Annals of Actuarial Science (2021), **15**, pp. 346–366
doi:10.1017/S1748499519000071

PAPER

A neural network extension of the Lee–Carter model to multiple populations

Ronald Richman[1]* and Mario V. Wüthrich[2]

[1] Actuarial Department, AIG South Africa, Johannesburg, Gauteng 2196, South Africa and [2] RiskLab, Department of Mathematics, ETH Zurich, 8092, Zurich, Switzerland
*Corresponding author. Email: ron@ronaldrichman.co.za

(Received 31 October 2018; revised 5 May 2019; accepted 8 May 2019; first published online 28 June 2019)

Abstract

The Lee–Carter (LC) model is a basic approach to forecasting mortality rates of a single population. Although extensions of the LC model to forecasting rates for multiple populations have recently been proposed, the structure of these extended models is hard to justify and the models are often difficult to calibrate, relying on customised optimisation schemes. Based on the paradigm of representation learning, we extend the LC model to multiple populations using neural networks, which automatically select an optimal model structure. We fit this model to mortality rates since 1950 for all countries in the Human Mortality Database and observe that the out-of-sample forecasting performance of the model is highly competitive.

Keywords: Mortality forecasting; Lee–Carter model; multiple populations; neural networks

1. Introduction

Understanding and quantifying mortality rates are fundamental to the study of the demography of human populations, and they are a basic input into actuarial calculations involving valuation and pricing of life insurance products. Since mortality rates have been observed to change over time, techniques to forecast future mortality rates are important within both demography and actuarial science. Two well-known examples of these techniques are the Lee–Carter (LC) (Lee & Carter 1992) and the Cairns–Blake–Dowd (CBD) (Cairns *et al.* 2006) models, which forecast mortality rates in two steps: firstly, a low dimensional summary of past mortality rates is constructed by fitting statistical models to historical mortality data, and secondly, future mortality rates are forecasted by extrapolating the summarised mortality rates into the future using time series models.

The LC and CBD models were applied originally to single populations. If forecasts for multiple populations were required, then the models were fit to each population separately. However, for several reasons, it seems reasonable to expect that a multi-population mortality forecasting model would produce more robust forecasts of future mortality rates than those produced by single-population models. If changes in mortality in a group of countries are due to common factors such as similar socioeconomic circumstances, shared improvements in public health and medical technology, then it makes sense to forecast the mortality rates for these countries as a group. Furthermore, mortality trends that are common to several populations would likely be captured with more statistical credibility in a model that relies on the experience of many countries, see Li & Lee (2005). Thus, Li & Lee (2005) recommend that multi-population models should be used even if the ultimate interest is only in forecasts for a single population. In addition, mortality forecasts from separate single-population models may diverge from each other, leading to implausible results if used in actuarial and demographic models, whereas a multi-population model can produce coherent forecasts. To this end, multi-population variants of the LC and CBD models have recently been developed.

In this work, we concentrate on the LC model and its multi-population extensions. We describe their design and fitting which have caused several challenges in the past. The original LC model cannot be fit as a regular regression model due to the lack of covariates, and, in the original paper, a Singular Value Decomposition (SVD) was applied to fit the model. More recently, regression approaches have been applied (Brouhns *et al.* 2002; Currie 2016) within the framework of non-linear regression models. Extending the SVD and regression frameworks to the multi-population case has proven challenging, and recent studies have resorted to elaborate optimisation schemes to fit these models, see for example, Danesi *et al.* (2015) or Enchev *et al.* (2017), or to relatively less well-known statistical techniques, such as Common Principal Components, as applied by Kleinow (2015). A further challenge is the significant amount of judgment that needs to be exercised when choosing the data on which to fit the multi-population models, so that similar countries are grouped together, in other words, it appears that the multi-population models developed to this point are not suitable for large scale mortality forecasting. Finally, and most significantly, the extension of the LC model to multiple populations can be accomplished in several ways, which we describe next, and it is not clear which of these extended models is optimal, or why. For example, Li & Lee (2005) propose to model a common age trend between populations, and fit a secondary population-specific LC-type model to the residuals of the common age model. Kleinow (2015) designs a so-called Common Age Effect (CAE) model, where the component of the LC model describing the change in mortality with time is held constant, but different period indices are fit for each population (for more examples of variations on LC models, see Chen & Millossovich (2018) and Danesi *et al.* (2015)). In a comparison of these models, and two other related variations, it was found that the CAE model fits better (Enchev *et al.* 2017) than the other models, but no theory seems to have been developed to explain these findings (we refer the reader to Villegas *et al.* (2017) for a partial explanation of the success of the CAE model).

