Neural networks for the joint development of individual payments and claim incurred

Łukasz Delong[1] **Mario V. Wüthrich**[2]

[1]SGH Warsaw School of Economics, Institute of Econometrics

Niepodległości 162, Warsaw 02-554, Poland

lukasz.delong@sgh.waw.pl

[2]ETH Zurich, RiskLab, Department of Mathematics

Rämistrasse 101, Zurich 8092, Switzerland

mario.wuethrich@math.ethz.ch

Abstract: The goal of this paper is to develop regression models and postulate distributions which can be used in practice to describe the joint development process of individual claim payments and claim incurred. We apply neural networks to estimate our regression models. As regressors we use the whole claim history of incremental payments and claim incurred, as well as any relevant feature information which is available to describe individual claims and their development characteristics. Our models are calibrated and tested on a real data set, and the results are benchmarked with the Chain-Ladder method. Our analysis focuses on the development of the so-called Reported But Not Settled (RBNS) claims.

Keywords: Neural networks, individual claims, Reported But Not Settled claims, claims simulations.

1 Introduction

Stochastic models for individual claims reserving have been introduced roughly 30 years ago in the work of Arjas (1989) and Norberg (1993, 1999). These papers introduce the stochastic framework of marked point processes for individual claims modeling. However, they provide little guidance on statistical aspects and on the application of these models to practical problems. Surprisingly little progress has been made in using these approaches for individual claims reserving. The proposed approaches are based on (semi-) parametric models, see Larsen (2007), Taylor et al. (2008), Zhao et al. (2009), Jessen et al. (2011), Pigeon et al. (2013) and Antonio and Plat (2014). Unfortunately, most of these proposals turn out to have too restrictive assumptions and they are not sufficiently practical in real data applications. Individual claims reserving has only started to become popular with the emergence of machine learning methods in insurance: some papers are based on the application of regression trees and gradient boosting techniques, see Wüthrich (2018), Lopez et al. (2019), De Felice and Moriconi (2019), Duval and Pigeon (2019) and Baudry and Robert (2019). Probably the most popular machine learning method, neural networks, has mainly been used on aggregate data, see Gabrielli (2020) and Kuo (2019). This seems surprising, because neural networks have gained a lot of attention in recent years due to their excellent performance on individual cases. A good introduction to neural networks and their application to insurance pricing can be found in Goodfellow et al. (2016), Ferrario et al. (2018) and Denuit et al. (2019). Neural networks have served at developing an individual claim history simulation machine, see Gabrielli and Wüthrich (2018), but these authors have not been following up their method for the purpose of claims reserving. Moreover, Gabrielli and Wüthrich (2018) only model claim payments but not claim incurred because of lack of the corresponding information. Clearly, claim incurred, together with past information about the claim development process, is important information for the prediction of future payments and should not be excluded a priori from regression models used for claim prediction.

The goal of this paper is two folds. Firstly, we would like to jointly model the development of individual claim payments and claim incurred, and secondly, we would like to explore neural networks for this task because they seem to be particularly suited for this problem. Our analysis focuses on the development of the so-called Reported But Not Settled (RBNS) claims. One can expect that the development process of individual RBNS claims could be characterized with regression models since claims reported with different features and claim histories should generate different cash flows in time and amount. Consequently, regression models for individual claims should improve reserving methods and provide more detailed information about claim developments and ultimate losses in portfolios (e.g. across segments or claim types).

In this paper we develop regression models and postulate distributions which can be used in practice to describe the joint development process of individual claim payments and claim incurred. We apply neural networks to estimate our regression models. As regressors we use the whole claim history of incremental payments and claim incurred, as well as any feature information which is available to describe individual claims and their development characteristics. Due to a large number of regressors we would like to use in individual claim prediction, neural networks seem to be the most appropriate choice for estimating the regression functions. The fit of our regression models and distributions is

tested on a real data set.

We see five contributions of this paper. The first contribution are the regression models themselves. We point out that regression models which describe the development process of claims from individual policies are fundamentally different than regression models which have been used for aggregate data. Hence, it is not possible to simply move the well-known chain-ladder type models to individual claims reserving. The second contribution is that we allow for non-Markovian dynamics of the claim development process and we test the Markovian assumption on a real data set. The standard approach in claims reserving is to only use the most recent information about the claim developments, hence, to assume a Markovian structure of the claim development process. It turns out that, in our practical example, the Markovian assumption is too strong and it is beneficial, in terms of prediction accuracy, to use the whole claim history in our regression models. Thanks to the application of neural networks, the claim history can be efficiently used for predictions of future claims. The third contribution is to jointly model incremental payments and claim incurred. From the paper by Quarg and Mack (2004), it is suggested that both payments and claim incurred should be used for estimating the reserve for the outstanding claim liabilities. Intuitively, the linear regression structures proposed by Quarg and Mack (2004) in their Munich chain-ladder model is not fully appropriate. Using the tools from neural networks, we can fit a better relation between future payments and claim incurred and past payments and claim incurred. To the best of our knowledge, there is only one paper by Kuo (2019) which uses neural networks with simultaneous consideration of payments and claim incurred. However, Kuo (2019) applies neural networks to aggregated payments and case reserves observed across all accident years and development years and multiple insurance companies. Consequently, claims from individual policies and models suited for individual claims reserving cannot be used. In Kuo (2019), recurrent neural networks with a mean-squared error loss function are calibrated to the whole available history of aggregated payments and case reserves. However, the non-Markovian assumption of the claim development process is not discussed in any aspect and is not validated in the paper. Moreover, there is no clear distinction in Kuo (2019) between random variables and their predictions, the latter typically being estimated expected values. This distinction is important, if these random variables are used as regressors in later periods. The fourth contribution of this paper is that we present a self-contained estimation procedure for our regression models and neural networks. We recommend to use the CANN approach of Schelldorfer and Wüthrich (2019) - we suggest to use generalized linear models (GLMs), generalized additive models (GAMs) or regression trees as initial predictions, or as the initial models, from which we start training the neural networks. We use multinomial/binomial cross-entropy and Gamma loss functions for calibrations of our neural networks. These loss functions are also related to the distributions of the variables which we would like to use for predictions in the claim development process. The goal in the simulation of outstanding claims is to model not only the mean response but also the distribution of the response. This might be a real challenge in practice since simple distributions, which are common in regression problems and focus on the mean response, may not fit real data sufficiently well. We aim at correctly modeling at least the first two moments of the distribution and we focus on modeling the mean as well as the dispersion of a response with a continuous distribution. In order to model extreme events, we separate large claims from attritional claims and use Pareto distributions for

claims above a high threshold. Finally, to the best of our knowledge this is the first paper in insurance data science where outstanding liabilities are predicted with neural networks and compared with classical chain-ladder estimates. This comparison on a real data set is the fifth contribution of this paper.

The remainder of this paper is organized as follows. In Section 2 we introduce all claim development models. In Section 3 we discuss the estimation procedure. The numerical example on real data set is investigated in Section 4, and in Section 5 we benchmark our results with the chain-ladder predictions. Finally, in Section 6 we conclude. All calculations were done in Keras, which is an open-source API to TensorFlow.

2 Models for individual claims development

Let $i \in \{1, 2, \ldots\}$ denote the accident period of the occurrence date of an insurance claim. The accident period can be measured in days, weeks, months, quarters or years. Let $j \in \{0, 1, 2, \ldots\}$ denote the reporting delay after the claim occurrence date, measured in the same time units as the accident period i. Consequently, $i + j$ denotes the reporting period (in days, weeks, months, quarters or years) of a particular claim. After a claim has been reported, it develops over its settlement time. Let $k \in \{0, 1, 2, \ldots\}$ measure the *development periods* of a reported claim, initialized to the respective reporting date $i + j$.

We investigate individual claims from individual insurance policies. Let $P_k^{i,j}$ denote the *incremental payment* in development period k for a claim of accident period i reported with delay j. The payment $P_k^{i,j}$ is made in calendar period $i + j + k$. Let $I_k^{i,j}$ denote the *claim incurred* at the end of development period k for a claim from accident period i reported with delay j. The claim incurred $I_k^{i,j}$ is observed at the end of calendar period $i + j + k$. We introduce the case reserve for an individual claim, which is a part of the claim incurred and corresponds to an individual claim amount estimate made by a claims adjuster. The *case reserve* at the end of development period k for a claim from accident period i reported with delay j is denoted by $R_k^{i,j}$, and it is given by

$$R_k^{i,j} = I_k^{i,j} - \sum_{l=0}^{k} P_l^{i,j}, \quad k = 0, 1, 2, \ldots.$$

The variables $P_k^{i,j}, I_k^{i,j}, R_k^{i,j}$ should also be indexed with unique individual claim identifier (claim number), for notational convenience this index is omitted for the moment.

At the reporting date of a claim we observe the first payment and the first evaluation of the total claim incurred, i.e. we have information $(P_0^{i,j}, I_0^{i,j})$. Note that $P_0^{i,j}$ can also be zero if there has not been made any payment in the initial development period $k = 0$. Having observed the values $(P_0^{i,j}, I_0^{i,j})$ means that the time point $i + j$ of the claim reporting has been fully observed and we aim at modeling the development $(P_k^{i,j}, I_k^{i,j})$ for all later time points $k = 1, 2, \ldots$. Our goal is to study the development process of RBNS claims, i.e. claims for which we have observed the initial state $(P_0^{i,j}, I_0^{i,j})$. In this paper we aim at modeling the two-dimensional process $(P_k^{i,j}, I_k^{i,j})_{k=1,2,\ldots}$, conditionally given the initial value $(P_0^{i,j}, I_0^{i,j})$. Since we are interested in individual claims reserving, the two-dimensional process $(P_k^{i,j}, I_k^{i,j})_{k=1,2,\ldots}$ is modeled for each single reported claim.

Let us define a filtration $(\mathcal{C}_k)_{k=0,1,\dots}$ which describes the history of payments and claim incurred on an individual claim. The individual claim filtration is defined by

$$\mathcal{C}_k^{i,j} = \sigma\left\{P_s^{i,j}, I_s^{i,j};\ 0 \le s \le k\right\}, \quad k = 0, 1, 2, \dots,$$

and it describes the history of payments and claim incurred for a claim from accident year i and reported with delay j. Moreover, to each individual claim we associate a vector of (static or dynamic) features, which we denote by $\boldsymbol{z}_k^{i,j}$. E.g. the vector $\boldsymbol{z}_k^{i,j}$ may include any feature of the insurance policy such as the line of business involved, the age of the injured person, the claim type, the accident period, the reporting delay etc. The information included in the filtration $\mathcal{C}_{k-1}^{i,j}$ and the features $\boldsymbol{z}_{k-1}^{i,j}$ are used as regressors (explanatory variables) in our regression problems to predict $(P_k^{i,j}, I_k^{i,j})$ in the next development period k.

2.1 Model 1: Occurrence of payments and changes in claim incurred

In the first step we model the indicators which indicate whether the policy generates a new payment and/or the value of the claim incurred is changed in the next development period. The severities of the incremental payments and the change in claim incurred, if needed, are modeled in the next subsections.

We define the indicator process $(\mathcal{I}_k^{i,j}, \mathcal{P}_k^{i,j})_{k=1,2,\dots}$ as follows:

$$\mathcal{I}_k^{i,j} = \mathbb{1}_{\{I_k^{i,j} - I_{k-1}^{i,j} \neq 0\}} \quad \text{and} \quad \mathcal{P}_k^{i,j} = \mathbb{1}_{\{P_k^{i,j} \neq 0\}}, \tag{2.1}$$

and we introduce a stochastic process $(Y_k^{i,j})_{k=1,2,\dots}$, which takes values in the set $\{0, 1, 2, 3\}$, by setting:

$$Y_k^{i,j} = 2\mathcal{I}_k^{i,j} + \mathcal{P}_k^{i,j} = \begin{cases} 0 & \text{if} \quad \mathcal{P}_k^{i,j} = 0 \text{ and } \mathcal{I}_k^{i,j} = 0, \\ 1 & \text{if} \quad \mathcal{P}_k^{i,j} = 1 \text{ and } \mathcal{I}_k^{i,j} = 0, \\ 2 & \text{if} \quad \mathcal{P}_k^{i,j} = 0 \text{ and } \mathcal{I}_k^{i,j} = 1, \\ 3 & \text{if} \quad \mathcal{P}_k^{i,j} = 1 \text{ and } \mathcal{I}_k^{i,j} = 1. \end{cases} \tag{2.2}$$

In order to model occurrences of payments and changes in claim incurred, we have to model the *conditional* probabilities for the sequence of random variables $(Y_k^{i,j})_{k=1,2,\dots}$. We should model the *conditional* probabilities since we expect the categorical distribution of $Y_k^{i,j}$ in development period k to depend on the individual claim history $\mathcal{C}_{k-1}^{i,j}$ and the individual claim feature $\boldsymbol{z}_{k-1}^{i,j}$.

We use a multinomial logistic regression to model these categorical conditional probabilities as follows:

$$\log\left(\frac{\mathbb{P}\left(Y_k^{i,j} = y \middle| \mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j}\right)}{\mathbb{P}\left(Y_k^{i,j} = 0 \middle| \mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j}\right)}\right) = f^y\big(\mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j}\big), \quad y = 1, 2, 3, \quad k \ge 1, \tag{2.3}$$

where we use three appropriate regression functions $f = (f^1, f^2, f^3)$. The regression model (2.3) is called Model 1.

A nowadays well-understood approach is to use a generalized linear model (GLM) with a linear additive function f_y, or a generalized additive model (GAM) with a non-linear additive function f_y, and selected predictor variables from the set $(\mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j})$. Alternatively, one can use regression trees where the best predictors from the set $(\mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j})$ are chosen as a part of the calibration and f_y is estimated as a piecewise constant function on subspaces defined by the predictors; in the context of individual claims reserving this has been considered in Wüthrich (2018), Lopez et al. (2019), De Felice and Moriconi (2019) and Duval and Pigeon (2019). A more advanced approach, gaining popularity among actuaries, is to use a neural network with a non-linear non-additive function f_y and including all predictor variables from the set $(\mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j})$, this has been considered in Kuo (2019) and Gabrielli (2020), however still on aggregated claims. We aim at fitting neural networks to individual claims.

Let us recall the categorical cross-entropy loss function for a sample of size n of independent random variables $\big(v_{\ell,1}, v_{\ell,2}, v_{\ell,3}, v_{\ell,4}\big)_{\ell=1,\ldots,n}$ with categorical distribution (4 categories) and estimated probabilities $\big(\hat{p}_{\ell,1}, \hat{p}_{\ell,2}, \hat{p}_{\ell,3}, \hat{p}_{\ell,4}\big)_{\ell=1,\ldots,n}$, it is given by

$$D_{\text{cat}} = -\frac{1}{n}\sum_{\ell=1}^{n}\sum_{k=1}^{4} v_{\ell,k}\log(\hat{p}_{\ell,k}). \tag{2.4}$$

The categorical cross-entropy loss function is closely related to the deviance loss function of the multinomial model. The deviance is equal to $2\sum_{\ell=1}^{n}\sum_{k=1}^{4} v_{\ell,k}\log(v_{\ell,k}/\hat{p}_{\ell,k})$. We remark that minimizing the cross-entropy is equivalent to minimizing the deviance, which is performed when we fit a categorical GLM/GAM or a categorical regression tree model.

If $Y_k^{i,j} = 0$, then we immediately know the values of the process $(P_k^{i,j}, I_k^{i,j})$ in the next development period k, because there is not any change in claim incurred and payments are equal to zero. Therefore, we only need to further model the cases $Y_k^{i,j} \in \{1,2,3\}$.

