果相差不大，集成模型的效果比单个残差修正模型的效果好，第二种集成方法比第一种集成方法效果更好。由此得到的集成模型在提高车险索赔频率预测效果的同时还保留了传统 *GLM* 模

型的解释力。

关键词：深度神经网络　广义线性模型　残差修正 集成模型

Abstract

Generalized linear model is the standard model of non life insurance actuarial, which is widely used in non life insurance rate determination and reserve evaluation. In the prediction of claim frequency, Poisson regression model and negative binomial regression model are commonly used. Although the generalized linear model has a good explanatory power, it can not well reflect the complex relationship between the data. The neural network model has a good model effect, which can well reveal the complex relationship between the data. The main goal of this paper is to combine GLM (generalized linear model) with neural network model to obtain better prediction effect and make the model have explanatory power at the same time.

This paper first introduces multiple claim frequency models and establishes Lasso-Gamlss (generalized additional models for location, scale and shape) model for the problem of model variable selection. Then it introduces multiple deep learning neural networks into non life insurance actuarial. The analysis results based on actual vehicle insurance data show that the effect of neural network model is better than that of traditional claim frequency model.

Based on the above results, combined with the generalized linear model and neural network model, CANN(combined actual neural network) model is introduced. Because GAM (generalized additive models) has

better fitting effect than generalized linear model, it is extended to CANN of generalized additive model. Because CNN (revolution neural network) model can automatically select variables, it has good model effect in the case of many variables, so it is extended to CANN of CNN model.

In order to improve the explanatory power of CANN model, this paper proposes an integrated model based on deep learning neural network to modify GLM residuals. Firstly, it studies from the perspective of theory, explains the advantages of the relevant integration model, and then establishes the integration model. Firstly, the generalized linear model is used to predict the loss data, and the residual between the predicted value and the observed value is calculated. Several neural network models are used to predict the loss data, and several models with relatively good results and the optimal parameters of the model are selected. Taking loss influencing factors as independent variables and residuals as dependent variables, the improved BP (back propagation), DNN (deep neural network) and CNN models are established to fit the residuals, and then the prediction value of the generalized linear model is modified by the residual prediction value of the deep neural network to obtain the prediction result. Based on the idea of integration, two kinds of integration models are established by selecting the appropriate model from the residual correction model. The first method is to integrate the residual correction model with DNN model, and the second method is to integrate the prediction model with linear

regression model. Through the empirical study, it is found that the residual correction model is better than the generalized linear model, and the result is similar to that of the neural network model. The effect of the integrated model is better than that of the single residual correction model, and the effect of the second integrated method is better than that of the first integrated method. The results show that the integrated model can improve the prediction effect of vehicle insurance claim frequency while retaining the explanatory power of traditional GLM model.

Key words: Deep Neural Network; Generalized Linear Model; Residual Correction; Integration Model

目录

1 引言

1.1 研究背景

保险被称作是社会的“助推器”和“稳定器”，不管是发展中的国家还是发达国家，都非常关注保险行业的稳定发展。而车险的保费是产险保费中最重要的一个模块。从1980年中国车险开始恢复经营，通过一段时间的发展，车险的保费在财产保险公司的保费中占举足轻重的地位。因为驾驶员参保的认识越来越强以及全国的机动车持有量越来越多，近几年车险保费的数据表现很出色，车险保费占比甚至超过70%，这足以见得对于财产保险业务而言，车险经营状况显得尤为重要。

我国车险保费收入的趋势呈现为逐年上升。公开数据显示，人保财险近三年的车险保费收入增速为1.6%、10.5%、3.9%，平安产险的同期增速分别为14.8%、6.6%和6.9%，太保产险的同期增速则为7.4%、7.5%和6%。可以看出，虽然2019年的保费增长的速度有所放缓，但是，总体的车险保费还是处于上升状态的。对于2019年车险保费收入增速慢慢放缓，有多方面的因素，像汽车逐步成为生活必须品以及国产车企不断升级更新产品，汽车的价格在下降。由此车险保费的增长也自然受限。这就要求保险公司有自己更好的产品，这样才能在市场越来越竞争的当下提升竞争力。

我国的车险行业起步较晚，当前我国车险行业存在的问题有竞争不正当、制度不全面等。从市场上车险费率厘定的表现上来看，费率厘定不能很好地贴合实际情况这个现象还是存在。但是，这几年里我国在车险行业中不断探索和改革，把车险费率改革成为市场化的保费。2020年9月7号，中国精算师协会发布了商业车险的基准纯风险保费表。与此同时，中国银保监会发布了《示范型商车险精算规定》。由此可见，建立更加符合实际情况、更加精确的车险定价模型是有一定的现实意义，这同样是促进我国车险行业平稳增长的重要保证。

1.2 研究的意义

在保险精算这个领域，学者们通过不断研究与应用，发现神经网络模型对比*GLM*来说有着更好的模型预测效果，解决了广义线性模型拟合能力不强的缺点。但是，对于保险行业来说，目前使用最多的车险模型还是广义线性模型。因为，

虽然广义线性模型相比于神经网络来说拟合效果相对差些，但是，广义线性模型有很强的模型解释力，而神经网络模型因为类似黑箱的结构导致其没有模型解释力，这在保险公司的产品定价中处于劣势。

因此本人的研究意义表现在：传统的车险定价模型多以 *GLM* 为主，*GLM* 模型在预测中的误差总是存在的。所以，使用神经网络模型来拟合残差序列，然后将残差序列的预测值加到 *GLM* 模型的预测值上就可以减小预测误差。由于误差的大小直接关系到模型的优劣，因此，通过矫正误差来提高 *GLM* 模型的预测精度是一种新的思路。同时，集成预测模型的效果也比单个预测模型的效果好，选择效果好的残差修正模型进行集成，应用到车险费率厘定中去，提高广义线性模型精度的同时还保留了模型的解释力，这有助于提升保险公司车险类保险产品的定价能力，提升产品优势，帮助保险公司针对不同的消费者提供个性化服务。

1.3 国内外研究的现状

1.3.1 国外研究现状

（1）广义线性模型在车险中的研究

Nelder 和 *Wedderburn*(1972)[1]介绍了广义线性模型，并且对于广义线性模型的模型构成和相关理论之知识都有详细介绍。以此为基础 *McCullagh* 和 *Nelder*(1989)[2]在非寿险精算中使用 *GLM*，使用真实的车险损失数据建模进行验证，并且还对轮船风浪保险定价建立广义线性模型，以此来阐述广义线性模型的实用性。从此之后，不少精算领域的学者开始研究广义线性模型，并对其进行完善和改进。对于车险索赔额和索赔频率这两方面的讨论一直都在进行。*Andrade*(1989)[3]对车险数据的索赔频率建立了广义线性模型，以此进行了定量分析。*Ohlsson* 和 *Johansson*(2010)[4]使用一组真实数据建立了 *GLM*，以此进行了相关的实证分析，实证结果表明 *GLM* 拟合效果好。*Garrid* 等(2016)[5]通过将广义线性模型拟合总索赔成本的边际频率和条件强度分量，并将索赔次数视为平均索赔金额的协变量，从而得出两者之间的相关性。这种模型除了易于实施外，还具有产生的纯保费是边际平均频率和修改后的边际平均强度的乘积这一优势。由此可见，在车险费率厘定的研究中广义线性模型有着极其广泛的使用。

（2）神经网络模型在保险中的研究

Brockett(1998)[6]通过建立 *Kohonen* 自组织竞争神经网络模型，来解决保险

理赔中人身伤害险的欺诈问题。*Guelman*(2012)[7]先对一组实际数据先进行交叉验证和欠抽样的处理，以此来降低数据不均衡带来的问题，然后通过对索赔强度和索赔频率分别建立梯度提升树 *GBT* 模型，最后得出的结果显示出， *GBT* 模型在保险数据的索赔频率和索赔强度预测方面，相对于 *GLM* 模型也是一种很好的选择。*Liu* 等(2014)[8]使用多分类 *AdaBoost* 树预测模型对保险数据中的索赔强度建立，并将其与两层的 *BP* 神经网络、*SVM*(*Support Vector Machine*)和广义线性模型进行相互比较，得出的结论是 多分类 *AdaBoost* 树模型在预测结果方面具有相对较小的方差和较好的预测精度。因为车险数据的特征都是发生频率低并且不均衡，同时在保险的费率厘定中，风险因子相互之间往往都是相乘关系，由此得出 *GBT* 并没有完全符合建立车险损失数据的预测建模。基于以上这个问题，*Lee* 等(2015)[9]提出了一种改进的预测模型，就是 *Delta Boosting* 预测模型。通过使用一组实际数据进行建模，对比模型的结果发现，在实际数据的应用中，与广义线性模型和梯度提升树模型 *GBT* 建立的预测结果相比，使用 *Delta Boosting* 方法建立的模型预测效果更好。之后，*Lee* 和 *Antonio*(2015)[10]还研究比较了广义可加模型、广义线性模型、决策树和神经网络等预测模型在进行索赔频率建模的效果，最后，得出的结果是神经网络预测模型虽然其预测结果具有较好的预测精度,但是，该模型在数据尾部有过拟合的问题。*Mzhavia*(2016)[11]使用神经网络预测模型来进行汽车驾驶人数据的风险分类，通过建模分析得出当选取输入层为 11 个神经元，输出层为 2 个神经元，激活函数是双曲正切函数，12 个神经元单隐藏层的后向传播神经网络模型时候，模型有很好的分类结果。*Xia* 等(2017)[12]建立了一种基于极端梯度提升的集成信用评分模型。实验结果表明:贝叶斯超参数优化方法优于随机搜索、网格搜索和人工搜索。此外，该模型在准确度、误差率、曲线下面积(*AUC*)测量和 *Brier* 评分等四个评价指标上均优于基线模型。*Wüthrich*(2018)[13]使用车联网保存的数据，运用 *Bottleneck* 神经网络学习算法，选择出驾驶行为当作一个变量,对索赔频率建立了服从泊松分布的广义可加模型。*Noll* 等(2018)[14]通过使用 *GLM*、神经网络、提升法和回归树等模型，对一组真实车险数据建立索赔频率预测模型,对这几个模型的结果和预测值进行对比分析，结果表 *GLM* 不能合适地解决特征变量之间的交互作用，而其他模型能够较好地处理这些交互影响。*Andrea Ferrario*(2018)[15]提供了一个教程，说明了在将深度

学习中的神经网络回归模型用于保险索赔频率数据时需要考虑的方面。*Richman Ronald*(2020)[17]建立了 Tweedie 复合泊松模型与神经网络模型的组合，并对组合模型进行集成，最后通过一组实际数据表明集成模型的效果好。

（3）国外相关文献评述

从国外广义线性模型的整体发展来看，开始 *GLM* 的出现和普及，这个阶段出现了广义线性模型并且不断的完善其理论基础。接着国外学者在对广义线性模型进行深入研究后，发现了其有些假设存在一定的缺陷，于是为了解决这些问题，对广义线性模型进行了扩展研究。神经网络模型等机器学习被提出并且受到了广泛的关注，学者们将神经网络等机器学习应用到车险领域中来。这个阶段学者们开始先将神经网络等机器学习应用到车险领域中，之后，比对广义线性模型与神经网络模型等机器学习模型的结果，发现神经网络等机器学习模型更加准确，并对其不断的研究。

1.3.2 国内研究现状

（1）广义线性模型在车险中的研究

当前国内保险公司对于车险费率厘定的方法也主要是广义线性模型。陈希孺(2002)[21]将广义线性模型正式引入国内车险的定价模型，并且详细阐述了广义线性模型理论。从此以后，国内学者开始围绕广义线性模型展开了众多的讨论。毛泽春和刘锦萼(2002)[22]使用广义线性模型对保险公司的一组真实数据进行了实证研究；孟生旺(2007)[23]研究了广义线性模型在精算领域的应用并且进行了分析，还通过实际数据分析把广义线性模型引入到车险费率的定价中；卢志义和刘乐平(2007)[24]对广义线性模型在精算领域的实际使用进行了综述；钟桢 (2010)[25]对一组车险损失数据建立了广义线性模型并对其进行了实证研究，还进行对比了损失强度的分布；张连增(2013)[26]实证分析了基于一组实际的车险损失数据建立的广义线性车险费率模型。孙维伟(2014)[27]使用真实车险数据建立 *GLM-Tweedie* 与 *GAM-Tweedie* 模型，对车险数据进行索赔额拟合并进行费率厘定。张连增(2017)[28]通过对国内外两组数据进行的实证分析，对比纯保费建模的两种方法在处理车险费率厘定问题时候的区别。

（2）神经网络在保险中的研究

随着机器学习算法的不断完善和推广，其在各种领域都得到一定的发展。保

险领域的学者也不断的研究如何使用机器学习建立符合保险数据的预测模型。傅鸿源等人(2008)[29]在工程保险费率确定中引入 *RBF* 神经网络模型。叶明华(2011)[30]使用神经网络建立了保险欺诈识别模型，基于车险数据，检验了神经网络模型在保险欺诈识别中的有效性。孟生旺(2012)[31]基于实际数据，在索赔频率方面得出神经网络的结果比 *GLM* 的结果要好。孟生旺等 (2017)[32]使用机器学习的方法对车险索赔频率和索赔金额进行建模预测，并与 *logistic* 广义线性回归模型和伽马广义线性回归模型进行对比，并为机器学习算法提出了一种新的保险损失建模方法。张连增等(2018)[33]使用机器学习中的回归树方法对车险的索赔频率进行预测建模。张连增等(2018)[34]在汽车保险的索赔预测中使用 SOM 神经网络对多个解释变量进行聚类得到综合变量，以此来减少解释变量，增加广义线性模型的拟合效果。张连增等(2019)[35]将 *Boosting* 算法与回归树结合的模型用于车险索赔频率的建模。孟生旺等(2019)[36]对案件的赔付状态、赔付金额使用随机森林和 *XGboost* 等机器学习算法进行建模，提升了准备金评估模型的准确度。张连增等(2019)[37]将提升算法分别与回归树模型和广义线性模型结合的模型用于车险索赔频率的建模。张碧怡等(2019)[38]利用随机森林和 *XGboost* 探讨了索赔频率风险因子重要性。

（3）基于神经网络的残差修正与集成

基于神经网络的残差修正方法，在工程方面使用较多。叶明全(2005)[39]通过季节性神经网络模型对 GM 的残差序列进行分析，提取其中的非线性成分作为预测时候的补偿项目，以进行残差修正，形成 *GMSANN* 叠合预测模型。王谷等(2006)[40] 将 *GM* 模型与神经网络模型集成，使用集成模型对交通量进行预测，结果表明集成模型的预测效果要优于单个模型的预测结果，能够弥补单个模型存在的缺点。刘玉兵(2007)[41]将 *GM* 模型与神经网络模型集成，使用集成模型对油液光谱分析参数进行预测，结果表明集成模型的预测效果要优于单个模型的预测结果，能够弥补单个模型存在的缺点。李艳昌(2007)[42]把基于神经网络修正的残差模型引入电力负荷的预测中。光辉(2008)[43]通过工程实例验证，基于神经网络的 *GM*（1，1）预测模型残差修正方法能有效提高预测精度，在工程中有较好的应用。孙金玲(2015)[44]对灰色神经网络进行残差修正，通过数据分析表明，残差修正模型相比灰色神经网络模型具有比较好的模型效果。毕建武(2015)[45]基于实

际瓦斯释放量的数据，先建立多元线性模型，然后使用 *RBF* 神经网络对残差序列拟合预测，最后利用 *RBF* 残差的预测结果对瓦斯涌出量预测结果进行补偿修正。得出的结果表明这种组合预测模型的预测效果比较好。陈卓(2018)[46]基于电力负荷数据，先建立深度神经网络与传统模型进行对比，结果表明深度神经网络预测效果好，接着建立多个残差修正的深度神经网络模型，最后将其进行集成。通过实证研究表明集成模型的预测效果好于单个模型的预测效果。李新琴等(2020)[47]先建立深度学习模型对文本进行分类，然后使用组合权重的方法将开始建立好的深度神经网络结果进行集成，最后使用真实的数据进行验证，结果显示集成模型能够进一步提高模型的分类性能。王守志等(2020)[48]提出了一个加权平均集成算法，通过实证研究表明集成学习算法的准确率比其它模型都要好。尹鹏博等(2021)[49]建立多个深度学习基础模型和一个随机森林算法的元模型，在基础模模型上进行预测得到输出结果，然后在元模型上进行二次训练集成模型以此来提升模型的检测准确度，最后的实证结果表明集成模型效果好。

（4）国内相关文献述评

通过国内文献发现在保险领域国内的文献与国外一样，学者们主要是将广义线性模型与神经网络等机器学习的方法相对比，并没有研究讨论将两个模型相互结合。并且对于国内学者使用的神经网络模型来说，并没有使用深度神经网络。以及对于神经网络模型中分类变量的处理和神经网络的选取没有给出具体办法。而对于残差修正模型和集成模型的思想都是使用在其他领域并且通过实证表明模型的效果比较好。残差修正和集成的思想并没有被使用到保险领域中。

1.4 研究的内容和结构安排

1.4.1 研究的内容

基于国内研究现状的不足，本文以构建车险广义线性模型与神经网络模型的残差修正集成模型为主要研究内容，具体包括以下几个内容：

（1）在数据处理方面引入了新的分类变量的处理方法，通过对比说明新方法的优缺点，给之后的研究提供更多的数据处理方法。

（2）介绍和建立了泊松广义线性模型、过离散模型、广义可加模型、*BP* 神经网络、改进 *BP* 神经网络、*DNN*、一维 *CNN*、*LSTM*(*Long Short-Term Memory*)神经网络预测模型，并基于模型变量选择问题建立了 *Lasso-GAMLSS* 模型。

（3）介绍并改进 *CANN* 模型。通过对比发现广义可加模型比广义线性模型有更好的效果，*CNN* 的结构能够自动的压缩变量。建立广义可加模型的 *CANN* 和 *CNN* 模型的 *CANN*

（4）建立了基于一维 *CNN*、*DNN* 和改进 *BP* 神经网络的残差预测模型，利用神经网络模型来拟合残差序列，将残差预测结果作为 *GLM* 预测结果的修正。

（5）将广义线性模型与残差预测模型组合成残差修正预测模型，提出车险数据预测的广义线性残差修正模型。对相关集成模型进行理论研究，说明集成模型的优点。选择模型效果比较好的残差修正模型进行集成。分别使用两种方法进行集成，第一种方法是对残差修正模型使用 *DNN* 模型进行集成；第二种方法是对预测模型使用线性回归模型进行集成。