It thus emerges that significant judgment needs to be applied when choosing the form of a multi-population mortality model, requiring, to borrow terminology from the machine learning literature, substantial manual “feature engineering”. In contrast, in this paper, we seek to offer an alternative multi-population mortality forecasting model that requires less manual feature engineering, can be fit to many populations simultaneously and can be fit using relatively standard optimisation approaches. This model is based on neural networks, which have recently been used to achieve a number of breakthrough results in the areas of computer vision, speech recognition and natural language processing tasks, see Bengio *et al.* (2013). Neural networks have been shown to automatically learn meaningful representations of the data to which they are applied, see for example, Guo & Berkhahn (2015) and Mikolov *et al.* (2013), and thus, our approach implements the paradigm of representation learning, which avoids manual feature engineering by using a neural network to derive automatically an optimal set of features from the input data. Modern software implementations (Allaire & Chollet 2018; Abadi *et al.* 2016) of the back-propagation algorithm (Rumelhart *et al.* 1986) allow these models to be fit easily in a number of different open-source software environments. A final advantage of our model is that the forecasts do not need to be derived using time series models, but are generated implicitly once the model has been fit to the historical data.

The remainder of this paper is organised as follows. Section 2 reviews the LC model and its extensions to multiple populations by Li & Lee (2005) and Kleinow (2015). Section 3 covers the basics of neural networks and shows how the LC model can be described as a neural network and fit using back-propagation. In particular, we then extend the LC model to the multi-population case using the deep neural network described in this paper. In Section 4, we fit the models to all data in the Human Mortality Database (HMD) (Wilmoth & Shkolnikov 2010) for the years

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

1950–1999, and forecast mortality rates for both genders up to the year 2016. We compare the in-sample fit of the neural network model to the variants of the LC model discussed above and the out-of-sample fit to the mortality rates in the years 2000–2016. In Section 5, we discuss strategies for improving the performance of the neural network approach. Finally, Section 6 concludes with a discussion and avenues for future research.

2. The LC Model and Extensions

The LC model defines the force of mortality as

$$\log\left(u_{x,t}\right) = a_x + b_x k_t \tag{1}$$

where $u_{x,t}$ is the force of mortality at age x in year t, a_x is the average log mortality at age x measured over the period in which the model is fit, b_x is the rate of change of the log mortality with time at age x and k_t is the time index for calendar year t. This model cannot be fit with the Generalised Linear Model (GLM) framework due to the multiplicative nature of the second term $b_x k_t$, which is comprised of two (latent) variables that must each be estimated from the data. Thus, two alternative approaches to fit the LC model have been developed. The original approach (Lee & Carter 1992) is to apply a SVD to the matrix of log mortality rates, centred by subtracting the average log mortality at each age from each row of the matrix. The first left and right vectors from the SVD provide values for b_x and k_t. A second approach (Brouhns *et al.* 2002) (illustrated in more detail in Currie (2016)) is to specify the LC model as a statistical model (on the assumption that log mortality rates are normally distributed, see also Remark 2.1). More specifically, a Generalised Non-linear Model (GNM) is used (see Turner & Firth (2007)), and then maximum likelihood method is applied to fit the model. The LC model is generally fit to historical mortality data, and the time index coefficients k_t are then forecast using a standard time series model to produce forecasts of future mortality rates.

Remark 2.1. The specification in (1) aims to model mortality rates directly. Another approach is to use Poisson rate regression to model counts of deaths, using the central exposed to risk as the exposure. For simplicity, we focus on the first case in this research, but note that our approach carries over to the case of Poisson rate regression.