2.2 Models 2 and 3: Claim severities of incremental payments

If $Y_k^{i,j} = 1$ or $Y_k^{i,j} = 3$, we have to model a non-zero payment in development period k for the claim under consideration. In practice, we observe both positive and negative incremental payments and we can expect that the behavior of positive and negative payments differs. By negative payments, we mean salvages and subrogations. Consequently, we can use a spliced distribution to model non-zero payments. We introduce the following sequences of random variables:

$$\begin{aligned} P_k^{i,j,(+)} &= P_k^{i,j} \big| \, Y_k^{i,j} \in \{1,3\}, P_k^{i,j} > 0, \quad k = 1,2,\ldots, \\ P_k^{i,j,(-)} &= -P_k^{i,j} \big| \, Y_k^{i,j} \in \{1,3\}, P_k^{i,j} < 0, \quad k = 1,2,\ldots, \end{aligned} \tag{2.5}$$

where the first sequence models positive incremental payments, given their occurrence, and the second sequence models negative recovery payments, again given their occurrence. We have to model the *conditional* probabilities of a positive and a negative payment together with the *conditional* distributions of the payments $(P_k^{i,j,(+)})_{k=1,2,\ldots}$ and $(P_k^{i,j,(-)})_{k=1,2,\ldots}$, respectively. As discussed in the previous subsection, the probabilities and the distributions of $P_k^{i,j,(+)}$ and $P_k^{i,j,(-)}$, respectively, in development period k should depend on the available information included in $\big(\mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j}\big)$. The probabilities and the distributions may

also depend on the change in claim incurred indicator $\mathcal{I}_k^{i,j}$, since the events $\{Y_k^{i,j}=1\}$ or $\{Y_k^{i,j}=3\}$ include two different scenarios for the change in claim incurred.

The conditional probabilities of a positive or a negative payment in development period k, respectively, given the occurrence of a non-zero payment, can be modeled with a binomial distribution. We use a binomial logistic regression model, and we set

$$\log\left(\frac{\mathbb{P}\left(P_k^{i,j}>0\middle|\, Y_k^{i,j}\in\{1,3\},\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right)}{\mathbb{P}\left(P_k^{i,j}<0\middle|\, Y_k^{i,j}\in\{1,3\},\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right)}\right)=f\big(\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\big),\quad k\geq 1. \tag{2.6}$$

The regression model (2.6) is called Model 2. The cross-entropy loss function for a sample of random variables with binomial distribution is calculated analogously to (2.4).

As far as distributions of $P_k^{i,j,(+)}, P_k^{i,j,(-)}$ are concerned, we can expect that the payments should have a skewed distribution. It is common in actuarial practice to model claim sizes with Gamma distributions and fit Gamma regression models to claims severities. For this reason, we start with assuming that $P_k^{i,j,(+)}$ and $P_k^{i,j,(-)}$ have Gamma distributions where we postulate the relationships:

$$\log\left(\mathbb{E}\left[P_k^{i,j,(+)}\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)=f\big(\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\big),\quad k\geq 1, \tag{2.7}$$

for an appropriate regression function f, and

$$\mathrm{Var}\left[P_k^{i,j,(+)}\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]=\psi\cdot\left(\mathbb{E}\left[P_k^{i,j,(+)}\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)^2,\quad k\geq 1, \tag{2.8}$$

where $\psi>0$ is called a dispersion coefficient. It has been observed in many data sets, including insurance data sets, that the dispersion coefficient may not be constant. In such cases, it should be modeled with a separate regression function. A classical approach to model dispersion coefficients in GLMs/GAMs is to use a second Gamma regression model with a response directly modeling the dispersion coefficient. Consequently, we assume that $P_k^{i,j,(+)}$ and $P_k^{i,j,(-)}$ have Gamma distributions where we postulate the relationships:

$$\begin{aligned}\log\left(\mathbb{E}\left[P_k^{i,j,(+)}\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right) &= f\big(\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\big),\quad k\geq 1, &(2.9)\\ \mathrm{Var}\left[P_k^{i,j,(+)}\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right] &= e^{\phi\left(\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right)}\\ &\quad\cdot\left(\mathbb{E}\left[P_k^{i,j,(+)}\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)^2,\quad k\geq 1, &(2.10)\end{aligned}$$

for another regression function ϕ. Let $Res_k^{i,j}$ denote the unscaled Pearson residual from the regression model (2.7)-(2.8). We assume that the squared residual $\left|Res_k^{i,j}\right|^2$ follows a Gamma regression with moments:

$$\log\left(\mathbb{E}\left[\left|Res_k^{i,j}\right|^2\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)=\phi\big(\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\big),\quad k\geq 1, \tag{2.11}$$

$$\mathrm{Var}\left[\left|Res_k^{i,j}\right|^2\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]=\vartheta\cdot\left(\mathbb{E}\left[\left|Res_k^{i,j}\right|^2\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)^2,\quad k\geq 1, \tag{2.12}$$

where $\vartheta>0$ denotes a dispersion coefficient for the Pearson residuals. Similar assumptions are made for negative payments. The regression model (2.9)-(2.12) is called Model 3_positive, or Model 3_negative if negative payments are modeled.

We recall that the unscaled deviance loss function for a sample of n independent random variables $(v_\ell)_{\ell=1,\ldots,n}$ with Gamma distribution with moments (2.9)-(2.10) and estimated expected values $(\hat{\mu}_\ell)_{\ell=1,\ldots,n}$ is given by

$$D_{\text{Gamma}}^{unscaled} = -\frac{1}{n}\sum_{\ell=1}^{n}\left(\log\left(\frac{v_\ell}{\hat{\mu}_\ell}\right) - \frac{v_\ell - \hat{\mu}_\ell}{\hat{\mu}_\ell}\right). \tag{2.13}$$

We omit the constant 2 and use the average deviance loss function compared to the traditional definition of the Gamma deviance in GLMs/GAMs/trees. The scaled deviance loss function, which takes into account the dispersion coefficient, is given by

$$D_{\text{Gamma}}^{scaled} = -\frac{1}{n}\sum_{\ell=1}^{n}\left(\frac{\log\left(\frac{v_\ell}{\hat{\mu}_\ell}\right) - \frac{v_\ell - \hat{\mu}_\ell}{\hat{\mu}_\ell}}{e^{\hat{\phi}_\ell}}\right), \tag{2.14}$$

where $\hat{\phi}_\ell$ denotes the prediction from (2.11)-(2.12) for observation $\ell = 1, \ldots, n$.

If we combine the binomial distribution for a positive or a negative payment with the Gamma distributions for the severities of positive or negative payments, we receive the distribution of the incremental payment $P_k^{i,j}|Y_k^{i,j} \in \{1,3\}$ in development period k.

We remark that if $Y_k^{i,j} = 1$, our modeling process is complete for period k and we can derive the values of the process $(P_{k+1}^{i,j}, I_{k+1}^{i,j})$ in the next development period $k+1$. If $Y_k^{i,j} = 3$, we have to model the change in claim incurred at the end of the development period k which is obviously related to the payment made in development period k. If $Y_k^{i,j} = 2$, the payment $P_k^{i,j} = 0$ is zero but we need to consider a change in claim incurred. The next two subsections deal with changes in claim incurred.

2.3 Model 4: Closing times

We consider the two remaining events $\{Y_k^{i,j} = 2\}$ and $\{Y_k^{i,j} = 3\}$ related to changes in claim incurred. We should differentiate between two cases: a) the value of the claim incurred changes and the case reserve is different from zero; b) the value of the claim incurred changes and the resulting case reserve is equal to zero. The latter is interpreted that the claim is closed at the end of period k, and we do not expect any further claim developments on that claim, unless such a claim is re-opened. Re-openings are assumed to only happen with a small probability (which still needs to be modeled). For claims that have zero case reserves at the end of period k we exactly know the change in claim incurred (i.e. in this case the severity of change in claim incurred does not need to be modeled). Therefore, it is reasonable to include an indicator for zero case reserves directly into all regression functions discussed in this section. Note that this indicator is already included indirectly since the regression functions are assumed to depend on the whole history of paid and incurred claims from which the value of the case reserve can be derived. However, a direct inclusion will allow the neural network to learn the relevant structure more quickly.

We introduce the following sequences of random variables:

$$\mathcal{R}_k^{i,j} = R_k^{i,j}\big|\, Y_k^{i,j} \in \{2,3\}, \quad k = 1, 2, \ldots. \tag{2.15}$$

The event $\{\mathcal{R}_k^{i,j}=0\}$ is interpreted as claim closing in development period k, given there is a change in claim incurred in the development period k. Of course, a claim can be re-opened in the future and this is modeled through probabilities (2.3). The *conditional* probability for the event $\{\mathcal{R}_k^{i,j}=0\}$ should depend on $(\mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j})$, as well as on $\mathcal{P}_k^{i,j}$ since the events $\{Y_k^{i,j}=2\}$ and $\{Y_k^{i,j}=3\}$ include two different scenarios for payment occurrences. Finally, if a payment is made, then the value of the payment $P_k^{i,j}$ could also have an impact on the probability of the event $\{\mathcal{R}_k^{i,j}=0\}$.

As Model 4, we use the binomial logistic regression model described by

$$\log\left(\frac{\mathbb{P}\left(\mathcal{R}_k^{i,j}=0\middle|\,\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right)}{\mathbb{P}\left(\mathcal{R}_k^{i,j}\neq 0\middle|\,\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right)}\right)=f\big(\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\big),\quad k\geq 1. \tag{2.16}$$

2.4 Model 5: Severities for claim incurred for open claims

We are left with changes in claim incurred in the non trivial scenario a) where the case reserve at the end of the development period is not equal to zero.

We introduce the sequence of random variables:

$$I_k^{i,j,(open)}=I_k^{i,j}|Y_k^{i,j}\in\{2,3\},R_k^{i,j}\neq 0,\quad k=1,2,\ldots. \tag{2.17}$$

As for payments, we assume that $I_k^{i,j,(open)}$ has a Gamma distribution with moments:

$$\log\left(\mathbb{E}\left[I_k^{i,j,(open)}\middle|\,\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)=f\big(\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\big),\quad k\geq 1, \tag{2.18}$$

for a suitable regression function f, and choosing another suitable regression function ϕ

$$\begin{aligned}\mathrm{Var}\left[I_k^{i,j,(open)}\middle|\,\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]&=e^{\phi\left(\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right)}\\&\cdot\left(\mathbb{E}\left[I_k^{i,j,(open)}\middle|\,\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)^2,\quad k\geq 1.\end{aligned} \tag{2.19}$$

Let $Res_k^{i,j}$ denote the unscaled Pearson residual from regression model (2.18)-(2.19). We assume that the squared residual $\left|Res_k^{i,j}\right|^2$ follows a Gamma regression with moments:

$$\log\left(\mathbb{E}\left[\left|Res_k^{i,j}\right|^2\middle|\,\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)=\phi\big(\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\big),\quad k\geq 1, \tag{2.20}$$

$$\mathrm{Var}\left[\left|Res_k^{i,j}\right|^2\middle|\,\mathcal{I}_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]=\vartheta\cdot\left(\mathbb{E}\left[\left|Res_k^{i,j}\right|^2\middle|\,\mathcal{P}_k^{i,j},P_k^{i,j},\mathcal{C}_{k-1}^{i,j},\boldsymbol{z}_{k-1}^{i,j}\right]\right)^2. \tag{2.21}$$

The set of predictors in (2.18)-(2.21) can be deduced in a similar way as in the previous sections. The regression model (2.18)-(2.21) is called Model 5. Negative claim incurred are assigned to data errors, henceforth, they are not modeled.

Combining the results from Sections 2.2-2.4, we can model claim incurred $I_k^{i,j}|Y_k^{i,j}\in\{2,3\}$ in development period k and we can derive the value of the process $(P_k^{i,j},I_k^{i,j})$ in the next development period k. The modeling process for the next development period is complete for all scenarios of potential claims developments.

2.5 Large incremental payments and claim incurred in Models 3 and 5

Typically, general insurance claims are skewed and heavy tailed. A traditional approach in actuarial pricing and reserving is to separate large claims from attritional claims. Large claims are usually identified by using techniques from Extreme Value Theory (EVT), and claims above a high threshold are modeled with Pareto distributions. In our case, we should model *large* incremental payments and *large* changes in claim incurred in each development period $k = 1, 2, \ldots$. By a large claim we understand a large incremental payment in Models 3 or a large change in claim incurred in Model 5.

Let d_k denote a threshold and $P_k^{i,j,(+)}$ denote a positive incremental payment in development period k. We postulate a Pareto tail

$$\begin{aligned}
&\mathbb{P}\Big(P_k^{i,j,(+)} - d_k > x \Big| P_k^{i,j,(+)} > d_k, \mathcal{I}_k^{i,j}, \mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j}\Big) \\
&= \left(\frac{\lambda\big(\mathcal{I}_k^{i,j}, \mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j}\big)}{\lambda\big(\mathcal{I}_k^{i,j}, \mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j}\big) + x}\right)^{\gamma\left(\mathcal{I}_k^{i,j}, \mathcal{C}_{k-1}^{i,j}, \boldsymbol{z}_{k-1}^{i,j}\right)}, \quad x > 0, \quad k \geq 1, \qquad (2.22)
\end{aligned}$$

where γ is called tail index. A similar model can be used for large negative incremental payments and large claim incurred. In the most general claim development model, regression models should be built for the probability of a large claim and the severity of the large claim (for λ and γ) since claims with particular features (e.g. body claims) are likely to have higher propensity to generate large claims in consecutive development periods. We refrain from doing this and choose a simpler approach. In each development period k, we set a fixed probability for the occurrence of a large claim, from which we deduce the threshold d_k, and estimate constant parameters (λ_k, γ_k) of the large claim distribution (2.22) using EVT. Next, based on descriptive statistics and knowledge about business, we determine key features in $\boldsymbol{z}_{k-1}^{i,j}$ which have high propensity to generate large claims and allocate large claims in simulations to claims with these features in the first place. Details are presented in Section 4.2, below.

Consequently, we use a spliced distribution for the response in Models 3 and 5. We use a binomial distribution to decide whether the claim is attritional or large. If the claim is attritional then we use the double Gamma regression model and the Gamma distribution to predict the response, otherwise we use the Pareto distribution for the response with additional information about the features which have higher propensity to generate a large claim.

3 Estimation approach

In this section, we explain the choices of the regression functions in our regression models (2.3), (2.6), (2.9), (2.11), (2.16), (2.18) and (2.20) for the claim developments, and how they can be estimated with neural networks. A remark is also given on the estimation of large claims.

As commented in Section 2, the vector $\boldsymbol{z}_{k-1}^{i,j}$ should include, among other features, the development period k under consideration. Since we model development of claims over consecutive development periods $k = 1, 2, \ldots$, it is natural to estimate a separate

regression function for each development period k. Hence, the regression functions f in our regression models are index with $k = 1, 2, \ldots$, and the goal is to estimate the regression function f_k for each $k = 1, 2, \ldots$.

Alternatively, we could use development period k as an explanatory variable in the regression functions, i.e. keep the variable k in $\boldsymbol{z}_{k-1}^{i,j}$. In our numerical example we are going to use quarterly data and because developments in different quarters may have a rather different structural form, we prefer the approach of separate regression functions for each development period, because this should have less difficulties to cope with structural differences from one to the next period. However, if we want to model claim developments over a large number of development periods (if the business is long-tailed), then this approach will be computationally intensive and slow. Moreover, we have less and less observations available for fitting separate neural networks in each later development period. Henceforth, for all later development periods, $k = K + 1, K + 2, \ldots$, we fit one neural network, denoted by f_{K+1}, where the development period k is included as a predictor in $\boldsymbol{z}_{k-1}^{i,j}$.