1.4.2 论文的结构安排

第一章主要对论文的研究背景和研究意义进行阐述说明，同时详细介绍了国内外在车险费率厘定、神经网络模型以及残差修正方面的研究现状。

第二章介绍索赔频率模型。通过对广义线性模型、过离散模型、广义可加模型和 Lasso-GAMLSS 模型的原理和结构的介绍，说明其在车险索赔频率的优势和不足。

第三章介绍神经网络模型。介绍广义线性模型和 BP 神经网络模型的原理和结构，基于 BP 神经网络的缺点提出泊松损失函数的 BP 神经网络，然后阐述一维 *CNN*、*DNN*、*LSTM* 深度神经网络的原理和结构，总结了神经网络模型在车险索赔频率中的优缺点。

第四章建立残差修正集成模型。先阐述 *CANN* 模型并根据广义可加模型和 *CNN* 模型将其进行扩展，指出模型的优缺点。接着基于 *CANN* 的缺点建立了 *GLM-改进 BP*、*GLM-DNN* 和 *GLM-CNN* 的残差修正预测模型。对相关集成模型进行理论研究，说明集成模型的优点。选择模型效果比较好的残差修正模型进行集成。分别使用两种方法进行集成，第一种方法是对残差修正模型使用 DNN 模型进行集成。第二种方法是对预测模型使用线性回归模型进行集成。

第五章实证研究。对车险索赔次数数据进行分析，展示数据各变量。对连续变量进行归一化处理，对分类变量，引入新的分类变量处理方法进行对比。详细说明神经网络模型对车险数据进行建模时候参数选取的步骤和方法。比较各算法

模型在数据中的效果和存在问题。

第六章对工作进行分析和总结，同时指出文本出现的不足以及能够进一步深入研究的问题。

1.5 本文的创新与不足

1.5.1 本文的创新点

本文主要对车险定价中索赔次数预测模型进行讨论，通过理论介绍和实证分析，验证了本文模型的可行性，其主要创新点总结如下：

（1）探讨 *Embedding* 算法与 *one-hot* 在分类变量处理方面的优缺点。针对变量选择问题提出了 *Lasso-GAMLSS* 模型。改进 *BP* 神经网络使其更加贴合车险索赔次数数据，将深度学习神经网络模型中的 *DNN*、一维 *CNN* 和 *LSTM* 使用到车险建模中，将 *CANN* 模型引入到车险建模中并对其进行扩展。

（2）在神经网络模型和传统的广义线性模型的基础上，提出了深度学习神经网络修正广义线性残差的模型，然后将多个残差修正模型进行集成，最后将建立的各个模型与传统广义线性模型一起运用到一组真实数据中去，通过对比多个结果发现，集成模型有更好的效果，说明新模型可行。在传统的车险索赔次数预测模型基础上提出了新的预测模型，具有一些创新意义。

（3）应用集成模型解决了神经网络模型不具有解释力的问题。同时讨论了针对数据集怎么选择合适的神经网络结构，具有一定应用价值。

1.5.2 本文的不足

本文通过广义线性模型与神经网络模型，提出新的集成模型，对索赔次数预测模型进行了讨论，虽然获得了一定的研究成果，但是因为个人能力、时间以及其他客观因素的影响，导致还有不少的问题可以进一步的讨论研究。

（1）将模型进行集成的方法有多种，本文只考虑了部分集成方法，未能全面比较集成算法。

（2）神经网络模型参数的选取关系到模型效果的好坏，对于参数选取的方法没有深入研究。

2 传统索赔频率预测模型

2.1 广义线性模型

广义线性模型结构如下：

（1）随机成分：被解释变量的观测值y_i为相互独立的随机变量，并且y_i服从指数分布族。

（2）系统成分：广义线性模型与线性回归模型一样，它们的线性预测值都可以用$\eta_i = x_i^T\beta$来表示。

（3）连接函数：使用连接函数$g(\cdot)$对被解释变量的均值进行变换之后等于线性预测值，也就是$g(\mu_i) = x_i^T\beta$。连接函数$g(\cdot)$是一个可导并且严格单调的函数，所以被解释变量的均值也可以写成是$\mu_i = \mathrm{h}(x_i^T\beta)$，其中$h(\cdot)$表示为连接函数$g(\cdot)$的逆函数，也称作其为响应函数。

广义线性模型的一般形式可以表示为：

$$\begin{cases} y_i \sim \text{均值为}\mu_i\text{指数分布族} \\ g(\mu_i) = x_i^T\beta \end{cases} \tag{2.1}$$

指数分布族有很多分布类型，像二项分布、正态分布、泊松分布、逆高斯分布、*Tweedie* 分布和伽马分布。

设随机变量 Y 服从指数分布族，那么其密度函数可以写成如下形式：

$$f(\mathrm{y},\theta,\phi) = exp\left[\frac{y\theta - b(\theta)}{\phi}\omega + c(y,\phi)\right] \tag{2.2}$$

在上式（2.2）中，$b(\theta)$与$c(y,\phi)$为已知函数，对于指数分布族中的不同分布，它们具有不同的形式。确定分布的具体形式是函数$b(\theta)$，而发挥标准化作用的函数是$c(y,\phi)$，即保证密度函数在它支撑集上的积分等于 1。θ被称作自然参数或者正则参数，它与分布的均值μ有关关系。离散参数为ϕ，它与分布的均值没有关系，只与分布的方差有关系。

如果连接函数能够使$g(\mu) = \theta$成立，那么就称该连接函数 g 是正则连接函数。对于指数分布族里的分布，因为$\mu = b'(\theta)$，所以，可以得出函数$b'(\cdot)$的反函数就是正则连接函数。考虑到汽车保险数据的特征，常常用到的是 *Logit* 连接函数$g(\mu) = ln[\mu/(1-\mu)]$、对数连接函数$g(\mu) = ln(\mu)$。

假设被解释变量 Y 密度函数满足式（2.2），其服从指数分布，那么观测值

（$Y_1, Y_2, \cdots, Y_n$）的对数似然函数可以表示成：

$$\ell = \sum_{i=1}^{n} \left[\frac{y_i\theta_i - b(\theta_i)}{\phi} \omega_i + c(y_i, \phi) \right] \tag{2.3}$$

因为$c(y_i, \phi)$与回归系数没有关系，所以对于广义线性模型，回归参数β_j（$j = 0,1,\cdots,k$）的的极大似然估计就是下面方程的解：

$$\frac{\partial \ell}{\partial \beta_j} = 0 \tag{2.4}$$

通过链式法则，可以知道：

$$\frac{\partial \ell}{\partial \beta_j} = \frac{\partial \ell_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j} \tag{2.5}$$

又因为$\mu_i = \mathrm{b}'(\theta_i)$，$\upsilon(\mu_i) = \mathrm{b}''(\theta_i)$，$g(\mu_i) = \sum_{j=0}^{p} x_{ij}^T \beta_j$，所以上述式子可以改写成为：

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\omega_i (y_i - \mu_i) x_{ji}}{\upsilon(\mu_i) g'(\mu_i)} = 0 \tag{2.6}$$

汽车保险的损失次数数据一般为取值是非负整数的随机变量，其往往是服从负二项分布或者是泊松分布。当损失次数的分布是服从泊松分布，那么就可以建立分布假设是泊松分布的广义线性回归，也被称为泊松回归模型，其指数分布族的密度函数是：

$$f(\mathrm{y};\mu) = exp\{y ln\mu - \mu - ln\Gamma(y+1)\} \tag{2.7}$$

其中，泊松回归模型的均值是μ，方差也是μ。由此可见泊松回归模型的方差与期望都是相同的形式。

当观测数据出现过离散现象的时候，数据损失次数的均值小于方差的时候，泊松回归模型就不在适合。就要建立过离散的预测模型。负二项回归模型是比较经常使用的过离散预测模型。有两种比较常见的负二项回归模型形式，在实际的应用中，负二项Ⅰ型分布比较常用。一般的负二项回归模型形式可以表示如下：

$$\begin{cases} y_i \sim NB(\mu_i, \sigma) \\ g(\mu_i) = \eta_i = x_i^T \beta \end{cases} \tag{2.8}$$

概率函数和参数形式如下：

$$f(y;\mathrm{r},\mathrm{p}) = \frac{\Gamma(y+1/\sigma)}{\Gamma(y+1)\Gamma(1/\sigma)} \left(\frac{1}{1+\sigma\mu} \right)^{1/\sigma} \left(\frac{\sigma\mu}{1+\sigma\mu} \right)^{y} \tag{2.9}$$

其中，正则连接函数是$g(\mu) = \ln \frac{\sigma\mu}{1+\sigma\mu}$, 均值为$\mu$，方差为$\mu + \sigma\mu^2$。

2.2 广义可加模型

为了使模型的解释变量和被解释变量之间建立非线性的关系，对下式（2.10）多元线性模型进行推广

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon_i \tag{2.10}$$

常见的推广方法为用$f_i(\cdot)$一个光滑的非线性函数替换$\beta_i x_i$。于是模型能够写成如下式子：

$$y_i = \beta_0 + \sum_{j=1}^{p} f_i(x_i) + \varepsilon_i \tag{2.11}$$

在线性模型的基础上，进行推广，使用一个非线性函数对自变量进行替换，同时保持模型的可加性，该模型就是可加模型。而将可加模型的框架使用在广义线性模型上的时候，就能够得到广义可加模型。

$$\begin{cases} y_i \sim \text{均值为}\mu_i\text{指数分布族} \\ g(\mu_i) = x_i^T \beta + \sum_{j=1}^{p} f_i(x_i) \end{cases} \tag{2.12}$$

其中，$g(\cdot)$为广义可加模型的连接函数，$f_i(\cdot)$是光滑函数。

2.3 *Lasso-GAMLSS*

Ribgy 针对广义线性模型和广义可加模型的不足，提出了基于尺度、形状、位置参数的广义可加模型 *GAMLSS*。该模型相对于广义线性模型和广义可加模型来说，其提供了多种分布种类，能够更好的刻画数据。被解释变量不在只是服从指数分布族，能够服从其他的分布，例如对数正态分布等。系统性成分引入了半参数或者非参数部分以及随机效应部分，丰富了模型系统性成分的内容。模型的结构如下：

$$g_k(\theta_k) = \eta_k = X_k \beta_k + \sum_{j=1}^{J_k} Z_{jk} \gamma_{jk} \tag{2.13}$$

其中，$g_k(\cdot)(k=1,2,\cdots,p)$是已知的连接函数。$\theta_k(k=1,2,\cdots,p)$是由 p 个参数组成的向量。在给定θ的条件下，观测值$y_i(i=1,2,\cdots,n)$相互独立，概率函数为$f(y_i|\theta)$。X_k是一个已知$n \times J_k$′的设计矩阵，Z_{jk}是一个$n \times q_{jk}$的矩阵，β_k是参数向量，γ_{jk}是一个q_{jk}维正态分布，满足$\gamma_{jk} \sim N_{qjk}\left(0, G_{jk}^{-1}\right)$，$G_{jk}^{-1}$是对称矩阵$G_{jk}$的广义逆矩阵。当 k=1，$Z_{jk}$=0 的时候，$g_1(\theta_1) = X_1 \beta_1$这就是常见的 *GLM*。

对上述式子参数的估计采用最大化惩罚对数似然函数，见下式：

$$l_p(\beta,\gamma) = \sum_{i=1}^{n} \ln f\,(y_i|\beta,\gamma) - \frac{1}{2}\sum_{k=1}^{p}\sum_{j=1}^{J_k} \gamma_{jk}^T G_{jk} \gamma_{jk} \tag{2.14}$$

上式子(2.14)能够用 *Rigby-Stasinopoulos*(*RS*)算法或者 *Cole-Green*(*CG*)算法进行运算得出。

考虑到 *GAMLSS* 模型在变量过多的时候，也会出现 *GLM* 模型类似的情况。考虑将 *Lasso* 引入到 *GAMLSS* 模型中，*Lasso-GAMLSS* 似然函数如下式子：

$$(\hat{\beta},\lambda) = argmin\{l_p(\beta,\gamma) + \lambda\sum_{j=0}^{p}|\beta_i|\} \quad (2.15)$$

其中，λ为调和参数，常用的选取调和参数的方法有交叉验证法、广义交叉验证法等。本文使用广义交叉验证法。

2.4 *GLM* 在索赔频率预测中的优缺点

根据车险索赔频率数据的特点，介绍了泊松分布的广义线性模型以及过离散模型，可以发现这两模型对参数估计的结果有直观的解释。而这两模型的缺点是建模开始前需要明确解释变量与被解释变量之间的函数关系，函数形式有限，而且，对于解释变量之间的相互关系，广义线性模型没法自动识别，这就会导致建模过程比较耗费时间。在广义可加模型中，非线性关系使得模型预测得更加准确，但是在多变量得情况下，模型会忽略有意义的交互项，由此会增加模型的估计方差。*Lasso-GAMLSS* 模型有着更多的参数，能够更好的刻画数据，但是，如果模型对变量的选择不准确，那么就会丢失有用信息导致模型不准确。针对对以上缺点，本文讨论神经网络模型在索赔频率预测中的应用。

3 神经网络模型

3.1 *BP* 神经网络模型

BP 神经网络一般的函数形式如下表示：

$$y = f\left(\omega_0 + \sum_{j=1}^{m} \omega_j \cdot g\left(\omega_{0j} + \sum_{i=1}^{n} \omega_{ij} x_i\right)\right) \tag{3.1}$$

在式（3.1）中，ω_0表示神经网络的截距项，ω_{0j}是第 j 个神经元的截距项；ω_j表示第j个神经元到输出变量的权重；ω_{ij}表示输入变量到神经元的权重；x_i代表输入变量；m 代表隐藏层中神经元的个数。

在式（3.1）中，神经网络模型的权重可以基于多种算法将其求出，例如让误差平方和函数达到最小的条件来进行估计，也就是最小化损失函数$D = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$。其中，$y_i$为观察值，$\hat{y}_i$为神经网络模型的拟合值。

3.2 改进的 BP 神经网络

在实际运用中，上边展示的 *BP* 神经网络模型往往因为种种原因不能很好的行进拟合，所以为了适应车险的数据，假设其服从泊松分布并且使用梯度下降法对其进行求解。式子如下：

$$\log \lambda(x) = \beta_0 + \sum_{j=1}^{q} \beta_j z_j(x) \tag{3.2}$$

当j=1，2，…,q，隐藏层的神经元$z_j(x)$如下：

$$z_j(x) = \phi\left(\omega_{j,0} + \sum_{l=1}^{d} \omega_{j,l} x_l\right) \tag{3.3}$$

以及在频率服从泊松分布的假设下：

$$N_i \sim Poi(\lambda(x_i)v_i)$$

得到观测值 D 的泊松损失函数：

$$\mathcal{L}(D,\theta) = \frac{1}{n}\sum_{i=1}^{n} 2N_i \left[\frac{\lambda(x_i)v_i}{N_i} - 1 - \log\left(\frac{\lambda(x_i)v_i}{N_i}\right)\right] \tag{3.4}$$

可以看出，损失函数的最大似然估计由最小化神经网络参数θ提供，使用梯度下降法进行求解。假设到达算法的$\theta^{(t)}$这个位置，那么下一步的局部最优移动由下式给出：

$$\theta^{(t+1)} = \theta^{(t)} - \varrho_t \nabla_\theta \mathcal{L}\left(D, \theta^{(t)}\right) \tag{3.5}$$

通过适当地微调学习率$\varrho_t > 0$，使得一阶项仍然是$\mathcal{L}\left(D,\theta^{(t)}\right)$的泰勒展开中的主导项。该算法的迭代(对于极小调节的学习率)将收敛到$\mathcal{L}(D,\theta)$的局部最小值，并且可以探索算法的不同起始点来分析潜在的不同局部最小值。一旦损失函数的减少不再显著，这个时候算法就开始潜在地过度拟合(在样本中)，应该停止迭代(提前停止)。神经网络模型的过度拟合可以使用验证样本进行监控。

3.3 深度神经网络

数据变量之间的关系往往是复杂多样的，而单一的 BP 神经网络结构不能很好的刻画其关系。深度神经网络模型不仅仅拥有更多的模型层数，还拥有多样的模型结构，能够更好的刻画数据变量之间的关系。接下来介绍 3 种结构的深度神经网络模型。

3.3.1 *DNN*

从深度神经网络按不同层的位置来进行划分，其内部的神经网络层能够分成三大类，输入层，隐藏层和输出层。一般的，对于深度神经网络模型结构来说第一层为输入层，最后一层为输出层，而中间的层数都为隐藏层。每一层和每一层之间为全连接，也就是第 *i* 层任意的一个神经元肯定会和第 *i*+1 层任意的一个神经元相互连接。虽然深度神经网络模型的结构看起来很复杂，但是从小的局部模型来看，它还是与感知机模型的结构一样，即一个线性关系$z = \sum \omega_i x_i + b$，再加上一个激活函数。

3.3.2 一维 *CNN*

在图像识别、语音识别等领域里，最受人关注的神经网络模型是深度卷积神经网络。该模型的主要优势是减少了手动选择特征的环节，与此同时，反向传播算法和权值共享确保了模型的泛化能力。图像对应的输入尺度为三维，分类时候选用的为二维 *CNN* 进行识别。而本文的数据是二维变量，所以选用一维的 *CNN*。具体结构如下：

提取数据集中局部区域信息是卷积层的主要功能。一维卷积只包含一个空间维度，具体形式如下：

$$X_{t,c}^{l+1} = \sum_{i=1}^{C_{in}} \sum_{k=1}^{K} W_{k,i,c}^{l} X_{s(t-1)+k,i}^{l} + B_C \qquad (3.6)$$

式（3.6）中，X^l是输入数据的二维矩阵，其大小为$L*C_{in}$，L 为输入的长度，

C_{in}为输入的通道个数。$X_{t,c}^{l+1}$是第 l+1 层中第 c 个通道的第 t 个参数；$W_{k,i,c}^{l}$是第 c 个卷积核中第 i 个通道中的第 k 个权重系数；卷积核中的偏置系数为B_C；卷积核的大小是 K，卷积核的步长为 s。

激活函数使得神经网络模型有了非线性性，让模型能够有更好的表达能力。*Relu* 函数具有收敛速度快等优点被广泛使用。当使用 *relu* 激活函数之后，输出结果为：

$$a^{[l]} = max\left(0,\ W^T a^{[l-1]} + b\right) \quad l=1,\ 2\cdots L-1 \tag{3.7}$$

式子（3.7）中，第 l 层的输出矩阵是$a^{[l]}$，神经网络模型中总层数（输入层除外）为 L。

池化层连接在卷积层之后，其作用是通过降采样的方式来减少网络中的参数数量。本文采用最大值池化层，其结构如下：

$$Y_{t,\ c}^{l+1} = \max_{(t-1)K+1 \le j \le tK} \left\{X_{j,c}^{l}\right\} \tag{3.8}$$