Extending the LC model to multiple populations generally involves adding terms to (1) that refer to some combination of the pooled mortality of the entire population and the specific mortality of each individual population. For example, Li & Lee (2005) model mortality as

$$\log\left(u_{x,t}\right) = a_x^i + b_x k_t + b_x^i k_t^i \tag{2}$$

where b_x^i and k_t^i are the rate of change of the log mortality with time and the time index, respectively, both for population i, which could, for example, refer to males and females in the same country, or populations from different countries. Li & Lee (2005) refer to this model as the Augmented Common Factor (ACF) model, which is fit in three steps – firstly, the population-specific average mortality a_x^i is calculated and subtracted from the matrix of mortality rates; secondly, the change with time of the pooled population mortality $b_x k_t$ is estimated; and, finally, the change with time of the residual population-specific mortality $b_x^i k_t^i$ is estimated from the residual matrix $\mu_{x,t} - a_x^i - b_x k_t$.

A variation of this model is the CAE model of Kleinow (2015), which we define in a simplified manner as follows:

$$\log\left(u_{x,t}\right) = a_x^i + b_x k_t^i \tag{3}$$

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

where the rate of change of the log mortality with time is pooled (the "CAE"), but the time index is population specific. This model was fit in Kleinow (2015) using Common Principal Components and in Enchev *et al.* (2017) using maximum likelihood techniques.

More terms relating to the change of the log mortality with time could be added to any of the models (1), (2) or (3). For example, the full CAE model presented in Kleinow (2015) is

$$\log\left(u_{x,t}\right) = a_x^i + b(1)_x k(1)_t^i + b(2)_x k(2)_t^i$$

where terms in parentheses indicate that, for example, $b(1)_x$ is the first change of mortality component, and so on. Another variation is the two-tier ACF model of Chen & Millossovich (2018), who include both gender-specific and gender- and population-specific terms in their model. Also, the specifications of the ACF and CAE models could be combined by including common factors and common effects within the same model.

Thus, the LC model can be extended using several different model specifications, but, to date, no theory has emerged explaining why these specifications may be more or less optimal for a particular set of mortality rates (though, some partial explanation is found in Villegas *et al.* (2017)). Therefore, we conclude that the specification of extended LC models appears somewhat arbitrary and depends on the judgment of the modeller, leading us to automate this process in the next section. Also, the extended LC models are not flexible enough to fit mortality data that are dissimilar from each other; thus, Li & Lee (2005) and Kleinow (2015) are forced to choose mortality data from regions that, a priori, would seem to have similar mortality experience. On the other hand, we aim to design a model with sufficient flexibility to model all mortality rates in the HMD since 1950 simultaneously, and turn specifically to neural networks, which have a sufficiently high representational capacity for this task.

3. Representation Learning and Neural Networks

We approach the extension of the LC model in (1) from two perspectives. Firstly, the LC model can be seen as a regression model, where mortality rates (the target or dependent variable) are predicted using features (independent variables) that, in the case of the original LC model, summarise the mortality rates in an optimal way. In other words, the LC model does not directly use the original features of the mortality data, such as age and year, but relies on a Principal Components Regression to derive new features, b_x and k_t, from the historical data. Similarly, the ACF and CAE extensions of the LC model rely on specific combinations of features derived from the aggregate, as well as the population-specific, mortality data. To extend the LC model, instead of subjectively deriving features from the mortality data, we turn to the paradigm of representation learning (Bengio *et al.* 2013), which is an approach that allows algorithms automatically to design a set of features that are optimal for a particular task. In particular, we utilise deep neural networks, which consist of multiple layers of non-linear functions that are optimised to transform input features of a regression model (in this case, age, calendar year, gender and region) into new representations that are optimally predictive with respect to the target variable.

Secondly, it can be seen that the LC model has the following specific functional form:

$$\log\left(u_{x,t}\right) = g(x) + h(x)i(t)$$

where we set

$$g(x) = \begin{cases} a_1 & \text{for } x = 1, \\ a_2 & \text{for } x = 2, \\ \vdots & \\ a_\omega & \text{for } x = \omega \end{cases}$$

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

for ω being the maximum age considered; and the functions $h(x)$ and $i(t)$ are also discrete functions described by b_x and k_t, respectively, see (1). Here too, rather than relying on manually specified features of the extended LC models, and the specific functional form of the LC model, we instead rely on neural networks to learn the function $\log(u_{x,t})$ directly from the features of the mortality data, by using age, calendar year, gender and region as predictors in a neural network (although, we maintain the discrete formulation in the form of embedding layers, which will be discussed later in this section). Thus, we utilise neural networks as universal function approximators, and refer the reader to Chapter 5 in Wüthrich & Buser (2016) for more details on the universality theorems underlying this choice.

