Believing the Bot - Model Risk in the Era of Deep Learning

Ronald Richman* Nicolai von Rummell† Mario V. Wüthrich‡

Version of August 29, 2019

Abstract

Deep Learning models are currently being introduced into business processes to support decision-making in insurance companies. At the same time model risk is recognized as an increasingly relevant field within the management of operational risk that tries to mitigate the risk of poor business decisions because of flawed models or inappropriate model use. In this paper we try to determine how Deep Learning models are different from established actuarial models currently in use in insurance companies and how these differences might necessitate changes in the model risk management framework. We analyse operational risk in the development and implementation of Deep Learning models using examples from pricing and mortality forecasting to illustrate specific model risks and controls to mitigate those risks. We discuss changes in model governance and the role that model risk managers could play in providing assurance on the appropriate use of Deep Learning models.

Keywords. Deep learning, Model Risk, Pricing, Mortality Forecasting, Insurance Modelling

1 Introduction

Deep learning refers to a modern approach to designing and fitting neural networks that recently has achieved state of the art results on machine learning problems in computer vision, natural language processing, machine translation and speech recognition, and has become the main avenue for solving unstructured data problems [1, 2]. In addition to unstructured data, deep learning approaches have produced promising results on structured data problems [3], as well as time series forecasting [4]. Modern neural networks are generally characterized by specialized architectures that are adapted to domain-specific problems, as well as by the depth of the networks, meaning to say, that these networks are composed of multiple layers of non-linear functions. Recently, deep learning techniques have been applied to problems within actuarial science such as pricing, reserving, analysis of telematics data and mortality forecasting. For a recent review of these applications, see [5]. The benefits of applying deep learning to actuarial problems have typically been greater accuracy than traditional actuarial or statistical techniques [6, 7, 8], the provision of more granular information than produced by traditional techniques [9], or the extension of actuarial modelling to new types of data [10] (however, we note that the greater accuracy has been achieved at an individual policy level, whereas, the portfolio

*QED Actuaries and Consultants, ronald.richman@qedact.com
†QED Actuaries and Consultants, nicolai.von.rummell@qedact.com
‡RiskLab, Department of Mathematics, ETH Zurich, mario.wuethrich@math.ethz.ch

averages may not necessarily be more accurate, see [11], which we discuss in more detail later). Considering these successes, it is likely that more applications will emerge in the literature, and actuaries will begin to use these techniques in practice.

Concurrent with the introduction of deep learning techniques, financial services providers such as banks and insurers are increasingly subject to regulation that demands the management of all categories of risks, including the risks posed by models. For example, in the United States guidance has been issued for banks requiring the management of model risk [12]. Other regulators require that models undergo a series of tests before use, for example in the Solvency II regulation of internal risk models, that are used for estimating capital requirements. In the South African context, regulation under the Solvency Assessment and Management (SAM) [13] requires that insurers justify why risk models are an accurate reflection of an insurer's risk profile on an annual basis in the Own Risk and Solvency Assessment (ORSA).

In addition to regulatory requirements, model risk is worthy of attention by actuaries and risk managers due to the potential for these risks to cause harm if not mitigated. For example, [14] provides a review of several well known cases of failures of models that led to significant financial and reputational damage for financial services providers.

Therefore, within the context of these developments in financial services, new and relatively less familiar modelling techniques such as deep learning will be subjected to scrutiny before their use is approved within insurance companies and, moreover, the ongoing use of these models needs to be conducted within a model risk management framework that is suitable for deep learning models. An insurer's customers whose policies have been priced using deep learning techniques may, for example, notice the application of more differentiated pricing and query how these prices have been derived, potentially posing some reputational risks. Finally, the risks posed by these models will need to be considered by the key stakeholders within insurers - actuaries, risk managers, Boards of Directors and regulators.

In this paper we study the specific model risks posed by deep learning techniques, and provide an initial view of the potential impacts and mitigating actions that can be taken to manage these risks, to enable the controlled adoption of these techniques by actuaries, risk managers and other insurance professionals. Therefore, the rest of the paper is organized as follows. In Section 2, we define what is meant by a model and provide background on the use of models by actuaries. In Section 3 we provide background on deep learning and discuss how it is different from other modelling approaches. Following these sections, in Section 4 we discuss the modelling process and how the output of models is used in insurance companies. In Section 5 we define model risk and discuss the management of these risks within an actuarial context. Section 6 presents the main body of the paper, in which we discuss how deep learning models are different from traditional actuarial models and examine the specific model risk posed by deep learning within the actuarial context. Building on this, in Section 7 we provide an analysis of two models in light of the discussion in previous sections. Finally, Section 8 provides conclusions and suggestions for future research.

2 Background, Definitions and Notation

Models are central to many aspects of practical actuarial work [15], which often involves specifying and parametrizing mathematical, statistical or economic models of financial systems. In

this study we adopt the operational definition of a model in [16], who defines a model (pg 2) as "a set of verifiable mathematical relationships or logical procedures which is used to represent observed, measurable real-world phenomena, to communicate alternative hypotheses about the causes of the phenomena, and to predict future behaviour of the phenomena for the purpose of decision-making". This general definition includes both explanatory and predictive modelling as two distinct goals (for more on the distinctions between these tasks, and the implications for practice, we refer the reader to [17]). Focusing on actuarial modelling, different studies have provided competing typologies of actuarial models. For example, [18] reviews several typologies, and then provides a comprehensive new classification, that includes, amongst others, descriptive and predictive models. Descriptive models describe the historical relationships between variables, and predictive models (which we focus on in this study) are those models that attempt to predict values of variables that are currently unknown, generally on the basis of other, explanatory, or known, variables which serve as inputs to the model. Of course, it is common in actuarial practice that sufficient data are sometimes not available to parametrize a predictive model fully and, in these cases, actuaries may resort to expert knowledge to set the parameters. Many standard actuarial models focus on prediction (even though the model building process may include elements of descriptive modelling), for example, pricing models in non-life [1] insurance attempt to predict the expected value of claims frequencies, claim costs or pure premiums based on the characteristics of an insurance policy. Ideally, when pricing non-life policies, we would like to know whether a particular insurance policy will have good, average or bad claims experience, and, based on this knowledge, allocate a low, middle or high price. Unfortunately, these explanatory variables of the likely claims experience are not available, and, therefore, we use proxies, such as the age of the policyholder or the type of motor vehicle, as explanatory variables that are input into a predictive model. It should be emphasized that these proxies are not causal, but they only represent an "average" relationship, in other words, not every young car driver is a bad driver, but on average younger drivers are worse drivers.

This definition of a predictive model also includes time series models, where the known variables are values of the variable which we seek to model that have occurred in previous time periods. For example, common mortality forecasting models, such as the Lee-Carter model [19], as well as Incurred But Not Reported (IBNR) reserving methods such as the Chain-ladder method, fall within the definition.

In more formal terms, the output of a predictive model M is an unknown (vector) variable $\hat{y}$ (we use $\hat{y}$ to denote that the true value of the variable y is unknown). Note that ideally, we would use the expected value of y, $E[y]$, for prediction if this quantity was known, however, since this value is unknown, we rely on the output of the predictive model $\hat{y}$. $\hat{y}$ is calculated on the basis of X, a matrix of known variables, which may be transformed in some manner using a set of functions T to a new matrix of variables X' (each function within T operates on at least one variable within X). For example, to achieve better predictions, T might specify that numerical features be logged or squared, and combinations of features might be calculated, leading to a potentially extended set of features, X', which is then used as input to the machine learning algorithm (the features in X' might also be reduced by T, if, for example, an algorithm such as Principal Components Analysis (PCA) is applied to X). X' is referred to as the feature matrix

[1] In this paper we refer to non-life insurance, which is also known as general insurance in the United Kingdom, property & casualty insurance in the United States and short-term insurance in South Africa.

within the machine learning literature and the design matrix within the statistical community. In the following, when discussing predictions, known variables and features relating to the ith policy, we extend the notation above by adding a subscript, and similarly, when discussing the jth known variable or feature, we again another subscript, thus, a prediction that is made for the ith policy using the jth feature will be represented as $\hat{y}_i = x'_{i,j}$.

In addition to X and T (which is generally defined explicitly, but can also be defined implicitly, as will be discussed in the section on deep learning), a model M consists of (at least) two more parts, a model specification S and a set of parameters θ. Considering the model specification S, we define A as the particular class of algorithms chosen by the modeller, for example linear models or decision trees and, where the class of algorithms requires the model to be defined explicitly by the modeller, we also include the explicit model definition within S as E. The parameter set θ of the model is a consequence of the model specification S. Thus, we define a predictive model as

$$M(X, T, S(A, E), \theta) = \hat{y}. \tag{2.1}$$

θ is usually obtained by optimizing an objective function, or loss function, L (as a complement to algorithm A), and different objective functions (as well as optimisation algorithms) may lead to different "optimal" parameters θ, and henceforth to different predictive models for $\hat{y}$. Here, we represent a generic loss function measuring the difference, in some sense, between the actual observations y and the predictions $\hat{y}$ as $L(y, \hat{y})$. A common example of an objective, or loss function, is the mean square error (MSE) criterion, but other functions, such as the mean absolute error (MAE) may also be specified. As an alternative, the values of θ might be determined using statistical techniques, such as maximum likelihood estimation (MLE).

Models used by actuaries are generally specified in one of two ways - some models, such as those used for pricing non-life business, are specified after investigation of empirical relationships found in datasets whereas other models are specified without reference to a particular dataset, but rather based on expert professional knowledge of particular actuarial tasks, for example, the Chain-ladder model is often used to derive IBNR reserves for accident periods that have experienced significant development.

2.1 The Traditional Approach

Predictive models built for actuarial tasks often rely on an explicit linear or, less often, an additive model specification (as examples, see [20] for the case of non-life pricing models, and [21] for mortality forecasting models). More formally, in most actuarial models the class of algorithms A is usually restricted to linear models. Thus, considering policy i, the model output y_i is related to the (vector) of known variables X_i using an explicit model definition E such that $\hat{y}_i = \sum_{j=1}^{n} \beta_j f_j(x_{i,j})$, where n is the number of columns of the feature matrix X, $x_{i,j}$ is the jth known variable (i.e. column j of X), β_j is the regression coefficient relating to variable $x_{i,j}$, and $f_j(\cdot)$ is a function of variable x_j. Within actuarial modelling, part of the task of model specification relates to selecting which known variables should be used within the matrix X', and the form in which these variables should enter the matrix. To specify a generalized linear models (GLMs), $f_j(\cdot)$ is chosen as the identity function (in other words, GLMs assume a linear relationship between the known variables and the unknown variable) and for specifying a generalized additive model (GAM), which assume that $\hat{y}$ is related to X_i by a smoothly varying

function, $f_j(\cdot)$ is chosen as a spline function (see [22] for more details on GAMs). Choices of the functions T might involve discretization of continuous variables, or, of particular importance within pricing models, multiplying known variables together to create interaction effects. We emphasize here that T must be manually specified in actuarial models (in contrast to deep learning models, which we examine later). To fit these models, part of the model specification S usually includes an error distribution for the predicted variable y, for example, frequency models often assume that y is distributed as a Poisson random variable. More specifically, in these cases a stochastic data generating process is also specified, which defines the distribution of a random variable of interest, Y. Ideally, we would use the expected values, $y = E[Y]$, as the prediction, but since the expected values are unknown, $E[Y]$ is approximated with estimates $\hat{y}$. In this case, once the model is specified, standard statistical techniques such as maximum likelihood estimation are used to find the parameters θ, which are, in the cases just described, the regression parameters β_j, $j \in \{1...n\}$ (or alternatively, a suitable loss function can be specified to achieve the same result). Thus, the form of a (linear or additive) predictive actuarial model can be expressed as:

$$M(X, T, S(A, \sum_{j=1}^{n} \beta_j f_j(x_j)), \theta) = \hat{y}. \tag{2.2}$$

To judge the quality of a model, actuaries will often consider residual plots and standard statistical hypothesis testing. For an example, we refer to [20] who describes the approach of building GLMs, using statistical tools used for inferential statistics (in other words, for the goal of "explaining" in the terminology of [17]), such as confidence intervals (see also [23] who take a similar approach). However, using the predictive performance of models as a criterion for model choice appears not to have been strongly emphasized in most of the literature on actuarial modelling, despite the intended use of most actuarial models for prediction (see [24] who discusses this in the context of statistical modelling). Put another way, predictive performance of models is not necessarily used as a key criteria to guide model specification.