式子（3.8）中，$Y_{t,\ c}^{l+1}$为第 l+1 层中第 c 个通道中的第 t 个值；池化区域的宽度为 K; $X_{j,c}^{l}$是第 l 层中第 c 个通道中的第 j 个神经元的值。

最后，有一个全连接层把所有的结果整合到一起，这个全连接层为最后的输出层。一般的设定与 *DNN* 的网络结构相同。

3.3.3 *LSTM*

LSTM 模型主要包含遗忘门、输入门和输出门。其中输入门能够暂时存储有关信息，输入信息传递到 LSTM 细胞单元，遗忘门控制信息通过量的大小，式子如下：

$$f_t = \sigma(\omega_{L1} \cdot [h_{t-1}, x_t] + b_{L1}) \tag{3.9}$$

上式（3.9）中，σ代表 *sigmoid* 函数；ω_{L1}代表权重；b_{L1}代表偏置项；h_{t-1}代表的是前一个单元输出；x_t则代表当前输入。

输入门控制着新信息是否被神经元记忆，式子如下：

$$i_t = \sigma(\omega_{L2} \cdot [h_{t-1}, x_t] + b_{L2}) \tag{3.10}$$

$$\widetilde{C_t} = \tanh(\omega_{L3} \cdot [h_{t-1}, x_t] + b_{L3}) \tag{3.11}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t} \tag{3.12}$$

上述三式中，ω_{L2}、b_{L2}分别是输入门的权重与偏置；ω_{L3}、b_{L3}分别是记忆单元的权重与偏置；*tanh* 代表激活函数；f_t、i_t分别对应的是遗忘门与输入门的输出；C_t代表的是记忆单元的输出；C_{t-1}代表的是上一个记忆单元的输出；$\widetilde{C_t}$代表的是激活函数 *tanh* 的输出。

记忆单元的输出C_t和输出门的输出o_t决定了 *LSTM* 单元的最终输出h_t，具体的式子如下：

$$o_t = \sigma(\omega_{L4} \cdot [h_{t-1}, x_t] + b_{L4}) \tag{3.13}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{3.14}$$

其中，b_{L4}、ω_{L4}分别代表输出们的偏置和权重。

3.4 神经网络模型在索赔频率预测中的优缺点

通过分析各个神经网络的结构，可以发现神经网络模型能够比较好的解决广义线性模型的缺点。在没法明确解释变量和被解释变量之间的函数关系时，神经网络模型能够基于观测值来建立解释变量和被解释变量之间的函数关系。同时，神经网络模型往往具有较好的拟合效果和预测能力。但是，神经网络的模型参数不具有解释力，模型的建立过程类似一个黑箱，而且，当模型参数过多的时候，会出现过拟合的现象。

4. GLM与神经网络的集成模型

不管是从定性的角度进行讨论还是从定量的角度进行研究，因为每一个预测方法的理论基础、方法机理和所使用的变量处理方式不尽相同，所以，这就导致了不同的方法有不同的特点。每种预测模型之间不是互相排斥的关系，反而在讨论有些问题的时候各种预测模型往往是互通的、互补的，大部分的情况下能够组合使用，这样不仅仅能够发挥每个模型的优点，还能充分挖掘相关性获得最好的预测结果。

残差修正预测模型就是把两种以上不一样的预测模型进行组合，全面考虑所选每个预测模型的优点，对其进行一定的组合，然后产生一个新的尽量包含所选预测模型优点的组合预测模型。这样做的目的是为了全面利用每个预测模型的优势，充分挖掘所研究问题的数据信息，更加真实地反映数据信息和特征，以达到提升模型预测精度的目的。在进行实际预测的时候，可能会遇到一种预测模型能够很好的反映出原始数据的一些特定性信息，但是有时候这样的预测模型预测误差比较大，可以通过其和另一个预测误差比较小的预测模型进行组合使用，以此来达到预测模型不仅仅能反映出原始数据所需的特定信息还能获得比较好的预测结果。一般情况下，组合预测模型能够增强预测结果的可靠性和模型的预测精度。

车险索赔次数数据会受到各类因素的影响。有些确定的影响因素会让车险索赔次数数据具有一定的规律，能够使用数学模型建立变量之间的关系并对其进行预测。但是，很多的影响因素是不确定的，对于车险预测模型而言，这些影响因素是出现残差的重要因素。通过阅读文献，知道深度神经网络模型在模型预测有更好的效果，也就是说该模型残差拟合效果更好。

基于此本文将先建立两种残差修正模型，一种是将广义线性模型的参数直接赋给深度神经网络，然后通过深度神经网络模型调整参数获得最终的预测模型。第二种是将广义线性模型的残差序列带入到深度神经网络中，获得残差的预测值，然后将预测的残差值作为预测结果的修正。接着选择合适的残差修正模型建立两种集成模型，一种是对残差修正模型进行集成。第二种是对预测模型进行集成。集成模型能够更好的发挥每个预测模型的优势，达到更好的预测效果。

4.1 *CANN*及其扩展

4.1.1*CANN*

CANN 是 Wüthrich 等[20]提出的一种模型，该模型把经典的广义线性模型 *GLM* 与深度神经网络模型相互结合。这种方法能够被使用的场景相当广泛，因为它适用于很多的参数回归模型问题，这能够获得两个模型的优点。这个想法是将广义线性模型 *GLM* 嵌入到一个 *DNN* 神经网络模型的架构中。为此将分别建立广义线性模型和神经网络模型，然后将其进行组合。模型结构如下图：

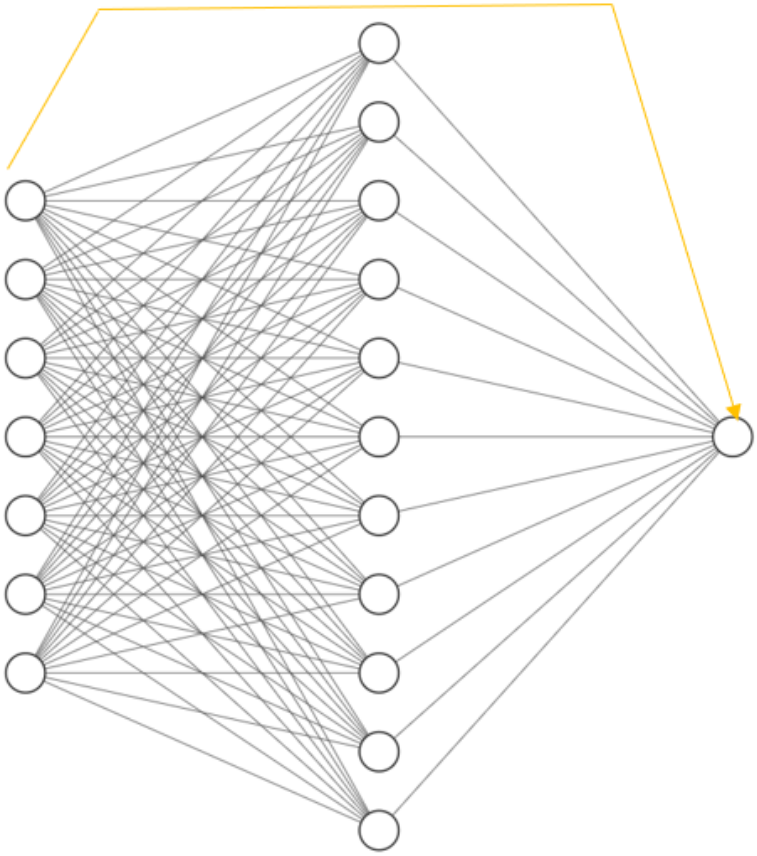

图 4.3 CANN 模型

它将 *GLM* 嵌入到 *DNN* 网络体系结构中，方法是将它打包到一个直接链接的所谓跳过连接中，从输入层到输出层，参见图 4.3 中的黄色箭头。*Skip* 连接用于深层网络是因为它们具有良好的校准特性，潜在地避免了梯度问题。

4.1.2*CANN* 扩展

本文将 Wüthrich 等[20]提出的 *CANN* 进行扩展，首先将该模型里面使用的

DNN 神经网络模型替换成 *CNN* 神经网络。*CNN* 模型相对于 *DNN* 模型来说，因其 *CNN* 结构的特征，能够自动进行特征提取，卷积层可以提取特征，卷积层中的卷积核（滤波器）真正发挥作用，通过卷积提取需要的特征，在变量较多的时候能够自动进行变量压缩。前面已经介绍过广义线性模型和 *CNN* 神经网络模型，现在直接进行使用。分别建立广义线性模型和 *CNN* 神经网络。假设两个模型都作用在相同的特征空间里，那么广义线性模型 *GLM* 提供的回归函数如下：

$$\lambda^{GLM}(x) = exp\langle \beta^{GLM}, x\rangle \tag{4.1}$$

式（4.1）中，这个广义线性模型能够被解释为不具有隐藏层和输出层且激活函数是指数函数的神经网络模型。*CNN* 神经网络作为该方法的第二个要素，选择深度为 *d* 的 *CNN* 神经网络模型，如下：

$$\lambda^{CNN}(x) = exp\langle \beta^{CNN}, z^{(d:1)}x\rangle \tag{4.2}$$

对于该方法，将这两个模型进行相互结合，定义该回归函数如下：

$$\lambda(x) = exp\{\langle \beta^{GLM}, x\rangle + \langle \beta^{CNN}, z^{(d:1)}(x)\rangle\} \tag{4.3}$$

CNN 神经网络梯度下降算法的初始参数可以精确的从广义线性模型提供的参数开始进行迭代。然后，梯度下降算法针对给定的损失函数开始使用深度神经网络来改进该广义线性模型网络体系结构。如果在梯度下降算法期间损失函数显著减小，那么就对广义线性模型进行改进，否则广义线性模型已经是足够好的了。

能选择训练或者不训练广义线性模型。在不训练广义线性模型的情况下，其一直保留在 *CANN* 模型中；在训练广义线性模型的情况下，通过神经网络部分对其进行修正。如果不想对广义线性模型进行训练，那么可以选择一个可训练的可信度权值α，其范围是(0,1)，它的大小能够改变广义线性模型对于整个模型的影响。式子如下：

$$\lambda(x) = exp\{\alpha\langle \beta^{GLM}, x\rangle + (1-\alpha)\langle \beta^{CNN}, z^{(d:1)}(x)\rangle\} \tag{4.4}$$

使用之前介绍的算法，很容易实现上述 *CANN* 回归模型。如果 *GLM* 参数β^{GLM}在式(4.3)中是不训练的，则泊松回归模型甚至能够更简单的获得其结果。在泊松回归模型中，有损失次数N_i，风险暴露v_i，解释变量x_i，式子如下：

$$N_i \sim Poi\big(exp\{\langle \hat{\beta}^{GLM}, x_i\rangle + \langle \beta^{CNN}, z^{(d:1)}(x_i)\rangle\}v_i\big)$$

$$= Poi\left(exp\langle\beta^{CNN}, z^{(d:1)}(x_i)\rangle\hat{v}_i\right) = Poi(\lambda^{CNN}(x_i)\hat{v}_i) \quad (4.5)$$

式子（4.5）中，定义$\hat{v}_i = exp\langle\hat{\beta}^{GLM}, x_i\rangle v_i$。因此，在这种情况下，仅用$\hat{v}_i$替换原始风险年暴露数$v_i$就能使得 *CANN* 回归模型校准与经典的前馈神经网络校准相同。

观察到在式子（4.5）中，使用广义线性模型的估计来修改风险年暴露v_i，该式提供了一种非常通用的方法来通过向原始回归模型添加神经网络特征来回溯测试(提升)任何回归模型，也可以使用任何其他回归模型代替广义线性模型。接着使用广义可加模型对广义线性模型进行替换，并且将其与 *DNN* 相结合构建 *CANN* 模型，模型结构如下：

$$\lambda(x) = exp\{f\langle x\rangle^{GAM} + \langle\beta^{DNN}, z^{(d:1)}(x)\rangle\} \quad (4.6)$$

广义可加模型有线性和非线性两部分组成，这两部分对广义可加模型的影响不同，将这两部分与神经网络结合时，为了获得更好的模型效果，建立 *CANN* 模型的时候，将广义可加模型的线性和非线性部分分开与神经网络模型进行结合。形式如下：

$$\lambda(x) = exp\left\{\langle\omega_1, \left(z_1^{(d:1)}\right)(x_1)\rangle, \langle\omega_2, \left(z_2^{(d:1)}\right)(x_2)\rangle\right\} \quad (4.7)$$

式子（4.7）中x_1向量为广义可加模型的线性解释变量，x_2向量为非线性解释变量。

4.2 残差修正模型

CANN 模型并没有实际解决模型没有解释力的问题，基于此本文提出神经网络模型修正的广义线性预测模型，具体如下：

神经网络模型修正的广义线性模型，其预测值y_i看成为广义线性模型的预测值y_{G}和神经网络模型的残差预测值ξ_{i}两部分组合而成，也就是$y_{\mathrm{i}} = y_{\mathrm{G}} + \xi_{\mathrm{i}}$。广义线性模型能够对分析车险数据各个变量对被解释变量的影响提供很好的帮助，再利用神经网络模型良好的数据挖掘能力，充分的挖掘变量和解释变量之间的关系，并且对广义线性模型进行补偿修正，以此弥补广义线性模型的不足之处，从而提高最终模型的预测精度。

基本步骤如下：

（1）对数据进行处理

（2）建立广义线性模型，然后建立残差序列。设原始数据为y，残差为ε，则有$\varepsilon_i = y_G(i) - y(i)$。其中，$y_G(i)$为广义线性模型对数据的预测值，$y(i)$为实际数据，$\varepsilon_i$为残差序列。

（3）以原始车险数据的影响因素为自变量，残差序列为被解释变量，利用神经网络模型的优势对变量之间的关系进行挖掘，更好的拟合模型。

（4）使用（3）中训练好的神经网络模型对残差进行预测，设神经网络残差预测值是ξ_{i}，使用ξ_{i}对广义线性模型的预测值$y_G(i)$进行修正，最后得到的模型预测结果是$y(i) = y_{\mathrm{G}}(i) + \xi_{\mathrm{i}}$。

从模型的整个预测过程来看，广义线性模型用来初步分析车险数据，同时提供模型的解释力。而神经网络模型进一步分析车险数据，通过使用神经网络对残差序列进行预测，然后使用预测残差值对广义线性模型结果进行修正，充分发挥了这 2 个模型的优点，不仅仅能够挖掘各个变量之间的深度关系，还能能够解释各个变量对被解释变量的影响力。模型的流程图见图 4.4。

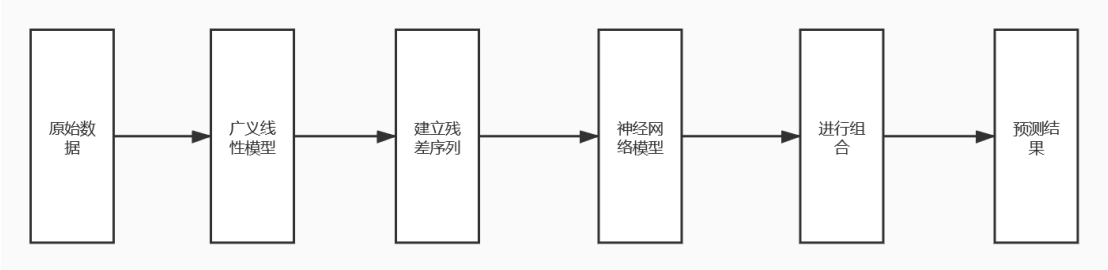

图 4.4 模型流程

基于上述建模流程，本文建立了 *GLM*-改进 *BP*、*GLM-CNN* 以及 *GLM-DNN* 这 3 个残差修正模型。具体模型图见图 4.5

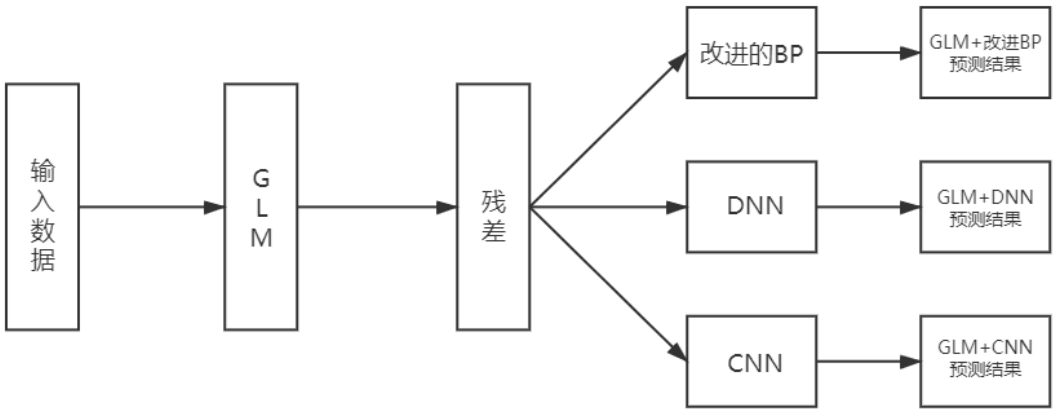

图 4.5 残差修正模型

4.3 集成模型

4.3.1 理论研究

集成模型一般会比单个模型的预测效果更好，并且对于神经网络模型来说不同的模型得到不同的模型参数但是可能有相同的结果，这对于模型的选择是有困难的，集成模型能解决上述问题。本节部分理论证明参考 *Richman Ronald* (2020)[17]。

通过之前章节的介绍，对模型 i 的参数θ_i建立模型。模型 i 是泊松广义线性模型与神经网络模型的混合模型。模型如下：

$$\theta(x_i) = \left(\kappa_p'\right)^{-1}\left(\mu(x_i)\right) = \left(\kappa_p'\right)^{-1}\left(g^{-1}\left(\langle\beta^{(d+1)}, \left(Z^{(d)} \circ \cdots \circ Z^{(1)}\right)(x_i)\rangle\right)\right) \quad (4.8)$$