We define our neural network model in two steps. We aim to fit the model to all mortality rates since 1950 in the HMD, and, therefore, our feature space is comprised of year of death, age of last birthday before death, region and gender. We model the year of death as a numerical input to the neural network, which allows us to extrapolate mortality rates beyond the observed range. The region and gender features are categorical, and we choose to treat the age variable as categorical as well. We model these categorical variables using an embedding layer (Bengio *et al.* 2003) (see Section 3 in Richman (2018) for a review). This embedding layer maps each category in the categorical feature to a low dimensional vector, the parameters of which are learned when the model is fit. Thus, for example, for the region feature, and similarly for the other categorical variables, we consider the embedding

$$f(region) = \begin{cases} \mathbf{region}_1 & \text{for } region = region_1, \\ \mathbf{region}_2 & \text{for } region = region_2, \\ \vdots & \\ \mathbf{region}_n & \text{for } region = region_n \end{cases}$$

where we assume to have n regions denoted by $\mathrm{REG} = \{region_1, \ldots, region_n\}$, and where $f : \mathrm{REG} \to \mathbb{R}^d$ is a function that maps a particular region, say $region_1$, to a real-valued d-dimensional vector $\mathbf{region}_1$, with d being a hyper-parameter defining the dimension of each embedding layer.

Remark 3.1. We note that if the dimension d of the embedding vector is 1, then the embedding layer reduces to nothing more than the normal treatment of categorical variables, where every categorical label may have its own parameter. This fact is further exploited below to describe how the LC model and extensions thereof can be fit using back-propagation, see Remark 3.2 below.

Once embedding vectors have been defined for each categorical variable, these are concatenated into a single feature vector $\mathbf{feature}_{t,x,i,j} = (t, \mathbf{age}'_x, \mathbf{region}'_i, \mathbf{gender}'_j)'$ which is used as input to the neural network in order to predict the force of mortality in year t, at age x for region i and gender j.

The basic form of the neural network we propose is a deep feed-forward network given by the ridge functions

$$\begin{aligned} \mathbf{Z}^1 &= \sigma_0(\mathbf{c}_0 + B_0' \, \mathbf{feature}_{t,x,i,j}) && (4) \\ \mathbf{Z}^2 &= \sigma_1(\mathbf{c}_1 + B_1' \, \mathbf{Z}^1) && (5) \\ u_{x,t} &= \sigma_2(c_2 + B_2' \, \mathbf{Z}^2) && (6) \end{aligned}$$

where B_0, B_1 and B_2 are weight matrices, $\mathbf{c}_0$, $\mathbf{c}_1$ and c_2 are intercepts and σ_0, σ_1 and σ_2 are the (non-linear) activation functions of the neural network. Note that (4) has to be understood component-wise, and it describes a hidden layer. That is, $\mathbf{Z}^1$ is a vector of intermediate variables calculated by applying the non-linear activation function σ_0 to linear combinations of the input features with weight matrix B_0 and intercepts $\mathbf{c}_0$. After calculating another set of intermediate variables, $\mathbf{Z}^2$, mortality rates are calculated using these intermediate variables $\mathbf{Z}^2$ as explanatory variables in a

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

final regression model (6). Since the weights defining the hidden variables are learned during the optimisation of the network, optimal combinations of the input features can be learned by the network. The model has been structured to learn both a representation of the input data using embedding layers, which have sufficient capacity to incorporate the large number of observations in the HMD dataset, and interactions between these representations are learned at the next layers of the network, freeing us from specifying exactly how the region and gender terms interact with the other terms, and simplifying the model specification compared to the ACF and CAE models noted above.