On the other hand, a different criteria that has been strongly emphasized within actuarial modelling is the stability of the predictions over time, and especially, the stability of the models when new information becomes available. A desirable property of models used within insurance is that predicted values behave relatively smoothly over time, in other words, do not react too sensitively to new observations, unless the new information in these observations is a contradiction to the old predictions. For example, within the context of pricing, policyholders expect to pay a relatively constant price over time, unless a serious event or change in risk factors happens. We note that most machine learning approaches do not focus on this property of stability, although some of these techniques can be adapted to produce stable estimates, see for example [25] in the context of regression trees.

We refer to this approach as the *traditional* actuarial paradigm in the rest of this paper (but note that other studies use different terminology, for example [24], in a discussion of various approaches to statistical modelling, refers to this paradigm as the "data modelling culture").

2.2 The Machine Learning Approach

Notably, another predictive modelling paradigm, which we call the *machine learning* paradigm, has developed in recent decades alongside the traditional approach just described, which differs

in its approach in a number of key aspects. For a thorough explication, we refer the reader to [24], who describes "two cultures" of modelling, the first being similar to the traditional approach just described (the "data modelling culture"), and another approach termed the "algorithmic modelling culture". Instead of working with models M with an explicit model specification E, including a stochastic generating process for the data, the machine learning paradigm shifts the focus to building algorithms that operate on the features X' to produce the model output $\hat{y}$. Thus, within the machine learning paradigm, there is not an explicit model definition E, but this is rather performed implicitly, by the class of algorithms, A, selected by the modeller. In other words, the explicit link between X and y (including the modeller's prior knowledge of the link) is not specified by the modeller, whereas the class of algorithms is. Expressing this in the formal manner above, in the machine learning approach

$$M(X, T, S(A, \tilde{E}), \theta) = \hat{y}, \tag{2.3}$$

where $\tilde{E}$ denotes the implicit specification of the model.
Many algorithms have been proposed in the machine learning literature, but two of the most successful (judged by predictive accuracy on unseen data) are gradient boosted trees [26] and neural networks.
Machine learning generally follows a three-stage process: firstly, in a step known as "feature engineering", the data to which the algorithm is applied is manually transformed into a more useful representation (by applying a function $T(\cdot)$ to the known variables X to derive the feature matrix X'). Note that this step is not only applicable to the machine learning approach, but also applies to some tasks performed by actuaries in the traditional approach, as described above. The feature engineering step relies on the expert domain knowledge of the user of the algorithm, which allows for the specification of features that are useful for the problem at hand; [27] describe this process as "a way to take advantage of human ingenuity and prior knowledge". Secondly, in the "feature selection" step, the most useful features are selected, and lastly, in the "training" step, the machine learning technique is calibrated to the transformed data. We also note that some machine learning techniques combine the second and third steps, by performing feature selection as well as training in a single step, such as the LASSO (least absolute shrinkage and selection operator) technique of [28], which is actually a statistical technique.

2.3 Differences between the Traditional and Machine Learning Approaches

As opposed to the assumptions of linearity or additivity in the traditional actuarial approach, these algorithms assume that the known variables X are related to the output y in complex, non-linear ways that may involve the interaction of multiple variables. To capture this relationship, machine learning models are often more complex and less transparent than traditional models. Breiman characterizes this as the "Occam dilemma" (pg 208): "Accuracy generally requires more complex prediction methods. Simple and interpretable functions do not make the most accurate predictors". Breiman's Occam dilemma is of particular relevance to actuarial modelling, which has long favoured relatively simple predictive models, which is a bias which may constrain the accuracy of predictions made by actuaries. Despite this, models based on the alternative paradigm of machine learning have recently been introduced to the actuarial literature, with applications in pricing [29, 30], reserving [31] and modelling and forecasting of mortality [32, 33].

Another notable difference of machine learning algorithms from traditional techniques is referred to by Breiman as the (pg 206) "multiplicity of good models". This refers to the observation that, when applying machine learning algorithms, one can often find many different models with similar error rates as measured by the loss function, but very different specifications and parameters. These differences may result in considerable real-world consequences, for example charging a different premium to a particular policyholder. Since traditional actuarial modelling typically involves a convex optimisation problem, there is usually a unique "best" model because the convex optimisation problem has a unique minimum. In contrast, machine learning techniques are often not convex optimisation problems and, as a consequence, many "sufficiently good" models might be found, and it is not clear, on the basis of the loss function, which of these models should be chosen. Furthermore, if regularization techniques such as early stopping are used when fitting machine learning models, then these models will depend on the seed of the random number generator used when fitting the model, which is not a desirable property of models (this point is discussed in more detail in the following section). In summary, moving from traditional actuarial modelling to machine learning means that there is no longer a unique "best" model, and many of these differences appear to relate to the under-determination of models due to an insufficiently specified loss function. We return to this point in Section 4.

3 Neural Networks and Deep Learning

3.1 Background

Neural networks are a relatively old technique, compared to other machine learning approaches such as support vector machines [34] and boosting, with some of the earliest research performed during the 1950s by [35]. Despite this early start, only recently a modern approach to designing, fitting and using neural networks has revitalized the field, leading to state of the art results in several areas of machine learning. This recent approach, reviewed in [2] and [36], is referred to in the machine learning literature as Deep Learning, and is characterized by the principle of representation learning, which we summarize briefly.

The deep learning literature criticizes the practice of feature engineering on several grounds, which we label as the complexity, effort and expert knowledge arguments. [1] and [37] note that, even within the context of a single machine learning problem, it is often difficult to know which features are relevant to extract from data due to the complexity of the problem, and that the problem is exacerbated when dealing with many categories of problems at once, such as classifying images into one of several categories, or when dealing with high dimensional data. [27] and [1] note that the feature engineering process is labour intensive, thus limiting the scope and applicability of machine learning, and, lastly, [2] note that the feature engineering approach depends on having suitable prior knowledge about the problem, the attainment of which sometimes requires the multi-decade focus of communities of researchers.

Before addressing the response to these criticisms, we briefly consider their importance in the context of actuarial modelling. Usually relatively limited data is collected for insurance policies and, furthermore, privacy regulations may also limit data collection. Also, actuaries have had at least 30 years of experience in modelling insurance related problems using statistical techniques, and have built the prior knowledge to work with most types of data. Moreover, since insurance is a competitive market, where it is necessary to watch what competitors are doing to avoid adverse

selection, similar institutional knowledge has been built. Thus, the complexity, effort and prior knowledge arguments might be considered somewhat less important for actuarial modelling. On the other hand, we consider that for many actuarial problems addressed at a large-scale, the appropriate techniques appear not yet to have been developed, for example, techniques leveraging multiple lines of business when reserving, or forecasting simultaneously the mortality of multiple populations (and indeed, applying deep learning to these problems has resulted in better performance than traditional techniques, see, for example, [6, 7]). Also, new sources of data that are more complex than traditional types of data, such as data from telematics and wearable devices, are now becoming available to actuaries, and the relevant techniques to deal with these types of data are still being developed. Finally, we consider that recent studies have demonstrated the relative ease with which traditional techniques can be outperformed using machine learning, see for example [29, 38] in the context of pricing. Thus, we believe that the arguments discussed here are applicable to problems of actuarial modelling.

The response to these criticisms of the feature engineering approach is the paradigm of representation learning, which is concerned with the study of algorithms which are designed to discover automatically the optimal representations needed for a machine learning task [2]. In other words, representation learning algorithms automate the specification of the function $T(\cdot)$ applied to the known variables X, to derive the feature matrix X'. An example of such an algorithm, that is familiar to actuaries, is PCA, which extracts features from input data by constructing a new, orthogonal representation of the input data that summarizes the greatest variance in the dataset (for techniques that are less familiar to actuaries, we refer the reader to the review in [27]). These features can then be used within a regression model, for example, in the simplest form, a linear regression model, leading to the Principal Components Regression (PCR) technique. However, applying relatively simple techniques such as PCA to complicated high-dimensional data, such as images, text and audio, often produces sub-optimal results; see for example [40] who tries to summarize the MNIST dataset [41] (which consists of images of hand-written digits) using PCA with limited success, meaning to say that the resulting features do not capture the differences between the digits well. This failure is understood by [1] to result from simple techniques failing to isolate the relevant "factors of variation" that explain complex data, such as images, text and speech. Of course, since PCA is a linear technique (by nature), it is not suitable to solve complex non-linear problems, and this result of [40] is not entirely surprising.

Deep learning is a representation learning technique that seeks to extend the paradigm of representation learning to these types of complex data, by structuring machine learning algorithms to learn hierarchal representations of data that combine multiple levels of simpler representations [1]. The goal of applying this technique is that the learned features at the top of the hierarchy are able to capture abstract features of the data. Thus, modern neural networks are often deep, meaning to say, that these neural networks are constructed as the application of non-linear functions multiple times to the data input into the neural network, allowing for the expression of complicated non-linear relationships between the input data and the output of the network. Returning to the example of the MNIST data, after applying a deep network, [40] found that the learned features captured the differences between images of different digits well. The successful implementation of representation learning for many types of machine learning problems is what distinguishes deep learning from other modern machine learning techniques and approaches.

To express the concept of representation learning more formally, in the deep learning approach

$$M(X, \tilde{T}, S(A, \tilde{E}), \theta) = \hat{y}, \tag{3.1}$$

where $\tilde{T}$ denotes the implicit specification of the set of functions T which transform the known variables X to the feature matrix X'.
The basic form of a two-layer deep neural network is given by the following equations:

$$\mathbf{Z}^1 = \sigma_0(\mathbf{c}_0 + B_0' \mathbf{X}), \tag{3.2}$$

$$\mathbf{Z}^2 = \sigma_1(\mathbf{c}_1 + B_1' \mathbf{Z}^1), \tag{3.3}$$

$$y = \sigma_2(c_2 + B_2' \mathbf{Z}^2), \tag{3.4}$$

where $\mathbf{Z}^1$ and $\mathbf{Z}^2$ represent the two hidden layers of the network, X is the feature matrix, B_0, B_1 and B_2 are weight matrices, $\mathbf{c}_0$, $\mathbf{c}_1$ and c_2 are intercepts, and where σ_0, σ_1 and σ_2 are (the typically) non-linear activation functions of the neural network. Common choices for the activation functions are the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ or the rectified linear unit (abbreviated to ReLu) $\sigma(z) = max(0, z)$, where $z \in \mathbb{R}$.
The network structure given in (3.2) can be understood in the following manner. In the first layer, (3.2), an affine transformation of the input variables X is performed, followed by a non-linear transformation using the function σ_0, producing a set of intermediate variables, $\mathbf{Z}^1$ (in other words, the network has performed an initial step of representation learning). The variables $\mathbf{Z}^1$ could be used immediately in the output layer of the network, and this algorithm would then be a so-called shallow neural network. Adding another layer, (3.3), creates a so-called deep network, which allows the network to perform another step of representation learning, producing another set of intermediate variables $\mathbf{Z}^2$, which are then used as input to the last layer of the network, (3.4), producing the output $\hat{y}$. Modern neural network models are usually deep, containing up to thousands of layers in some computer vision applications. For tabular data, models up to five layers deep have appeared in the literature, but are often only two or three layers deep. In terms of width, common choices for smaller, less complex data are 32 or 64 neurons, and fore more complicated data, wider networks are used.
We now briefly consider how to fit the parameters θ of the model: firstly, each of the parameters in the network (the intercepts and weight matrices) are randomly initialized. Secondly, a loss function $L(y, \hat{y})$ is specified, measuring the distance between the output of the network, $\hat{y}$, and the observed outcomes y in the training set. The goal of the fitting process is to minimize this loss function, and, to do this, gradient descent is applied: the derivatives of the loss function with respect to the parameters of the network are estimated (using back-propagation) and then, lastly, the parameters of the network are updated using these derivatives, to produce a somewhat smaller loss. The second and last steps are repeated until the network produces accurate predictions on the training set. The predictive accuracy of the network is then assessed on unseen data in the test set. For a more detailed explanation of the techniques used for training a neural network - back-propagation [42] and gradient descent - we refer to [1].
The network specification in (3.2)-(3.4) only consists of two layers; deeper networks can be specified by adding more intermediate layers, however, doing so makes the network harder to fit. Having described the basic structure of a deep neural network, we note that most of the state of the art results achieved by deep learning rely on specialized layers being incorporated into