3.1 Initial regression models

Our goal is to train neural networks in the five regression models which describe the claim development process. We follow the CANN approach. We start with a simple regression model M_0. The predictions from the simple regression model M_0 are next used as a regressor in the neural network. As initial model we choose a simple one with few features and without interactions. The adequacy of this initial model M_0 is then validated during the training process of the neural network and other important features and interactions will be added. Usually, it should be beneficial to start training a neural network even from a simple model. Simple models can give us information about the process and reduce the loss function at the initial training step of the neural network compared to the loss function for a model generated with completely random weights.

As initial models we can use GLMs, GAMs or regression trees. GLMs are fast in estimation and providing predictions, but we have to spend more time on pre-processing the regressors as linear link functions are unlikely to hold over the whole range of the regressors. GAMs can efficiently handle non-linear relations between the response and the regressors. Consequently, the pre-processing of the regressors is less important for GAMs. However, the estimation and prediction is typically slow for GAMs. If we fit regression trees, then continuous regressors are optimally discretized and piecewise constant relations between the response and the regressors are estimated. Estimation and prediction are also fast for trees. It is hard to recommend a particular type of model as an initial model in the CANN approach. One should investigate the costs of estimation and prediction for the initial model M_0 versus the benefits from the reduction in the loss function at the initial step of training the neural network and, consequently, a lower number of epochs used in training the neural network. It may happen that we should resign from fitting an initial model and just initiate the neural network with the constant equal to the sample mean of the response (this corresponds to the initial model equal to a homogeneous model). Let us remark that in order to model the claim development process for RBNS claims, we have to fit regression models and do predictions for many development periods. Hence, the costs and benefits of different estimation strategies should be balanced.

Since the idea is to start with simple models, we can just use two predictor variables which we include in our regression functions f_k, for $k = 1, 2, \ldots, K+1$. From the individual claim history $\mathcal{C}_{k-1}^{i,j}$ we only use cumulative payments defined by $CP_{k-1}^{i,j} = \sum_{l=0}^{k-1} P_l^{i,j}$ and claim incurred $I_{k-1}^{i,j}$ as predictor variables. By this choice, we assume that there is a Markovian structure in the claims development process and only the most recent information about the claim is relevant for the next step predictions, this is similar to Quarg and Mack (2004).

3.2 Feed-forward neural networks

For each regression model considered and development period $k = 1, 2, \ldots, K+1$ investigated, we fit two neural networks - the so-called 0^{th} neural network (denoted by NN_0) and the so-called main neural network (denoted by NN_1). For a general introduction to neural networks we refer to Goodfellow et al. (2016).

In 0^{th} neural networks NN_0:

- We use all predictor variables which we choose for M_0, i.e. from $\mathcal{C}_{k-1}^{i,j}$ we use cumulative payments $CP_{k-1}^{i,j}$ and claim incurred $I_{k-1}^{i,j}$.
- As discussed in the previous section, we include the indicators $\mathcal{I}_k^{i,j}, \mathcal{P}_k^{i,j}$ and the incremental payment $P_k^{i,j}$ as regressors in the regression functions f_k, for $k = 1, 2, \ldots, K+1$.
- We also include the indicator $\mathbb{1}_{\{R_k^{i,j}=0\}}$. As discussed above, the status of the claim is an important predictor variable for the claim development process and we want to use it directly as a regressor in the regression functions, i.e. we guide the network more directly to the study of this variable.
- As far as the vector of additional features $\boldsymbol{z}_{k-1}^{i,j}$ is concerned, we include all available claims features such as accident quarter, reporting delay, claim segment, claim type and claim origin. These variables are also natural regressors whose predictive power in the claim development process we would like to test. We do not include accident year. Since claims in latter development periods come only from earlier accident years, the predictions for latter accident years in latter development periods would be based on extrapolation of the calibrated regression functions beyond the training set which we want to avoid. Moreover, we want to make the predictions based on the observed individual claim history. Yet, we include accident quarters to directly model seasonality effects. When we fit the regression function f_{K+1} to all development periods latter than K, then $\boldsymbol{z}_{k-1}^{i,j}$ also includes the development period k as a regressor.
- Finally, we use the prediction from the initial model M_0 as a regressor in the neural network.

When fitting neural network NN_0, we still assume, as in M_0, that there is a Markovian structure in the claim development process and only the most recent information about the claim is relevant for the next step prediction. However, we extend the set of predictors

by using all available features of the claim, model non-linear relationships and interactions between the features. Neural network NN_0 should improve the initial, simple regression model and we now fine tune the parameters of the neural network so that we result in a good neural network regression model. Comparing deviance loss functions on the same validation sets, we can test the predictive power of a simple model, where only two, potentially most important, regressors are used, versus the neural network, where more regressors are used including custom-made regressors such as the status of the claim and the indicator of a positive payment. Moreover, applying neural network NN_0, we can efficiently handle many regressors, non-linear relationships and interactions, and we should improve the predictive power of the regression model.

In main neural networks NN_1:

- We use all predictor variables which we choose for NN_0,
- We add all variables included in the individual claim history $\mathcal{C}_{k-1}^{i,j}$ as predictors in our regression models for the development period k. Hence, we relax the Markovian assumption postulated in neural network NN_0. We now assume that the whole claim development history is relevant for the prediction of the claim development process. We remark that we keep the cumulative payments $CP_{k-1}^{i,j}$ in the regression functions even though the whole history $(P_l^{i,j})_{l=0,\ldots,k-1}$ is included in the regression functions. This approach is possible since neural networks do not (directly) suffer from possible collinearity effects. The reason why we keep the cumulative payments in the regression functions is that we want to extend the set of predictors compared to M_0 and NN_0, i.e. we nest the models w.r.t. the predictor variables involved. Another reason is that the calibration of a neural network may be faster if we provide feature information in the right structure.

Comparing the deviance loss functions for the 0^{th} neural network and the main neural network on a validation set, we can verify whether the inclusion of the whole history of the claim development process improves the prediction of the payments and the claim incurred in the next development period. In other words, we can test the Markovian assumption for the development process of individual payments and claim incurred. Finally, the best neural network among NN_0 and NN_1 can be chosen.

3.3 Estimation of the neural networks

Let $\boldsymbol{x}_\ell \in \mathbb{R}^{q_0}$ denote a vector of predictors which characterizes an individual observation ℓ, excluding the prediction from the initial model M_0. In the categorical case, we model the probability $p_{\ell,a}$ that observation ℓ is in class a, for $a \in \mathcal{A}$; in the continuous case, we model the expected value μ_ℓ of the response of observation ℓ. Let $\left(\hat{p}_{\ell,a}^{\text{init}}\right)_{a\in\mathcal{A}}$, respectively $\hat{\mu}_\ell^{\text{init}}$, denote the corresponding estimations from the initial model M_0.

We train a neural network with M hidden layers and $q_m \in \mathbb{N}$ hidden neurons in layers $m = 1, \ldots, M$. We use standard notation for neural networks. We define the network layers:

$$\begin{aligned}
\boldsymbol{b} \in \mathbb{R}^{q_{m-1}} \;\mapsto\; \theta^m(\boldsymbol{b}) &= \left(\theta_1^m(\boldsymbol{b}), \ldots, \theta_{q_m}^m(\boldsymbol{b})\right)' \in \mathbb{R}^{q_m}, \quad m = 1, \ldots, M, \\
\boldsymbol{b} \in \mathbb{R}^{q_{m-1}} \;\mapsto\; \theta_r^m(\boldsymbol{b}) &= \varphi\big(c_r^m + \langle \boldsymbol{w}_r^m, \boldsymbol{b}\rangle\big), \quad r = 1, \ldots, q_m,
\end{aligned}$$

where φ denotes the (non-linear) activation function, c_r^m denotes the bias term (the constant), $\boldsymbol{w}_r$ denotes the network weights and $\boldsymbol{b}$ denotes a vector of predictors. For layers $m = 1, \ldots, M$, we use the hyperbolic tangent activation function for φ. Finally, the mapping

$$\boldsymbol{b} \in \mathbb{R}^{q_0} \;\mapsto\; c^{M+1} + \Big\langle \boldsymbol{w}^{M+1}, \left(\theta^M \circ \cdots \circ \theta^1\right)(\boldsymbol{b}) \Big\rangle,$$

gives us the prediction in the output layer $M+1$ with linear activation function and the output of dimension 1.

For the Gamma regressions, we use the exponential activation function with an output of dimension 1 in layer $M+1$. We model the expected values of an individual case ℓ by

$$\begin{aligned} \mu_\ell &= e^{f(\boldsymbol{x}_\ell)} \\ f(\boldsymbol{x}_\ell) &= c^{M+1} + \alpha \log\left(\hat{\mu}_\ell^{\text{init}}\right) + \beta \Big\langle \boldsymbol{w}^{M+1}, \left(\theta^M \circ \cdots \circ \theta^1\right)(\boldsymbol{x}_\ell) \Big\rangle, \end{aligned} \tag{3.1}$$

and we shall use the log-link for the initial model (GLM or GAM) to be consistent with the exponential output activation. If the initial model is the best choice for our data set, then the fitting algorithm should choose $\alpha = 1$ and $c^{M+1} = \beta = 0$. These values are provided as initial weights for (c^{M+1}, α, β) to the fitting algorithm. The weights (c^{M+1}, α, β) are kept trainable so that the neural network can diminish the effect of the initial prediction from M_0 on the response. The structure of the predictor in (3.1) can be achieved by using concatenation of two layers in Keras: the first layer contains the 1-dimensional prediction from the initial model, the second layer gives the 1-dimensional output in layer $M+1$ from M hidden layers and linear activation function in layer $M+1$.

For the categorical regressions with $A = dim(\mathcal{A})$ classes, we use the softmax activation function with output of dimension A in layer $M+1$. We model the (softmax) probabilities for a single case ℓ as follows

$$\begin{aligned} p_{\ell,a} &= \frac{e^{f_a(\boldsymbol{x}_\ell)}}{\sum_{u \in \mathcal{A}} e^{f_u(\boldsymbol{x}_\ell)}}, \quad a \in \mathcal{A}, \\ f_a(\boldsymbol{x}_\ell) &= c_a^{M+1} + \sum_{u \in \mathcal{A}} \alpha_u \log\left(\hat{p}_{\ell,u}^{\text{init}} / \hat{p}_{\ell,a^*}^{\text{init}}\right) \\ &\quad + \sum_{u \in \mathcal{A}} \beta_u \Big\langle \boldsymbol{w}_u^{M+1}, \left(\theta_u^M \circ \cdots \circ \theta_u^1\right)(\boldsymbol{x}_\ell) \Big\rangle, \quad a \in \mathcal{A}, \end{aligned} \tag{3.2}$$

where $a^* \in \mathcal{A}$ denotes the reference level. As reference level a^* we choose, as usual, the class with the highest empirical probability. We use the logit-link for the initial model (GLM or GAM). If we stick to the initial model, we should end up with the neural network with $\alpha_a = 1$ and $\alpha_u = 0, u \neq a, c_u^{M+1}, \beta_u = 0, u \in \mathcal{A}$. These values are provided as initial weights for $(c_a^{M+1}, \alpha_a, \beta_a)_{a \in \mathcal{A}}$ and the weights $(c_a^{M+1}, \alpha_a, \beta_a)_{a \in \mathcal{A}}$ are trained in the fitting algorithm. The structure of the predictor in (3.2) can be again achieved by using concatenation of two layers in Keras: the first layer contains the A-dimensional predictions across A classes from the initial model, the second layer gives the A-dimensional output in layer $M+1$ from M hidden layers and linear activation function in layer $M+1$.

Let us remark that if we resign from fitting an initial regression model M_0, then we use a homogeneous model M_0 which means that we initiate the neural network by setting

the weight c^{M+1}, respectively the weights $(c_a^{M+1})_{a\in\mathcal{A}}$, equal to the logarithm of the sample mean of the response, or the sample log-odd ratios for the categorical classes.

Neural networks NN_0 and NN_1 for Models 1-5 are fitted by minimizing the categorical cross-entropy or the unscaled deviance loss of the regression model on a validation set. As far as fitting of double Gamma regressions in Models 3 and 5 is concerned, we use a simplified approach. We first fit a neural network NN_0 or NN_1 to the mean of the response assuming Gamma distribution with constant dispersion coefficient, i.e. we consider (2.7)-(2.8). Next, we calculate the Pearson residuals and we fit a second neural network NN_0 or NN_1 for the dispersion coefficient, i.e. we consider a second Gamma regression with constant dispersion coefficient (2.11)-(2.12). We do not iterate the estimation process as it is done when we fit double GLMs/GAMs. Even after one iteration we observe improvements in the residuals and more iterations will slow the calibration of the models.

We use drop-out probabilities as a regularization technique. In order to guarantee that the mean prediction is equal to the sample mean of the response, we perform the bias regularization technique proposed in Wüthrich (2020). For each categorical Model 1,2 and 4, we estimate a multinomial or binomial GLM with canonical link function (logit-link) where we use neurons from the last output layer estimated for the neural network as regressors in the GLM. For Gamma Models 3 and 5, we simply scale the predictions from the neural networks to obtain the correct sample mean of the response (this also includes the predictions for the dispersion coefficients and we assume that the average of the predictions for the dispersion coefficients is equal to the constant dispersion coefficient estimated with the method presented below).

The cross-entropy loss function (2.4) is directly available in Keras. The Gamma unscaled deviance loss function (2.13) is not available but can be coded as a custom loss function. Since the constant dispersion coefficient in the Gamma regression model is estimated independently of the regression function for the mean response, we can use the Gamma unscaled deviance loss when fitting neural networks in Models 3 and 5. The constant dispersion coefficient for the Gamma regression neural network is estimated with the following approach: We estimate the dispersion coefficient in the initial model M_0 using Pearson residuals and traditional estimation techniques from GLMs/GAMs. The dispersion coefficient for the neural network is estimated as the dispersion coefficient for M_0 scaled with a factor which shows an improvement in the prediction accuracy measured with the reduction in the loss functions on a validation set. This approach seems to be reasonable since drop-out probabilities and an early stopping rule are implemented, hence the number of degrees of freedom is not clear. Since the estimate of the dispersion coefficient in M_0 is based on Pearson residuals, not on deviance residuals, we use a loss function based on Pearson residuals to scale the dispersion coefficient from M_0.

3.4 Transformations of variables for the neural networks

It is known that data should be pre-processed before training a neural network. We now turn to the discussion of the pre-processing of continuous regressors and the corresponding responses, as well as to the coding schemes for categorical variables.

- We advise to transform all skewed continuous predictors, except the predictions

from the initial model M_0, by applying the logarithmic transformation:

$$T(x) = \begin{cases} -\log(1-x) & \text{if} \quad x < 0, \\ 0 \quad \text{if} \quad x = 0, \\ \log(1+x) & \text{if} \quad x > 0. \end{cases} \tag{3.3}$$

The logarithmic transformation produces a more balanced regressor with much less skewed values. We do not need the logarithmic transformation when fitting GLMs/GAMs/trees, but for a unified approach, the logarithmic transformation is also applied to regressors when the GLM/GAM/tree is fitted. Moreover, the logarithmic transformation reduces the collinearity effect between cumulative payments and claim incurred. Let us remark that we do not have to apply the logarithmic transformation to predictors which are not skewed. In our numerical example the logarithmic transformation is not applied to development periods.

- After the logarithmic transformation (3.3) is applied (if necessary), we normalize the regressors. We apply the *MinMaxScaler* transformation in each development period where we fit a new regression model. We note that we should not apply the *MinMaxScaler* transformation to skewed regressors, hence a logarithmic transformation is first required. If we directly apply the *MinMaxScaler* transformation to a skewed regressor, then majority of values of the transformed regressor will concentrate close to 0, as the range of a skewed regressor in the sample is likely to be very large. Taking logarithm first and next applying the *MinMaxScaler* transformation will give us a more balanced values in the interval $[-1, 1]$.