其中，$\kappa_p = exp(\theta)$, 该函数是光滑且严格凸函数。g()是连接函数。参数采用极大似然法进行估计。能够通过最大化似然函数或最小化相应的偏差函数损失函数进行估计。本文使用最小化偏差损失函数，因为其与梯度下降法中最小化目标函数的原理相似。在上述模型假设下，独立随机变量Y_i的平均偏差损失写成下式：

$$\begin{aligned}\mathcal{L}(D,\beta) = \frac{2}{n}\sum_{i=1}^{n} v_i\left[Y_i\left(\kappa_p'\right)^{-1}(Y_i) - \kappa_p\left(\left(\kappa_p'\right)^{-1}(Y_i)\right) - Y_i\theta_i + \right.\\ \left.\kappa_p(\theta_i)\right] = \frac{2}{n}\sum_{i=1}^{n} v_i\left[Y_i\left(\kappa_p'\right)^{-1}(Y_i) - \kappa_p\left(\left(\kappa_p'\right)^{-1}(Y_i)\right) - \right.\\ \left. Y_i\left(\kappa_p'\right)^{-1}(\mu_i) + \kappa_p\left(\left(\kappa_p'\right)^{-1}(\mu_i)\right)\right] \quad (4.9)\end{aligned}$$

数据集$D = \{(Y_i, x_i, v_i); i = 1, \ldots, n\}$, $\theta_i = \theta(x_i, \beta)$。对于任何模型 i 的均值参数μ_i可以写成如下：

$$\delta(Y_i,\mu_i)=2v_i\left[Y_i\left(\kappa_p'\right)^{-1}(Y_i)-\kappa_p\left(\left(\kappa_p'\right)^{-1}(Y_i)\right)-Y_i\left(\kappa_p'\right)^{-1}(\mu_i)+\kappa_p\left(\left(\kappa_p'\right)^{-1}(\mu_i)\right)\right] \tag{4.10}$$

$\delta(Y_i,\mu_i)$被称作是关于Y_i均值参数μ_i的一个估计。梯度下降法通过优化网络参数β使目标函数$\mathcal{L}(D,\beta)$变小。为了防止模型过拟合往往会给定一个规则将其提前停止。但是，这种方法取决于梯度下降算法的起始点，而且由于每次运算的起始点不同，最后得到的参数结果也不一样。这就导致了如何选择模型的问题，而集成模型能够解决上述问题。

对于单个模型 i，假设$\widehat{\mu_i}$是观测值Y_i的预测因子（均值参数$\mu_i=E[Y_i]$）。假设预测因子$\widehat{\mu_i}$和观测值Y_i是独立的。通过平均损失来进行泛化分析，见下式：

$$E\left(\delta(Y_i,\widehat{\mu_i})\right) \tag{4.11}$$

对于上式，通常假设$\widehat{\mu_i}$使得上述式子是有限的。

命题 1 观测值Y_i服从泊松分布，假设$\widehat{\mu_i}$是$\mu_i=\kappa_p'$的无偏估计，与Y_i相互独立。那么有如下损失函数关系：

$$E[\delta(Y_i,\widehat{\mu_i})]\geq E[\delta(Y_i,\mu_i)] \tag{4.12}$$

证明:通过单个偏差的平均值来计算预期的泛化损失

$$E[\delta(Y_i,\widehat{\mu_i})]=2v_iE\left[Y_i\left(\kappa_p'\right)^{-1}(Y_i)-\kappa_p\left(\left(\kappa_p'\right)^{-1}(Y_i)\right)-Y_i\left(\kappa_p'\right)^{-1}(\widehat{\mu_i})+\kappa_p\left(\left(\kappa_p'\right)^{-1}(\widehat{\mu_i})\right)\right] \tag{4.13}$$

式(4.13)加上并减去$2v_iE\left[Y_i\left(\kappa_p'\right)^{-1}(\mu_i)-\kappa_p\left(\left(\kappa_p'\right)^{-1}(\mu_i)\right)\right]$得下式：

$$E[\delta(Y_i,\widehat{\mu_i})]=E[\delta(Y_i,\mu_i)]+2v_iE\left[Y_i\left(\kappa_p'\right)^{-1}(\mu_i)-\kappa_p\left(\left(\kappa_p'\right)^{-1}(\mu_i)\right)-Y_i\left(\kappa_p'\right)^{-1}(\widehat{\mu_i})+\kappa_p\left(\left(\kappa_p'\right)^{-1}(\widehat{\mu_i})\right)\right] \tag{4.14}$$

令$h_p(m)=Y_i\left(\kappa_p'\right)^{-1}(m)-\kappa_p\left(\left(\kappa_p'\right)^{-1}(m)\right)$，其中$m>0$，则式(4.14)写成如下：

$$E[\delta(Y_i,\widehat{\mu_i})]=E[\delta(Y_i,\mu_i)]+2v_i\left(h_p(\mu_i)-E\left[h_p(\widehat{\mu_i})\right]\right) \tag{4.15}$$

因为$\kappa_p(\theta)=exp(\theta)$,所以$h_p(m)$写成下式

$$h_p(m)=Y_i\log(m)-m \tag{4.16}$$

对式(4.16)求二阶导如下：

$$\frac{\partial^2 h_p(m)}{\partial^2 m} = -\frac{Y_i}{m^2} < 0 \tag{4.17}$$

通过式(4.17)，可以得出h_p函数是一个下凸函数，对式(4.15)使用 Jensen 不等式得到下式：

$$E[\delta(Y_i,\widehat{\mu_i})] = E[\delta(Y_i,\mu_i)] + 2v_i\left(h_p(\mu_i) - E\left[h_p(\widehat{\mu_i})\right]\right) \geq E[\delta(Y_i,\mu_i)] + 2v_i\left(h_p(\mu_i) - h_p(E[\widehat{\mu_i}])\right) \tag{4.18}$$

因为假设$\widehat{\mu_i}$是μ_i的无偏估计，所以$E[\delta(Y_i,\mu_i)] + 2v_i\left(h_p(\mu_i) - h_p(E[\widehat{\mu_i}])\right) = E[\delta(Y_i,\mu_i)]$，由此可得$E[\delta(Y_i,\widehat{\mu_i})] \geq E[\delta(Y_i,\mu_i)]$。证毕。

假设独立同分布的$\hat{\mu}_i^{(j)}$是$\widehat{\mu_i}$的无偏估计量，定义集成预测因子如下：

$$\bar{\mu}_i^{(M)} = \frac{1}{M}\sum_{j=1}^{M}\hat{\mu}_i^{(j)} \tag{4.19}$$

命题2 假设独立同分布的$\hat{\mu}_i^{(j)}(j \geq 1)$满足命题1以及与$Y_i$相互独立。对于$M \geq 1$，有如下关系：

$$E\left[\delta\left(Y_i,\hat{\mu}_i^{(1)}\right)\right] \geq E\left[\delta\left(Y_i,\bar{\mu}_i^{(M)}\right)\right] \geq E\left[\delta\left(Y_i,\bar{\mu}_i^{(M+1)}\right)\right] \geq E[\delta(Y_i,\mu_i)] \tag{4.20}$$

证明：最后一个关系是明确的，因为集成的预测因子本身满足命题1的假设，并且满足之后相应的假设。因此，主要讨论 M≥1 各式子的关系。使用之前定义的函数h_p，有如下式子：

$$\begin{aligned} E\left[\delta\left(Y_i,\bar{\mu}_i^{(M)}\right)\right] &= E\left[\delta\left(Y_i,\bar{\mu}_i^{(M+1)}\right)\right] + 2v_i\left(E\left[h_p(\bar{\mu}_i^{M+1})\right] - E\left[h_p(\bar{\mu}_i^{M})\right]\right) \\ &\geq E\left[\delta\left(Y_i,\bar{\mu}_i^{(M+1)}\right)\right] + 2v_i\left(h_p\left(E\left(\bar{\mu}_i^{M+1}\right)\right) - h_p\left(E\left(\bar{\mu}_i^{M}\right)\right)\right) \\ &= E\left[\delta\left(Y_i,\bar{\mu}_i^{(M+1)}\right)\right] + 2v_i\left(h_p\left(E\left[\frac{1}{M+1}\sum_{j=1}^{M+1}\hat{\mu}_i^{(j)}\right]\right) - h_p\left(E\left[\frac{1}{M}\sum_{j=1}^{M}\hat{\mu}_i^{(j)}\right]\right)\right) \\ &= E\left[\delta\left(Y_i,\bar{\mu}_i^{(M+1)}\right)\right] \end{aligned} \tag{4.22}$$

证明过程与命题1相似。其中，将 Jensen 不等式应用于凹函数h_p就能证明不等式。证明完毕。

命题2说明集成模型是有效的，即集成的独立同分布预测因子式（4.19）导

致预期泛化损失单调递减。

命题 3 假设独立同分布的$\hat{\mu}_i^{(j)}$ $(j\geq 1)$满足命题 1，并且与Y_i相互独立。$E[\delta(Y_i,\hat{\mu}_i)]$有一致可积的上界。有下式关系：

$$\lim_{M\to\infty} E\left[\delta\left(Y_i,\bar{\mu}_i^{(M)}\right)\right]=E\left[\lim_{M\to\infty}\delta\left(Y_i,\bar{\mu}_i^{(M)}\right)\right]=E[\delta(Y_i,\mu_i)] \tag{4.23}$$

证明：通过命题 1 可以得出如下式子：

$$E\left[\delta\left(Y_i,\bar{\mu}_i^{(M)}\right)\right]=2v_i(E\left[Y_i\left(\kappa_p'\right)^{-1}(Y_i)\right]-E\left[\kappa_p\left(\left(\kappa_p'\right)^{-1}(Y_i)\right)\right]-$$
$$E\left[h_p\left(\bar{\mu}_i^{(M)}\right)\right]) \tag{4.24}$$

证明命题 3 只用考虑式(4.23)最后一项就可以。通过大数定理，可以得出$\lim_{M\to\infty} h_p(\bar{\mu}_i^{(M)})=h_p(\mu_i)$。因为$\hat{\mu}_i^{(j)}$是独立同分布的并且也是无偏估计量。同时，$h_p$函数为有界函数(式(4.17)可知)，这就满足控制收敛定理。

命题 4 假设服从独立同分布的$\hat{\mu}_i^{(j)}$满足命题 1 并且独立于Y_i。$\hat{\mu}_i$有有限的二阶矩。有以下关系式：

$$M^{1/2}\frac{h_p\left(\bar{\mu}_i^{(M)}\right)-h_p(\mu_i)}{h_p'(\mu_i)Var(\hat{\mu}_i)^{1/2}}\Longrightarrow\mathcal{N}(0,1),M\longrightarrow\infty \tag{4.25}$$

命题 4 说明了集成预测模型的收敛速度如何转化成为偏差损失函数的收敛速度。

证明:使用泰勒展开式：

$$h_p\left(\bar{\mu}_i^{(M)}\right)=h_p(\mu_i)+h_p'(\mu_i)\left(\bar{\mu}_i^{(M)}-\mu_i\right)+\frac{1}{2!}h_p''(m)\left(\bar{\mu}_i^{(M)}-\mu_i\right)^2 \tag{4.26}$$

其中，m 在$\bar{\mu}_i^{(M)}$和μ_i之间。通过上式可以得到如下关系式：

$$M^{1/2}\frac{h_p\left(\bar{\mu}_i^{(M)}\right)-h_p(\mu_i)}{h_p'(\mu_i)Var(\hat{\mu}_i)^{1/2}}=M^{1/2}\frac{\bar{\mu}_i^{(M)}-\mu_i}{Var(\hat{\mu}_i)^{1/2}}+M^{1/2}\frac{h_p''(m)\left(\bar{\mu}_i^{(M)}-\mu_i\right)^2}{2!\ h_p'(\mu_i)Var(\hat{\mu}_i)^{1/2}} \tag{4.27}$$

其中，对于等式(4.27)右边的第一项,当$M\longrightarrow\infty$时，使用中心极限定理可以得出$M^{1/2}\frac{\bar{\mu}_i^{(M)}-\mu_i}{Var(\hat{\mu}_i)^{1/2}}\Longrightarrow\mathcal{N}(0,1)$。所以，只需要证明等式最后一项以概率收敛到零，即可证明命题 4。考虑$A_M=\left\{\left|\bar{\mu}_i^{(M)}-\mu_i\right|>M^{-3/8}\right\}=\left\{M^{1/2}\left|\bar{\mu}_i^{(M)}-\mu_i\right|>M^{1/8}\right\}$。根据中心极限定理可以得到$\lim_{M\to\infty} P[A_M]=0$。对于无穷小量$\varepsilon>0$.有下关系式：

$$\lim_{M\to\infty} P\left[M^{1/2} \max_{-\left|\bar{\mu}_i^{(M)}-\mu_i\right|\le\xi\le\left|\bar{\mu}_i^{(M)}-\mu_i\right|} \left|h_p''(\mu_i+\xi)\right|\left(\bar{\mu}_i^{(M)}-\mu_i\right)^2 > \varepsilon\right] \le$$

$$\lim_{M\to\infty} P\left[M^{1/2} \max_{-\left|\bar{\mu}_i^{(M)}-\mu_i\right|\le\xi\le\left|\bar{\mu}_i^{(M)}-\mu_i\right|} \left|h_p''(\mu_i+\xi)\right|\left(\bar{\mu}_i^{(M)}-\mu_i\right)^2 > \varepsilon, A_M^c\right] +$$

$$\lim_{M\to\infty} P[A_M] = 0 \qquad (4.28)$$

上述命题1-4说明泊松广义线性模型与神经网络组合的集成模型效果好，接着将模型泛化，设原始样本为y，训练T个模型，然后集成为一个强模型。T个模型得到的强模型为T个模型估计的期望，见下式：

$$\phi_A(x) = E_T\phi(x,\mathrm{T}) \qquad (4.29)$$

观测值y与每个单个模型之间的差，见下式：

$$E_T\left(y-\phi(x,\mathrm{T})\right)^2 \qquad (4.30)$$

式子（4.30）展开之后得到下式：

$$y^2 - 2yE_T\phi(x,\mathrm{T}) + E_T\phi^2(x,\mathrm{T}) \qquad (4.31)$$

而观测值y与强模型的差值如下：

$$\left(y-\phi_A(x)\right)^2 \qquad (4.32)$$

式子（4.32）展开后得到下式：

$$y^2 - 2y\phi_A(x) + \left(\phi_A(x)\right)^2 \qquad (4.33)$$

而将式子（4.29）带入式子（4.33）中，得到下式：

$$y^2 - 2yE_T\phi(x,\mathrm{T}) + \left(E_T\phi(x,\mathrm{T})\right)^2 \qquad (4.34)$$

而式子（4.31）与式子（4.34）做差就得到下式：

$$E_T\left(y-\phi(x,\mathrm{T})\right)^2 - \left(y-\phi_A(x)\right)^2 = E_T\phi^2(x,\mathrm{T}) - \left(E_T\phi(x,\mathrm{T})\right)^2 \qquad (4.35)$$

通过方差与期望的公式$D(x) = E(x^2) - \left(E(x)\right)^2$大于零可以得出下式：

$$E_T\left(y-\phi(x,\mathrm{T})\right)^2 \ge \left(y-\phi_A(x)\right)^2 \qquad (4.36)$$

4.3.2 模型建立

通过上述理论研究得出单个模型估计值与观测值的差异大于等于统计平均得出的强模型估计值与观测值的差异，也就是强模型有更好的模型效果。对比残差修正方法得到的 3 个残差修正模型的效果，为了结合各模型的优点，选择将

GLM-DNN 与 *GLM-CNN* 再次进行集成。本文提出两种模型集成的方式，第一种是对残差修正模型进行集成；第二种是对预测模型进行集成。

通过上述证明，可以知道对于第一种集成方法，对残差修正模型进行集成之后得到的残差修正集成模型要优于单个的残差修正模型，最后残差修正集成模型得到的预测结果要优于单个残差修正模型的预测结果；对于第二种集成方法，对预测模型进行集成之后得到的预测集成模型要优于单个的预测模型。

对于整合集成模型中各个模型的方法，一般采用平均法。平均法的优点是减少了误差，这是因为该方法分配相同的权重给各模型。而在实际情况中，每个预测模型的预测效果好坏是有差异的，对于误差较大的模型要减小其权重，对于误差较小的模型要加大其权重，在这样的情况下平均法就不太适合了。考虑到预测值与真实值存在一定的线性关系，对于两种集成方法使用不同的方式进行集成，对于第一种集成方法使用 *DNN* 进行集成，因为第一种方法是对残差模型进行集成不需要模型有解释力，而神经网络模型能够得到更好的模型效果；第二种使用线性回归进行集成，线性回归模型可以根据不同模型效果给与其不同的权重，这使得最终集成模型的效果更好，目的是在保留模型解释力的条件下，还能有更好的模型效果。

对于第一种集成方法，通过建立 *DNN* 模型，将两组模型输出的残差预测结果和真实值作为输入数据，利用深度神经网络来寻找这两个残差预测结果和真实值的联系。基于各模型的预测效果，深度神经网络进行分析，然后将两个残差预测结果进行集成作为最终的修正残差，最后将其与广义线性模型的预测值加总，获得最终的预测结果，模型流程图见 4.6。

基本步骤如下：

（1）对数据进行处理

（2）建立广义线性模型，然后建立残差序列。设原始数据为 y，残差为ε，则有$\varepsilon_i = y_G(i) - y(i)$。其中，$y_G(i)$为广义线性模型对数据的预测值，$y(i)$为实际数据，$\varepsilon_i$为残差序列。

（3）以原始数据的影响因素为自变量，残差序列为被解释变量，使用之前选择好的 *CNN* 和 *DNN* 神经网络，利用神经网络模型的优势对变量之间的关系进行挖掘，更好的拟合模型。

（4）使用（3）中训练好的 *CNN* 和 *DNN* 神经网络模型对残差进行预测，设神经网络残差预测值是ξ_{CNN}和ξ_{DNN}，再次将ξ_{CNN}和ξ_{DNN}作为解释变量，ε_i为被解释变量输入到 DNN 神经网络模型，对残差修正模型进行集成。

（5）使用（4）中训练好的残差修正集成模型对残差进行预测，设残差的预测结果为ξ_{i}。使用ξ_{i}对广义线性模型的预测值$y_G(i)$进行修正，最后得到的模型预测结果是$y(i)=y_{\mathrm{G}}(i)+\xi_{\mathrm{i}}$。

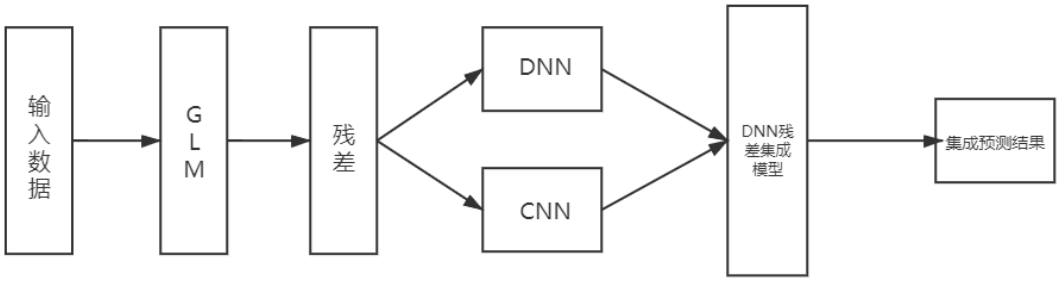

图 4.6 残差修正模型集成

对于第二种集成方法，通过建立线性回归模型，将两组残差修正模型输出的最终预测值和观测值作为输入数据，使用线性回归来探寻两个预测值和观测值的联系。通过分析各个模型预测值的效果，确定各个系数，然后将预测值进行线性组合作为最终结果，模型流程图见 4.7。