the network. For example, convolutional layers [43] are used for computer vision applications and recurrent layers, such as Long Short Term Memory (LSTM) networks, are used for natural language processing (NLP). These layers incorporate prior knowledge about the structure of the data being modelled and lead to major performance gains. For a comprehensive overview of the types of specialized layers, we refer the reader to [1], and, within an actuarial context, to [5].
In this study we focus on embedding layers [44], which have been shown in the literature to achieve excellent predictive performance when modelling categorical data (or numerical data that can be expressed as categorical data) (see [3] for a general example, and [5, 7] for actuarial examples). Consider a vector $\mathbf{C}$ containing measurements relating to one of N classes, in other words, $C_i \in \{1, ..., N\}$. The classical approach to representing this data is using so-called dummy encoding, or one-hot encoding in the machine learning literature, in other words, each class $n \in N$ is mapped to a vector $K_n \in \mathbb{R}^N$ containing zeroes in each element of the vector, except for the nth element, which contains 1 (the unit vectors in Euclidean space); dummy coding reduces this representation by one dimension. As an alternative to this approach, an embedding layer of dimension k is a set of mappings for each class $n \in N$ to a numerical vector $K_n \in \mathbb{R}^k$, where the k values of K_n are free parameters that are fit together with the rest of the neural network, and where the dimension k is typically (much) smaller than N. Thus, instead of representing categorical data using the sparse vectors produced by dummy coding and one-hot encoding, respectively, embedding layers allow for representation learning to take place directly for categorical variables, by encoding these variables as dense numerical vectors.

3.2 Applying Deep Learning to Actuarial Modelling

We now briefly consider some potential pitfalls when applying deep learning techniques to actuarial modelling.
The current state of the art in fitting deep neural networks relies on randomly initializing the parameters of these networks, and then, as mentioned above, applying the technique of gradient descent to minimize the loss function. Gradient descent is not, however, applied until the parameters of the network converge, but rather the performance of the network on an alternative set of data is tracked, and once performance on this data stops improving, then the process of gradient descent is stopped. This is referred to in the literature as "early stopping", and, because of the random initialization of parameters, the parameter set found using early stopping may vary each time the network is fit. Other sources of randomness when training neural networks are the typically random selection of the training and validation sets, the addition of dropout [45], where neurons of the network are randomly set to zero to regularize the network, and the random selection of batches of data from the training set on which to fit the network (stochastic gradient descent). Due to these issues, the solutions found when fitting deep neural networks vary each time the network is fit, and depend, to some extent, on the seed chosen for the random number generator. For an illustration of the problem in the context of mortality forecasting, we refer the reader to [7]. Seemingly, a helpful solution to this problem is to average the predictions of several neural networks, providing more stability and usually resulting in better predictive performance. While model averaging works well in practice, it seems that more theoretical understanding is required, in particular, whether model averaging leads to enough stability that results from neural networks can be successfully communicated to senior management, Boards of Directors and regulators of insurance companies.

A different issue, discussed in [11], is that while deep neural networks fit and predict individual data points well, nonetheless, these models may fail to reproduce portfolio averages (this is also a problem with other popular machine learning models, for example gradient boosted trees, see [46]). For example, in pricing, neural networks may produce better individual prices, but fail to reproduce the overall price level of a portfolio (for another example in the context of life insurance experience analysis, see [47]). This problem does not occur with traditional GLM models, which reproduce the same portfolio averages as homogenous models (in other words, models that do not differentiate between policies) in an unbiased manner. An intuitive understanding of the problem is that if a model is optimised to predict individual data points well, nonetheless, these predictions may contain a small bias, which in aggregate lead to portfolio averages not being reproduced. While seemingly not a problem in other domains, such as computer vision, this is a serious problem for actuarial modelling. Recently, [11] has proposed two solutions, firstly, using the features from a deep neural network within a GLM, and secondly, adding a term to the network's loss function which penalizes the bias of the model in aggregate.
Finally, as mentioned above, in common with other machine learning techniques, deep neural networks fit using gradient descent are not guaranteed to provide stable predictions over time, and this is an open problem that requires further research.
Having described three potential approaches to modelling in this and the previous section, we now focus on how models are used within insurance companies.

4 Modelling in Insurance Companies

The previous sections have discussed the concept of a model as an abstract simplification of reality, that is built to provide relevant information for decision-making. As a prelude to the next sections discussing model risk, we now discuss the way in which modelling is conducted within insurance companies; from the design, calibration and validation of models, to the actual implementation and operation. These processes must be performed before model output can be used for business decisions, which is the purpose of modelling in insurance companies (for more detail, see [48]). We consider five main aspects of the modelling process leading up to the use of the model in making decisions (note that these simplified steps are not necessarily performed in the order given below):

1. Design of models is choosing the various parts of the model defined above, in other words, specifying the known variables X, the prediction task $\hat{y}$, the set of functions T which produce the feature matrix (for traditional and machine learning models, whereas for deep learning models T is specified implicitly as part of the model calibration) and the model specification S. We only consider the model M to be fully specified once these items have been determined.

2. Calibration is the determination of the parameters θ, and the method of calibration is generally determined by the class of algorithms, A, that have been chosen as part of the model specification, as discussed above. Whereas in traditional actuarial models, the calibration might occur on all of the data available to the modeller, for deep learning models calibration is performed on the training set (and later validated on the test set). Furthermore, while actuarial models may not have an explicit loss function, or might rely

on the default loss function used to fit GLMs (the deviance function), the choice of loss functions is much broader with deep learning models. For example, instead of the common Mean Square Error (MSE) loss function, which is quite sensitive to outliers, one might also specify that the Mean Absolute Error (MAE) loss function be used. As mentioned before, another key difference is that actuarial models generally lead to convex optimisation problems, whereas in many machine learning models the optimisation problem typically is non-convex and there are no unique optimal models.

3. Validation is the step of the modelling process that ensures the outputs of the model are suitable for use in decision-making, by assessing the model's accuracy and robustness by performing tests and analysis during the model's design and calibration [48]. Key differences in the modelling process for deep learning models are that:

    - extra attention needs to be given to deep learning models to ensure results are consistent over time and across different training runs;
    - the accuracy of these models is generally assessed by recalculating the loss function on unseen data in the test set, whereas validation of models built using the traditional approach does not strongly emphasize this step (although in classical statistics one uses, for instance, the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) which try to correct decision-making for the model complexity involved).

4. Implementation is the technical configuration and set up of a model in a suitable platform so that the model can be run to produce output.

5. Operation is the production of model predictions $\hat{y}$ using the model implementation.

These steps lead to a model output which is an input into the business decision-making process.

4.1 Choice of Loss Function

Many of the loss functions in common use for calibrating machine learning models are relatively simple, and do not fully encapsulate the real-world objectives of modelling. Hence, these simple loss functions will not uniquely identify a single "best" model, as discussed above, leading to a set of models that all might be sufficiently well-calibrated (as measured by the loss function) for use in insurance decision-making. Moreover, often the choice of a specific loss function is difficult to connect to the application or decisions that are made based on the model results. For example, in the context of life insurance experience analysis, [47] find that models optimised using the Poisson deviance loss function do not necessarily produce unbiased estimates of the mortality and morbidity rates, in other words, the accuracy of forecasts as measured by the actual versus expected metric (which is used by actuaries as a measure of the accuracy of a set of rates), was not competitive. This particular problem is addressed in detail in [11], however, the more general correspondence between loss functions and the goals of insurance modelling has not been addressed in detail. An unsuitable choice of loss function, for a particular problem, would result in a sub-optimal set of models that are calibrated. We address the risk management consequences of this in Section 6.

4.2 Use of Model Output

The manner in which model output is used will depend on the purpose for which the modelling has been designed. The output of a non-life pricing model built to predict claims frequency (an example is discussed later in Section 7.1) is a prediction of claims frequency for a particular policy, which will usually be combined with another estimate of claims severity to derive an estimate of the predicted costs of insurance losses for each policy. Before using these loss costs directly, other items of commercial interest, such as expense and capital loadings, and profit margins, will be added to derive the premium charged to policyholders (for an extended discussion of these considerations, see [49]). This theoretically correct premium is often called the "technical premium", which may be different from the final premium which is charged to the policyholder, which is often called the "commercial premium". Differences between the technical and commercial premiums might result from a number factors. In general, the technical premium will consider all relevant information, regardless of any privacy and legal constraints and thus, the technical premium reflects the most actuarially correct price that can be determined. The commercial premium might be a less accurate price which is only based on information and risk drivers that are allowed to be considered. Furthermore, commercial considerations might be used to modify the technical premium, say to ensure that the prices charged to new policyholders are in line with those charged to existing policyholders and also the pricing offered by competing insurers should be considered. Tracking the technical premium allows an insurance company to assess the quality of its commercial premiums and its insurance portfolio.

The outputs of a mortality forecasting model (an example is considered later in Section 7.2) will also not be used directly, but rather as an input into an assumption setting process. For example, the mortality improvement rates implied by the model for a specific population might be applied to mortality rates derived for a subset of that population that holds annuities with a particular company. Whereas the pricing model is likely to be used on an ongoing basis directly within a system that produces prices provided to potential policyholders, the mortality forecasting model will usually be used at set intervals when assumptions are updated, without the model outputs being communicated directly to policyholders.

In both of these cases, though, the model output is used as an input into a decision-making process that includes considerations other than the predictive performance of the models, which has been quantified in the steps discussed above. These other considerations could be conceptualized as extra constraints on the loss function used to calibrate the models. For consideration of this point in a general context, see [50] who discusses how qualitative constraints, such as legality or ethics, are difficult to express using a loss function, thus giving rise to concerns of model interpretability, which allows users to assess (usually in an heuristic manner) how well models might meet these extra constraints.

In summary, in investigating the risks posed by deep learning models, both the characteristics of these models and how they are likely to be used in a decision-making process must be considered.

5 Introduction to Model Risk Management

We consider model risk management within the broader context of enterprise risk management (ERM), which is a framework for managing all risks of an organisation holistically to achieve business objectives, reduce earning volatility and maximize the value of the enterprise [51]. We

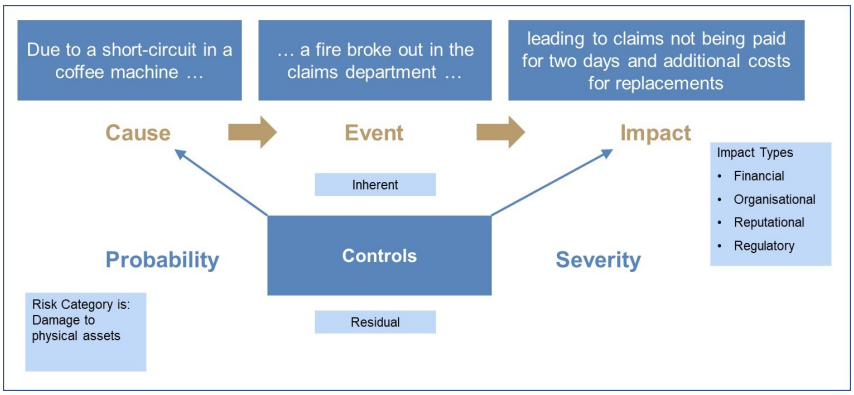

Figure 1: To illustrate the key concepts of ERM, the diagram shows a graphical depiction of an example of the risk classification used in the paper, and the application of controls to this risk.

provide a brief overview of some terminology used within ERM, before focussing on model risk. In this paper we define risks as potential events that may result in a loss for an organisation, with an impact that could be either financial, organisational, reputational or regulatory. The event will have a clearly defined cause that is stated in the description of the risk. Various classifications of risks have been proposed, for example, see [52] who discuss the difference between the event-based method of classifying risks, compared to the cause-based method. Here, we define a risk category as all risks that share one of an impact type (for example, "reputational risk"), an event or event type (for example, "market risk" or "damage to physical assets") or a common cause of the event (for example, "cyber risk"). Particular risks within risk categories are managed by implementing controls, which are put in place to prevent (reduce the frequency with which a risk occurs), detect and correct (reduce the severity of a risk once it has occurred) a risk, see [53] for more details (controls that combine these aims can also be implemented). Before controls are applied, the risk is referred to as "inherent"; after controls, the remaining risks are referred to as "residual". We illustrate our view of ERM in Figure 1.