- The predictions from the initial model M_0, treated as a regressor, are not normalized. We initiate the training algorithm by using the bias zero and attaching weight one to the prediction from the initial model and weight zero to the neural network's output from the M-th hidden layer. If the initial predictions were normalized, we had to up-date, accordingly, the bias and the initial weight given to the initial prediction in order to start the training process of a neural network from the initial model.

- The categorical regressors are coded with the one-hot encoding procedure. In our numerical example the categorical regressors can take only up to 5 levels so there is no need to use embedding layers. The categorical regressors for GLMs/GAMs/trees can be coded with standard factor procedure with dummy variables.

- The responses for the Gamma regressions are scaled so that their empirical means are equal one. The scaling of the response with its sample mean is less important than pre-processing of regressors since the predictions from the initial model and the weights initialized in the training process of a neural network already set the proper scale for the response. The scaling of the response for the Gamma regression is performed in each development period where we fit a new regression model. Pearson residuals used for estimating dispersion coefficients are not scaled and the Gamma neural networks for dispersions are initiated at the point equal to the constant dispersion coefficient.

- Large incremental payments and large claim incurred are removed in each development period from the training set before Gamma regression models are fitted. Pareto distributions are fitted to the large claims on the original scale.

For more details we refer to Wüthrich (2019).

3.5 Estimation of the large claims distributions

In each development period investigated, $k = 1, 2, \ldots, K+1$, we fit Pareto distributions to large incremental payments and large claim incurred. In each case, we identify the threshold, above which the claim is interpreted as a large claim, and estimate the tail index of large claims. The threshold d_k and the tail index γ_k are chosen with classical methods by graphical inspection of the Pareto quantile plot and the altHill plot on the logarithmic scale. The tail index is estimated with the Hill estimator. Since we are interested in modeling large claims, we should only consider very high quantiles. The parameter λ_k of the Pareto distribution (2.22) is estimated with the method of moments, once γ_k is found.

Let us remark that the model for large claims is based on a conditional distribution given the claim exceeds the threshold. Hence, Models 3 and 5 should be reformulated as conditional models for attritional claims given that the claim is below the threshold. However, the deviance for truncated Gamma distribution is not tractable. Hence, we calibrate neural networks for Gamma regression models using the unconditional Gamma deviance (2.13). In our simulations, attritional incremental payments and attritional claim incurred generated with the Gamma regression models are truncated at the appropriate thresholds.

4 Numerical example

In this section we present the fitting results of our neural networks to real observations. We analyze the Markovian assumptions for the real claim development time series, the goodness-of-fit of the assumed Gamma distributions and investigate the predictive power of our neural networks in claims reserving. Supporting figures are presented in the Appendix. In Section 5 we compare the ultimate payments from RBNS claims predicted with our regression models with the ultimate payments from RBNS claims predicted with the classical Chain-Ladder method.

It is known from neural network theory that one hidden layer networks are universal approximators, see Cybenko (1989) and Hornik et al. (1989). However, deep neural networks with multiple hidden layers may faster capture interactions between variables and better stabilize the calibration process, in general, see Grohs et al. (2019) for some analytical convergence results. We expect that for earlier development periods it is possible to fit deeper neural networks. However, for later development periods we have to use shallow neural networks due to the limited number of observations and increasing number of predictors related to the whole claim history. Moreover, to prevent neural networks from overfitting, we should use early stopping rules and higher drop-out probabilities if the number of calibrated parameters is large compared to the number of available observations.

4.1 Description of the insurance portfolio

We have a data set consisting of $1,332,495$ individual claims. The data set describes the development processes of claims with accident dates and reporting dates both between January 2005 and December 2018 (the latter being called the cutoff date). It contains incremental payments and case reserves on a monthly basis for all reported claims, starting from the reporting date. We remark that a claim is included in our regression modeling in development period k if the development of that claim in development period k can be observed before the cutoff date of December 2018. This includes claims that were closed at an earlier date, because these claims may be re-opened. The data set also includes information about claim features. We have information about claim type (property or bodily injury), claim segment (5 segments) and claim origin (where the claim was caused - Poland or abroad). We recall that the regressors, which we use in this study, are described in Section 3.2.

In the first step, we applied data pre-processing as follows:

- We scaled incremental payments and case reserves with a constant in order to anonymize the results.
- We removed the claims for which we identified inconsistencies in accident dates, reporting dates and settlement months. In a company, one should apply data cleaning to such claims by doing further investigation with involved persons.
- We investigated incremental payments and case reserves. In order to simplify the modeling process for the purpose of the paper, we only model positive payments and resign from modeling negative payments (salvages and subrogations). Consequently, we do not fit Model 2 and Model 3_negative in this paper. For simplicity, negative payments were set to zero in the data set.
- We ended up with a cleaned data set consisting of $1,331,856$ individual claims. We removed less than 0.1% of the claims from the date set. The change in the aggregate payments historically observed was 1.2%, mainly caused by removal of recoveries.
- The claims incurred were calculated for all claims.

For computational reasons, we fit the neural networks on quarterly data. The unit of the accident period, the reporting period and the development period is 3 months. Consequently, $P_k^{i,j}$ denotes the sum of incremental monthly payments made in the k-th quarter starting from the reporting date, and $I_k^{i,j}$ denotes the value of the claim incurred at the end of the k-th quarter starting from the reporting date (in quarterly units) for a specific claim. We deal with $k = 1, 2, \ldots, 55$, since we have claims from 14 accident years in the data set.

Table 4.1 presents the number of observations which we can be used for fitting our regression models in consecutive development periods $k \geq 1$. It is clear that the number of available observations decreases with the development period. This means that we are not able to estimate separate regression models for all later development periods $k = 1, 2, \ldots, 55$. We decide to fit separate neural networks for $k = 1, \ldots, 16$, and one

Development	Model 1				Model 3_positive	Model 4		Model 5
k	Case 0	Case 1	Case 2	Case 3	Payments	Case 0	Case 1	Claims incurred
1	321,618	87,926	123,745	762,394	850,320	82,430	803,709	82,430
2	1,094,980	10,386	46,495	114,158	124,544	39,053	121,600	39,053
4	1,148,925	1,573	28,366	24,109	25,682	23,650	28,825	23,650
8	1,062,232	360	18,056	6,589	6,949	16,801	7,844	16,801
12	948,516	157	16,619	3,485	3,642	15,563	4,541	15,563
14	893,918	139	13,311	2,102	2,241	10,954	4,459	10,954
16	842,072	86	9,155	1,162	1,248	7,511	2,806	7,511
20	732,314	40	3,328	559	599	1,990	1,897	1,990
30	433,175	8	137	89	97	113	113	113
40	170,629	0	20	9	9	20	9	20
45	88,120	0	10	2	2	8	4	8
50	35,571	0	4	1	1	1	4	1

Table 4.1: Number of observations available for estimating the regression models for development periods $k \geq 1$.

neural network for all development periods $k = 17, \ldots, 55$, for each Model 1, 3_positive, 4, 5. The neural network for the last development periods is denoted as a neural network for development period $k = 17$. The empirical probability that the incremental payment is zero and the claim incurred do not change (Case 0) in Model 1 in period $k = 16$ is 98.78%, and increases above 99% for later periods. We use 99% as a threshold probability in Model 1 for fitting separate neural networks in each development period.

In the next subsections we present the estimation results of our regression models for the selected development periods $k = 2, 8, 16, 17$. We work with neural networks with 2 hidden layers. The rule of thumb is that the number of neurons in the first layer should be between the number of regressors and three times the number of regressors. The largest number of regressors is in Models 4 and 5, where we have 20 regressors plus regressors related to the claim history in the number of 2 times the development period considered. We investigate two choices for the number of neurons in the first layer: $q_1 = (20 + 2 \cdot \mathit{development\_period}) \cdot 2$ and $q_1 = (20 + 2 \cdot \mathit{development\_period}) \cdot 1.5$. In the second layer we choose, respectively, $q_2 = 10$ and $q_2 = 5$.

The data set is split to a training set and a validation set. We use a random split where 90% of the observations are allocated to the training set. If we fit a categorical regression model (Model 1 or Model 4), then we use a stratified sampling and the proportions of the classes in the training set are the same as in the whole data set. In training the neural networks, we use a batch size of $10,000$ observations, 500 epochs for the categorical regressions, $1,000$ epochs for the Gamma regressions, and drop-out probabilities from 1% to 30%. Our experiments show that we need less epochs to train the categorical regressions than the Gamma regressions. We use a low learning rate at 0.001. We also test higher learning rates and we do not observe improvements in calibrations.

4.2 Predictions in development period $k = 2$

In development period $k = 2$ we have many observations available for fitting each regression model, see Table 4.1. Moreover, the number of predictors, which we want to use in neural networks NN_1, is small due to the short history of the claim development. Hence, we expect that we can fit deep neural networks in development period $k = 2$ with a large number of trainable parameters and a low drop-out probability. We choose drop-out probability at 1%.

First, we analyze large claims, i.e. large incremental payments and large changes in claim incurred. We use Hill plots and we try to find the most stable region in the Hill's

estimates of the tail index, see Figure A.1. We choose the probability of a large claim at 1% for incremental payments and claim incurred. The estimated tail indices are presented in A.18. In Figure A.2 we study the proportions of claim features in the whole data set available for estimation of the model under investigation and in the data set consisting of large claims. We can conclude that the proportion of claims from Segment 1, bodily injury claims and claims from abroad increase in the subset of large claims. This conclusion is better pronounced in later development periods in Figures A.6, A.10 and A.14. Consequently, in simulations of ultimate payments, we assume that 75% of large incremental claims are allocated in the first place to bodily injury claims and claims from abroad, and 75% of large changes in claim incurred are allocated in the first place to Segment 1, bodily injury claims and claims from abroad. Remaining large claims are allocated to random claims. The proportion 75% is an arbitrary number.

Next, we fit Models 1, 3_positive, 4, 5. We should decide which model should be chosen as an initial model M_0 from which we start training our neural networks. It turns out that GAMs and regression trees with claim incurred in the last development period and cumulative payments do not reduce significantly the loss function on the training set compared to a homogeneous model for Models 1 and 4. Hence, we decide to initiate the training of the neural networks for Models 1 and 4 starting from empirical estimates. For Models 3_positive and 5 we use regression trees with claim incurred in the last period and cumulative payments as initial models M_0 since we observe a significant decrease in the loss function on the training set compared to a homogeneous model. We initiate the training of the neural networks for Models 3_positive and 5 starting from the estimates produced by the regression tree. We could start with GAMs as M_0 and reduce the loss function at the initial training step of neural networks even more. Yet, regression trees are faster for predictions than GAMs. The disadvantage is that we have to train Gamma neural networks for more epochs.

The cross-entropy and deviance loss functions on the validation sets are presented in Table 4.2 and Figure A.3. As a benchmark, we also provide the loss functions in the GAMs where we use claim incurred in the last development period and cumulative payments as the only regressors. The predictions from M_0 and GAM can be improved with neural networks NN_0. We can conclude that not only the cumulative payments and the claim incurred in the last development period are useful for predictions in the next development period, but we should also use other claim features and interactions between the predictors in claims reserving. Moreover, we can observe that the use of the whole claim history also improves the predictions. Of course, for $k = 2$, the use of the whole claim history means to use the two historical values of the incremental payments and the claim incurred observed in $k = 0$ and $k = 1$. Hence, the regression models are not significantly enhanced by additional information about the claim development process. The models' enhancement is much bigger for later development periods where, we show in the next subsections, we should also use the whole claim history, see Tables 4.3, 4.4 and Figures A.7, A.11. It is clear that the predictions are improved in all Models 1, 3_positive, 4, 5 when we add incremental payments and claim incurred (observed in all past development periods) as regressors in the regression functions, i.e. when we switch from neural network NN_0 to neural network NN_1. Consequently, the Markovian assumption should be rejected for the claim development processes in our data set. Finally, deeper neural networks are preferred in $k = 2$ since we have a large number of observations at our disposal and there

is no danger that we overfit the models (by appropriately early stopping and drop-out probabilities).

	D_{GAM}	D_{NN_0}	D_{NN_1}	$1-\frac{DNN_0}{D_{\text{GAM}}}$	$1-\frac{DNN_1}{D_{\text{GAM}}}$
Model 1: $q=(48,10)$	0.3865	0.3243	0.3206	16.11%	**17.06%**
Model 1: $q=(36,5)$	0.3865	0.3249	0.3229	15.95%	16.46%
Model 3_positive: $q=(48,10)$	0.5309	0.4842	0.4762	8.79%	**10.30%**
Model 3_positive: $q=(36,5)$	0.5309	0.4880	0.4823	8.08%	9.14%
Model 4: $q=(48,10)$	0.4900	0.1895	0.1888	61.32%	**61.47%**
Model 4: $q=(36,5)$	0.4900	0.2125	0.2117	56.63%	56.79%
Model 5: $q=(48,5)$	0.1714	0.1263	0.1250	26.34%	**27.10%**
Model 5: $q=(36,5)$	0.1714	0.1311	0.1304	23.54%	23.94%

Table 4.2: Minimal cross-entropy and deviance loss functions on validation sets observed during the training of the neural networks in $k=2$.

We now validate the distributional assumption that the incremental payments and the claim incurred are from Gamma distributions with the moments, respectively, given by (2.9)-(2.10), (2.18)-(2.19). As in GLMs/GAMs, we use Pearson and deviance residuals.

If the response v_ℓ follows Gamma distribution with true expected value μ_ℓ and dispersion coefficient ψ_ℓ, then the Pearson residual, after adding 1, i.e. $Res_\ell+1=\frac{v_\ell-\mu_\ell}{\mu_\ell}+1$ has a Gamma distribution with expected value equal to 1 and dispersion ψ_ℓ. We can use a version of a QQ normal plot to verify the Gamma distributional assumption. If Res_ℓ denotes an observation from the Gamma distribution with true expected value equal to 1 and dispersion ψ_ℓ, in our case Res_ℓ denotes the Pearson residual from the fitted Gamma regression model, then $F_{\Gamma(1,\psi_\ell)}(Res_\ell)$ has a uniform distribution, where $F_{\Gamma(1,\psi)}$ denotes the cumulative Gamma distribution function for parameters 1 and ψ. Consequently, $\Phi^{-1}\big(F_{\Gamma(1,\psi_\ell)}(Res_\ell)\big)$ has a normal distribution, where Φ^{-1} denotes the inverse of the standard normal distribution function. If the Gamma distribution is the correct distribution for the response, then we should observe a straight line in the QQ-plot for the Pearson residuals from the fitted model. Let us recall that the dispersion coefficients are estimated for individual claims with the second Gamma regression model. The mean values of the dispersion coefficients estimated for individual claims in Models 3_positive and 5, which coincide with the constant dispersion coefficients in the first Gamma regression model, are presented in Figure A.18.

We also investigate the so-called Tukey-Anscombe plot where we plot the deviance residuals from the fitted Gamma regression model, scaled with the dispersion coefficients, against the log-predictions of the response in the fitted model. If the mean and variance assumptions for the distribution of the response are correct, then the deviance residuals should be fluctuate randomly around the horizontal line through zero and the deviation in the residuals should be constant.

The QQ normal plots and Tukey-Anscombe plots for the Gamma regressions fitted with NN_1 are presented in Figure A.4. We observe that the left tail is underestimated for incremental payments and both right and left tail are underestimated for claim incurred. If we look at later development periods, see Figures A.8, A.12, A.16, we can conclude that the Gamma distribution can be accepted for incremental payments. The choice of the Gamma distribution for claim incurred can be questioned and an improvement would be desirable. In practice, it is difficult to get a perfect fit to a Gamma distribution

(with constant or varying dispersion). We should at least aim at correctly modeling the mean response and the dispersion with two neural networks. If we investigate deviance residuals, then we can conclude that the first two moments of the response are correctly modeled for incremental payments and claim incurred in all development periods. This property should be sufficient for projecting the aggregate ultimate payments in a large portfolio, estimating the best estimate and not-extreme quantiles of the aggregate ultimate payments. We would like to point out that varying dispersion improves the deviance residuals and the second neural network for the dispersion coefficient in Model 3_positive and 5 is indeed required.