基本步骤如下：

（1）对数据进行处理

（2）建立广义线性模型，然后建立残差序列。设原始数据为 y，残差为ε，则有$\varepsilon_i=y_G(i)-y(i)$。其中，$y_G(i)$为广义线性模型对数据的预测值，$y(i)$为实际数据，$\varepsilon_i$为残差序列。

（3）以原始数据的影响因素为自变量，残差序列为被解释变量，使用之前选择好的 *CNN* 和 *DNN* 神经网络，利用神经网络模型的优势对变量之间的关系进行挖掘，更好的拟合模型。

（4）使用（3）中训练好的 *CNN* 和 *DNN* 神经网络模型对残差进行预测，设神经网络残差预测值是ξ_{CNN}和ξ_{DNN}。使用ξ_{CNN}和ξ_{DNN}对广义线性模型的预测值$y_G(i)$进行修正，得到的模型预测结果$y_{CLM-CNN}(i)=y_{\mathrm{G}}(i)+\xi_{\mathrm{CNN}}$和

$y_{CLM-DNN}(i) = y_{\mathrm{G}}(i) + \xi_{\mathrm{DNN}}$。

（5）再次将模型的预测结果$y_{CLM-CNN}$和$y_{CLM-DNN}$作为解释变量，y 为被解释变量输入到线性回归模型中，对模型进行集成。然后使用训练好的集成模型对观测值进行预测，最后的预测结果就是模型结果。

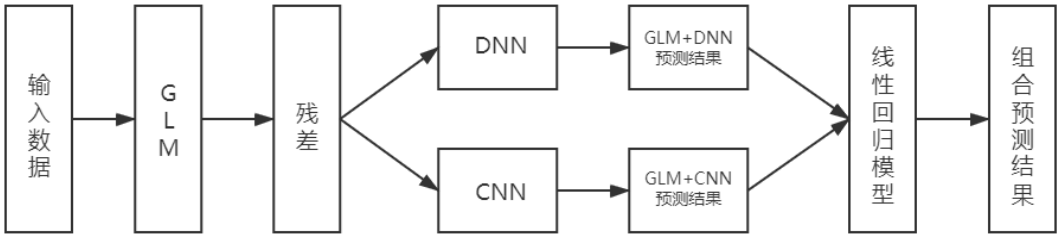

图 4.7 预测模型集成

第一种集成方法的优点是使用 *DNN* 进行集成比平均法能够获得更好的模型效果。第二种集成方法的优点是线性回归模型不仅可以给各模型分配系数，保留模型的解释力，还能提供一个常数项，以此对加权后的结果进行一个调节。

5 实证研究

5.1 数据介绍和分析

本章使用两组真实的车险数据进行数据分析，第一组数据来源于 *R* 语言 *CASdatasets* 数据包中的 *freMTPL2freq*，第二组数据来源参考文献风险模型[50]。因限本文的篇幅而且两组数据的处理方式方式相同，参考文献已经对其数据进行了详细介绍，所以本章主要介绍第一组数据，没有说明的情况下默认使用第一组数据。第一组数据集包含了法国汽车第三方责任(*MTPL*)保险组合，在一个会计年度中观察到相应的索赔计数。该数据集包含了 678013 条数据，每一条数据有 12 个变量，分别是：保单号、索赔数量、风险暴露数、地区代码、汽车功率、车龄、司机年龄、惩奖水平、汽车品牌、司机常驻人口密度、地区、汽车燃料种类。接下来分别对每一个变量进行解释：

（1）保单号(*IDpol*)：每一个保单的唯一标识；

（2）索赔次数（*ClaimNb*）:保单在记录中的索赔次数；

（3）风险暴露（*Exposure*）:以年为单位的总风险暴露；

（4）地区代码（*Area*）：分类变量，一共 6 个类别。从农村地区到城市中心；

（5）汽车功率（*VehPower*）:按分汽车功率大小类进行排序；

（6）车辆年龄（*VehAge*）： 车辆年龄，以年数表示；

（7）司机年龄（*DrivAge*）：以年分（在法国，人们可以在 18 岁时开车）；

（8）惩奖等级（*BonusMalus*）：100 表示没有奖惩，150 表示保费惩罚，也就是变为初始保费的 1.5 倍；

（9）汽车品牌（*VehBrand*）:分类变量，一共 11 个类别。按汽车品牌的大类分，不进行品牌下种类的细分；

（10）汽车燃料种类（*VehGas*）：分类变量，一共 2 个类别。汽车使用的燃料是汽油、柴油或普通汽油；

（11）司机常驻人口密度（*Density*）：汽车司机居住城市的居民密度（每平方公里居民数）；

（12）地区（*Region*）：分类变量，一共 21 个类别。法国的政策区域（基于 1970-2015 年分类）

知道了各个变量代表的意思之后，接下来就进行数据的描述统计：

表 5.1 索赔次数与保单数的关系

索赔次数	0	1	2	3	4	5	6	8	9	11	16
保单	643953	32178	1784	82	7	2	1	1	1	3	1

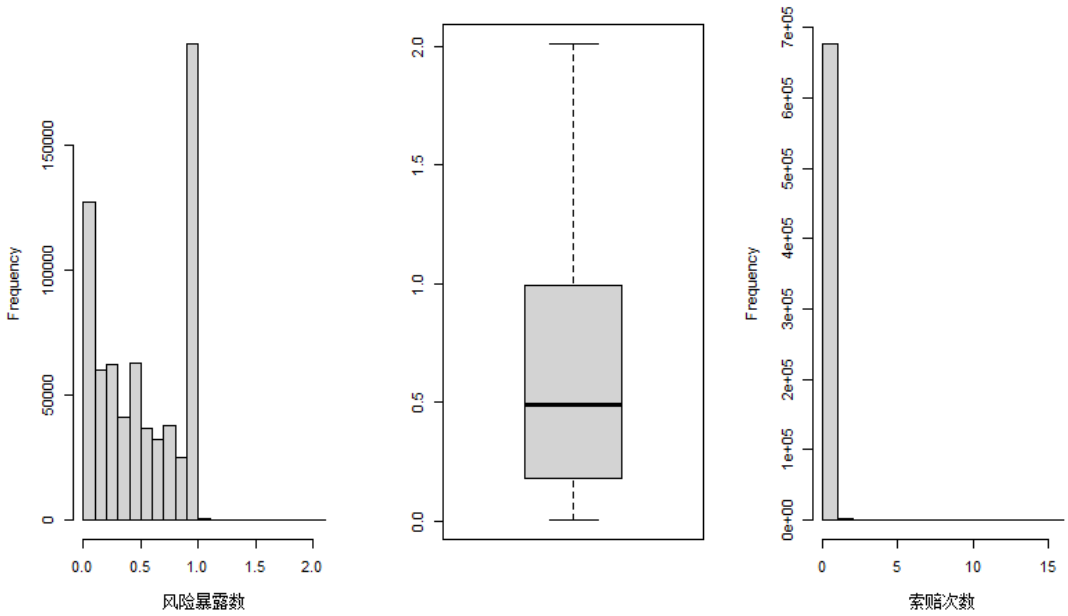

图 5.1 (左)风险暴露直方图，(中)风险暴露箱线图，(右)观察到的索赔次数

在表 5.1 中，提供了索赔次数和保单数据的概述，在图 5.1 中，提供了风险暴露数的直方图和箱线图以及观察到的索赔次数的直方图。从这些统计数据中，观察到数据有许多风险暴露数小于 1 年的暴露，有 1224 个暴露数大于 1 年的保单；有最小暴露数为 1 天的保单。对大于一年的风险暴露数进行修正(通过将它们设置为 1)，因为认为这是由于数据错误引起的(因为所有观察都在一个会计年度内)。对于索赔的次数，观察到有 4 份保单的索赔次数大于 9，最大索赔次数为 16，见表 5.1。重新定义了这些保单的索赔次数，将保单中索赔次数大于 4 的索赔次数设置为 4，因为这些大值很可能是数据错误。

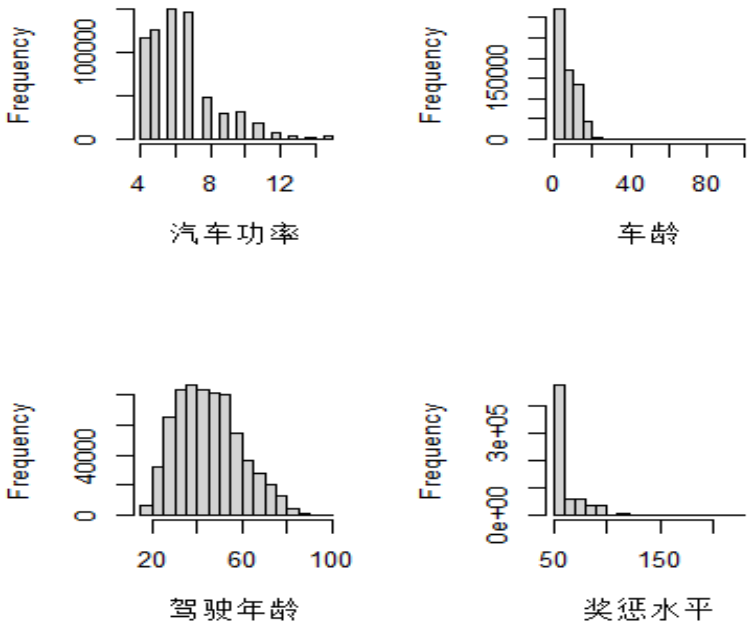

图 5.2 影响因素 1

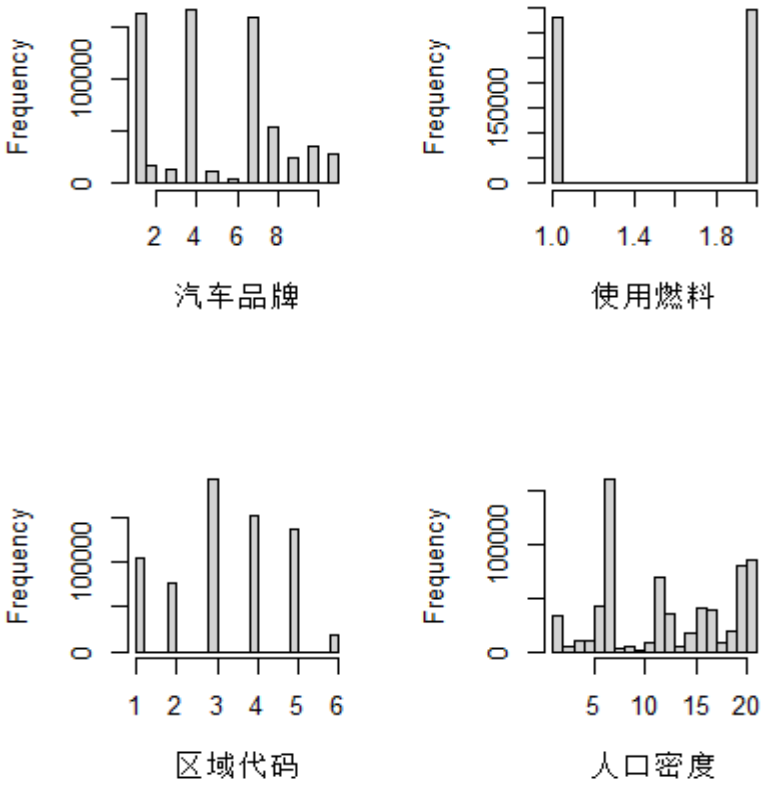

图 5.3 影响因素 2

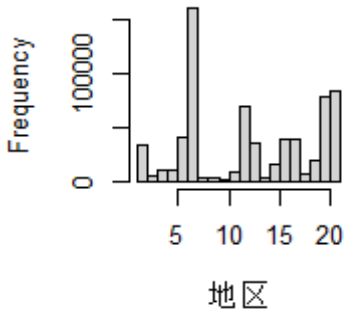

图 5.4 影响因素 3

通过图 5.2、图 5.3、图 5.4 所展示 9 个风险变量的索赔次数条形图简要分析可以得出以下结论：

（1）在汽车功率这个变量中，汽车功率小于 7 的类别索赔次数最多。在汽车功率大于 7 之后，索赔次数出现了断崖式的下降。在功率大于 12 之后，索赔次数下降趋于平缓；

（2）在车龄这个变量中，索赔次数主要集中在车龄为 20 年以前，并且主要以 10 年之前的为主。极个别出现大于 20 年的数据；

（3）在驾驶年龄这个变量中，索赔次数呈现右偏型。一开始索赔次数不大，随着驾驶年龄变大索赔次数不断变大，到达 57 左右随着驾驶年龄变大索赔次数不断变小；

（4）在惩奖等级这个变量中，可以明显的看出惩奖励等级越大，索赔次数越小，说明惩奖系统对于索赔次数有着比较明显的作用；

（5）在汽车燃料这个变量中，发现使用这两种汽车燃料的保单，其索赔次数相差不大；

（6）对于汽车品牌这个变量，可以看出来个别类别的汽车品牌，其索赔次数远远大于其他汽车品牌的索赔次数，并且索赔次数多的汽车品牌类别之间的索赔次数大体相等；

（7）对于区域代码这个变量，可以从图中看出每个区域的索赔次数相差不大；

（8）对于司机常住城市人口密度和地区这两个变量来说，都表现为部分分类变量的索赔次数远大于其他分类变量的索赔次数。

接下来给出变量的相关系数表，进行变量共线性的问题讨论。

表 5.2 变量相关系数

	地区	汽车功率	车龄	驾龄	惩奖金	人口密度
地区		0.00	-0.10	-0.05	0.12	0.59
汽车功率	-0.01		-0.10	0.03	-0.08	0.04
车龄	-0.01	0.00		-0.06	0.08	-0.09
驾龄	-0.05	0.04	-0.08		-0.48	0.00
惩奖等级	0.14	-0.07	0.08	-0.57		0.08
人口密度	0.98	-0.01	-0.10	-0.04	0.14	

表 5.2 中，左下的三角为 *Spearman* 相关系数，右上三角为 *Pearson* 相关系数。从这两个相关系数中能够发现，这两个相关系数都显示出司机常住城市的人口密度与地区成正相关，惩奖等级与驾龄成负相关。这两点不难理解，一个地区的人口多了，那么这个地区的汽车拥有量就会变多以及出租车也会变多。而一个人驾驶年龄越久，就越小心，受罚的可能性就越小。

5.2 数据处理

5.2.1 广义线性模型的数据处理

对原始数据进行如下处理：

(1)对于地区代码这个解释变量，选择将其变成连续型的特征变量。$\{A,\ldots,F\}\rightarrow\{1,\ldots,6\}$;

(2)对于汽车功率这个解释变量，对汽车功率大于等于 9 的变量进行合并，最后得到 6 个分类变量；

(3)对于车龄这个解释变量，定义了 3 个分类组$[0,1),[1,10],(10,\infty)$;

(4)对于驾龄这个解释变量，定义了 7 个如下分类组$[18,21),[21,26),[26,31),[31,41),[41,51),[51,71),[71,\infty)$;

(5)对于惩奖等级这个解释变量，定义其为连续对数线性特征分量(上限为值 150)；

(6)对于汽车品牌这个变量，将其定义为一共有 11 个类别的分类变量；

(7)对于汽车燃料这个变量，就是一个二分类的变量；

(8)对于司机常住城市人口密度这个变量，将其进行对数特征变换；

(9)对于地区这个变量，一共有21个分类组。

因此，考虑了3个连续特征成分(面积代码，惩奖金额，对数人口密度)，1个二分类特征变量(汽车燃料)和5个分类特征成分(汽车功率，车龄，驾龄，汽车品牌，地区)。对于广义线性模型，将分类特征变量转换为数值型变量。这可以通过虚拟编码来实现(虚拟编码不同于机器学习中的 *one-hot* 编码，因为后者不使用参考标签)。例如，如果通过虚拟编码转换7个分类驾驶员的年龄类别，考虑表5.3中给出的二进制表。

表 5.3 驾驶员年龄虚拟变量表

[18,21)	1	0	0	0	0	0	0
[21,26)	0	1	0	0	0	0	0
[26,31)	0	0	1	0	0	0	0
[31,41)	0	0	0	1	0	0	0
[41,51)	0	0	0	0	1	0	0
[51,71)	0	0	0	0	0	1	0
[71,∞)	0	0	0	0	0	0	1

从表5.3中，可以看出由于这7个分类驾驶员的年龄变量完全由一个6维特征向量确定，该6维特征向量只有0和至多一个等于1的分变量，因此可以用这样的6维特征向量(初始化为一个标签)来替换分类驾驶员的年龄类别。对所有分类特征组件应用虚拟编码并收集所有的数据，就能获得以下特征空间：

$$\chi \subset [1,6] \times \{0,1\}^5 \times \{0,1\}^2 \times \{0,1\}^6 \times [50,150] \times \{0,1\}^{10} \times \{0,1\} \times [0,11] \times \{0,1\}^{20}$$

也就是说，获得了一个d=1+5+2+6+1+10+1+1+20=47维的特征空间。根据目的不同，其他特征编码方案也可能是有用的，例如，用于比较给定分类特征分量的不同类别的赫尔默特对比编码。如果希望特征变量的分组之间以非乘法方式进行交互，需要不同的特征进行数据的预处理。表5.3中汽车功率、车龄和驾龄的分类类别的选择是基于之前研究学者的观点。还可以通过以下方式做出数据变量分类的选择，对于不同的特征成分能够使用回归树或者其他算法进行选取。

5.2.2 神经网络模型的数据处理

对于无序分类特征变量，通常应用虚拟编码或 *one-hot* 编码来获得其数字特征，分类变量的虚拟编码在广义线性模型的数据处理中有做介绍，这里就不在讨论了。理论上，如果选择一个足够好的神经网络模型，数据里连续型的变量就不用进行预处理。但是，在实际建的神经网络模型运算中，连续型特征变量也需要预处理，使得他们都在相似的尺度上，并且各个变量的数据在该尺度上足够均匀的分布。这是因为人们经常使用梯度下降算法进行神经网络模型的拟合。而这个算法要求所有要素组件都位于同一尺度上，否则该算法将无法正常工作（也称之为梯度问题）。因此，对所有特征变量应用适当的严格单调变换。最简单的单调变换由以下映射(称为MinMaxScaler)给出：

$$x_l \mapsto x_l^* = 2\frac{x_l - min\, x_l}{max\, x_l - min\, x_l} - 1 \in [-1,1] \quad (5.1)$$