5.1 Definition of Model Risk

Model Risk is a particular risk category, defined as those events where a business decision is made based on model errors or the inappropriate use of a model [12], resulting in a loss to an organisation. Based on EU regulation, model risk can be defined as "the potential loss an institution may incur, as a consequence of decisions that could be principally based on the output of internal models, due to errors in the development, implementation or use of such models" [54]. As discussed in the introduction to this section, the definition follows a cause-event-impact structure. The event is a (poor) decision that could be based on the outputs of a model $\hat{y}$, errors in the development or design of the model M, the (practical) implementation including the operation of the model or the use of the model, with a potential loss as a consequence of the risk event. For the purpose of this paper, the impact type is not further specified, but could be any impact type from financial, organisational, reputational or regulatory (generally, to quantify the impact of a model risk occurring, it would be necessary to understand and measure all the consequences of the decision that was made). Therefore, model risk is the class of all risks that share the event of making a business decision with unintended consequences because of an

erroneous, or poorly calibrated, model or the misuse of the model output.
To illustrate the definition we consider the example of the 'London Whale' incident in which JP Morgan lost £6bn in trades and was fined £1bn, as described in [14]. The trading losses were due to JP Morgan shorting synthetic Credit Default Swaps (CDS) derivatives, in other words, betting on an upturn of the market, while in fact, in 2012, the European debt crisis occurred and markets experienced a downturn. Various broader risk management issues contributed to the risk event occurring including the breaching of risk controls, which flagged the trades as substantially more risky than the limits allowed, by the Chief Investment Officer. As a consequence, the calculation of the VaR was changed, however, the updated spreadsheet erroneously underestimated the risk by half, and, as a result, decisions were made that allowed continuing those trades. To make matters worse, other market participants noticed and took opposing positions to JP Morgan. When the debt crisis occurred, trade losses for JP Morgan amounted to £6bn. In a simplified representation, the risk event is the decision to continue with trades that were betting on an upturn of the markets caused by a misrepresentation of the risk due to a spreadsheet calculation error, which resulted in financial losses. For more case studies on model risk see [55], [14] and for some machine or deep learning examples see [56].

5.2 Model Risk Classification

For a systematic approach to classify the different types of model risk we refer to [48]. First, we distinguish between *Structural Risk* and *Operational Risk* as the causes of the event of model risk. Structural Risk refers to issues relating to the model directly (in other words, steps 1-3 of the modelling process defined above) and includes Specification Risk, in other words, the model specification $S(A, E)$, including choosing an inappropriate class of algorithms or an unsuitable candidate for the specific algorithm. It also includes other model assumptions as simplifications of the real world. Also part of the Structural Risk is Parameter Risk (θ) which manifests differently for traditional models, which have a unique optimal parameter set, compared to machine learning models. For traditional models, parameter risk includes calibration errors and errors in parameter estimation through the choice of an unsuitable data set or undetected changes in the data or specific estimation methods used. In addition to these issues, parameter risk for machine learning models may manifest in more ways, due the issue of many "sufficiently good" models that was discussed before, that cannot be differentiated with the relatively simple loss functions used in machine learning, as well as due to the issue of the random training process of machine learning models. The last set of risks relating to the model itself (and not the modelling process) is Numerical and Simulation Errors which we will not focus on in this article. The controls that are part of model validation are those that mitigate the Structural Risk.
Operational Risk refers to the last two stages of the modelling process defined above, including risks in the IT implementation of the model, the inappropriate usage of the model, insufficient data or data of insufficient quality being input into the model (X,T) to produce predictions or other types of process risk, for example using an older, outdated version of a model.

5.3 Model Risk Management Framework

All the activities of an organisation to manage model risk should be guided by a model risk management framework. This framework itself generally forms a part of an organisation's overall

ERM approach. The model risk framework includes the model risk management process (an example of which is described below) but also the principles guiding the process, the related documents (for example, the operational risk management policy, the IT governance policy and terms of reference for a model committee) and the roles and responsibilities of all relevant parties pertaining to the management of model risk. It further includes a description of the resources an organisation allocates to the management of model risk, for example, requirements for the skills and qualification of people that are eligible to be model owners.

The general risk management process is usually seen to involve the steps of Identification, Assessment, Managing, Monitoring and Reporting, and we view these steps as directly applicable to managing model risk. As in other risk management activities, the principle of proportionality applies [53], in other words, the framework for managing model risk needs to adequate in comparison to the size, nature and complexity of the organisation and the use of the model. For a slightly different approach to the model risk management process see [55].

Identification of model risk starts with documenting which models exist in an organisation, what purpose they are used for (Model Use) and which business decisions are based on these models. This will allow an assessment of the criticality of the model according to the relevance of the decisions being based on it. The model inventory would also capture any planned updates, reviews or other development work. Sometimes models have been built with one purpose in mind and are being used now for another application. Thus, if a model inventory is being kept up-to-date, the inventory will allow an assessment of the potential misuse of a model. It also identifies the model owner as person who is responsible to manage the model risk. The scope of the model inventory will depend on the definition of a model and, indeed, a definition that is too rigid might exclude certain tools that are used for decision-making. The model inventory would also typically include an indication of the complexity of the model, including the IT infrastructure used, and the number of people involved in the development, maintenance and running of the model [48].

Assessment of model risk can be done in various ways; from qualitative, such as ranking models on a subjective scale to more sophisticated approaches. Although some quantitative approaches to assessing model risk have been proposed, see for example [57], most companies use a simple frequency and severity approach, which is a subjective assessment of these components of model risk that is performed by conducting interviews with modelling experts and managers. Typically, the strength of the control environment is assessed by quantifying the frequency and severity of the inherent risk, in other words, before the application of relevant controls, and the residual risk, after allowing for the mitigating effect of controls that are currently in place. The frequency of the assessment of the model risk (whether quarterly or annually) will determine the frequency of the model risk management cycle.

The assessment of model risk allows the organisation to determine whether the current risk is within the risk appetite of the organisation. Some companies express their risk appetite relative to each model application or as part of the operational risk framework, for example, as a maximum permissible risk score based on the frequency/severity assessment described above. For example, a qualitative risk appetite might be expressed as a requirement for all business critical models to comply fully with model risk management requirements (such as annual independent review and calibration reports). Another example is that a semi-quantitative risk appetite might require that a model's risk assessment score should not exceed a total of, say,

10 (calculated as frequency x severity, with both rated on a scale from 1 to 5). Alternatively, the risk appetite for model risk might not be made explicit, but is left implicit and can only be observed through the behaviour and decisions made by the parties involved in managing model risk. If this is the case, then, as in other areas of risk management, this can lead to inconsistent practices and application of controls over time and across an organisation.
If risk appetite is defined, the assessment of model risk can be compared to the risk appetite and management actions can be taken if the model risk is currently outside of risk appetite. The successful implementation of those management actions are usually monitored through a risk committee or other governance forum. The monitoring of management actions and the results from all other steps in the model risk management process are reported back into the organisation through the governance structures in place.

5.4 Control Environment

The model risk management framework includes all activities an organisation performs to manage model risk. The model owner, together with the teams operating the model, are generally responsible for the majority of the controls relating to the structural risk of the model, as well as the effectiveness of the overall control environment, and thus have responsibility for reducing the frequency of model risks occurring. Based on the classification of model risk above, the following examples of controls can be implemented to mitigate the risks of traditional actuarial models. While the model risk management framework and process are largely unaffected when moving from traditional actuarial models to deep learning models, the specific controls that are performed by the model owner will need to change, as we discuss in the next section.

- Data Risk
  - Assessment of the input data X to determine sufficient data quantity and quality i.e. clean and fit for purpose, defined metrics and reporting formats to track data quality of the inputs
  - Documentation and review of the manipulation or initial transformation T of the input data X
- Decision Risk
  - Training and communication of the model to model users and decision makers
  - Identify areas of human intervention (e.g. judgements) and what triggers intervention
  - Clear documentation showing that underlying assumptions, and results and appropriateness of the model have been discussed at Board level
  - When complex models are used for decision-making, simpler toy models should be used as a control to back-test reasonability of the complex models
  - Defining clear performance metrics to assess the predictive accuracy of the model
- Specification Risk
  - Clear definition of the problem or business question the model is going to solve, documentation and review of the model specification and how it relates to the business question

  - Model development framework in place
  - Peer review of the model specification i.e. is the model suited to produce the required outcome
  - Governance (committee, minutes) in place to review model specification and related decisions in model development

- Parameter Risk
  - Calculate confidence intervals, sensitivities and stress tests
  - Reasonability checks on the input and output

- Overall Controls
  - Testing of output results against defined criteria (validation of outputs)
  - Sign-off of model calibration process (more important for machine learning models) and model development (more important for traditional models) by model owner at each development cycle
  - Annual review of the model to ensure it is still applicable
  - Version control of the model and its developments

6 Model Risk Management Aspects for Deep Learning Models

To understand how the specific controls for deep learning models will need to change, we first summarize the main differences between the traditional and deep learning approaches, covering both the difference in models and the modelling process.

6.1 Actuarial Models and Deep Learning

Compared with both the traditional and machine learning approaches discussed in Section 2, the *deep learning* approach is different in several aspects. Whereas machine learning models may have an implicit model specification $\tilde{E}$, this specification relies on manual feature engineering to find the set of functions T to transform the known variables X into the feature set X', and similarly actuarial models that are explicitly specified will also depend on manual feature engineering. Thus, in both the actuarial and machine learning approaches, the potential models found using the algorithm are constrained by the feature engineering performed by the modeller. In contrast, models built using deep learning automate the production of the extended feature set X' using the implicit set of transformations $\tilde{T}$, in a manner that depends on the specification of the various layers available to the modeller and, thus, deep learning models have fewer constraints on the model specifications S that might be fit (the requirement to specify manually the optimal architecture for the network means that the feature engineering step is not completely automated). Therefore, compared to the traditional actuarial approach, two key differences of deep learning models are apparent: firstly, the shift to an implicit model specification that is no longer in the direct control of the modeller (this difference is shared with other machine learning algorithms), and secondly, due to the flexibility with which deep learning may specify the feature

matrix X', a larger space of potential models is explored (even compared to the machine learning approach). Whereas these two differences relate to the models themselves, others aspects of the modelling process create a relevant difference from the traditional approach. On the one hand, this is the focus within the machine and deep learning approaches on predictive accuracy, quantified using an explicit loss function, which is not necessarily the case in the traditional approach (which is often more descriptively focussed). On the other hand, the technical implementation of deep learning models lead to specific risks that need to be addressed, for example, the stochastic training of these models. We address each of these aspects in turn in the following sections.