4.3 Predictions in development period $k = 8$

We present the same tables and plots as in the previous section, see Table 4.3 and Figures A.5, A.6, A.7, A.8. The conclusions are the same. We choose the probability of a large claim at 1%. We could differentiate the probabilities of large claims in development periods, but, if possible, we try to choose the same probability for large claims in all development periods. The drop-out probability is still set at 1%. We can deduce that the cumulative payments and the claim incurred in the last development period are not sufficient, except in Model 5, for precise predictions of the claim development processes. The Markovian assumption has to be rejected for the claim development processes. The fit of a Gamma distribution to incremental payments and claim incurred is acceptable. We prefer larger neural networks. Yet, the need to apply the early stopping rule, in order not to overfit Model 3_positive, can be observed in Figure A.7.

	D_{GAM}	D_{NN_0}	D_{NN_1}	$1 - \frac{DNN_0}{D_{\text{GAM}}}$	$1 - \frac{DNN_1}{D_{\text{GAM}}}$
Model 1: $q = (72, 10)$	0.1086	0.0499	0.0487	54.03%	**55.17%**
Model 1: $q = (54, 5)$	0.1086	0.0503	0.0488	53.73%	55.03%
Model 3_positive: $q = (72, 10)$	0.6190	0.5763	0.5593	6.90%	**9.64%**
Model 3_positive: $q = (54, 5)$	0.6190	0.5707	0.5657	7.80%	8.61%
Model 4: $q = (72, 10)$	0.5533	0.1941	0.1742	64.92%	**68.52%**
Model 4: $q = (54, 5)$	0.5533	0.1939	0.1792	64.96%	67.61%
Model 5: $q = (72, 10)$	0.1152	0.0349	0.0390	**69.71%**	66.17%
Model 5: $q = (54, 5)$	0.1152	0.0431	0.0424	62.61%	63.19%

Table 4.3: Minimal cross-entropy and deviance loss functions on validation sets observed during the training of the neural networks in $k = 8$.

4.4 Predictions in development period $k = 16$

In development period $k = 16$ (4 years), the set of predictors which we want to use in neural network NN_1 is rather large. We want to use 51 or 52 regressors in total (where 32 regressors are related to incremental payments and claim incurred observed in the past development periods). At the same time, the number of observations is relatively small for Models 3_positive. Since the number of neurons in the first hidden layer should be at least equal to the number of regressors, the number of trainable parameters in Models 3_positive quickly approaches, and exceeds, the number of observations when we increase the number of neurons in NN_1, which we would do to improve the fit of the neural

network. Consequently, the loss function may, already after few iterations, increase on a validation set and the fitting algorithm may produce a poor prediction model. In order to prevent over-fitting, we can apply regularization techniques (early stopping may not be sufficient). We decide to increase the drop-out probability to get stable calibration results for Model 3_positive. We choose 30% in $k = 16$. In fact, we decide to increase the drop-out probability from 1% to 30% for Model 3_positive starting from the development period $k = 11$.

We also modify the probability of a large claim in Model 3_positive. The reason is again a low number of observed payments. We choose 2% in development period $k = 16$, and 1.5% in $k = 15$. Hill plots, Pareto quantile plots and proportions of claim features in the two data sets of all claims and large claims are presented in Figures A.9, A.10.

The results of fitting neural networks for Models 1, 3_positive, 4, 5 are presented in Table 4.4 and Figure A.11. The conclusions are the same as in the previous sections. The larger neural network is also preferred for Model 3_positive, since it is regularized with a large drop-out probability. However, when we re-run the fitting algorithm then it turns out that in many runs the smaller network is preferred, see Figure A.17. Hence, we decide to switch to smaller neural networks for Model 3_positive starting from development period $k = 11$. For Models 1, 4 an 5 we use larger neural networks in all development periods.

	D_{GAM}	D_{NN_0}	D_{NN_1}	$1 - \frac{D_{NN_0}}{D_{\text{GAM}}}$	$1 - \frac{D_{NN_1}}{D_{\text{GAM}}}$
Model 1: $q = (104, 10)$	0.0653	0.0132	0.0130	79.84%	**80.10%**
Model 1: $q = (78, 5)$	0.0653	0.0136	0.0132	79.15%	79.85%
Model 3_positive: $q = (104, 10)$	0.5425	0.4988	0.4781	8.05%	**11.87%**
Model 3_positive: $q = (78, 5)$	0.5425	0.4972	0.4943	8.35%	8.88%
Model 4: $q = (104, 10)$	0.5350	0.2540	0.2417	52.51%	**54.82%**
Model 4: $q = (78, 5)$	0.5350	0.2569	0.2568	51.98%	51.99%
Model 5: $q = (104, 10)$	0.0347	0.0297	0.0285	14.41%	**17.96%**
Model 5: $q = (78, 5)$	0.0347	0.0319	0.0305	8.23%	12.22%

Table 4.4: Minimal cross-entropy and deviance loss functions on validation sets observed during the training of the neural networks in $k = 16$.

The residuals are presented in Figure A.12. The fit of Gamma distributions is acceptable. Yet, the estimated Gamma distribution now overestimates both tails of the claim incurred.

4.5 Predictions beyond development period $k = 16$

All claims with development periods bigger than 16 quarters are grouped into one data set and one neural network with development period information as a regressor is fitted. Due to this grouping, we cannot fit neural networks NN_1 since claims in different development periods have different lengths of claim history. Hence, we fit neural network NN_0. This is justified since for latter development periods we expect that only the most recent claim history should have the greatest power in the prediction of claim development. Moreover, the results from the previous sections show that neural networks NN_1 are only slightly better than NN_0 in predictions on the validation sets.

Hill plots, Pareto quantile plots and proportions of claim features in two data sets of all claims and large claims are presented in Figures A.13, A.14. We choose the probability

of a large incremental payment at 3%, and the probability of a large claim incurred at 0.5%. The probability of a large payment is higher than in the previous models. This agrees with intuition as heavier claims occur at later development periods.

The results of training the neural networks are presented in Table 4.5 and Figure A.15. It is clear that NN_0 has better predictive power than M_0 and GAM, and we prefer larger neural networks. Residuals are presented in Figure A.16. We conclude that the fit is acceptable.

	D_{GAM}	D_{NN_0}	$1 - \frac{DNN_0}{D_{\text{GAM}}}$
Model 1: $q = (48, 10)$	0.0176	0.0048	**73.013%**
Model 1: $q = (36, 5)$	0.0176	0.0048	73.012%
Model 3_positive: $q = (48, 10)$	0.6044	0.5555	**8.08%**
Model 3_positive: $q = (36, 5)$	0.6044	0.5656	6.42%
Model 4: $q = (48, 10)$	0.6431	0.3880	**39.67%**
Model 4: $q = (36, 5)$	0.6431	0.4506	29.94%
Model 5: $q = (48, 10)$	0.0712	0.0300	**57.84%**
Model 5: $q = (36, 5)$	0.0712	0.0344	51.77%

Table 4.5: Minimal cross-entropy and deviance loss functions on validation sets observed during the training of the neural networks in $k = 17$.

5 Projection to ultimate payments for RBNS claims

In the first step, we use Chain-Ladder techniques to estimate the ultimate payments for RBNS claims. In Table 5.1 we present the Chain-Ladder estimates of the development factors calculated on quarterly data for aggregate payments and the numbers of reported claims. We use all observations in the period $2005 - 2018$. We can conclude that we deal with an insurance portfolio with long-tailed development patterns, where late development factors for payments are still bigger than 1.

Development	1	2	3	5	10	20	30	40	50
CL factor: payments	2.124071	1.167995	1.083916	1.035170	1.015100	1.005782	1.003623	1.002830	1.001223
CL factor: no. of claims	1.121514	1.024738	1.011397	1.003963	1.001042	1.000209	1.000317	1.000355	1.000311

Table 5.1: Chain-Ladder (CL) estimates of development factors for aggregate payments and the number of reported claims.

We calculate the Chain-Ladder predictions for the aggregate ultimate payments and the ultimate number of reported claims per each accident quarter. Next, we calculate the average ultimate payment per accident quarter by diving these two predictions by each other. In order to differentiate the ultimate payment by the reporting delay of the claim, we calculate the mean values of the claims incurred available at the end of December 2018 per reporting delay and we regress the mean value of the claims incurred versus the reporting delay. We use a GAM regression model to obtain the scaling factors for ultimate payments for claims reported with reporting delays compared to the ultimate payment for a claim reported without a delay. Using the average ultimate payments per accident quarter and reporting delay and the number of claims to be reported in the future periods, we can calculate the IBNYR (Incurred But Not Yet Reported) and RBNS reserves. The aggregate ultimate payments for RBNS claims per accident quarter are calculated as the

predicted aggregate ultimate payments for the portfolio minus the predicted number of claims to be reported in the future periods times the average ultimate payment taking into account the reporting delay of the claims (IBNYR). The RBNS reserve is calculated by subtracting the aggregate payments up to December 2018 from the projected aggregate ultimate payments for the RBNS claims. We remark that this is still a rough estimate of the RBNS reserve to receive numbers that are comparable to our individual claim reserving method. The accident years from 2009 till 2018 cover 98.98% of the total RBNS reserve of all accident years $2005 - 2018$, so we only focus on these years. The results are presented in Table 5.2.

Once Models 1, 3_positive, 4 and 5 are calibrated, we can use them to simulate incremental payments in consecutive development periods (quarters) and generate ultimate payments for RBNS claims. Since we deal with a very large portfolio of claims, we do not have to perform many simulations to receive stable results, unless we are interested in a very high quantile of aggregate ultimate payments. We use 50 simulations, which is sufficient for the results we present below. We consider the claims from the years $2009 - 2018$ but we model their development over the full 55 development periods.

	RBNS reserve				
Accident year	Mean	25th quantile	75th quantile	Chain-Ladder	Case reserve
2009	1.26	1.13	1.39	1.35	0.65
2010	1.51	1.32	1.68	1.97	1.56
2011	2.29	2.03	2.39	2.66	3.14
2012	3.65	3.26	3.78	3.85	3.63
2013	5.10	4.69	5.18	5.11	5.07
2014	5.99	5.43	6.35	6.61	7.00
2015	8.27	7.57	8.36	9.36	6.90
2016	11.91	11.48	12.27	13.43	10.81
2017	17.20	16.67	17.80	19.62	11.65
2018	35.09	34.64	35.59	39.77	28.02
All	92.27	88.21	94.78	103.74	78.43

Table 5.2: Simulations results and Chain-Ladder (CL) estimates (in MM).

In Figure 5.1 we present histograms of the aggregate ultimate payments for the whole insurance portfolio, as well as for segments and different claim features (property versus bodily injury and Poland versus abroad). Such results can help the insurer to improve claims reserving as the insurer can now set claims reserves for policies grouped by claim features. Even at the portfolio level, we expect that the mean prediction from our models should be more precise than the Chain-Ladder estimate derived from aggregate data since our models are calibrated based on detail information about the claim development process, they can cope with seasonality and with changes in portfolio mix. In our case, the proportion of body claims in the portfolio decreases from 10% in 2005 to 5.5% in 2018, and the proportion of the claims caused abroad decreases from 5.7% to 2.6%. Since body claims and claims from abroad are expected to be the most severe, the Chain-Ladder estimates might be overestimated for the recent accident periods. Apart from calculating the best estimate of liabilities, the histograms in Figure 5.1 can also help the insurer to understand which claim features generate the most uncertainty of future payments. Such results can be used to calculate risk adjustments in IFRS 17 or define a safety buffer in claims reserves. We remark that there is a peak in the right tail of the histogram of the

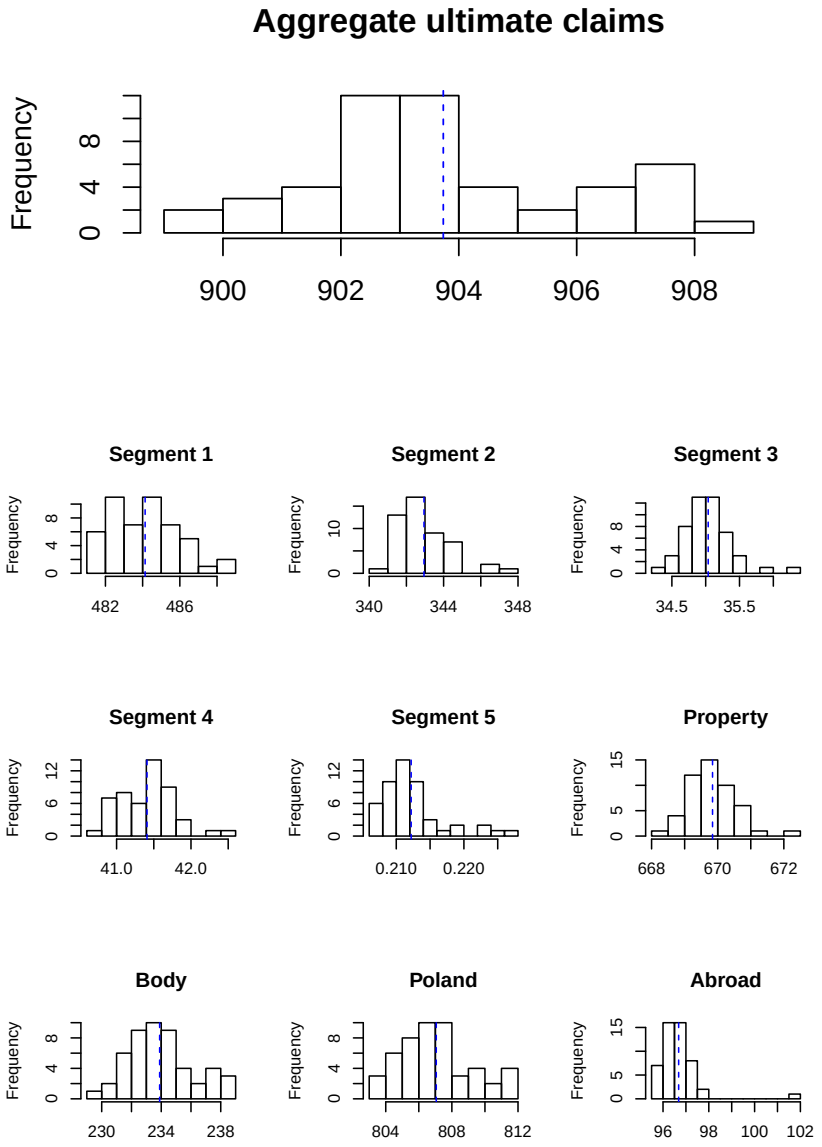

Figure 5.1: Histograms of the aggregate ultimate payments from the RBNS claims (in MM). The dashed lines indicate the mean value from the simulations.

aggregate ultimate payments. We conclude that the portfolio is exposed to large claims, which mainly occur in late development periods from body claims and claims from abroad.