The model in (4)–(6) has, to this point, been described in general terms. The specific choices that we make are choosing a dimension of $d = 5$ for each of the embedding layers in the feature vector $\textbf{feature}_{t,x,i,j}$, and 128 neurons in the intermediate layers of the network. The activation functions considered for the first two layers of the network are the Rectified Linear Unit (ReLU) (Nair & Hinton 2010) or the hyperbolic tangent function (tanh), and a sigmoid function σ_2 for the last layer. Between the intermediate layers, we add dropout layers (Hinton *et al.* 2012) to regularise the network, with the probability of each neuron being switched off set at $p = 0.05$. We test variations on this network with extra intermediate layers, as described in Section 4, and, in this case, also add batch normalisation layers (Ioffe & Szegedy 2015). To fit these models, we rely on the back-propagation algorithm, as implemented in the TensorFlow software package (Abadi *et al.* 2016), accessed via the Keras library in R, we refer to Chollet (2015). The R code for a variation of this model is provided in Listing 2 in Appendix A.

Remark 3.2. We present the original LC model (1) as a version of a network model similar to network (4)–(6). To this end, we consider a fixed gender j in a given region i, and, thus, the regression model only depends on the age feature x and the year feature t. Let $\textbf{year}_t \in \mathbb{R}$ denote the one-dimensional embedding vector ($d = 1$) of the year feature t, and let $\textbf{age}_x = (\textbf{age}(1)_x, \textbf{age}(2)_x)' \in \mathbb{R}^2$ denote the two-dimensional embedding vector ($d = 2$) of the age feature x. Remark that in the network model considered in (7), we interpret the two-dimensional age embedding as two one-dimensional embeddings. The original LC model is then received as

$$u(x, t) = \exp\left\{\textbf{age}(1)_x + \textbf{age}(2)_x\,\textbf{year}_t\right\} \tag{7}$$

This model is illustrated in Listing 1: On lines 4–6 we define the one-dimensional embedding vector for the year feature t, and on lines 8–14 we define the two-dimensional embedding vector for the age feature x. These two embeddings involve trainable parameters of the same dimensions as k_t, a_x and b_x in (1). Finally, on lines 16–25 we merge all terms to the required network architecture (7); note that this modelling part does not involve further trainable parameters.

Remark that the ACF and CAE models can be formulated in a similar manner.

Remark 3.3. We close this section by noting that the specification of the neural network could be extended to include other numerical or categorical features. For example, another embedding layer representing the geographical region of each country could be added. Furthermore, although we show later that the model specified above succeeds in capturing cohort effects to some extent, one might choose also to add an explicit cohort effect to the network. This could be added as another embedding layer; however, one would need then to derive manually values for the cohorts that are outside of the sample used to fit the network. Another option is to add a numerical input representing cohort to the network. We leave the implementation of these ideas for potential future research.

4. Fitting the Models and Results

To fit the models discussed, we divide the data of the HMD into training and test sets, defining the training set as the mortality rate observations at ages 0–99 occurring in the years before 2000, and

Listing 1: LC model as a network with embeddings.

```
Year <- layer_input(shape = c(1), dtype = ’int32’, name = ’Year’)
Age  <- layer_input(shape = c(1), dtype = ’int32’, name = ’Age’)

Year_embed = Year %>%
  layer_embedding(input_dim=50, output_dim=1, input_length=1, name = ’Year_embed’) %>%
  keras::layer_flatten()

Age_embed_1 = Age %>%
  layer_embedding(input_dim=100, output_dim=1, input_length=1, name = ’Age_embed_1’) %>%
  keras::layer_flatten()

Age_embed_2 = Age %>%
  layer_embedding(input_dim=100, output_dim=1, input_length=1, name = ’Age_embed_2’) %>%
  keras::layer_flatten()

one_init = initializer_ones()

Year_effect = list(Age_embed_2,Year_embed) %>% layer_multiply()

main_output = list(Age_embed_1, Year_effect) %>% layer_add() %>%
  layer_dense(1, kernel_initializer = one_init, use_bias = FALSE, trainable = FALSE,
              activation = "linear") %>%
  layer_lambda(function(x) exp(x), name = ’main_output’)

model <- keras_model(inputs = c(Year, Age), outputs = c(main_output))
```

the test set is chosen as the observations in the years 2000–2016. The models are fit only to those countries in the HMD that have at least 10 years of data before year 2000; all of the 41 countries considered are listed in Table A.1 in Appendix A (at the time of performing the analysis, mortality rates for Korea had not yet been added to the HMD). Of these 41 countries, Croatia (HRV) was excluded since data were only available after the year 2002, Germany (DEUTNP) was excluded as data were only available since 1990 (i.e. 1 year of data were missing in the training set) and Chile (CHL) was excluded since data were only available since 1992. Thus, 38 of the 41 countries are included in the analysis that follows, and, since observations are for both sexes, we aim to forecast 76 distinct sets of mortality rates.