6.2 Implicit Model Specification

In this section, we consider the issues arising from the implicit model specification $\tilde{E}$ that is guided by the choice of machine learning algorithm A. Since the model specification is now implicit, and relatively unconstrained, the actuary may no longer "understand" the model that has been fit, even though its predictive performance has been quantified. Thus, many of the controls applied to reduce the specification risk of models are no longer applicable, because they rely on the interpretability of the models. These issues result from the so-called "black-box" nature of machine learning models [24], where the transformation of known variables to outputs is typically highly non-linear. For example, the documentation and peer review of a machine learning model can no longer rely on inspecting particular coefficients of a model to ensure these are in line with expectations, since the interpretation of these coefficients is no longer straightforward (or the model does not rely on coefficients, for example, in the case of decision trees or ensembles of models). Another example of a control that no longer works is that building an understanding of how the model is answering the business question at hand is no longer applicable, since we only specify the model architecture without an explicit specification of model structure. To understand these issues in more detail, we turn to the framework of [50] who describes the issues resulting from the "black-box" problem by categorizing the aspects of the problem into issues of transparency compared to post-hoc interpretability. Transparency refers to those criteria which relate to the model itself, and is applied at the level of the entire model (simulatability), each of its components (decomposability) and the training algorithm (algorithmic transparency), whereas post-hoc interpretability relates to the extraction of information from learned models to confer useful understanding for practitioners.

6.2.1 Transparency

Many of the controls on the model risk of traditional actuarial models operate by allowing the actuary to step through the calculations required by a model to produce its predictions (simulatibility) or to investigate particular aspects of one part of the model in isolation, say a particular model coefficient (decomposability). This allows the actuary to ensure that the model is operating as intended to solve the business problem, and also to ensure that the model and its parts accord with professional expectations. Although we are not aware of studies showing how risk is controlled through this process, we nonetheless consider these controls as effective, because they allow the actuary to apply heuristics and other professional knowledge to benchmark whether the model is correct. Without the expert knowledge allowing actuaries

to benchmark traditional models, we believe that even these models would be considered non-transparent, and controls would be ineffective. However, we note that the problem of non-transparency, as also the ability to control model risk, is not necessarily only related to machine or deep learning models. For example, the GLMs used currently for non-life pricing in advanced markets may contain upwards of 50 known variables, meaning that even a traditional model may not be simulatable, whereas a sparse LASSO machine learning model [28] may be much more transparent. Thus, the issue of model transparency is beginning to affect even traditional actuarial models, and is not specific to machine or deep learning models, even though the problem is exacerbated with these types of models. Another issue relates to the new problem of algorithmic transparency, where the algorithms used to fit deep learning models may not result in a stable model being fit, leading to uncertainty whether a particular fitted deep learning model is correct, or optimal.

We believe that the viable way of controlling model risks arising from non-transparent model specification is to build the expert knowledge of new classes of algorithms, and the heuristics which enable practitioners to benchmark whether a particular model meets professional expectations. This means that the actuarial profession needs to understand the theory of how deep neural networks should be built in order to ensure a sound model specification, as well as the practical experience of working with these models to be able to judge if they are working optimally.

In addition to this, more simplistic "toy" models can be used as controls to ensure that complex models are functioning appropriately by providing a benchmark for the aggregated results of the complex model. For example, if individual claims reserving is performed on granular data to estimate claim specific reserves, then the Chain-ladder method could be applied to the aggregated data to assess whether the overall reserve level is correct.

Another approach would be to separate a deep neural network model into two parts, a first part that performs feature engineering, and a second part that uses these learned features in a GLM. Referring back to the neural network introduced in Section 3, it is apparent that layers of the neural network appearing before the last layer, in other words, (3.2) and (3.3), perform feature engineering, whereas the last layer of the network, (3.4) is a (generalized) linear model using these learned representations as input. Thus, by inspecting the learned representations in the penultimate layer of the network, it becomes possible to achieve an element of decomposability even for a deep neural network. Usually, the dimension of the penultimate layer of the network will be so large as to preclude inspecting the values taken by individual neurons, however, if the dimension is reduced, say by using PCA, then some insights into what the network has learned may be gained. If multiple networks have been trained, and used together as an ensemble, then inspecting these representations from each network should provide further insights into the variability of the training process. Thus, as part of controlling the model risks of deep neural networks, we recommend that the learned representations be inspected. This recommendation comes with the caveat that usually the first dimension produced by PCA is readily interpretable, but subsequent dimensions are (much) harder to interpret.

A different model risk arising from deep learning models is that it is much harder to set subjectively the value of the output of a deep learning model based on expert judgement, compared to the traditional approach, since these models are not decomposable. For example, an actuary might have a strong belief that theft frequency is higher for some models of car and modify the

parameters of a GLM to reflect this, whereas this is not as straightforward with a deep neural network. Limiting the application of expert prior knowledge may increase model risk, to the extent that better decisions would have been made if this had been included. On the other hand, reducing manual intervention in predictive models has been shown in some contexts to increase predictive accuracy (possibly by reducing the impact of subjective bias) and we refer to Section 4 in [58] for an example in the context of time series forecasting. If including expert judgement within a model is considered to increase predictive accuracy, then the loss function of the neural network could be modified to allow instances of prior knowledge to be included, or else, a model architecture such as that suggested by [59] could be used. In this latter proposal, the input layer of the network is connected directly to the output with a so-called "skip connection" that can be used to transport certain signals, including potential expert judgement, more directly to the output of the network.

A final consideration relates to predicting outside the range of the input data X. A linear model, such as a GLM, might be used successfully to make predictions outside the range of the input data, by relying on the linear assumption of the model specification. However, it would appear that performing the same task with deep neural networks poses more model risk due to their highly non-linear specification.

In summary, due to issues of non-transparency, there is increased model risk, however, this risk appears to be manageable if the controls are changed for the specific class of algorithm used, and as experience in working with these algorithms increases.

6.2.2 Post-hoc Interpretability

Post-hoc interpretability relates to the extraction of information relevant to decision-making from a model through various techniques, such as visualization of what the model has learned, examining sets of examples or investigating how model inputs are related to outputs (local explanations). Many of these techniques are currently used by actuaries in practice for traditional actuarial models and also, many techniques applicable to linear models are equally applicable on machine and deep learning models. Where this is not the case, new techniques have been developed where traditional techniques are not applicable. Examples of these new techniques are Locally Interpretable Model Explanation (LIME) [60], Shapley Additive Explanations (SHAP) [61] and partial dependence plots [26]. From a model risk perspective, post-hoc interpretability allows actuaries to benchmark whether the relationships captured by models, as well as the model results, are sensible, by applying expert knowledge. Since both traditional and deep learning models are equally amenable to post-hoc interpretation, we do not believe there is a need to change model risk controls relating to post-hoc interpretability.

6.3 Representation Learning

While the topic discussed in the previous section applies both to machine and deep learning models, in this section we consider the issue of representation learning which is specific to deep learning models. In other words, here we consider the change in risk due to the automated feature engineering that deep learning models perform ($\tilde{T}$). The issue of representation learning means that deep neural networks are even more susceptible to the black-box problem than machine learning models. Whereas some control over the final model specification is retained within the machine learning approach (since S is constrained by the extent of X'), this is no longer the

case with deep learning. Thus, the challenge to build suitable professional knowledge to mitigate the model risks arising from deep neural networks is even more pronounced than in the case of machine learning models. For example, the deep learning approach relies on the specification of specialized architectures to perform representation learning optimally, thus, there is a new requirement to understand these specialized layers used within deep learning.
Apart from this, two additional problems arise due to the representation learning performed by deep neural networks: firstly, the issue of potential model bias, and, secondly, the possibility that the representations learned by deep neural networks become too specialized with respect to a particular dataset, and thus do not generalize well to new data, or become susceptible to adversarial tricks.

6.3.1 Bias

By applying representation learning to a dataset, deep neural networks may learn representations of the input data that discriminate in ways that are illegal or considered to be unethical by the organisation using the model. For example, within the European Union, it is illegal to include gender as a rating factor for insurance, and some organisations may consider other rating variables, such as credit ratings or education, to be inadmissible from an ethical perspective. In traditional actuarial modelling, the variables used within the model (X') are fully transparent and controlled by the modeller, therefore, if a certain variable should not be used within a model, it is trivial to exclude it. Similarly, machine learning models rely on an explicit set of transformations T to derive the features X' used for the model, allowing the modeller to control what enters the model. However, deep neural network may specify $\tilde{T}$ in such a manner that unwanted representations are recovered by the model, therefore, even if the unwanted variables are excluded from X, the output of the model may nevertheless be biased. For example, [62] show that gender is associated with the usage of a motor vehicle, thus a deep neural network provided with usage information from a telematics dataset may be able to reconstruct a proxy for gender in its representations.
Another way this may occur may be not only due to representation learning being performed on the training data X, but also due to previously unrecognised bias exhibited when collecting training data. For example, as reported in the media, Amazon recently scrapped its algorithm to support recruitment that the company has been using since 2014. The algorithm was built to support the recruitment process by selecting the top candidates based on their resumes only. It was trained using past applications for jobs to Amazon and since the training set was predominantly male (as a reflection of male dominance in the technology industry), the algorithm inherited the bias from the historical data and learned to penalize female applicants. The learned representation was based on key words in the resume that indicated a female applicant, for example, being captain of a women's sports team. Editing the algorithm for those instances could not ensure that the bias was removed and therefore, the algorithm was scrapped.
A potential solution to the problem of bias is to test the output of deep neural networks against predetermined categories of unwanted bias, to investigate if the network has discovered unwanted representations, thus moving prior constraints from the design phase of the model, to the validation stage. Another solution is to calculate the technical price using a full model including, for example, gender and then compare the technical price to the commercial price, calculated using a model that does not include gender, to ensure that the commercial price does not vary

in the same manner as the technical price.
If bias were identified in this manner, a secondary model could be applied to remove it; another solution is to include constraints on the loss function of the neural network to enforce equality between genders or other categories included in the model. While some data may already be recorded allowing this control to be performed (for example, gender), other categories, such as ethnicity, may not be captured, leading to an increased difficulty in recognizing the bias and elevated model risk.

6.3.2 Unintended Consequences of Representation Learning

The representations learned by a deep neural network are guided by the training data supplied to the model. There is a risk that, although the model produces high predictive accuracy using these representations, nonetheless, the model has learned a representation that does not generalize beyond the current data, leading to unwanted consequences. This issue may manifest due to a changing environment, for example, motor insurance data from a time before driving assistance tools were developed is probably of little use for models built for the current environment. Due to the ability of deep learning to produce representations that are highly adapted to a particular dataset, it is fair to assume that the model risk posed by this problem is greater than in the case of traditional models. Outside of insurance modelling, this has been illustrated by, for example, [60] who discusses the application of a deep neural network to identify the species of a dog, in which it was found that model had learned to associate an image's containing snow with huskies. In this case, although the model was able to predict the huskies with high accuracy, it was using a coincidental linkage in the training data, instead of learning a representation that would generalize to all circumstances. Similarly, [50] discusses the case of an adversarial attack on a deep neural network, whereby an image is perturbed by adding pixels that are imperceptible by a human, but change the classification produced by the model. In both these cases, a human has robust prior information which indicates that the representation that has been learned is incorrect.
Whether this issue is directly applicable to actuarial models is not entirely clear, but, as noted before, it would seem that the model risk is heightened and this stresses the need to inspect the learned representations of the model to ensure that these could generalize well outside the data used to train the model. For examples of this, see [7] and [38] who provide interpretations of embedding layers learned by neural networks.
In summary, these aspects of representation learning lead to an increased model risk when using deep learning, as compared to other machine learning techniques, and further research of this issue in the context of actuarial modelling is required [2].