In Figure 5.2 we find boxplots of the aggregated ultimate payments per accident year. Deviations in the simulated payments are small which justifies a small number of simulations, see also 25th and 75th quantiles in Table 5.2. In Figure 5.2 we compare the ultimate payments for the RBNS claims simulated with our model with the results estimated with the Chain-Ladder technique, and in Table 5.2 we compare the RBNS reserves. We can observe that our model produces lower numbers for years $2009 - 2018$ than the Chain-Ladder estimates. This observation may be explained with the change of

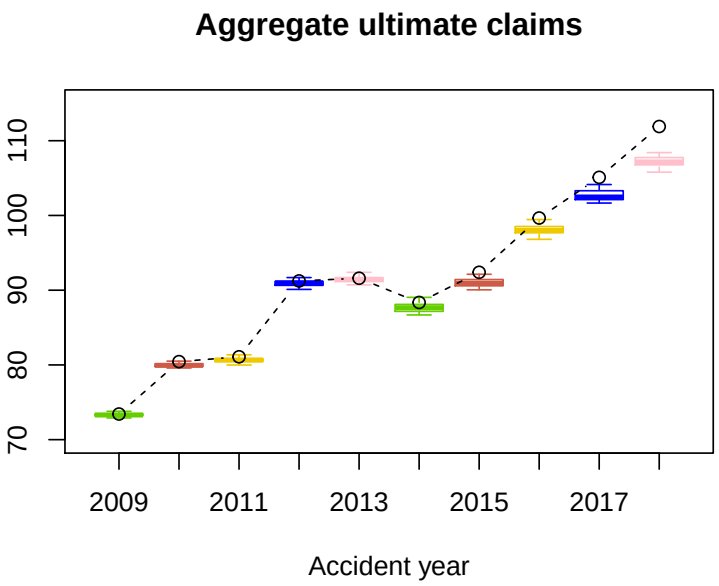

Figure 5.2: Boxplots of the aggregate ultimate payments from the RBNS claims (in MM). The dots indicate the Chain-Ladder estimates.

the proportion of body claims and claims from abroad in the insurance portfolio, which we discuss above. Moreover, the Chain-Ladder development factors for old accident years and early development periods tend to be systematically higher than the development factors for recent accident years, see Figure A.19 where dark blue points cumulate in the top left corner of the triangle, especially for the first development period for the old accident years. Consequently, the Chain-Ladder development factors tend to overestimate the future payments.

We can also compare the company's case reserves with the RBNS reserves estimated with our model, see Table 5.2. The comparison is not obvious since case reserves depend on company's claims reserving policy. For some old accident years the case reserves are above the RBNS reserves. This might be caused by the fact that for old claims, which are not closed, the case reserve is not revaluated and is kept unchanged until the full settlement is made. For recent accident years the case reserves are below the RBNS reserves, as expected.

We estimate the expected ultimate payment for an individual claim, see Figure 5.3. It agrees with our intuition that the expected ultimate payment for a property claim is lower than for bodily injury one. It also agrees with our intuition that the expected ultimate payment for a claim caused in Poland is lower than for a claim caused abroad. We can also confirm that the expected ultimate payment for an individual claim increases with the reporting delay of the claim.

Finally, we check stability of our calibration procedure. We re-calibrate all neural networks and we re-simulate the developments of all reported claims. When refitting the neural networks, we also change the split of the data of training and validation sets and, this time, we use 80% of observations as training set and the remaining 20% as test set. The results are presented in Table 5.3. Comparing Tables 5.2 and 5.3, we can conclude

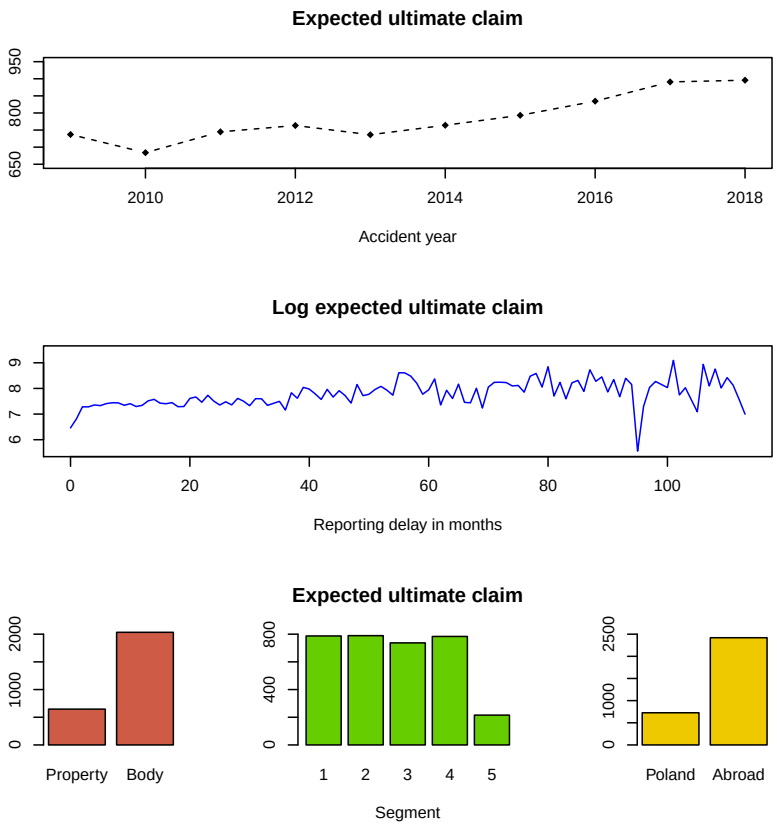

Figure 5.3: Expected aggregate ultimate payments from an individual claim.

that the calibration procedure is very stable on the aggregate level (see RBNS reserves of 89.80 vs. 92.27). Bigger changes are only observed on older accident years where RBNS are very low and, thus, only marginally contribute to the overall RBNS reserves. This uncertainty on older claims is, of course, caused by the fact that payments for these claims are very rare because most of the claims have already been settled.

6 Conclusions

We have developed regression models and postulated distributions which can be used in practice to describe the joint development process of individual claim payments and claim incurred. We have applied neural networks to estimate our regression models. The models can improve claims reserving techniques for RBNS claims and provide additional information about the risk factors which trigger the future payments. Our regression models estimated with individual claims can be used at any level of granularity, from the portfolio level to the policy level. Consequently, the RBNS reserve, which is traditionally calculated at the portfolio level, can be now directly allocated to sub-portfolios/segments/units of accounts in our modelling framework. Finally, since the projected ultimate payments can be split among claims features, the result can also be used to improve pricing and estimate the ultimate loss.

	RBNS reserve				
Accident year	Mean	25th quantile	75th quantile	Chain-Ladder	Case reserve
2009	0.94	0.79	1.08	1.35	0.65
2010	1.31	1.18	1.36	1.97	1.56
2011	1.94	1.74	2.09	2.66	3.14
2012	3.11	2.94	3.32	3.85	3.63
2013	4.51	4.29	4.71	5.11	5.07
2014	5.49	5.17	5.78	6.61	7.00
2015	7.78	7.43	8.02	9.36	6.90
2016	11.40	10.84	11.79	13.43	10.81
2017	17.16	16.83	17.44	19.62	11.65
2018	36.17	35.64	36.68	39.77	28.02
All	89.80	86.87	92.28	103.74	78.43

Table 5.3: Simulations results and Chain-Ladder (CL) estimates (in MM) - a new re-calibration of NNs and a new simulation run.

Acknowledgments: We would like to thank Marcin Szatkowski for helping in analyzing the insurance portfolio. The research of Ł.Delong is financially supported with grant NCN 2018/31/B/HS4/02150.

References

Antonio, K., Plat, R. (2014). Micro-level stochastic loss reserving for general insurance. *Scandinavian Actuarial Journal* **2014/7**, 649-669.

Arjas, E. (1989). The claims reserving problem in non-life insurance: some structural ideas. *ASTIN Bulletin* **19/2**, 139-152.

Baudry, M., Robert, C.Y. (2019). A machine learning approach for individual claims reserving in insurance. *Applied Stochastic Models in Business and Industry* **35/5**, 1127-1155.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* **2**, 303-314.

De Felice, M., Moriconi, F. (2019). Claim watching and individual claims reserving using classification and regression trees. *Risks* **7/4**, 102.

Denuit, M., Hainaut, D., Trufin, J. (2019). *Effective Statistical Learning Methods for Actuaries III: Neural Networks and Extensions*. Springer

Duval, F., Pigeon, M. (2019). Individual loss reserving using a gradient boosting-based approach *Risks* **7/3**, 79.

Ferrario, A., Noll, A., Wüthrich, M.V. (2018). Insights from inside neural networks. *SSRN Manuscript* ID 3226852. Version November 14, 2018.

Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics* **29/5**, 1189-1232.

Gabrielli, A. (2020). A neural network boosted double overdispersed Poisson claims reserving model. *ASTIN Bulletin* **50/1**, xx-xx.

Gabrielli, A., Wüthrich, M.V. (2018). An individual claims history simulation machine. *Risks* **6/2**, 29-43.

Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press.

Grohs, P., Perekrestenko,D., Elbrächter, D., Bölcskei, H. (2019). Deep neural network approximation theory. Submitted to *IEEE Transactions on Information Theory* (invited paper).

Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359-366.

Jessen, A.H., Mikosch, T., Samorodnitsky, G. (2011). Prediction of outstanding payments in a Poisson cluster model. *Scandinavian Actuarial Journal* **2011/3**, 214-237.

Kuo, K. (2019). DeepTriangle: A deep learning approach to loss reserving. Risks **7/3**, 97.

Larsen, C.R. (2007). An individual claims reserving model. *ASTIN Bulletin* **37/1**, 113-132.

Lopez, O., Milhaud, X., Thérond, P.-E. (2019). A tree-based algorithm adapted to microlevel reserving and long development claims. *ASTIN Bulletin* **49/3**, 741-762.

Norberg, R. (1993). Prediction of outstanding liabilities in non-life insurance. *ASTIN Bulletin* **23/1**, 95-115.

Norberg, R. (1999). Prediction of outstanding liabilities II. Model variations and extensions. *ASTIN Bulletin* **29/1**, 5-25.

Pigeon, M., Antonio, K., Denuit, M. (2013). Individual loss reserving with the multivariate skew normal framework. *ASTIN Bulletin* **43/3**, 399-428.

Quarg, G., Mack, T. (2004). Munich chain ladder. *Blätter DGVFM* **XXVI**, 597-630

Schelldorfer, J., Wüthrich, M.V. (2019). Nesting classical actuarial models into neural networks. *SSRN Manuscript* ID 3320525.

Taylor, G., McGuire, G., Sullivan, J. (2008). Individual claim loss reserving conditioned by case estimates. *Annals of Actuarial Science* **3/1-2**, 215-256.

Wüthrich, M.V. (2018). Machine learning in individual claims reserving. *Scandinavian Actuarial Journal* **2018/6**, 465-480.

Wüthrich, M.V. (2019). From generalized linear models to neural networks, and back. *SSRN Preprint* ID 3491790.

Wüthrich, M. V. (2020). Bias regularization in neural networks for generalized insurance pricing. To appear in *European Actuarial Journal*.

Zhao, X.B., Zhou, X., Wang, J.L. (2009). Semiparametric model for prediction of individual claim loss reserving. *Insurance: Mathematics and Economics* **45**, 1-8.

A Figures

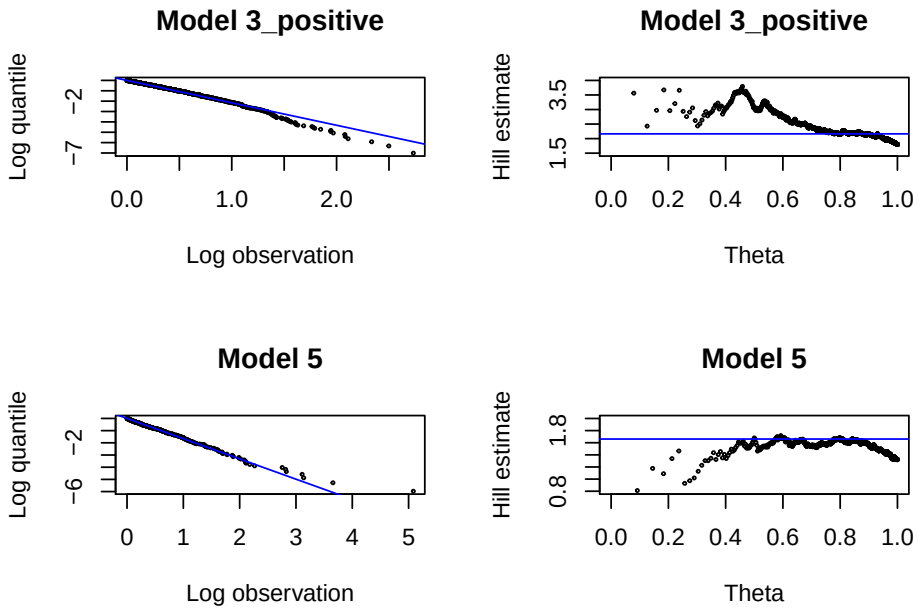

Figure A.1: Pareto quantile plots and altHill plots for incremental payments and changes in claim incurred in $k = 2$.

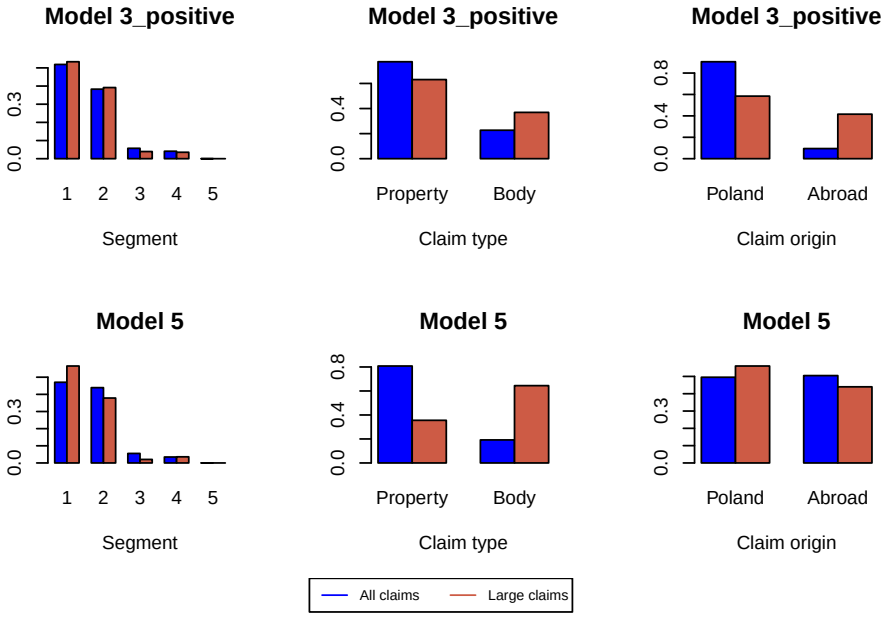

Figure A.2: Proportions of claim features in the whole data set and in the data set of large claims in $k = 2$.