其中，式子(5.1)中的最小值和最大值对应于x_l的最小值和最大值(假设它是有界的)。实际上，最小x_l或最大x_l可能是未知的。在这种情况下，可以选择观测数据中相应的最小值或最大值。

5.2.3 *Embedding* 算法

在机器学习算法中，对于分类变量的处理，通常使用 *one-hot* 编码来解决，该编码方法也被统计学家称作是虚拟编码。在使用这种编码方法对分类变量进行编码时候，分类特征的向量被扩展为一组指标变量，除了与出现在其类别相关的一个指标变量之外，所有指标变量都被设置为零向量的每一列。对于这样的编码，所具有的最大好处就是，不管分类变量中有多少个种类，都能使用 0 和 1 在一个一维的数组里表示出来。并且不同的种类绝对不一样，这样表达变量的能力就很强，同时容易计算。但是，因为这个算法把分类变量中的每一个种类都从新编码成新的一维数组，当分类变量特别多或者分类变量中的种类特别多的时侯，*one-hot* 编码将会产生稀疏的高维矩阵(即许多列为零)，过度占用资源。同时，因为其完全的独立，那么它表达种类之间相互的关联特征能力几乎为 0。这就会导致拟合模型的时候会有潜在困难。

在这样一个背景下，*Embedding* 算法就被提出了。*NPL* 领域很重要的发明之一就是提出了 *Embedding* 算法。它把 *one-hot* 的稀疏矩阵，通过一定的线性变换（在 *CNN* 中使用全连接层进行变换，这也被称做查表操作），成为了一个密集矩阵，并且还让相互独立的向量变成了有相互联系的向量。以文字变换为例，密集

矩阵用 N 个特征来表示所有的文字，在这个密集矩阵中，表面上代表着密集矩阵与每个字的一一对应关系，实际上还包含着字与字之间，词与词之间以及句子与句子之间的内在关系。从稀疏矩阵再到密集矩阵的过程，就叫做 *Embedding*，很多人也把它称作查表，因为它们之间的关系是一一映射的。更为重要的是，这种关系在反向传播的算法过程中，是一直不停的在更新，因此能够在多次 *epoch* 后，使得这个关系变的相对成熟，也就是能够相对正确的解释句子语义以及每个语句之间的关系。*Embedding* 层所有权重参数组合的式子就是这个相对正确的关系。

对于分类变量，使用 *Embedding* 算法而不使用 *one-hot* 编码，这能够获得更好的结果。*Embedding* 算法有两个优点：第一，与具有高维稀疏矩阵的 *one-hot* 编码相比，尺寸相对较小。第二，能够检查变量与变量之间的相似性，并能够提供额外的解释。

基于 *Embedding* 的优点，将其引入到精算领域，针对本文数据，建立两个 3 层深度神经网络，一个是基于 One-Hot 进行分类变量处理，一个是基于 *Embedding* 算法进行的数据分类变量的处理。结果如下表：

表格 5.4 分类方法的对比

网络结构	参数个数	样本内损失	样本外损失	运行时间
DNN-One Hot	8706	29.53894	30.33916	253.28
DNN-Embedding	6947	30.92316	30.7859	170.39

表 5.4 中建立的两个 3 层深度神经网络模型中各隐藏层的神经元为 64，64，32。从中能够看出 *DNN-Embedding* 的参数个数比 *DNN-One Hot* 的参数个数少，运行时间比较快。这是因为当数据量特别大的时候，模型参数过多会导致运算时间过长。综上所述，*Embedding* 算法在神经网络建模中有提高模型运算效率的作用。

从表 5.4 中可以看到 *DNN-One Hot* 有着更好的样本内损失，这得益于该模型有着更多的参数。并且样本外损失比 *DNN-Embedding* 模型也要小。也就是说两种处理方式得到的模型对于样本内外的数据， *DNN-One Hot* 更好一些。

因为本文将与广义线性模型做对比，数据的处理方式尽量一致以及 *one-hot* 有着更好的模型效果，所以对数据进行以下的处理。对于分类的变量，都使用虚

拟变量进行处理(因为 *CANN* 模型的特殊性，在 *CANN* 模型中使用 *Embedding* 进行分类变量的处理)。对于连续型的变量，对地区代码、汽车功率、车龄、驾龄、惩奖金额和司机常住城市人口密度进行 *MinMaxScaler* 处理。

5.3 传统索赔频率模型结果

广义线性模型作为基准模型，是最先选择的保险定价工具。其易于管理，并且在模型的解释方面非常容易。接下将使用广义线性模型对数据进行解释。

将数据分成训练集 *D* 和测试集 *T*，训练集有 610212 个数据，测试集有 67801 个数据。因为对于车险，假设其服从泊松分布。对于目标函数，选择泊松损失函数，函数如下：

$$\mathcal{L}(D,\lambda)=\frac{1}{n}\sum_{i=1}^{n}2N_i\left[\frac{\lambda(x_i)v_i}{N_i}-1-\log\left(\frac{\lambda(x_i)v_i}{N_i}\right)\right] \tag{5.2}$$

对于广义线性模型来说，如果知道到某一个解释变量的回归系数是已经知道的，那么就能够将其设定成为抵消项，也就是说，广义线性模型的线性预测项可以写成是：

$$\eta_i = x_i^T\beta + offset \tag{5.3}$$

在使用对数连接函数的情况下，广义线性模型的均值预测值能够写成是：

$$\mu_i = exp(x_i^T\beta + offset) \tag{5.4}$$

在车险数据中，索赔频率等于索赔次数y_i和车年数n_i之比。本文的数据中暴露数可以看成是车年数。在广义线性模型中连接函数是对数函数的情况下，索赔频率模型可以表示成：

$$\ln\frac{\mu_i}{n_i} = x_i^T\beta \tag{5.5}$$

其中，μ_i代表数据中第 *i* 个风险类别的期望索赔次数，μ_i/n_i表示期望索赔频率。式子（5.5）经过简单变换之后，能够得到如下：

$$\ln\mu_i = \eta_i = x_i^T\beta + \ln n_i \tag{5.6}$$

在式子（5.6）中，所谓的抵消项就是$\ln n_i$，在本文中，把暴露数当成为抵消项。在式子（5.5）中，因变量是索赔频率，而在式子（5.6）中，因变量是索赔次数。但是，这两个模型是等价的，也就是说它们的参估计值都β是完全相同的。

由此可以看出，抵消项能够看作为一项无需估计它参数的已知项。而抵消项的作用只是相当于对广义线性模型的截距项进行了平移，因此只对其截距项的估

计值能够产生影响，对于模型的其他参数估计值和拟合值都没有任何影响。而对模型截距项的影响，能够通过之后的模型对其进行修正。

在有抵消项的情形下，期望索赔次数将与车年数（暴露数）成比例，也就是：

$$\mu_i = n_i \times exp(x_i^T \beta) \quad (5.7)$$

上述式子表明，期望索赔次数等于期望索赔频率与车年数（暴露数）的乘积，而$exp(x_i^T \beta)$则表示索赔频率。由此可见，在索赔次数模型中引入抵消项，等价于用车年数（暴露数）加权。

将广义线性GLM1模型拟合到训练数据集D中。结果如下表5.5所示：

表 5.5 广义线性模型GLM1结果

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-3.9855092	0.0385681	-103.337	< 2e-16 ***
VehPowerGLM5	0.1814126	0.0191785	9.459	< 2e-16 ***
VehPowerGLM6	0.2237008	0.0190419	11.748	< 2e-16 ***
VehPowerGLM7	0.1298414	0.0188687	6.881	5.93e-12 ***
VehPowerGLM8	-0.0797704	0.0285639	-2.793	0.00523 **
VehPowerGLM9	0.2090839	0.0210762	9.920	< 2e-16 ***
VehAgeGLM1	1.1806886	0.0170797	69.128	< 2e-16 ***
VehAgeGLM3	-0.2040749	0.0140632	-14.511	< 2e-16 ***
DrivAgeGLM1	0.0719483	0.0448778	1.603	0.10889
DrivAgeGLM2	-0.2982647	0.0277763	-10.738	< 2e-16 ***
DrivAgeGLM3	-0.4301756	0.0231830	-18.556	< 2e-16 ***
DrivAgeGLM4	-0.2944111	0.0166332	-17.700	< 2e-16 ***
DrivAgeGLM6	-0.0668375	0.0149825	-4.461	8.16e-06 ***
DrivAgeGLM7	0.0037769	0.0236307	0.160	0.87301
BonusMalusGLM	0.0229107	0.0003562	64.319	< 2e-16 ***
VehBrandB10	0.0148531	0.0389361	0.381	0.70285
VehBrandB11	0.0897171	0.0417910	2.147	0.03181 *
VehBrandB12	0.1237492	0.0184915	6.692	2.20e-11 ***
VehBrandB13	0.0329271	0.0436955	0.754	0.45111
VehBrandB14	-0.1141285	0.0826650	-1.381	0.16740
VehBrandB2	0.0026290	0.0161538	0.163	0.87071
VehBrandB3	0.0118635	0.0230092	0.516	0.60614
VehBrandB4	0.0033196	0.0313467	0.106	0.91566
VehBrandB5	0.0825986	0.0261558	3.158	0.00159 **
VehBrandB6	-0.0061362	0.0301808	-0.203	0.83889
VehGasRegular	0.0701407	0.0119754	5.857	4.71e-09 ***
DensityGLM	0.0319770	0.0126584	2.526	0.01153 *
RegionAlsace	-0.0659157	0.0898473	-0.734	0.46317
RegionAquitaine	-0.1620067	0.0313068	-5.175	2.28e-07 ***

续表 5.5 广义线性模型GLM1结果

RegionAuvergne	-0.3763319	0.0776813	-4.845	1.27e-06 ***
RegionBasse-Normandie	-0.0466064	0.0433273	-1.076	0.28207
RegionBourgogne	-0.0618216	0.0475883	-1.299	0.19391
RegionBretagne	0.0274194	0.0232686	1.178	0.23864
RegionChampagne-Ardenne	0.0530132	0.0846385	0.626	0.53109
RegionCorse	0.0158941	0.0697132	0.228	0.81965
RegionFranche-Comte	-0.1796346	0.1406210	-1.277	0.20145
RegionHaute-Normandie	-0.1282915	0.0612606	-2.094	0.03624 *
RegionIle-de-France	-0.1404677	0.0246800	-5.692	1.26e-08 ***
RegionLanguedoc-Roussillon	-0.1203208	0.0298662	-4.029	5.61e-05 ***
RegionLimousin	0.1018224	0.0672010	1.515	0.12972
RegionMidi-Pyrenees	-0.2046846	0.0426690	-4.797	1.61e-06 ***
RegionNord-Pas-de-Calais	-0.2270013	0.0282004	-8.050	8.31e-16 ***
RegionPays-de-la-Loire	-0.0818968	0.0263491	-3.108	0.00188 **
RegionPicardie	-0.0152729	0.0522428	-0.292	0.77002
RegionPoitou-Charentes	-0.1166341	0.0355628	-3.280	0.00104 **
RegionProvence-Alpes-Cotes-D'Azur	-0.0938449	0.0218867	-4.288	1.80e-05 ***
RegionRhone-Alpes	0.0085635	0.0197477	0.434	0.66455
AreaGLM	0.0174052	0.0169782	1.025	0.30529

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

表 5.6 广义线性模型GLM1结果的方差分析

	Df	Deviance
VehPowerGLM	5	142.9
VehAgeGLM	2	4969.2
DrivAgeGLM	6	1020.0
BonusMalusGLM	1	3741.1
VehBrand	10	58.5
VehGas	1	60.7
DensityGLM	1	134.1
Region	20	181.7
AreaGLM	1	1.1

从表5.5中，对输出的结果详细分析表明，除了地区代码之外，所有考虑的特征分量都是重要的。如果把地区代码看作一个分类变量而不是一个连续变量，也会得到同样的结果。从表5.6中，可以看出通过一个接一个地添加一个特征组来减少样本内损失。它还显示，在已经包含所有其他功能组件之后，不需要地区代码这个特征变量。由此得出结论，可能会在广义线性模型中删除地区代码这个特征变量，这并不令人惊讶，因为该特征变量与司机常住城市的人口密度特征变量有很强的共线性。同样，也要探讨是否可以删除汽车品牌这个解释变量，因为其在表5.6中，因为其仅在样本内损失中提供相对小的效果。所以考虑以下三种模式，同时建立过离散模型和广义可加模型进行比较。

（1）*GLM*1：在建立广义线性模型的时候，考虑所有的解释变量；

（2）*GLM*2：相对于*GLM*1，在建立广义线性模型的时候，去除地区代码这个解释变量；

（3）*GLM*3：相对于*GLM*1，在建立广义线性模型的时候，去除地区和汽车品牌这两个解释变量。

表 5.7 传统索赔频率模型结果对比

	AIC	样本内	样本外
*GLM*1	254284	31.40828	30.91210
GLM2	254283	31.40845	30.91210
GLM3	254332	31.41983	30.92185
过离散模型		31.39702	30.89507
广义可加模型		31.27437	30.76346

在表5.7中给出了这五个模型的结果。从表5.7中先对比样本外损失结果，

可以看出这五个模型中前四个模型结果相差不大，广义可加模型的效果是最好的。这是因为广义可加模型结构的特性，能够更好的对数据进行拟合。虽然过离散模型的效果比三个广义线性模型的效果好，但是样本内外的损失函数相差不大。第三个模型与前两个有点差距，但是第一个模型与第二个模型差距很微小。得出这样的结果并不奇怪，因为如果差异太大，模型就会是一个过度拟合的迹象。考虑到 *Akaike* 的信息准则(*AIC*)，其引入了对过度拟合的惩罚项(以模拟样本外的损失结果)，具有最小 AIC 值的模型应该是优选的。在模型中，第二个模型的 *AIC* 准则略微好点。然而，在样本外的损失结果方面，发现第二个模型与第一个模型一样。但是，在样本内的损失结果方面，第一个模型比第二个模型要好。由此可以得出，第一个模型与第二个模型具有相同的样本外性能，同时，第一个模型样本内的性能更好。因此，在这里无法从 AIC 和样本内外的损失结果分析中获得明确的建议。同时考虑之后的分析，将使用第一个模型作为基准模型。从表 5.7 中也能够看出，第三个模型相对于其他两个模型是没有竞争力的。

直接给出第二组数据的传统索赔频率模型对比结果：

表 5.8 传统索赔频率模型对比

模型	MSE	泊松偏差
泊松广义线性模型	11252.63	26.35094
过离散模型	18346.39	26.97199
广义可加模型	10654.39	26.35094
Lasso-GAMLSS	12164.19	26.35094

通过表 5.8 可以得出，对于泊松偏差来讲，除了过离散模型之外，其它几个模型的效果一样，并且与过离散模型的效果相差不大。考虑 *MSE*，广义线性模型的效果与广义可加模型效果相差不大，过离散模型模型表现的并不好，而新建立的 *Lasso-GAMLSS* 模型表现的并没有非常好，这与使用数据的变量并不是很多有关。由此，通过传统索赔频率模型在两组数据上的建模效果来说，在将传统索赔频率模型与神经网络模型进行结合的时候，选择将广义线性模型与神经网络模型进行结合。

5.4 神经网络模型

5.4.1 *BP* 神经网络模型结果

BP 神经网络是目前学者们用的最多的神经网络，本文选取的神经网络迭代次数为 100、批量数为 10000。为了与之前建立的广义线性模型变量保持一直，选取除了风险暴露变量之外的所有变量进行分析，最终获得如下结果。

表格 5.9 *GLM* 与 *BP* 神经网络模型结果对比

	AIC	样本内	样本外
GLM1	254284	31.40828	30.91210
GLM2	254283	31.40845	30.91210
GLM3	254332	31.41983	30.92185
过离散模型		31.39702	30.89507
广义可加模型		31.27437	30.76346
BP 神经网络		31.23728	30.59765

通过表 5.9，能够看出 *BP* 神经网络的样本内结果比广义线性模型、过离散模型和广义可加模型的样本内结果效果好，这表明了 *BP* 神经网络模型在训练数据集上拟合效果好。考虑样本外损失函数，也能发现 *BP* 神经网络的结果要好，这表明该模型的泛化能力好。广义可加模型样本内损失函数的结果与 *BP* 神经网络模型的结果相差不大，这是因为广义可加模型有非线性的结构，能够很好的拟合数据，但是该模型的泛化能力相对差一些。综上所述，*BP* 神经网络模型的效果要好于广义线性模型、过离散模型和广义可加模型的效果。

5.4.2 改进的 *BP* 神经网络结果

对 *BP* 神经网络的改进主要是使用了泊松损失函数进行神经网络的迭代，改进了其损失函数。虽然泊松损失函数并不是常见的几种损失函数，但是实际的数据往往是复杂的，这就需要选择适当的损失函数进行建模。而本文的数据是车险数据，对于车险数据的索赔频率进行建模，通常对车险的索赔频率数据采用泊松模型假设。自然选择在这个模型假设下的目标函数就是泊松偏差损失了。

为了避免我们的主观性而造成模型不精确的结果，建立了泊松损失函数的神经网络模型和平方损失函数的神经网络进行对比。结果如下表：

表 5.10 损失函数的不同

损失函数	周期	批量数	样本内损失	样本外损失
BP 神经网络	100	10000	31.23728	30.59765
改进神经网络	100	10000	31.20338	30.51043

定义了两个全连接前反馈的神经网络模型，除了使用不同的损失函数，为这两个神经网络模型提供了相同的初始参数。从表 5.10 中可以观察出，对于模型拟合来说，改进的 *BP* 神经网络模型要比 *BP* 神经网络模型效果好点，该模型的样本内损失要小于未改进的；对于模型的泛化能力来说，同样是改进的 BP 神经网络样本外的损失函数小。综上所述，改进的 *BP* 神经网络模型更符合车险索赔次数建模。

5.4.3 *DNN* 神经网络

（1）选取最优算法

希望找到足够好的估计参数方法，它不光在训练样本中有良好的模型解释力，在其他数据集上也具有良好的性能(模型的泛化能力)。但是在实际运用中，通常在神经网络模型的算法优化中存在大量的工作，这就需要我们对于数据集选择出一个相对好的算法，从而进行神经网络模型的算法优化。对比几种在神经网络建模时候常用的模型优化算法，并且将其运用到 *epochs*=100，*batch size*=10000 的全连接前反馈深度神经网络模型中，该模型使用的是双曲线正切激活函数和指数激活函数的输出层，比较每个算法在模型里的结果，从而选出一个适合的优化算法。