6.4 Loss Functions and Predictive Accuracy

The selection of models based primarily on predictive performance on unseen data has not been emphasized strongly within actuarial practice, despite the importance of this within the machine and deep learning approach (although we note that the AIC and BIC criteria have been discussed in the actuarial and statistical literature). Within the modelling process, several techniques are

[2] Another area that could be explored further is whether actuarial modelling might be made more robust through the use of adversarial techniques, such as randomly perturbing feature values. Anecdotally, similar techniques appear to have enhanced modelling in the context of the Porto Seguro Kaggle claim prediction competition.

used at different stages to enhance predictive accuracy. Firstly, the measurement of predictive performance by defining an explicit loss function allows for models to be selected according to predictive accuracy, and thus offers a (somewhat) objective manner for choosing an optimal model (if the test data is not related to the data that the model will be applied to, then this ranking may be misleading). Secondly, the strict separation of training and test data, allows the generalization performance of the models to be tested, or, if there is not enough data for this split, then a technique such as cross-validation can be applied. We believe that these techniques enhance the predictive accuracy of models and thus reduce model risk, compared to approaches that do not formalize this. A different consideration is that deep learning techniques have been shown to produce more accurate predictions than traditional machine learning techniques, both for actuarial and wider machine learning tasks, as discussed in the introduction. This has particularly been the case in complex problems, when there is no obvious model specification or when it is not obvious which features to derive from the data. In other words, deep learning techniques have been successful in addressing the complexity argument of [1] and [37]. Therefore, when applied to certain types of problem, we believe that deep learning techniques will reduce model risk by increasing predictive accuracy.

On the other hand, as mentioned before, there are difficulties in determining a loss function that connects suitably to the real-world goals of modelling in insurance, leading to risks that must be considered in the model risk management framework. To mitigate these risks, sufficient justification of why a particular loss function is suitable for a problem must be given, and if this connection cannot be clearly made, then there is increased model risk in using the machine or deep learning model. Furthermore, insufficient research on more richly specified loss functions has been performed and, as a consequence, the model risk management framework must consider the risks posed by the range of "sufficiency good" models as measured by the relatively simple loss functions in current use. Since these models can vary substantially in terms of both their structure and the information that can be extracted by post-hoc interpretations of these models, the level of comfort that can be gained by applying post-hoc interpretability techniques must be questioned. To mitigate this increased model risk, it appears reasonable that the choice of the final model used should have a set of interpretations that are in line with prior expert knowledge of the problem, or, more optimally, enhancements to the loss function should be made to narrow the range of acceptable models.

Another consideration relates to confidence bounds, which are helpful for determining the confidence with which a model prediction can be trusted, and are also useful, in some situations, for determining capital requirements. Whereas most traditional models provide point estimates as well as confidence bounds, confidence bounds for machine learning models have received (much) less attention, and further research into these methods is warranted.

On balance, we consider that the relatively increased model risk of deep learning techniques resulting from the implicit model specification and representation learning may be somewhat remediated by the emphasis on predictive ability, to the extent that the loss function is well aligned to the modelling problem. The quantification of predictive accuracy is not limited in its application to machine or deep learning models, and to reduce model risk, should also be applied to traditional techniques. On the other hand, until sufficient professional knowledge about these techniques is accumulated (including the development of robust confidence bounds), there may be increased risk that these techniques are applied in error, leading to erroneous business

decisions being made.

6.5 Stability and Consistency of Neural Networks

Another issue discussed in this paper is the stability and consistency of neural network solutions. The dependency of the final model on the random seed used in the random number generator has the consequence that overall model performance, and individual model predictions, might differ substantially over different model training attempts. Furthermore, the stability of model results as new observations are made has not been investigated in detail for deep learning applied to insurance modelling. These properties are not ideal, and form a serious obstacle in using the model results practically since models and model results are expected to be stable and enable consistent decision-making in insurance management over time. Furthermore, policyholders, regulators and other stakeholders have similar expectations of relative stability. From a risk management perspective, models and model results that are unstable are usually deemed unacceptable for decision-making purposes, and claims of model outperformance that cannot be reproduced are unlikely to be accepted. Some of these issues are familiar from traditional actuarial modelling, for example, after a large loss occurs unexpectedly on a line of business, severe reserve adjustments might result. Nonetheless, the issues are more pronounced for machine and deep learning models. To some extent, the issue of models depending on the random seed can be mitigated by ensembling many trained models together, thus providing a more stable estimate of model performance, with the added advantage that ensembles of deep neural networks appear to outperform single models. If stability will be introduced by relying on an ensemble, then the model risk of this solution should be investigated by risk managers by considering the overall volatility of the results, the number of runs that are needed to attain stability and the whether more stable results might be attained if other model architectures were to be tested. Also, risk managers should be aware of the potential for model results to be inconsistent over time. To test whether complex models are behaving reasonably consistently over time, a simpler "toy" model from class of algorithms known to be consistent could be fit to updated data, and the predictions from this model analysed. If these predictions are as unstable as those produced by a deep neural network, then some comfort can be gained on the neural network predictions. It is important to note that as yet, no solutions have been studied in detail for this problem, and this is an area requiring future research.

7 Applied Model Risk Considerations

We now provide two examples of deep neural networks applied to the problems of pricing of non-life business and mortality forecasting. In both of these examples, the class of algorithms, A, has been chosen to be neural networks and therefore, there is neither an explicit model specification E, nor a set of transformation functions T, as both of these modelling activities are performed implicitly by the neural network. Subsequent to introducing these models, we analyse the model risk according to the structure of the previous section.

7.1 Example 1 - Pricing

Pricing of non-life insurance policies is generally performed using GLM methods, fitting separate models for claims frequency and severity. Recently, [29] explored the application of several machine learning techniques to a French Motor Third Party Liability (MTPL) dataset accompanying [63], finding that the machine learning techniques outperform the relatively simple GLM fit as a baseline model (the outperformance here is on an individual policy level, averaged over the whole portfolio. In practice, one also needs to make sure that the average price levels of the portfolio are reproduced, as described in [11]). Further studies of this dataset using deep neural networks are in [5] and [38], who use embedding layers to model this dataset, and find that including embedding layers within neural networks results in better performance compared to networks without embedding layers.

Here, we apply a 5 layer deep neural network with 32 neurons in each layer, embedding layers of dimension 5 for each categorical variable in the dataset and ReLu activation functions. Between each layer, we apply dropout regularization [45], which is a standard technique to regularize the network (in other words, to ensure the network generalizes well to the test data by preventing the network from over-fitting to the training data) and batch normalization [64], which is another standard technique to both regularize the network and make optimisation more efficient. The Keras code (using the R package) [65] to fit this network is provided in Listings 1-2 in the appendix. The known variables X are shown in Table 1.

Table 1: Extract of known variables X from the French MTPL dataset

	Area	VehPower	VehAge	DrivAge	BonusMalus	VehBrand	VehGas	Density	Region
1	B	4	11	23	68	B2	Regular	96	R93
2	E	6	11	27	90	B2	Diesel	2,740	R22
3	E	5	0	70	50	B12	Regular	3,075	R72
4	D	12	4	41	50	B10	Regular	1,313	R24
5	A	7	2	61	56	B1	Diesel	11	R24

In this pricing model, it is neither possible to step through the calculations, due to the complexity of the network structure and its depth, nor is it possible to examine all of the model coefficients, since these do not necessarily have a straightforward interpretation as would be the case in a GLM. However, by applying the decomposability technique of Section 6, it is possible to understand what the model has learned and to which features the model coefficients of the last layer relate.

As an example, the representations in the last layer of the deep neural network were analysed. PCA was applied to reduce the dimension of the 32 dimensional representation to a single dimension, in other words, the first principal component score was isolated. The average value of the score was calculated for groups defined according to the age at which a driver was recorded in the data, and whether the policy related to a low, medium, high or very high density area. Lastly, the averages were smoothed using the LOESS (locally estimated scatterplot smoothing) technique. The results of this analysis are shown in the left panel of Figure 2 and the right panel of the figure shows the average raw claims frequencies calculated in the same groups. It can be observed that the representation learned by the model has a similar shape to the claims frequencies, and that by scaling and shifting this representation (refer to equation (3.4)), the observed

claims frequencies will be reproduced closely. The representation accords well with known facts about motor claims: higher values have been estimated for younger drivers and drivers in higher density areas have higher values compared to those in lower density areas. Somewhat less intuitive are the values of the representation for the highest density area at the oldest ages, which continue to decrease with age in similar manner to the observed claims frequencies, whereas the representations for the other areas begin to rise with age at around age 80. Furthermore, the values of the representations for the medium and high density areas also begin to increase, whereas the observed claims rates in these groups decrease. Since the volumes of data at these advanced ages are low, it would be advisable to investigate further whether predictive accuracy might be improved if these values were modified using expert judgement, in particular, since the low volumes may imply that over-fitting to individual observations has occurred despite the application of regularization. An alternative to modifying the representation might be to only use the predictions from the network up to a specific age, say age 80, after which a traditional model might be applied.

Although we do not report in detail on the results of further analysis here, we note briefly that this network was fit another 20 times and the learned representations were visualized for each training run. In almost every case, similar patterns to those noted in 2 were found, for the majority of the age range. The behaviour at these higher ages was more variable, giving more weight to the argument that the behaviour of the network at these ages needs more investigation.

To review the model specification, which is now implicit compared to a GLM, it is necessary to perform a peer review of the model architecture. Practically, this means assessing whether the width of the model layers, the activation functions, and dropout layers have been selected in an appropriate manner. Thus, a peer review is possible, but will take a different form from traditional models. A further set of controls would be to investigate that the model output accords with expectations, for example, by examining the frequency of a claim produced by the model for certain policy records, or by examining visualizations of frequencies against known variables, such as driver age or population density, see, for example [29] who provide examples of how these visualizations could be produced. Considering whether the deep neural network has learned representations that might be illegal or considered unethical by the organisation, requires the organisation to articulate these upfront. For example, an organisation might consider it unethical to discriminate by ethnicity, and a feature highly correlated with ethnicity may have been reconstructed using some of the variables within the dataset. In this example, the data needed to verify this is not available, but the dataset could be augmented with other data sources, giving, for example, the most likely ethnicity in certain regions, in other words, the control groups may be chosen in a stratified way representing different characteristics. In addition, even without this data to allow for explicit analysis, expert knowledge could determine if the learned representations are likely to be highly correlated with unwanted biases. Connected to this is the problem of unintended consequences of representation learning, for example, whereas we might intend for the model to learn the frequency of claims using vehicle brand as a proxy for power-to-weight ratio, and region as a proxy for weather, instead the model may learn a proxy for socio-economic status based on the combination of these two factors. In an unchanging environment, this would probably not be an issue, but if regional socio-economic circumstances are changing rapidly, this may well become problematic.