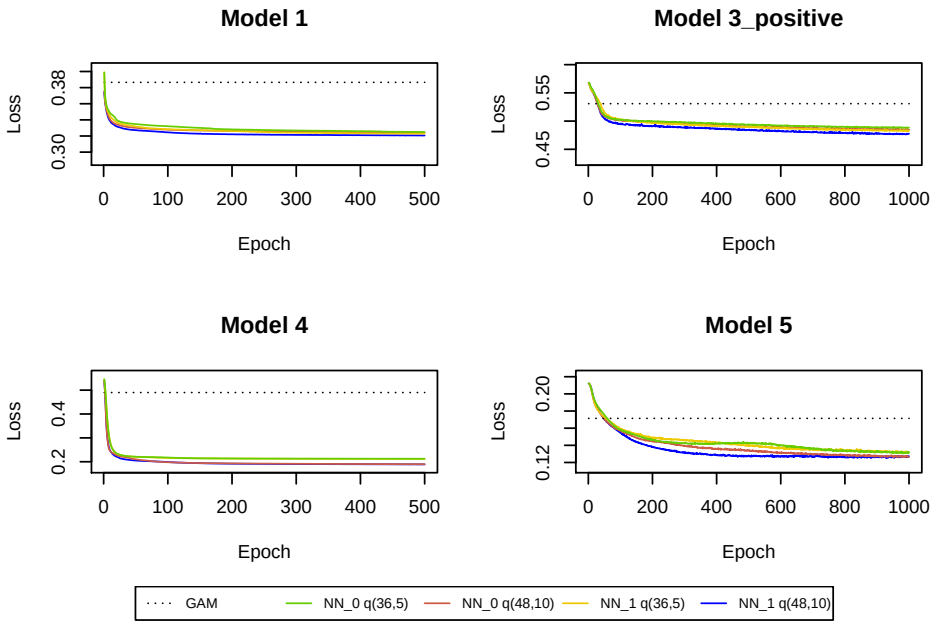

Figure A.3: Cross-entropy and deviance loss functions on validation sets observed during the training of the neural networks in $k = 2$.

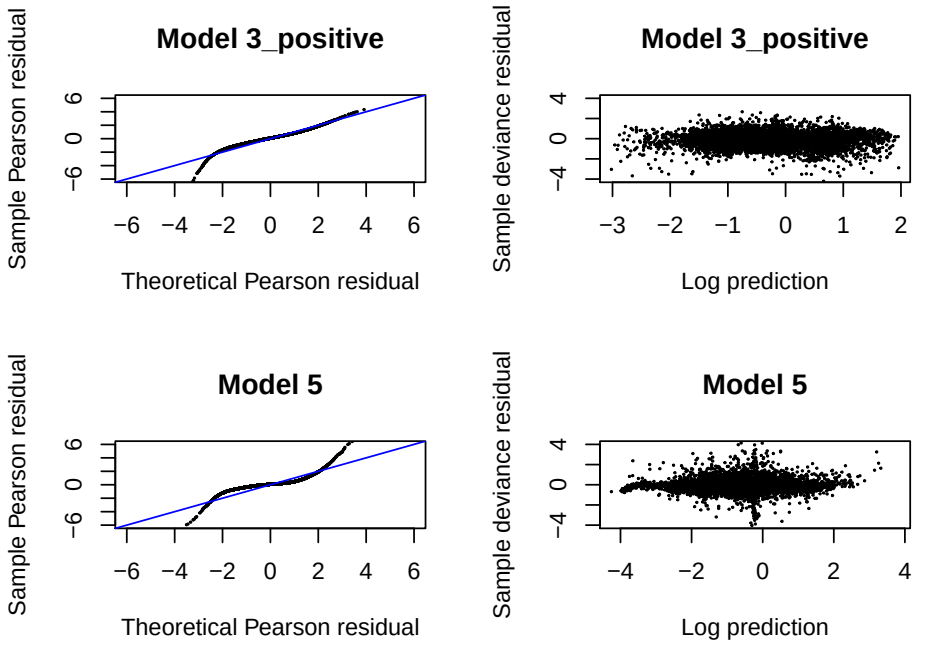

Figure A.4: QQ normal plots and Tukey-Anscombe plots in Models 3_positive and 5 fitted with neural networks NN_1 in $k = 2$

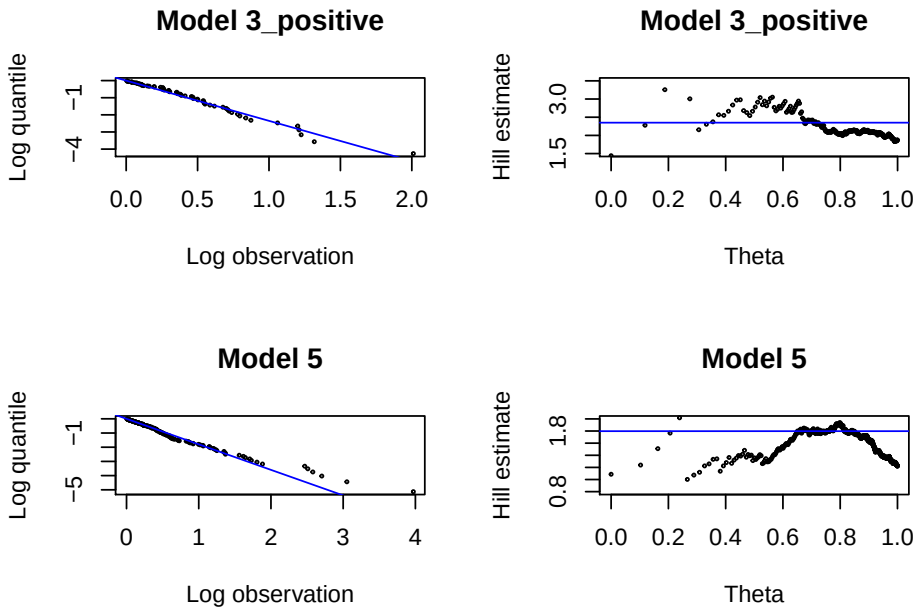

Figure A.5: Pareto quantile plots and altHill plots for incremental payments and changes in claim incurred in $k = 8$.

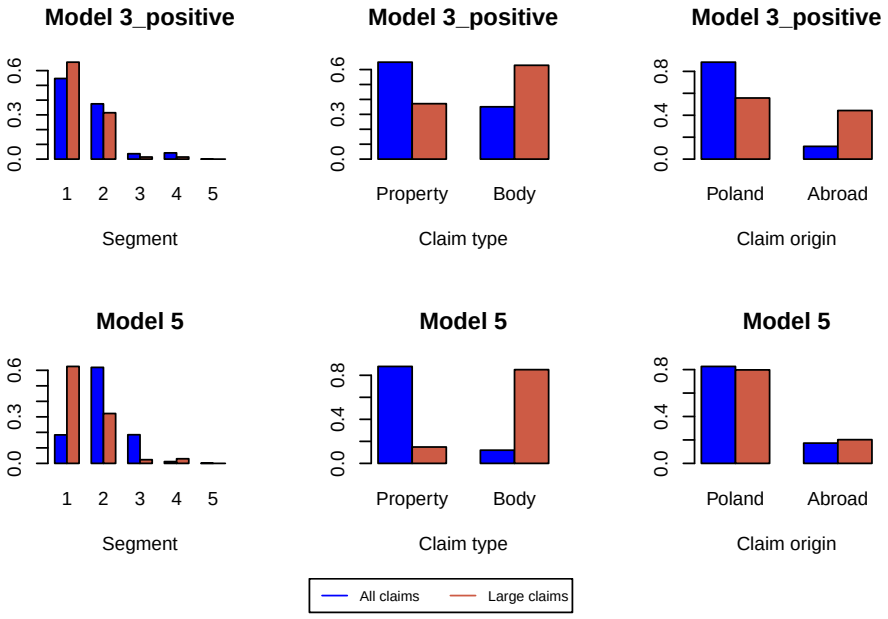

Figure A.6: Proportions of claim features in the whole data set and in the data set of large claims in $k = 8$.

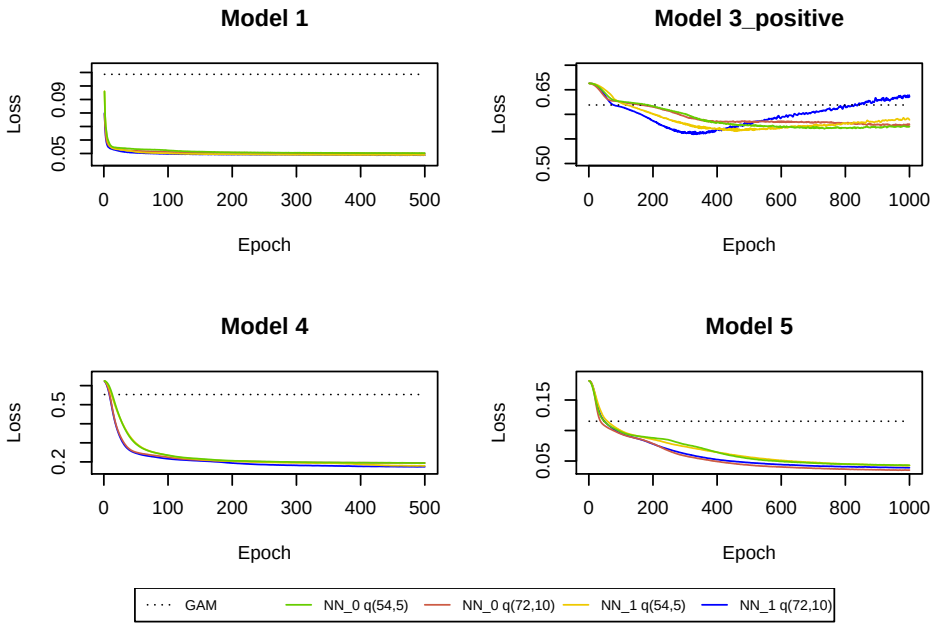

Figure A.7: Cross-entropy and deviance loss functions on validation sets observed during the training of the neural networks in $k = 8$.

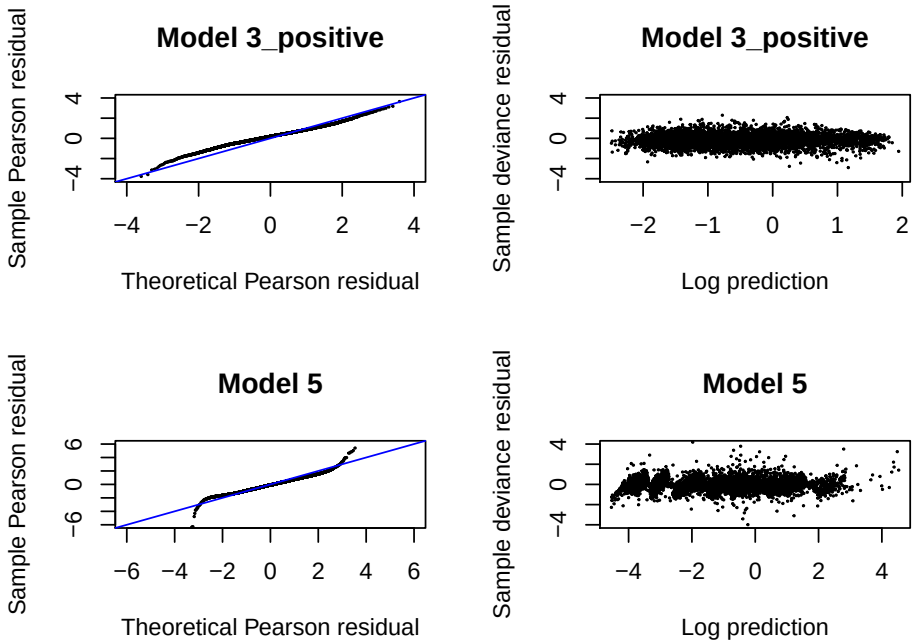

Figure A.8: QQ normal plots and Tukey-Anscombe plots in Models 3_positive and 5 fitted with neural networks NN_1 in $k = 8$

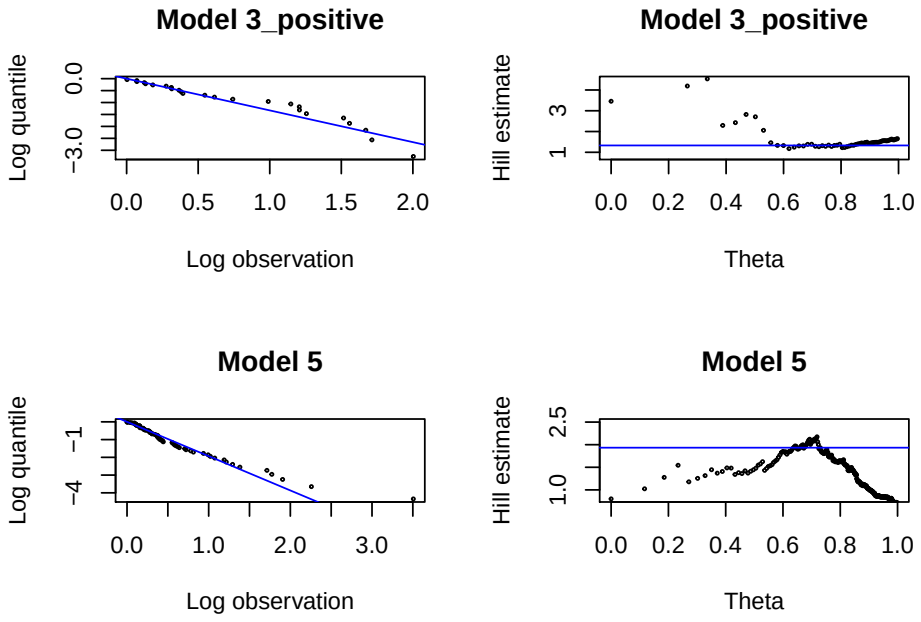

Figure A.9: Pareto quantile plots and altHill plots for incremental payments and changes in claim incurred in $k = 16$.

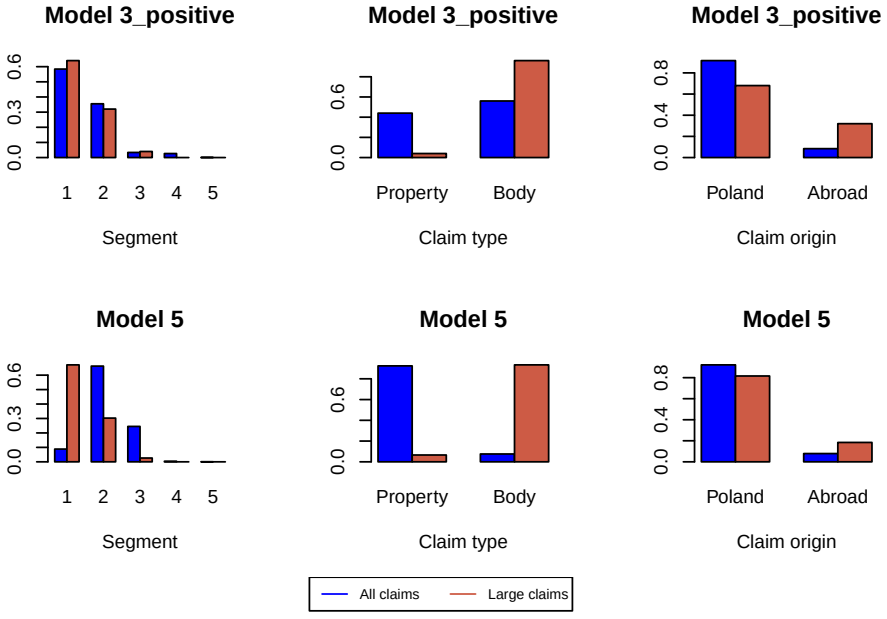

Figure A.10: Proportions of claim features in the whole data set and in the data set of large claims in $k = 16$.