表 5.11 *DNN* 优化算法的不同

优化算法	周期	批量数	样本外损失	样本内损失
nadam	100	10000	31.31618	30.56957
sgd	100	10000	31.65830	30.87616
adagrad	100	10000	32.50806	31.66138
rmsprop	100	10000	31.20338	30.51043
adam	100	10000	31.24727	30.51727
adamax	100	10000	31.39768	30.66738

对于表 5.11 中每个优化算法，都使用了相同的初始参数(这为算法提供了合理的起始值)。从表 5.11 中可以知道，改进的算法，如“*rmsprop*”或“*nadam*”，比普通的随机梯度下降法“*sgd*”提供了相对比较好的收敛结果。同时，随机梯度下降法相对于改进的算法需要用的时间也更久一点，这是因为随机梯度下降法用于学习速率等。因此，对于之后的神经网络模型将进行指定使用“*rmsprop*”这个神经网络模型优化算法。

（2）神经网络模型训练次数的选取

周期（*epochs*）代表使用神经网络模型遍历整个训练数据的次数，而批量（*batch size*）表示在每一次使用神经网络模型遍历整个训练数据时候，每一步需要用到多少个子样本。由此可以看出，如果预先设定的批量大小恰好等于观测数n，那么在神经网络模型的一个周期里只需要做一步就能得出结果。而如果预先设定的批量大小恰好等于 1，那么在神经网络模型的一个周期里就需要做 n 步，这样才能得出结果。需要值得注意的方面是，在大数据样本的时候，需要预先设定比较小的批量。这是因为，在大数据样本中将会有许多的观测值，要想把所有数据的梯度有效的计算出来，这是不太可行的。所以，将整个数据集随机分为若干个小批量。

表 5.12 *DNN* 批量数的不同

优化算法	周期	批量数	算法步骤	总时间	平均步长时间	样本内损失	样本外损失
rmsprop	10	610212	1	8.96s	0.8960s	94.71158	94.55170
rmsprop	10	122043	5	8.86s	0.1772s	35.05282	34.36937
rmsprop	10	61022	10	8.41s	0.0841s	31.91518	31.07715
rmsprop	10	12205	50	8.69s	0.0174s	31.47842	30.71334
rmsprop	10	6103	100	11.15s	0.0112s	31.43510	30.68765
rmsprop	10	1221	500	20.83s	0.0042s	31.35428	30.65733
rmsprop	10	611	1000	30.31s	0.0030s	31.36630	30.66361
rmsprop	10	123	5000	93.84s	0.0019s	31.43589	30.65131

选择不同批量大小的神经网络模型来进行分析，比对不同批量的大小对于模型精度的影响，同时还考虑了不同批量对于模型运算时间影响的问题。在比较不同批量结果的时候，使用相同优化器 *rmsprop* 并且具有相同初始参数的全连接深度神经网络进行建立模型。从表 5.12 中可以观察出，对于最大批量 n = 610212 的模型，能够在一个周期内完成一个算法步骤。而对于批量为 123 的模型，能够在一个周期内完成 5000 个算法步骤。对于最大批量，需要计算整个数据集的梯度，因此需要 0.8960 秒。而对于批量为 123 的模型，计算 123 个样本的梯度，平均需要 0.0019 秒。由此可见，后者当然要快得多，但是另一方面，在一个周期内，需要计算 5000 个批量样本的梯度来遍历整个数据集。这就会导致每个周期需要花费 9.3840 秒。所以，对于最后的模型总用时来说，批量为 123 的模型比最大批量的模型运算时间长了 10 倍多。由此得出，对于模型的总运行时间来

说，需要在批量大小与需要执行的每个周期内算法步骤数量之间进行权衡，以便找到一个适合的批量，让模型在一定的时期内遍历整个数据集。在预设的批量中，模型的最小运算时间为8.41秒(10个周期)，批量是61022。

在表5.12中，预先设定遍历整个数据集需要的周期为10，对于不同的批量模型，将其运行模型的时间与最后得出模型的拟合质量相比较。通过观察表5.12，可以看出在周期都是10的不同批量神经网络模型下，最佳样本外性能的模型是在批量为1221下实现的，该批量模型对应每个周期500个算法步长以及模型的总运行时间为20.83秒。在较大批量和较大批量之间进行权衡。如果模型的批量数太小，那么模型会考虑太多具有不均衡数据特性的批次。模型的调参在于微调批量数、在模型运行时间与性能之间找到平衡点。

表 5.13 *DNN*不同周期数

优化算法	周期	批量算法	算法步数	样本内损失	样本外损失
rmsprop	104	610212	1	31.96803	31.14705
rmsprop	105	122043	5	31.35021	30.65125
rmsprop	112	61022	10	31.29303	30.58821
rmsprop	108	12205	50	31.06005	30.38722
rmsprop	85	6103	100	31.02939	30.36155
rmsprop	45	1221	500	31.20536	30.54661
rmsprop	31	611	1000	31.15303	30.44365
rmsprop	10	123	5000	31.43589	30.65131

在表5.13中，提供了与上一个表类似的分析，这次控制模型的批量数与表5.12中的一样，但是调整相应的周期，以此来对比不同的模型结果。从表5.13中可以看出，在固定了批量数之后，增加模型的周期数，这样都使得模型结果有了更好的效果。尤其是对于批量数为610212的模型来说，增加了模型的周期数大幅度的增加了模型的性能。以样本内外的损失函数来说，建立的神经网络模型的最佳批量数大小大约是在6103左右。这是因为，当模型的批量数在12205时候，模型的样本内损失函数是31.06005。当模型的批量数在6103时候，模型的样本内损失函数是31.02939。而当模型的批量数在1221时候，模型的样本内损失函数是 31.20536。模型的样本内损失函数呈现一个先递减后递增的趋势。所以，之后选取批量数为10000。

将第二组数据建立的DNN模型与传统索赔频率模型比较，结果如下：

表 5.14 索赔频率模型与 DNN 模型对比结果

模型	MSE	泊松偏差
泊松广义线性模型	11252.63	26.35094
过离散模型	18346.39	26.97199
广义可加模型	10654.39	26.35094
Lasso-GAMLSS	12164.19	26.35094
DNN	8868.442	26.57702

通过表 5.14 可知，对于泊松偏差而言，虽然 DNN 模型的效果要略差于传统的索赔频率模型，但是结果相差不大。而且考虑 MSE，DNN 模型比其它几个模型的 MSE 小的多，说明神经网络模型比其它几个模型的效果好。通过两组数据都说明神经网络模型比传统索赔频率模型效果好。将神经网络与传统索赔频率模型结合来改进传统索赔频率模型是可行的。

5.4.4 一维 *CNN* 与 *LSTM*

表 5.15 一层 *CNN* 不同激活函数和池化层的对比

激活函数	池化层内核数	样本内损失	样本外损失
Tanh	7	31.45461	30.76302
Relu	7	31.36843	30.67415
Relu	5	31.48496	30.81307

通过表 5.15 可以看出，池化层都是 7 的情况下，激活函数是 *Relu* 的 CNN 比激活函数是 *Tanh* 的 *CNN* 样本内外的损失函数小，这表明使用 *Relu* 激活函数的 *CNN* 有着更好的模型效果。而对于激活函数都是 *Relu* 的情况下，池化层内核数选择 7 比选择 5 有着更小样本内外的损失函数。基于此，选择激活函数为 *Relu* 以及池化层内核数为 7。

表 5.16 *LSTM* 模型结果

周期	样本内损失函数	样本外损失函数
10	32.11452	31.31662
30	33.76792	32.77956

通过观察表 5.16 发现 *LSTM* 模型的效果不如广义线性模型的效果好，并且模型的运行时间较长、很早就出现了过拟合情况。这可能是由数据特性和模型结构导致的。*LSTM* 模型因为其独特的门结构适合观测值间有一定关系的数据，而车险数据并不像时序数据一样有着很强的相互关系，因此的到的模型结果并没有广义线性模型的好。基于此，接下来的模型比较中将剔除 *LSTM* 模型。

5.4.5 改进的 *BP* 神经网络与深度神经网络对比

将建立改进的 *BP* 神经网络模型和深度神经网络模型对数据进行建模，从而比较出那个模型更适合于我们的数据集。

在之前讨论的影响神经网络模型因素方面，没有讨论神经网络结构的参数，也就是神经网络有多少个神经元。并不代表其不重要，这是因为一个神经网络模型中隐藏的神经元越多，对样本损失的微调就越多。而隐藏的神经元越多，它们执行的任务就越多，并且在隐藏的神经元中诱发的异质性就越多。但是，如果神经元太少，那么就会在第一隐藏层中会丢失太多信息。通常一个好的选择是第一隐藏层的神经元大于输入的变量数。

对于所有的模型，选择相同的批量数，其大小为 10000；使用 *Keras* 中提供的相同优化算法；每层设置相同的隐藏神经元；使用相同的标准进行数据的初始化。模型的最终结果如表 5.17 所示。

表 5.17 网络结构不同

模型	周期	样本内损失	样本外损失
改进 *BP*	100	31.16068	30.46526
DNN-2 层	100	30.99971	30.57733
CNN-2 层	100	31.03148	30.42279
DNN-3 层	100	30.68862	30.33634
DNN-4 层	100	30.41888	30.17696
改进 *BP*	200	30.96445	30.36851
DNN-2 层	200	30.59085	30.43962
CNN-2 层	200	30.59713	30.01586
DNN-3 层	200	30.10133	30.35148
DNN-4 层	200	29.84240	30.33229
改进 *BP*	300	30.89281	30.30904
DNN-2 层	300	30.05414	30.13674
CNN-2 层	300	30.53096	30.05111

续表 5.17 网络结构不同

DNN-3 层	300	29.53894	30.33916
DNN-4 层	300	29.22989	30.44069
改进 *BP*	400	30.60932	30.25013
DNN-2 层	400	29.89533	30.09274
CNN-2 层	400	31.34400	30.85425
DNN-3 层	400	29.00050	30.58517
DNN-4 层	400	28.78125	30.85203

从分析表 5.17 的改进 *BP* 神经网络开始。从表中可以看到，对于 400 个周期的改进 *BP* 神经网络在数据建模结果方面似乎具有最佳的样本内和样本外效果。可以看出，在一定条件下，模型的精度是随着周期数增加而增加的。但是，这并不代表我们可以设置很大的周期数。这是因为当设置很大的周期数时候，就需要花费更多的时间去进行拟合。如果数据集很大并且模型结构复杂这将花费我们巨大的时间。并且，当周期数过大的时候，模型就不光光只是浪费时间的问题了，甚至还会出现模型过拟合的问题。这个问题不光是周期数会给模型带来的问题，神经网络模型的结构也会带来这样的麻烦。虽然在我们建立的改进 *BP* 网络模型上并没有出现过拟合的问题，但是，并不代表接下来的深度神经网络模型不会出现这种问题。基于此，接下来将开始讨论深度神经网络模型。

如果考虑 2 个隐藏层的深度神经网络，这样就会得到一个比改进 BP 神经网络更好的模型结果。通过观察不同周期的 2 层神经网络模型可以发现，其模型的效果也是随着周期数不断增加而变好的，并且在周期数达到 400 的时候，样本外的损失并没有增加还是在降低。如果使用周期数为 400 的 2 层神经网络模型，相对于其他周期数的 2 层模型来说，能够获得的损失会更小，从这个意义上说，模型效果它更好。然而，深度神经网络需要较多的运行时间，因为它需要运算更多的梯度。在具有 2 个隐藏层的深度神经网络模型里，首先，用第一个隐藏层的神经元分离输入变量的所有特性，其次，将该信息压缩到第二个隐藏层的神经元，让其相互作用。这种压缩参数的方法通常用于深度神经网络模型，但它并不完全成立。因为，当有许多复杂的相互作用，可能需要第二个隐藏层的神经元大于第一个隐藏层的神经元，以此在模型校准中获得更为良好的收敛特性。

接着考虑 3 个隐藏层的深度神经网络模型。发现随着周期数不断增加，3 个隐藏层深度神经网络模型的样本内效果不断变好，但是，样本外的效果先增加后

降低。这就出现了之前所说的模型过拟合问题。改进的 *BP* 神经网络模型和 2 层的神经网络模型没有出现模型过拟合的情况可能是因为模型预设的神经元多，层数少的神经网络在 400 周期内没有机会出现模型过拟合的情况。

最后，考虑具有 4 隐藏层的神经网络模型。对于周期数为 100 的时候，发现 *DNN*-4 模型的效果是建立的深度神经网络模型中效果最好的。可以看到 *DNN*-1 最好的模型效果为周期数 400 的时候，这个时候的 *DNN*-1 的模型效果还是比 *DNN*-4 的 100 周期模型效果差点。接着观察到随着周期数增加，4 个隐藏层的神经网络模型是所有模型里面样本内效果最好的，但是，它也出现了模型过拟合现象，它样本外模型效果也是先增加后减少，而且出现过拟合现象更早，在周期为 300 的时候就出现了。

表 5.17 中使用的两层 *CNN* 池化数为 7 和 2，激活函数为 *Relu*。通过观察表 5.17 能够得出，随着周期数不断增加，模型的样本内外的损失函数先减少后增加，模型效果要优于广义线性模型。在 200 周期的时候，模型的样本外损失函数最小，模型的泛化能力最好。在 300 周期的时候，模型的样本内损失函数最小，有着较好的模型效果。在 400 周期的时候，模型出现过拟合的情况。

理论上只要浅层神经网络模型的周期数够大，该模型也是足够的。然而，深度神经网络模型允许用更少的神经元实现更多的建模灵活性，并且根据潜在的问题，能够获得更好的收敛行为。况且，当浅层神经网络的周期数特别大的时候，其运算时间并不会比在周期数较小的深度神经网络运算时间快。所以在建立模型的时候，更倾向于深度神经网络模型。

5.5 残差修正模型和集成模型

5.5.1 残差修正模型

经过建立广义线性模型以及对相关数据的预测之后，能够得到残差结果，将其确定为神经网络模型的因变量。经过上边的初步选择，大体上确定了在组合模型中使用深度神经网络模型。但是为了严谨，还是做了改进之后的 *BP* 神经网络，结果输出如下表：

表 5.18 残差修正模型效果

模型	周期	样本内损失	样本外损失
GLM-改进 *BP*	100	31.37015	30.88197

续表 5.18 残差修正模型效果

GLM-DNN-2层	100	30.85707	30.51109
GLM-CNN-2层	100	31.25737	30.79895
GLM-DNN-3层	100	31.22245	30.86669
GLM-DNN-4层	100	30.88045	30.56228
CANN(CNN)-2层	100	30.58679	30.41374
CANN(GAM)-3层	100	31.04971	30.51198
CANN(DNN)-4层	100	30.75151	30.42003
GLM-改进 *BP*	200	31.33011	30.87902
GLM-DNN-2层	200	30.89661	30.68386
GLM-CNN-2层	200	31.06904	30.62026
GLM-DNN-3层	200	31.17849	30.82141
GLM-DNN-4层	200	30.73015	30.80799
CANN(CNN)-2层	200	30.38884	30.40296
CANN(GAM)-3层	200	30.77764	30.26941
CANN(DNN)-4层	200	30.61834	30.24212
GLM-改进 *BP*	300	31.30231	30.87407
GLM-DNN-2层	300	30.63964	30.48265
GLM-CNN-2层	300	31.02667	30.56790
GLM-DNN-3层	300	30.46061	30.42904
GLM-DNN-4层	300	30.50762	30.43331
CANN(CNN)-2层	300	30.25264	30.27645
CANN(GAM)-3层	300	30.6311	30.18716
CANN(DNN)-4层	300	30.48451	30.22477
GLM-改进 *BP*	400	31.28918	30.95687
GLM-DNN-2层	400	30.56870	30.44036
GLM-CNN-2层	400	30.8901	30.5106
GLM-DNN-3层	400	30.55891	30.42775
GLM-DNN-4层	400	30.21804	30.48231
CANN(CNN)-2层	400	30.25711	30.66106
CANN(GAM)-3层	400	30.63837	30.23012
CANN(DNN)-4层	400	30.45642	30.27709
广义线性模型	-	31.40828	30.91210

对于表5.18，先考虑*GLM*-改进*BP*残差修正神经网络。对于*GLM*-改进BP残差修正神经网络模型来说，样本内的损失都是小于广义线性模型的，也就是说，对于训练数据来说，*GLM*-改进*BP*残差修正神经网络模型的效果比广义线性模型好。但是，在考虑样本外的损失时候，当*GLM*-改进*BP*残差修正神经网络模型周期数过大的时候，就会出现过拟合的现象。这个时候，模型效果的提升也不是很明显，还要承受模型过拟合的问题。所以，不建议选取浅层神经网络对广义线性模型的残差进行拟合。

接着考虑 *k*=2 的 *GLM-DNN* 残差修正神经网络模型。对比 *GLM*-改进 *BP* 残差修正神经网络模型，两层的 *GLM-DNN* 残差修正神经网络模型不管在样本内还是样本外都有很好的效果。这说明 *DNN* 神经网络模型比改进 *BP* 神经网络模型好。在考虑周期数方面，虽然在 200 周期的时候模型效果变差，但是随着周期数增加样本内外的损失函数还是降低了。这说明对于 2 层的 *GLM-DNN* 残差修正神经网络模型来说，增加周期数还是能够使模型的效果变好。

考虑 *k*=3 的 *GLM-DNN* 残差修正神经网络模型。对比 2 层的 *GLM-DNN* 残差修正神经网络模型来说，虽然 3 层的模型在周期数较小的时候，模型效果表现较差，但是当周期数变大之后，模型效果还是相对较好的。对于周期数不断增加，3 层的 *GLM-DNN* 残差修正神经网络模型的样本内损失函数出现了先降低在升高的情况。在周期数为 300 的时候，样本内损失函数最低。但是，对于样本外的损失函数来说，一直随着周期数增加而不断降低。虽然在 400 周期的时候样本外的损失函数比 300 周期的时候小，但是差距微小，样本内的损失函数差距比它大，所以选择 300 周期。