Balancing these issues, the accuracy of the deep neural network as quantified by the Poisson

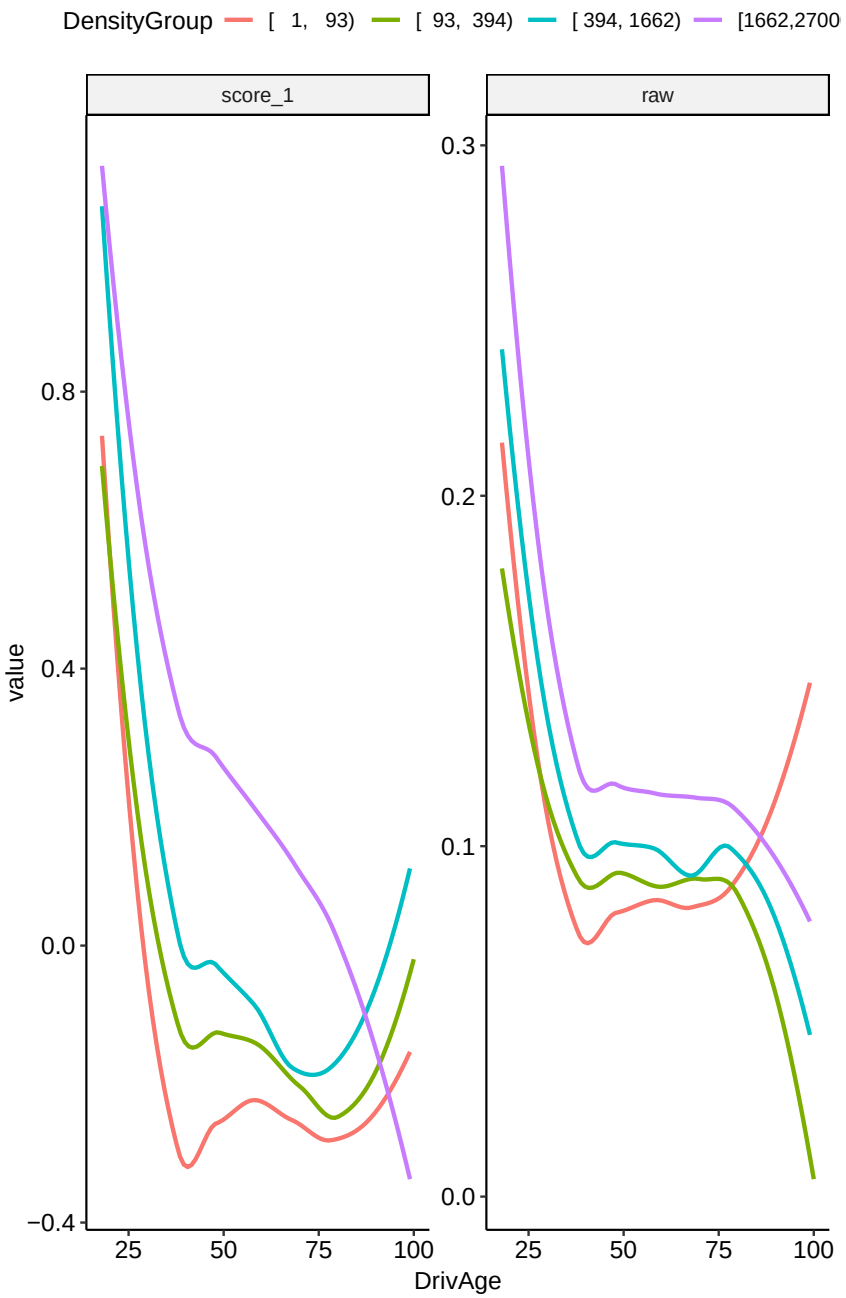

Figure 2: Learned representations from the pricing example discussed in Section 7.1. The left panel shows the average representation value, calculated for each driver age and density group separately, after reducing the dimensionality of the learned representation from 32 to 1. The right panel shows the corresponding observed claims frequencies in each group. The density groups are low, medium, high and very high, and the numerical values defining each group are shown at the top of the figure.

deviance is better than that of other traditional and machine learning models [5], so, to the extent that the model will be applied to similar data, it is likely that the company will achieve more optimal results (in terms of more accurate prices for individual policyholders) using the deep neural network.

7.2 Example 2 - Mortality Forecasting

Mortality forecasting is an important step in setting mortality improvement assumptions for use in actuarial valuations. Many univariate forecasting models have been proposed, with fundamental contributions made by [19] and [66]. Recently, a multi-population model using deep neural networks was proposed by [7], who forecast the mortality experience of 38 countries, for both genders (in other words, 76 separate populations were considered) in the Human Mortality Database [67] over the period 2000-2015, and found that the model outperformed other competing approaches for 51 out of 76 populations (see Table 6 of [7]). Here, we consider the DEEP5 model of [7], which consisted of a 5 layer deep neural network with 128 neurons in each layer, embedding layers of 5 dimensions for each categorical variable in the dataset and ReLU activation functions, as well as dropout and batch normalization. The Keras code to fit a variant of this network is provided in Listing 2 of [7]. The known variables X are shown in Table 2.

Table 2: Extract of known variables X from the Human Mortality Database

	Year	Age	Country	Gender
1	1990	73	UKR	Male
2	1965	31	BEL	Male
3	2010	87	FRATNP	Male
4	1990	7	RUS	Female
5	1959	51	LTU	Female

Comparing this example to the pricing model considered earlier, it is clear that this model is more complex, and in terms of practical use, is likely to have much greater financial importance to a life insurance company, due to the long term nature of the liabilities, and the longer time scale over which experience investigations can be performed. However, the risks and necessary controls when moving from a traditional mortality forecasting model, such as the Lee-Carter model, to a deep neural network are very similar to those described for the previous example of a pricing model. For example, inspecting the features in the penultimate layer of the model can be performed in the same manner, once the dimensionality of the 128 features is reduced using a suitable dimension reduction algorithm, such as PCA. The results of applying this technique to a single training run of this model are shown in Figure 3. It can be observed that the learned representations follow the familiar shape of a life table, with mortality improvement being noticeable as lower values at each age in the year 2000 compared to the year 2010.

The peer review of the network architecture should be performed in exactly the same way as before. The issue of unwanted bias is probably less pronounced in this example, but controls could in principle be performed in the same way.

Having considered the similarities between the risk management of both models, we note that

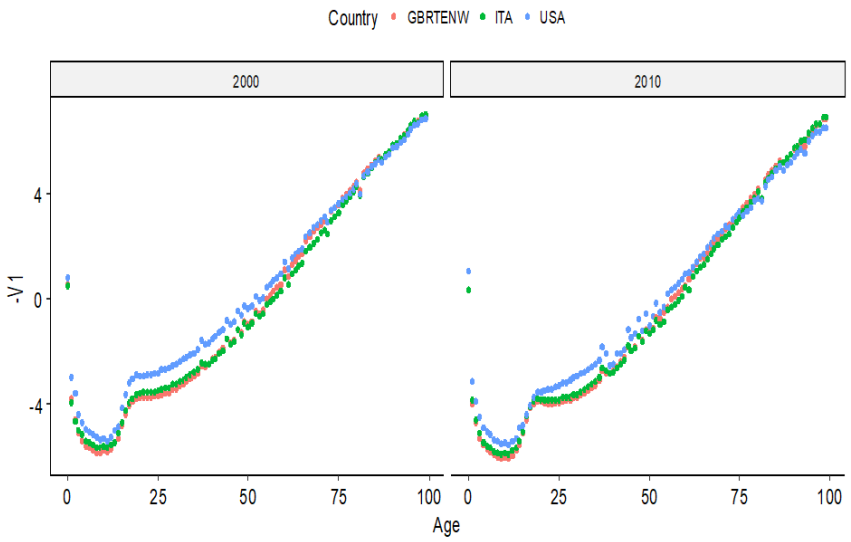

Figure 3: Learned representations from the mortality forecasting example discussed in Section 7.2. The left panel shows the learned representation values, for each of the United Kingdom, Italy and the United States in the year 2000, after reducing the dimensionality of the learned representation from 128 to 1. The right panel shows the corresponding learned representations in the year 2010. Note that the representations were reoriented by multiplying the values by -1 and that no smoothing was applied to avoid distorting key features of the life table.

the main difference between the mortality forecasting and pricing models is that the while the pricing model is used to predict unseen observations within a portfolio, the mortality forecasting model is used to extrapolate mortality assumptions beyond the last year of observed mortality data (in this example, training data was used up to the year 2000). Therefore, if the slope and curvature of the forecasting model at this observation boundary are estimated inaccurately, then the extrapolation will fail completely, thus, special controls are needed to ensure that the model is functioning correctly at this point. Since there are no explicit slope and curvature parameters in this neural network, this task is harder than in a traditional forecasting model. One potential method of investigating the behaviour of the network is to back-solve the equivalent parameters of a simpler mortality forecasting model to gain a better understanding of the extrapolation performed by the model, and we refer the reader to [39] who illustrate this for recurrent neural networks.
As a final consideration, predictive accuracy of the model has been shown in [7] to be better than traditional models, indicating a reduced model risk from that perspective.

8 Discussion and Conclusions

In this paper, we have formalized the definition of predictive models used in practice for actuarial work and contrasted these models to machine learning and deep learning models by defining a common framework to express models of each type. We have also considered the modelling process and what might be different when utilizing deep learning models. The paper has then considered the technical challenges of applying deep learning models, such as the potential for

model calibrations to be influenced by the random elements of the fitting process, the potential lack of stability of model results over time and the possible bias that neural networks might exhibit when predicting portfolio averages. By examining the way in which these models differ from each other, we have identified where model risk might be higher, compared to traditional actuarial models, but also where it might be lower. In particular, we have identified that additional risk is introduced by implicit model specification ($\tilde{E}$), in common with machine learning techniques, and by the representation learning (implicit feature engineering) performed by deep neural networks ($\tilde{T}$). As a result, deep neural networks are likely to be less transparent than traditional actuarial models (although some of these issues are now becoming apparent even in complex traditional models) and may suffer from unwanted bias, leading to increased model risk. To mitigate these model risks to some extent, we have suggested controls in the body of the text, including peer review of the models and examination of learned representations and outputs of these models. Furthermore, we have considered how risk managers might tackle some of the technical challenges mentioned above, and mitigate the attaching model risk. Since these controls and techniques need further development (as we discuss further below), we consider that currently, an important control for risk managers is to consider the outputs of simpler, traditional models to benchmark the results of deep learning models at an aggregate level.

On the other hand, we believe that as deep learning techniques achieve greater accuracy on some tasks than is possible using traditional actuarial and machine learning techniques, the overall model risk faced by organisations using deep learning techniques will be reduced due to the greater accuracy and consequently, better decisions, that will be possible.

Having provided an initial analysis of the model risk of deep learning techniques, several new areas of research can be identified for future development. To enhance post-hoc interpretability, we showed in Section 7.1 how the learned representations of a deep neural network can be inspected and interpreted. Future research should demonstrate this technique on more examples, and seek to understand its relationship to other machine learning interpretability techniques. More analysis of other machine learning interpretability techniques and their suitability for actuarial application is also required.

The application of deep learning to insurance modelling will benefit from research considering how the predictions of these models might be stabilized as new observations are used to recalibrate the models and also from further consideration of bias reduction techniques, extending the work of [11]. We believe that a better theoretical understanding of the ensemble properties of neural networks will help alleviate some of the problems induced by the random elements of the training process.

In Section 6.2.1, we suggested that the increased model risk introduced by the decreased transparency of machine and deep learning models can be mitigated by expert peer review, which will ensure that the models are specified, trained and operated according to best practice. To allow this, the actuarial profession needs to build a suitable body of knowledge amongst its practitioners, enabling them to build and operate machine and deep learning models in a safe manner. An initial step would be to design a new actuarial curriculum covering these topics as part of basic actuarial training, and we refer the interested reader to [68] as an example of how actuarial associations may tackle this topic.

Acknowledgements

We thank Jaco van der Merwe for his review and comments which helped to improve the quality of the manuscript. The first two authors thank several of their colleagues at QED for their helpful suggestions.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[3] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," *arXiv*, arXiv:1604.06737, 2016.

[4] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 competition: results, findings, conclusion and way forward," *International Journal of Forecasting*, 2018.

[5] R. Richman, AI in actuarial science. *SSRN Manuscript* ID 3218082. Version July 24, 2018.

[6] A. Gabrielli, R. Richman, and M.V. Wüthrich, "Neural network embedding of the over-dispersed poisson reserving model," *Scandinavian Actuarial Journal*, pp. 1–29, 2019.

[7] R. Richman and M.V. Wüthrich, "A neural network extension of the Lee–Carter model to multiple populations," *Annals of Actuarial Science*, p. 1–21.

[8] K. Kuo, "DeepTriangle: A deep learning approach to loss reserving," *arXiv*, arXiv:1804.09253, 2018.

[9] M.V. Wüthrich, "Neural networks applied to chain-ladder reserving." *European Actuarial Journal*, vol. 8, no. 2, pp. 407–436, 2019.

[10] G. Gao and M.V. Wüthrich, "Feature extraction from telematics car driving heatmaps." *European Actuarial Journal*, vol. 8, no. 2, pp. 383–406, 2018.

[11] M.V. Wüthrich, "Bias regularization in neural network models for general insurance pricing," *SSRN Manuscript* ID 3347177, 2019.

[12] United States Federal Reserve, "SR 11-7: Guidance on model risk management," 2011.

[13] South African Reserve Bank, "Prudential Standard GOI 3.1 - Own Risk and Solvency Assessment for Insurers," 2018.