In cases when particular mortality rates were recorded as zero or were missing, we impute the missing mortality rate using the average rate at that age across all countries, for that gender, in that year.

We first describe in the next subsection how the LC, ACF and CAE models are fit, and discuss the results of each method separately, before comparing these results to those of the deep neural network approach.

Throughout, we use the (out-of-sample) mean squared error (MSE) as the metric to compare the fit of the models. As discussed above, the LC model was originally fit using the SVD procedure to perform principal components analysis (PCA). It is well known that PCA can be expressed as an optimisation problem, in which the MSE between the original data, and the data reconstructed using an approximating linear sub-space is minimised, see, for example, Chapter 18.3 in Efron & Hastie (2016); thus, the use of the MSE criterion is consistent with the history of mortality forecasting. Furthermore, minimising the MSE is equivalent to maximising the likelihood of a Gaussian distribution (i.e. we assume that conditional on the mean produced by each mortality model, the forecast mortality rates are Gaussian), and, given the observed regularity of mortality rates, this distributional assumption appears reasonable. Therefore, in addition to comparing models using the MSE, we also use the MSE as the optimisation objective of the neural network models (although other functions, such as the mean absolute error could also be considered).

Table 1. Out-of-sample performance of three methods of fitting the Lee–Carter model; MSE values are multiplied by 10^4

	Model	Average MSE	Median MSE	Best performance
1	LC_GNM	7.19	4.51	5
2	LC_SVD	5.50	2.48	40
3	LC_BP	5.19	2.60	31

We note that other approaches to comparing mortality models have been proposed, for example, see Dowd *et al.* (2010) for a review of recent advances and new approaches for comparing ex-post density mortality forecasts; however, since our main aim in this work is to introduce a deep neural network for mortality forecasting, we refrain from giving the other approaches here.

4.1 Baseline models

The basic LC model was fit for each country and gender in the HMD separately, using mortality rates up to and including the year 1999. Following the literature, we fit the LC models by applying the SVD procedure (denoted LC_SVD in the following) and using GNMs, which were fit under the assumption that mortality rates follow a Gaussian distribution (LC_GNM). We found that in some cases, the GNM models did not converge, and in these cases we do not report the results of the GNM method. The LC models were also fit via back-propagation (LC_BP), using the code in Listing 1 to define the network in Keras and using the RMSProp optimiser to minimise the in-sample MSE of the predicted mortality rates before year 2000. In this case, separate parameters were fit for each country and gender, but the LC models were fit jointly to the entire dataset at once. Mortality rates in the years after 1999 were forecast, for each of the three LC models, by extrapolating k_t using a random walk with drift, as determined by the random walk function in the Forecast package in R (Hyndman *et al.* 2015).

The out-of-sample performance of these LC models is shown in Table 1. A model is defined as having the "Best Performance" when it achieves the lowest MSE on the test set for a particular country and gender. We also report the overall mean and median of the MSEs on the test set for each model.

We observe that the models fit with GNMs did not perform well out-of-sample, beating those fit with SVD and back-propagation only in 5 of 76 instances. This may be because the combination of the relatively inflexible LC model specification together with the assumption that mortality rates are distributed as Gaussian random variables with constant variance over all ages is too restrictive for the GNM fitting algorithm to converge; for example, at older ages with fewer lives exposed to risk we expect a higher variance of the mortality rates than at younger ages. Of the models shown in Table 1, those fit with SVD have a lower median MSE and outperform those fit with back-propagation, although we note that the LC models fit with back-propagation produce fewer extreme outliers compared to those fit with SVD. In what follows, we select the SVD model as the benchmark forecast against which we compare the ACF and CAE models.

Next, we fit the ACF and CAE models. Therefore, we grouped the countries in the HMD subjectively into geographical regions based on proximity, as shown in Table A.1 that appears in Appendix A. These regions are then used to describe and derive a common mortality trend.