考虑 *k*=4 的 *GLM-DNN* 残差修正神经网络模型。4 层的 *GLM-DNN* 残差修正神经网络模型也是随着周期数的增加，样本内的损失函数减小的。而对于样本外的损失函数，出现了先减少后增加的过拟合现象。对比浅层的残差修正神经网络模型和 2 层的 *GLM-DNN* 残差修正神经网络模型，4 层的 *GLM-DNN* 残差修正神经网络模型在周期数较大的时候，模型的效果总是优于这两个模型。对比 3 层的 *GLM-DNN* 残差修正神经网络模型来说，周期数为 400 的时候 3 层和 4 层都出现了过拟合的现象，而在周期数为 300 的时候，3 层深度神经网络模型样本内外的损失函数都小于 4 层深度神经网络模型的损失函数。综上所述，对于组合模型来说，建议选择 3 层 300 周期数的深度神经网络模型。

接着考虑 *CANN（CAM）*-3 层和 *CANN（DNN）*-4 层的神经网络，因为这两个神经网络的特点，使的其在运算时间方面比深度神经网络快速。通过观察能够发现 *CANN（DNN）*-4 层模型的拟合效果一直随着周期数不断增加而增加，在周期数从 300 到 400 的时候，增加幅度已经微小了，因为样本内的损失函数变化不大。而对于样本外的损失函数来说，在周期数 300 之前一直降低，说明模型效果好，周期数到了 400 之后，出现了过拟合现象。由此可以得出，*CANN（DNN）*

-4层模型在300周期的时候模型效果最好。对于*CANN（CAM）*-3层神经网络模型来说，随着周期数不断增加，样本内损失函数先减小后增大，样本外损失函数也是一样的趋势，在300周期的时候模型效果最好。并且，样本外的损失函数低于样本内的损失函数，说明模型有比较好的泛化能力。对比两个模型可以看出，*CANN（CAM）*-3 层的样本内损失函数比较大，说明模型在样本内效果不如*CANN(DNN)*好，但是对于样本外损失函数来说，*CANN（CAM）*-3层模型结果相对要小，说明该模型具有比较好的模型泛化能力。

而对于*CANN（CNN）*-2层模型，因为*0ne-hot*算法比*Embedding*算法在本文选择的数据上效果好并且因为*CNN*结构的限制，所以该模型只有2层以及选择*one-hot*算法处理。该模型样本内外的损失大小随着周期数不断增加先变小再增加，在周期数为300的时候模型效果最好。与*CANN（DNN）*-4层模型相比在周期数为100的时候，*CANN（CNN）*-2层模型效果好，随着周期数不断增加，*CANN（CNN）*-2层模型的样本内损失一直小于*CANN（DNN）*-4层模型，说明该模型在本样内有很好的效果。但是，对于样本外损失来说，该模型样本外损失随着周期数不断增加逐渐大于*CANN（DNN）*-4层模型，说明该模型的模型泛化能力差。这可能是由于该模型的结构导致

在周期数为300的时候，对比*CANN（DNN）*-4层模型与3层*GLM-DNN*残差修正神经网络模型，发现3层深度神经网络模型的模型拟合效果好，但是模型的泛化能力稍显不足。对于样本外的微小差值，并不影响最后选择*GLM-DNN*残差修正神经网络模型，因为*CANN（DNN）*-4层模型没有解决模型没有解释力这个问题，而*GLM-DNN*残差修正神经网络模型拟合效果更好，模型泛化能力也相差不大，最重要的是模型具有解释力。

最后考虑*GLM-CNN*-2层的神经网络，通过表5.18发现随着周期数不断增加，模型的样本内外的损失函数都在减少，模型一直在不断优化，但是到400周期的时候，样本外损失减小的趋势变缓，表明模型快要出现过拟合的情况。在200周期的时候模型效果比*GLM-DNN*-2层的神经网络模型效果好。在表5.18中各周期内该模型都优于广义线性模型。

5.5.2 集成模型

通过表5.18发现*GLM*-改进*BP*神经网络与广义线性模型的效果相差不大，

所以在进行模型集成的时候并没有选取该模型。通过表5.18可以看出*GLM-DNN*残差修正模型在3层300周期的时候模型效果最好，2层的*GLM-CNN*在400周期的时候模型效果最好，但是与300周期模型的效果相差不大，为了集成模型结构的统一，选取周期数一样的3层*GLM-DNN*与2层*GLM-CNN*残差修正模型作为集成模型，结果输出见下表：

表 5.19 集成模型结果

集成模型	周期	样本内损失	样本外损失
残差修正模型集成	100	30.81649	30.46215
预测模型集成	100	30.78233	30.33525
残差修正模型集成	200	30.57638	30.29226
预测模型集成	200	30.42708	29.98697
残差修正模型集成	300	30.47066	30.32402
预测模型集成	300	30.19272	29.87251
残差修正模型集成	400	30.35881	30.34694
预测模型集成	400	30.23222	29.95488
广义线性模型	-	31.40828	30.91210

表5.19中周期数是选取的各残差修正模型的周期数。先考虑残差修正模型集成，通过表5.19可以看出，随着周期数不断增加样本内损失一直减小，样本外损失减小后增加，模型在超过200周期的时候出现了过拟合现象。接着考虑预测模型集成，随着周期数不断增加该模型样本内外的损失先减小在增加，在周期数为300的时候模型效果最好。对比两种集成模型可以发现，预测模型集成的方法整体结果都比残差修正模型集成方法好。

通过实证研究，本文得出集成模型比单独的模型效果好，同时也比广义线性模型效果好。最好的集成模型为预测集成模型，通过表5.19可知，该模型在100到400周期之内，无论是样本内还是样本外的损失函数结果都比广义线性模型的效果好。由此，广义线性模型能够提供各参数的模型解释，残差修正集成模型能够使其有更好的模型效果。

6 结论及展望

6.1 结论

车险模型的精确度一直都是财险保险公司重点关注的要素之一。而随着我国车险改革的不断深入，车险模型就显得更为重要了。虽然现在出现的神经网络模型能够提供更好的模型精度但是无法提供模型各变量的解释力，因为神经网络模型类似黑箱结构无法知道每个变量的具体参数值。如果在保险公司制作产品环节中起着重要作用的车险模型没有解释力，那么保险公司就不能够很好的从整体把控产品，这就会导致保险公司在市场竞争中处于不利地位。所以加强车险数据预测方法及技术的研究，提高车险模型预测的精度以及保留模型的解释力，对提高保险公司市场竞争力具有重要意义。

本文主要工作和结论如下：

（1）本文引入 *Embedding* 算法与 *0ne-hot* 算法进行对比，通过实证简要说明 *Embedding* 算法在分类变量中的优点。并基于模型变量选择问题建立了 *Lasso-GAMLSS* 模型。

（2）从实证数据来看，广义线性模型能够输出各个解释变量的参数，神经网络模型缺少解释力。通过两组不同实际数据建模对比广义线性模型与神经网络模型的结果，我们能够发现神经网络模型有比较好的模型拟合和预测精度，这验证了神经网络模型的有效性。

（3）通过数据分析发现，车险索赔次数数据符合泊松分布，于是对 BP 神经网络进行改进，将数据进行处理使其结构符合一维 *CNN* 模型，建立 *DNN*、一维 *CNN*、*LSTM* 深度学习的神经网络。通过实际数据验证发现，改进的 BP 神经网络效果比之前的要好，DNN 与一维 CNN 神经网络模型比其他网络模型效果好，*LSTM* 模型因为自身结构模型效果并不好。

（4）提出了神经网络修正广义线性残差的模型。介绍了 *CANN* 模型，然后对其进行了一定的扩展，第一个扩展模型是使用 *CNN* 神经网络替换 *CANN* 结构中 *DNN* 神经网络模型；第二个扩展模型是用广义可加模型对 *CANN* 模型中的广义线性模型进行替换。接着基于 *CANN* 的缺点提出了残差修正模型。深度神经网络修正广义线性残差模型相对于 *CANN* 模型来说，它不仅具有简单的模型结构还有不错的模型效果。同时，深度神经网络还能解决 *BP* 神经网络的一些问题。

首先利用广义线性模型对车险数据进行初步预测，计算预测值与观测值的残差，以车险损失频率影响因素为自变量以及残差为因变量，使用深度神经网络进行拟合，最后使用深度神经网络的残差预测值对广义线性回归分析的预测值进行修正就得到了最终的预测结果。通过实证研究发现新建立的残差修正模型不管是样本内还是样本外的损失函数都比广义线性模型小，也就是说修正残差模型的效果比广义线性模型，提高了车险模型精度的同时还保留了模型的解释力。

（5）将广义线性模型与神经网络构成的组合模型进行集成，对其进行理论研究，说明集成模型的优点。基于集成学习的思想将多个残差修正模型进行集成能够获得比单个残差修正模型更好的结果。因此本文选择2个残差修正模型，将其建立两种集成模型。第一种是对残差修正模型使用DNN模型进行集成。第二种是对预测模型使用线性回归模型进行集成。实证结果表明，多个残差修正模型的线性回归集成的模型效果比单个残差修正模型效果好。残差修正集成模型便于推广和实际应用，对保险公司的车险建模有一定的参考价值。

6.2 展望

怎样使用更高效、解释力更强的预测模型来对车险保费定价具有一定的重要意义。本文基于神经网络对广义线性模型进行修正的方法，虽然取得一定的成果，但是还有很多的问题可以进一步研究。

（1）考虑更多集成方式，进一步讨论集成模型

（2）广义可加模型的效果比广义线性模型好，但是本文建立的广义可加模型的*CANN*模型效果并不是很好，可以进一步研究其组合模型的结构，达到更好的模型效果。

（3）在神经网络中还有其它的网络结构，研究讨论将其加入到集成模型中。

参考文献

[1] Nelder J A, Wedderburn R W M. Generalized linear model[J]. Journal of the Royal Statistical Society, 1972, 135(3): 370-384

[2] McCullagh P, Nelder J A. Generalized Linear Models[M]. New York: Chapman and Hall/CRC, 1989

[3] Andrade E, Silva J M. An application of generalized linear models to Portuguese motor insurance[C]. Proceedings XXI ASTIN Colloquium. New York, 1989

[4] Ohlsson E, Johansson B. Non-life Insurance Pricing with Generalized Linear Models[M]. Hedelberg: Springer, 2010

[5] Garrido J,Genest C,Schulz J,Generalized Linear Models for Dependent Freq uency and Severity of Insurance Claims[J].Insurance:Mathematics and Econo mics,2016,6(6):205-215

[6] Using Kohonen's Self-Organizing Feature Map to Uncover Automobile Bodily Injury Claims Fraud. Journal of Risk and Insurance, 1998, 65(2):245-274.

[7] Leo Guelman.Gradient boosting trees for auto insurance loss cost modeling and prediction[J].Expert Systems with Applications,2012,39(3):3659-3667

[8] Liu Yue,Wang Bingjie,Lv Shaogao.Using multi -class adaboost tree for pred iction frequency of auto insurance[J] Journal of Applied Finance and Bank ing,2014,4(5) :45.

[9]Lee SCK, Sheldon L.Delta Boosting Machine with Application to General Insurance[J]. North American Actuarial Journal, 2018, 22(3):1-21.

[10] Lee Simon,Antonio Katrien.Why high dimensional modeling in actuarial sc ience? ASTIN,AFIR/ERM and IACA Colloquia,Institute of Actuaries of Au stralia,2015:1-28.

[11] Mzhavia Tornike.Vehicle insurance claim data study and forecasting model using artificial neural networks [D].Tallinn University of Techinology,2016

[12] Yufei Xia, Chuanzhe Liu, YuYing Li, et al. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. 2017, 78:225-241.

[13] Mario V. Wüthrich. Neural networks applied to chain–ladder reserving[J]. European Actuarial Journal, 2018, 8(2):407-436.

[14] Noll A , Salzmann R , Wuthrich M V . Case Study: French Motor Third-Party Liability Claims[J]. Ssrn Electronic Journal, 2018.

[15] Ferrario A , Noll A , Wuthrich M V . Insights from Inside Neural Networks[J]. Ssrn Electronic Journal, 2018.

[16] Piet de Jong, Gillian Z. Heller. Generalized Linear Models for Insurance Data[M]. Cambridge University Press, 2008.

[17] Richman Ronald,Wüthrich Mario V.. Nagging Predictors[J]. Risks,2020,8(3).

[18] Zöchbauer P.Data Science in Non -Life Pricing: Predicting Claims Frequencies using Tree -Based Models[D].ETH Zürich: Department of Mathematics,2016

[19] Andrea Gabrielli, Ronald Richman, Mario V. Wüthrich. Neural network embedding of the over-dispersed Poisson reserving model. 2020, 2020(1):1-29.

[20] Wuthrich M V , Merz M . EDITORIAL: YES, WE CANN![J]. Astin Bulletin,

2018,49(1):1-3.
[21]陈希孺.广义线性模型(一)[J].数理统计与管理,2002(05):54-61.
[22]毛泽春,刘锦萼.广义线性模型与保费点数计价系统[J].统计研究,2002(06):23-27.
[23]孟生旺.非寿险分类费率模型及其参数估计[J].数理统计与管理,2007(04):584-588.
[24]卢志义,刘乐平.广义线性模型在非寿险精算中的应用及其研究进展[J].统计与信息论坛,2007(04):26-31.
[25]钟桢,孟生旺.基于伽玛与对数正态分布假设下的广义线性模型的比较和应用[J].数理统计与管理,2010,29(03):430-436.
[26]张连增,吕定海.广义线性模型在非寿险费率分析中的应用[J].数理统计与管理,2013,32(05):903-909.
[27]孙维伟.基于Tweedie类分布的广义可加模型在车险费率厘定中的应用[J].天津商业大学学报,2014,34(01):60-67.
[28]张连增,谢厚谊.Tweedie 分布在车险费率厘定中的应用[J].保险研究,2017(01):80-90.
[29]傅鸿源,姚尧,李良.基于RBF网络的工程保险费率厘定研究[J].系统工程理论与实践,2008(07):169-172.
[30]叶明华.基于 BP 神经网络的保险欺诈识别研究——以中国机动车保险索赔为例[J].保险研究,2011(03):79-86.
[31]孟生旺.神经网络模型与车险索赔频率预测[J].统计研究,2012,29(03):22-26.
[32]孟生旺,李天博,高光远.基于机器学习算法的车险索赔概率与累积赔款预测[J].保险研究,2017(10):42-53.
[33]张连增,谢厚谊.回归树方法在车险索赔频率预测建模中的应用[J].保险研究,2018(01):101-111.
[34]张连增,王缔.保险大数据条件下车险费率厘定的研究——基于 SOM 神经网络方法的车险索赔强度建模[J].保险研究,2018(09):56-65.
[35]张连增,申晴.提升算法对传统车险索赔频率建模模型的改进——基于我国五省交强险保单数据[J].保险研究,2019(07):67-78.
[36]孟生旺,王海淘.基于机器学习算法的个体索赔准备金评估模型[J].保险研究,2019(09):88-101.
[37]张连增,申晴.泊松提升模型在中国车险索赔频率预测建模中的应用[J].统计与信息论坛,2019,34(09):27-34.
[38]张碧怡,肖宇谷,曾宇哲.车险定价中风险因子重要性测度的比较研究——基于集成学习方法和广义线性回归模型[J].保险研究,2019(10):73-83.
[39]叶明全,胡学钢.GM(1,1)残差修正的季节性神经网络预测模型及其应用[J].计算机工程与应用,2005(01):194-196.
[40]王谷,汪洋.神经网络修正灰色残差模型的交通量预测[J].交通标准化,2006(01):76-78.
[41]刘玉兵,陈亚忠,王晓东,李霞.BP 神经网络修正灰色残差组合模型方法在油液光谱分析中应用的研究[J].润滑与密封,2007(03):172-174.
[42]李艳昌,徐帅.基于神经网络修正的残差智能灰色模型在负荷预测中的应用[J].华东电力,2007(11):30-33.

[43]光辉,李国芬.基于神经网络的 GM(1,1)预测模型残差修正研究[J].城市勘测,2008(01):157-160.
[44]孙金岭,庞娟.基于残差修正的灰色神经网络在数据挖掘中的应用[J].吉林大学学报(理学版),2015,53(06):1263-1268.
[45]毕建武.基于多元回归残差 RBF 神经网络修正算法瓦斯涌出量预测研究[D].辽宁工程技术大学,2015.
[46]陈卓. 残差自修正深度学习集成神经网络在短期电力负荷预测中的应用[D].浙江大学,2018.
[47]李新琴,张鹏翔,史天运,李平.基于深度学习集成的高速铁路信号设备故障诊断方法[J].铁道学报,2020,42(12):97-105.
[48]王守志,奚歌,张福坤,刘金玉,耿振云,詹昊,张云姣.基于集成学习算法的黄河中游采砂信息提取[J].水利水电技术,2020,51(12):161-168.
[49]尹鹏博,彭成,潘伟民.基于集成学习的微博谣言早期检测[J].微电子学与计算机,2021,38(01):83-88.
[50]孟生旺.风险模型[M].清华大学出版社,2017

后记

时光匆匆而过，三年的研究生生活就将画句号，在兰州财经大学生活的这段时光虽然不长但尤其珍贵，使我从一个对精算懵懂的本科生变成了能够独立自主进行研究的硕士生。在我对精算的研究方向迷茫之际，是导师和师门师姐的帮助与指导，使我逐渐走进了保险精算领域。在研究过程中遇到了各种困难，但是导师和同学的帮助给予了向前的动力，为此向他们表示感谢。

首先，衷心感谢的人是我的导师孟生旺老师，很荣幸能成为孟老师的学生。孟老师学识渊博、思路新颖、研究范围广阔、对待科研仔细认真，脚踏实地。在我刚开始研究生的生活时，孟老师根据我个人情况指定学习计划并且每个寒暑假都不辞辛劳地给我进行辅导，讲授学科前沿知识，让我了解到保险精算的魅力。论文从最初的选题到最终定稿都是在孟老师耐心指导下完成的，您严谨治学的精神一直激励我不断前进，在将来，这对于我来说是相当可贵的。

其次，我要感谢我的第二位指导老师，孙景云老师。在学习保险精算的过程中，孙老师给我了许多学习指导和帮助，孙老师牺牲自己的休息时间不辞幸苦地辅导我们学习，针对问题进行全面细致的讲解，夏天的酷热和冬天的严寒都没能妨碍讨论班的正常进行。孙老师坚持不懈的讲授知识，使我储备了必要的研究知识。

感谢统计学院提供的研究平台和环境。感谢学院各老师精彩的授课让我深入理解并掌握了统计学的知识。感谢管青青师姐对我的关怀与鼓励，让我更好更快的适应了研究生生活。感谢同学与舍友，在学习之余带给我快乐。

还要感谢我的父母家人。从小到大，他们一直给我建立舒适的成长环境，他们对于我的鼓励和支持，让我在学习的道路上没有后顾之忧，让我有乐观向上的心态面对困难。

最后，感谢评阅老师的评语和答辩老师。