[14] A. Aggarwal, M.B. Beck, M. Cann, T. Ford, D. Georgescu, N. Morjaria, A. Smith, Y. Taylor, A. Tsanakas, L. Witts, *et al.*, "Model risk - daring to open up the black box," *British Actuarial Journal*, vol. 21, no. 2, pp. 229–296, 2016.

[15] A.S. Macdonald, "Current actuarial modeling practice and related issues and questions," *North American Actuarial Journal*, vol. 1, no. 3, pp. 24–35, 1997.

[16] W.S. Jewell, "Models in insurance: paradigms, puzzles, communications and revolutions," Technical Report, University of California, Berkley, Operations Research Center, 1980.

[17] G. Shmueli, "To explain or to predict?," *Statistical Science*, pp. 289–310, 2010.

[18] R. Thomson, "A typology of models used in actuarial science," *South African Actuarial Journal*, vol. 6, no. 1, pp. 19–36, 2006.

[19] R.D. Lee and L.R. Carter, "Modeling and forecasting US mortality," *Journal of the American Statistical Association*, vol. 87, no. 419, pp. 659–671, 1992.

[20] E. Ohlsson and B. Johansson, *Non-Life Insurance Pricing with Generalized Linear Models*, vol. 2. Springer, 2010.

[21] I.D. Currie, "On fitting generalized linear and non-linear models of mortality," *Scandinavian Actuarial Journal*, vol. 2016, no. 4, pp. 356–383, 2016.

[22] S. Wood, *Generalized Additive Models: an Introduction with R*. Chapman and Hall/CRC, 2017.

[23] P. De Jong and G.Z. Heller, "Generalized linear models for insurance data," 2008.

[24] L. Breiman, "Statistical modeling: the two cultures (with comments and a rejoinder by the author)," *Statistical Science*, vol. 16, no. 3, pp. 199–231, 2001.

[25] M.V. Wüthrich, "Price stability in regression tree calibrations," in *Proceedings of 2017 China International Conference on Insurance and Risk Management*, pp. 749–762, Tsinghua University Press, 2017.

[26] J. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[28] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[29] A. Noll, R. Salzmann, and M.V. Wüthrich, "Case study: French motor third-party liability claims," *SSRN Manuscript* ID 3164764, 2018.

[30] G. Gao, S. Meng, and M.V. Wüthrich, "Claims frequency modeling using telematics car driving data." *Scandinavian Actuarial Journal*, vol. 2019, no. 2, pp. 143–162, 2018.

[31] M.V. Wüthrich, "Machine learning in individual claims reserving," *Scandinavian Actuarial Journal*, vol. 2018, no. 6, pp. 1–16, 2018.

[32] P. Deprez, P. Shevchenko, and M.V. Wüthrich, "Machine learning techniques for mortality modeling," *European Actuarial Journal*, vol. 7, no. 2, pp. 337–352, 2017.

[33] A. Nigri, S. Levantesi, M. Marino, S. Scognamiglio, and F. Perla, "A deep learning integrated Lee-Carter model," *Risks*, vol. 7, no. 1, p. 33, 2019.

[34] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[35] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, p. 386, 1958.

[36] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[37] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[38] J. Schelldorfer and M.V. Wüthrich, "Nesting classical actuarial models into neural networks," *SSRN Manuscript* ID 3320525, 2019.

[39] R. Richman and M.V. Wüthrich, "Lee and Carter go Machine Learning: Recurrent Neural Networks," *SSRN Manuscript* ID 3441030, 2019.

[40] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[41] Y. LeCun, C. Corinna, C. Burges, "The MNIST database of handwritten digits," *Database*.

[42] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[44] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[46] N. Zumel, "An ad-hoc method for calibrating uncalibrated models," *Blog post*, 2019.

[47] L. Rossouw, R. Richman, "Using machine learning to model claims experience and reporting delays for pricing and reserving," *Paper presented at the Actuarial Society of South Africa's 2019 Convention*, 2019.

[48] CRO Forum, "Leading practices in model management," 2017.

[49] P. Parodi, *Pricing in General Insurance*. CRC Press, 2014.

[50] Z.C. Lipton, "The mythos of model interpretability," *arXiv* arXiv:1606.03490, 2016.

[51] J. Lam, *Enterprise Risk Management: from Incentives to Controls*. John Wiley & Sons, 2014.

[52] P.O.J. Kelliher, D. Wilmot, J. Vij, and P.J.M. Klumpes, "A common risk classification system for the actuarial profession," *British Actuarial Journal*, vol. 18, no. 1, p. 91–121, 2013.

[53] HM Treasury, "The orange book: management of risk - principles and concepts," *London: HM Treasury*, 2004.

[54] European Parliament, Council of the European Union, "Directive 2013/36/EU of the European Parliament and of the Council of 26 June 2013 on access to the activity of credit institutions and the prudential supervision of credit institutions and investment firms, amending directive 2002/87/EC and repealing directives 2006/48/EC and 2006/49/EC Text with EEA relevance," 2006.

[55] R. Black, A. Tsanakas, A.D. Smith, M.B. Beck, I.D. Maclugash, J. Grewal, L. Witts, N. Morjaria, R. Green, and Z. Lim, "Model risk: illuminating the black box," *British Actuarial Journal*, vol. 23, 2018.

[56] CRO Forum, "Machine decisions: Governance of AI and big data analytics," 2019.

[57] P. Glasserman and X. Xu, "Robust risk measurement and model risk," *Quantitative Finance*, vol. 14, no. 1, pp. 29–58, 2014.

[58] R.J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2018.

[59] M.V. Wüthrich and M. Merz, "Yes, we CANN!," *ASTIN Bulletin: The Journal of the IAA*, vol. 49, no. 1, pp. 1–3, 2019.

[60] M.T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, ACM, 2016.

[61] S.M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

[62] M. Ayuso, M. Guillen, and A.Pérez-Marín, "Telematics and gender discrimination: some usage-based evidence on whether men's risk of accidents differs from women's," *Risks*, vol. 4, no. 2, p. 10, 2016.

[63] A. Charpentier, *Computational Actuarial Science with R*. CRC Press, 2014.

[64] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[65] J. Allaire and F. Chollet, *R Interface to Keras*. RStudio, Google, 2018.

[66] A.J.G. Cairns, D. Blake, and K. Dowd, "A two-factor model for stochastic mortality with parameter uncertainty: theory and calibration," *Journal of Risk & Insurance*, vol. 73, no. 4, pp. 687–718, 2006.

[67] J.R. Wilmoth and V. Shkolnikov, "Human mortality database," *University of California*, 2010.

[68] Data Science Working Group of the Swiss Association of Actuaries (SAA), "Data science strategy," tech. rep., Swiss Association of Actuaries (SAA), 2018.

A Appendix 1 - Keras Code for Pricing Model

This Appendix contains the code for the pricing model discussed in Section 7.1. In the first code excerpt, the embedding layers are set up and in the second excerpt, the rest of the network is defined [3]. The third excerpt provides the code to initialize a Keras model to produce the learned representations, and finally to analyse these using PCA.

Since the representations learned in the last layer of the network will be analysed, we label the last layer of the network as "ReluLayer", see Line 28 of Listing 2, which allows access to the last layer of the model. The code to set up the model accessing the last layer is shown on Lines 1-3 of Listing 3, Lines 9-11 show the analysis using PCA, and lastly, the average representation for each group is calculated and then illustrated on Lines 17-25.

[3]The network architecture presented here could be fine-tuned, for example, embedding layers of dimension 5 have been chosen for all variables, and this may by improved by tweaking the dimensions of the embedding layers for each variable.

Listing 1: Keras code for embedding layers.

```
Exposure <- layer_input(shape = c(1), dtype = 'float32', name = 'Exposure')

VehGas <- layer_input(shape = c(1), dtype = 'int32', name = 'VehGas')
VehGas_embed = VehGas %>%
  layer_embedding(input_dim = 2, output_dim = 5,input_length = 1, name = 'VehGas_embed') %>%
  keras::layer_flatten()

VehPower <- layer_input(shape = c(1), dtype = 'int32', name = 'VehPower')
VehPower_embed = VehPower %>%
  layer_embedding(input_dim = 12, output_dim = 5,input_length = 1, name = 'VehPower_embed') %>%
  keras::layer_flatten()

VehAge <- layer_input(shape = c(1), dtype = 'int32', name = 'VehAge')
VehAge_embed = VehAge %>%
  layer_embedding(input_dim = veh_age_dim, output_dim = 5,input_length = 1, name = 'VehAge_embed') %>%
  keras::layer_flatten()

DrivAge <- layer_input(shape = c(1), dtype = 'int32', name = 'DrivAge')
DrivAge_embed = DrivAge %>%
  layer_embedding(input_dim = driv_age_dim, output_dim =5,input_length = 1, name = 'DrivAge_embed') %>%
  keras::layer_flatten()

VehBrand <- layer_input(shape = c(1), dtype = 'int32', name = 'VehBrand')
VehBrand_embed = VehBrand %>%
  layer_embedding(input_dim = 11, output_dim = 5,input_length = 1, name = 'VehBrand_embed') %>%
  keras::layer_flatten()

Region <- layer_input(shape = c(1), dtype = 'int32', name = 'Region')
Region_embed = Region %>%
  layer_embedding(input_dim = 22, output_dim = 5,input_length = 1, name = 'Region_embed') %>%
  keras::layer_flatten()

Area <- layer_input(shape = c(1), dtype = 'int32', name = 'Area')
Area_embed = Area %>%
  layer_embedding(input_dim = 6, output_dim = 5,input_length = 1, name = 'Area_embed') %>%
  keras::layer_flatten()

BonMal <- layer_input(shape = c(1), dtype = 'int32', name = 'BonMal')
BonMal_embed = BonMal %>%
  layer_embedding(input_dim = bon_mal_dim, output_dim = 5,input_length = 1, name = 'BonMal_embed') %>%
  keras::layer_flatten()

Density <- layer_input(shape = c(1), dtype = 'int32', name = 'Density')
Density_embed = Density %>%
  layer_embedding(input_dim = density_dim, output_dim = 5,input_length = 1, name = 'Density_embed') %>%
  keras::layer_flatten()
```

Listing 2: Keras code for deep ReLu network.

```
embed_layer <- layer_concatenate(list(VehGas_embed,
                                      VehBrand_embed,
                                      VehPower_embed,
                                      VehAge_embed,
                                      DrivAge_embed,
                                      VehBrand_embed,
                                      Region_embed,
                                      Area_embed,
                                      BonMal_embed,
                                      Density_embed)) %>%
  layer_dropout(rate=0.05)

relu_layer=embed_layer %>%
  layer_dense(units = 32, activation = 'relu') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.1) %>%
  layer_dense(units = 32, activation = 'relu') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.1) %>%
  layer_dense(units = 32, activation = 'relu') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.1) %>%
  layer_dense(units = 32, activation = 'relu') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.1) %>%
  layer_dense(units = 32, activation = 'relu') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.1, name="ReluLayer")

main_output = relu_layer %>%
  layer_dense(units = 1, activation = 'sigmoid', name = 'main_output')

N <- layer_multiply(list(main_output,Exposure), name = 'N')
```

Listing 3: R code for accessing learned representations.

```
model_last <- keras_model(
  inputs = c(Exposure,VehGas, VehPower, VehAge,DrivAge,VehBrand,Region,Area,BonMal, Density),
  outputs = c(relu_layer))

learned = model_last
  %>% predict(x,batch_size = 256*8*4)
  %>% data.table()

pcas = learned
  %>% as.matrix()
  %>% princomp()

learn[,paste0("scores_",1):=data.table(pcas$scores[,1])]

learn[,DensityGroup := cut2(Density,g = 4)]

learn[,.(score_1 = mean(scores_1),
  raw= sum(ClaimNb)/sum(Exposure)),
  keyby = .(DrivAge,DensityGroup)] %>%

  melt(id.vars=c("DrivAge","DensityGroup")) %>%

  ggplot(aes(x=DrivAge ,y=value)) +
  geom_smooth(aes(color = DensityGroup),se=F) +
  facet_wrap(~variable, scales="free")+theme_pubr()
```