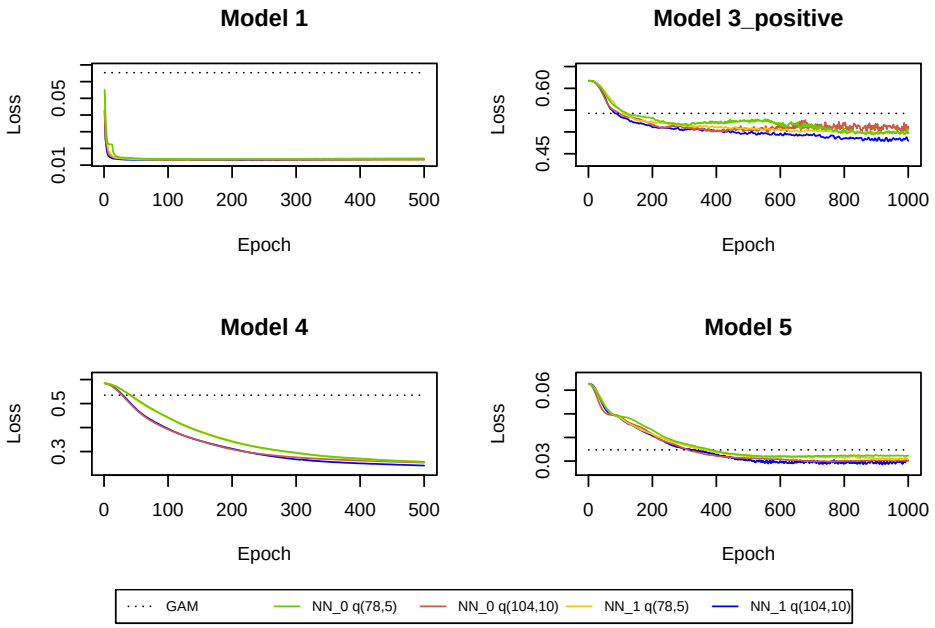

Figure A.11: Cross-entropy and deviance loss functions on validation sets observed during the training of the neural networks in $k = 16$.

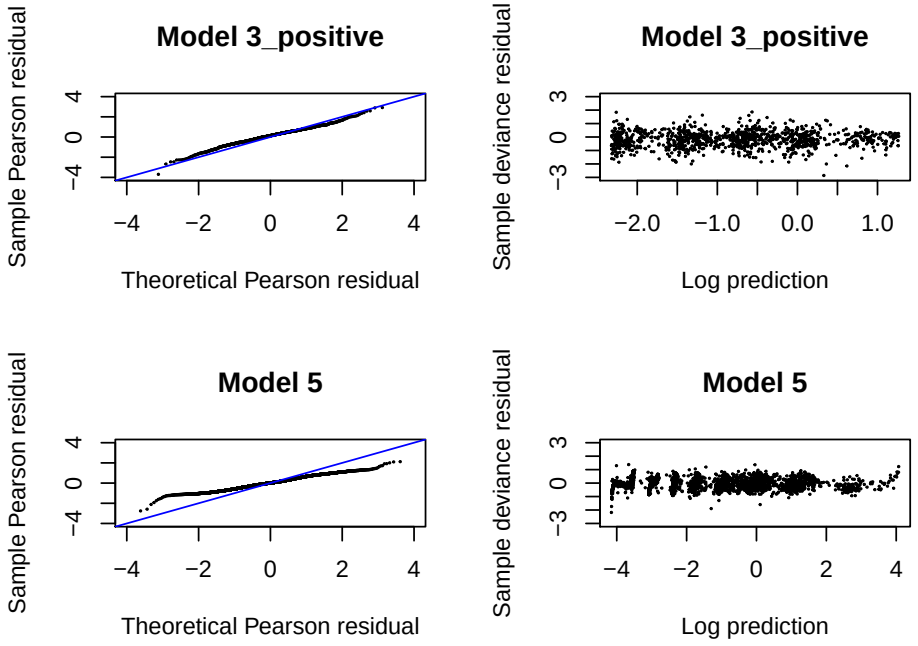

Figure A.12: QQ normal plots and Tukey-Anscombe plots in Models 3_positive and 5 fitted with neural networks NN_1 in $k = 16$

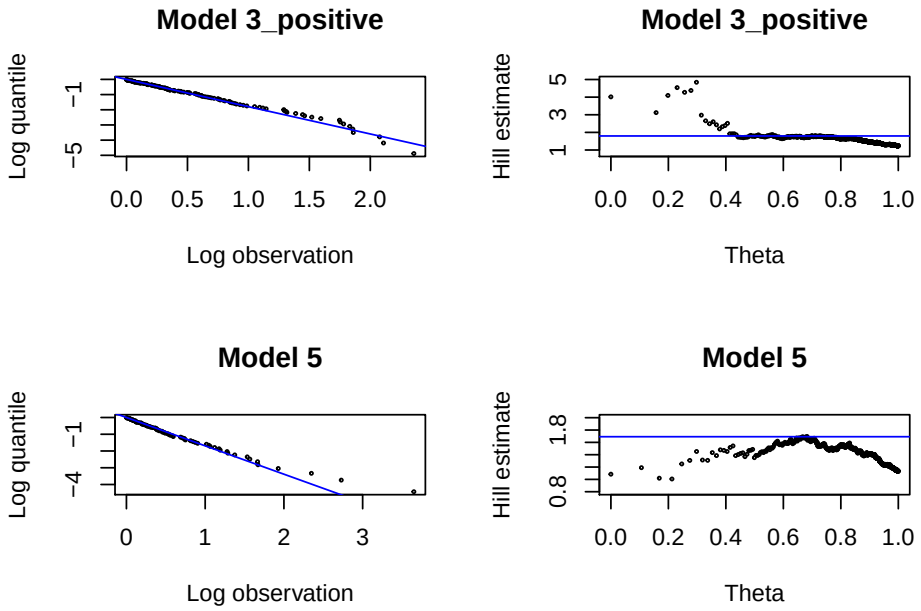

Figure A.13: Pareto quantile plots and altHill plots for incremental payments and changes in claim incurred in $k = 17$.

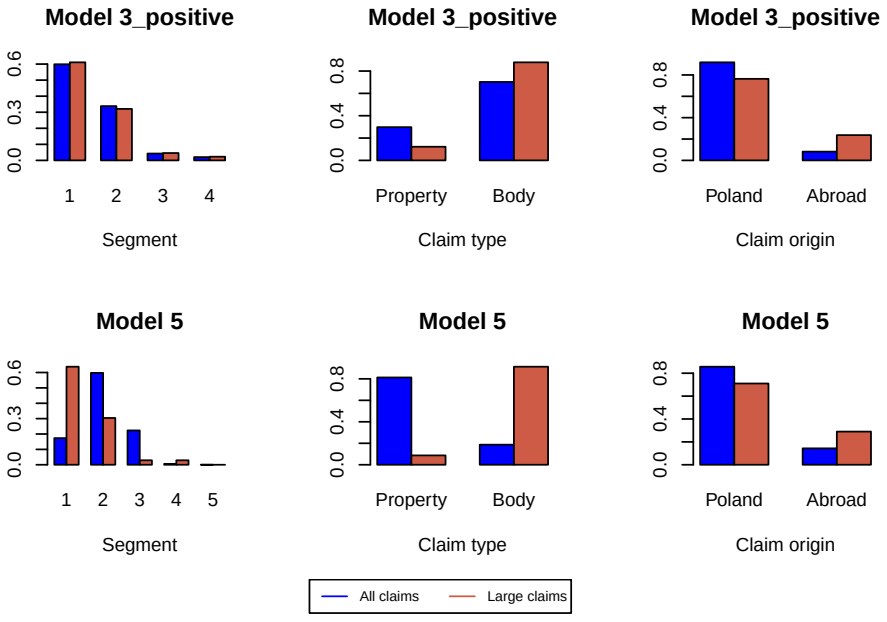

Figure A.14: Proportions of claim features in the whole data set and in the data set of large claims in $k = 17$.

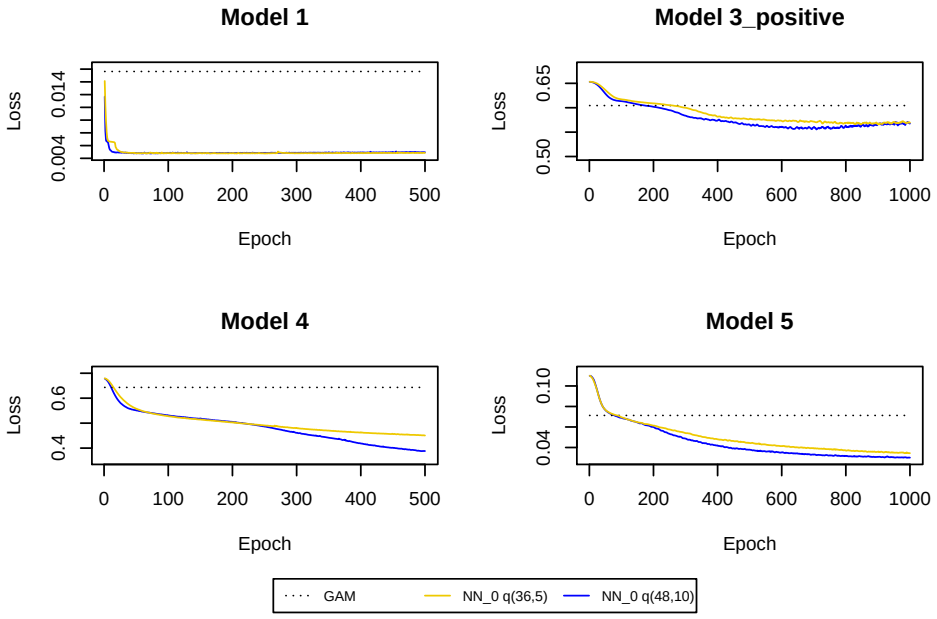

Figure A.15: Cross-entropy and deviance loss functions on validation sets observed during the training of the neural networks in $k = 17$.

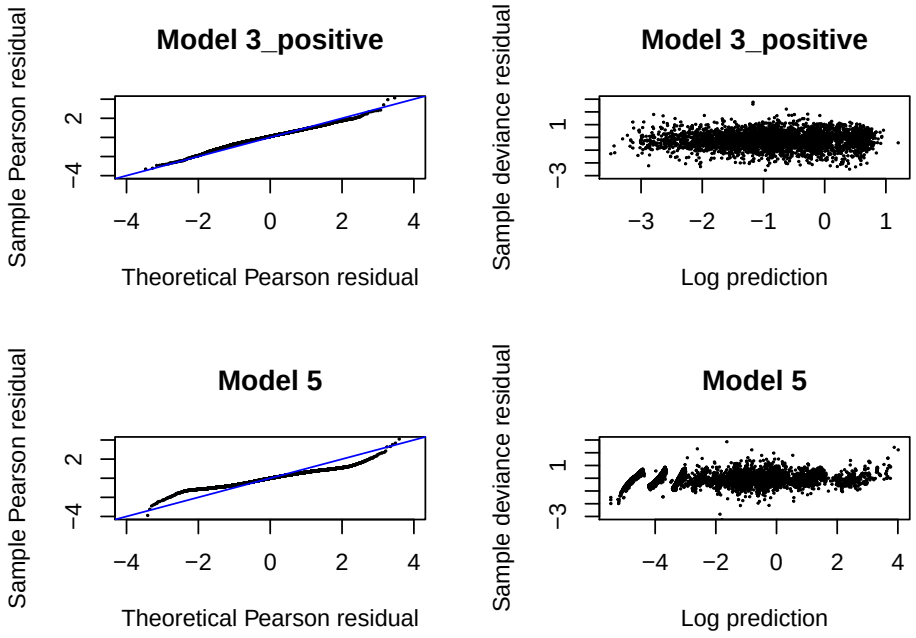

Figure A.16: QQ normal plots and Tukey-Anscombe plots in Models 3_positive and 5 fitted with neural networks NN_1 in $k = 17$

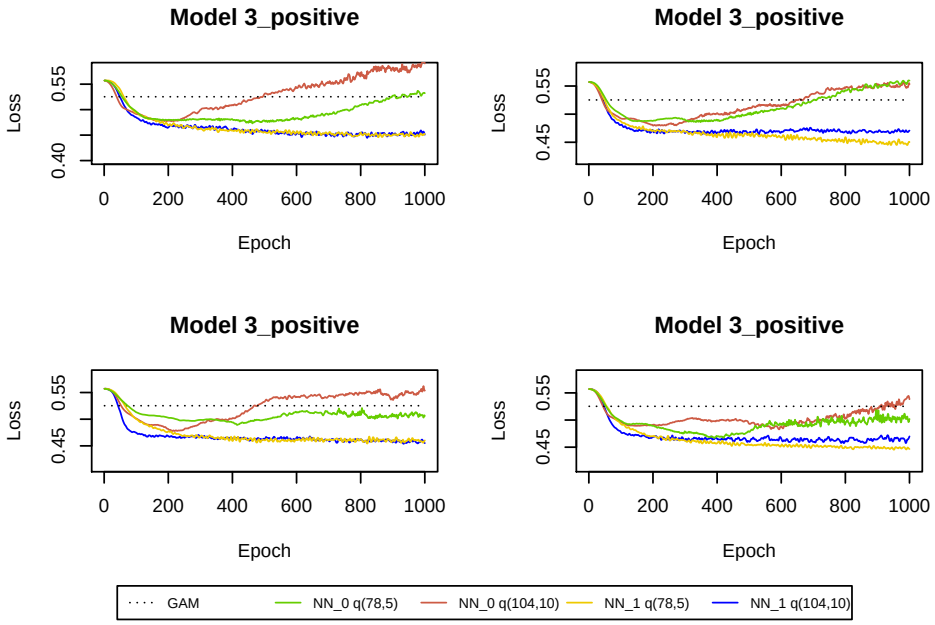

Figure A.17: Deviance loss functions on validation sets observed during the training of the neural networks in $k = 16$.

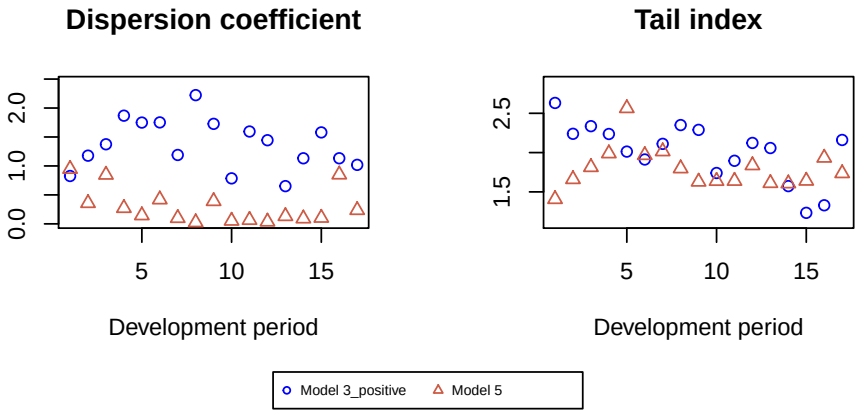

Figure A.18: Estimates of tail indices and constant dispersion coefficients in Models 3_positive and 5.

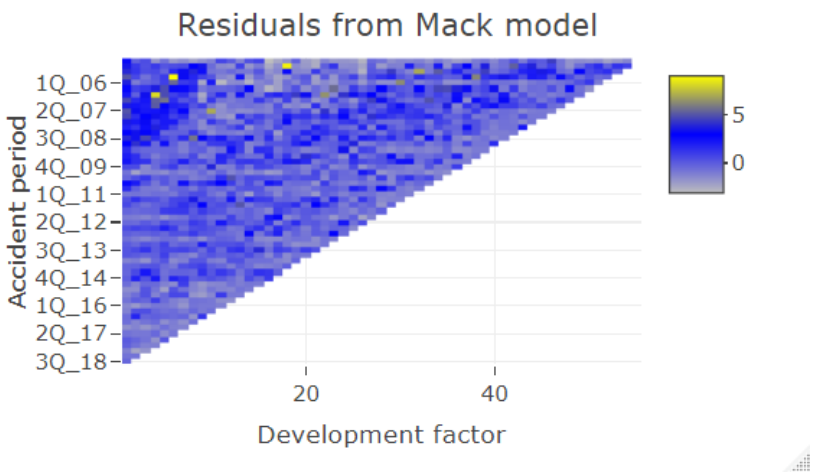

Figure A.19: Residuals from Mack model fitted to development factors in a run-off triangle.