The ACF model was fit according to the method in Li & Lee (2005), as well as using back-propagation. When using the method of Li & Lee (2005), the ACF model was fit for each region and gender separately, as follows. The average (log) mortality rate for each region and age was calculated and subtracted from the average regional mortality rates in each year, producing a matrix of centred regional mortality rates, on which the time effect (k_t) of the original LC model was fit by applying the SVD procedure. These regional LC models were used to produce first mortality rates after 1999 by forecasting using random walks with drift (denoted ACF_SVD_region in the following); note that these forecasts do not consider country-specific information, but only

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

Table 2. Out-of-sample performance of the Augmented Common Factor (ACF) model; MSE values are multiplied by 10^4

	Model	Average MSE	Median MSE	Best performance
1	LC_SVD	5.50	2.48	19
2	ACF_SVD_region	3.46	2.50	36
3	ACF_SVD_country	7.30	4.77	9
4	ACF_BP	6.12	3.00	12

information at the regional level according to Table A.1. Then, the fitted regional time effects ($b_x k_t$) were subtracted from the mortality rates in each country, producing a matrix of residuals, on which a second set of country-level LC models were fit, producing a country time effect (k_t^i), which was forecast again using a random walk with drift (Forecast package in R; Hyndman *et al.* 2015). Thus, two sets of forecasts were produced for each country using the ACF model, one at a regional level (ACF_SVD_region) and one adding a country-specific effect to the regional forecast (ACF_SVD_country).

Next, the ACF model was fit using back-propagation, by defining the network version of the ACF model in Keras and minimising the in-sample MSE of the predicted mortality rates using the RMSProp optimiser (ACF_BP). In this case, the regional and country effects were optimised jointly (i.e. the two-step procedure of Li & Lee (2005) was not required), and, in a similar way to the LC model, the ACF model was fit for all regions and genders simultaneously. The results are presented in Table 2.

Surprisingly, amongst the ACF models, the relatively simple forecasts produced using mortality rates at a regional level, which ignore country-specific information, significantly outperform the rest of the models, including the original LC model. Comparing the MSE of the regional forecasts to the LC model, it emerges that the LC models have a lower median MSE, but the regional forecasts have a much lower average MSE, implying that, for some countries where the original LC model does not perform well (i.e. produces a high out-of-sample MSE), the regional forecasts are more accurate, as shown in Figure 1. Similarly, the regional forecasts outperform both of the full ACF models (i.e. those incorporating country-specific information), although the ACF model fit with back-propagation appears to be better than the two-step model fit using SVD. Since these results indicate that only in some instances does the country-specific information incorporated into the ACF models increase predictive power, a credibility mixture between regional and country forecasts might produce more robust results than those shown here, but we do not pursue this further. On the other hand, the good performance of the regional approach may indicate that the subjective choice of the regions has been done in a reasonable way, providing bigger volumes on a regional level compared to the country level, and in turn giving more robust models.

Finally, the CAE models were fit. Firstly, a simple optimisation of the regional effects derived for the ACF model was attempted. As mentioned above, the SVD procedure was applied to the matrix of centred average regional mortality rates. The first and second left vectors from this SVD decomposition were used to provide values for the regional rate of change of mortality with time, $b(1)_x$ and $b(2)_x$. Then, country-specific time effects, $k(1)_t^i$ and $k(2)_t^i$, were derived using a GNM model, where the regional effects (as well as the country-specific average mortality rates) were held constant and entered the model as an offset. Mortality forecasts were derived by projecting the country-specific time effects using a random walk with drift. In the following, we report on the results of a CAE model with a single time component (CAE_SVD) and two time components (CAE2_SVD). The CAE models were also fit using back-propagation for all regions and genders simultaneously, in a similar fashion to that described for the ACF model (CAE_BP).

A comparison of the CAE models is shown in Table 3. We note that the CAE model with two time components did not produce reasonable forecasts of mortality for males in Eastern Europe, and these results were excluded from the table. In other words, the CAE2_SVD is less robust than

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

Table 3. Out-of-sample performance of the Common Age Effect (CAE) model; MSE values are multiplied by 10^4

	Model	Average MSE	Median MSE	Best performance
1	LC_SVD	5.50	2.48	33
2	CAE_SVD	4.76	2.35	13
3	CAE2_SVD	12.01	1.79	14
4	CAE2_BP	5.59	3.46	16

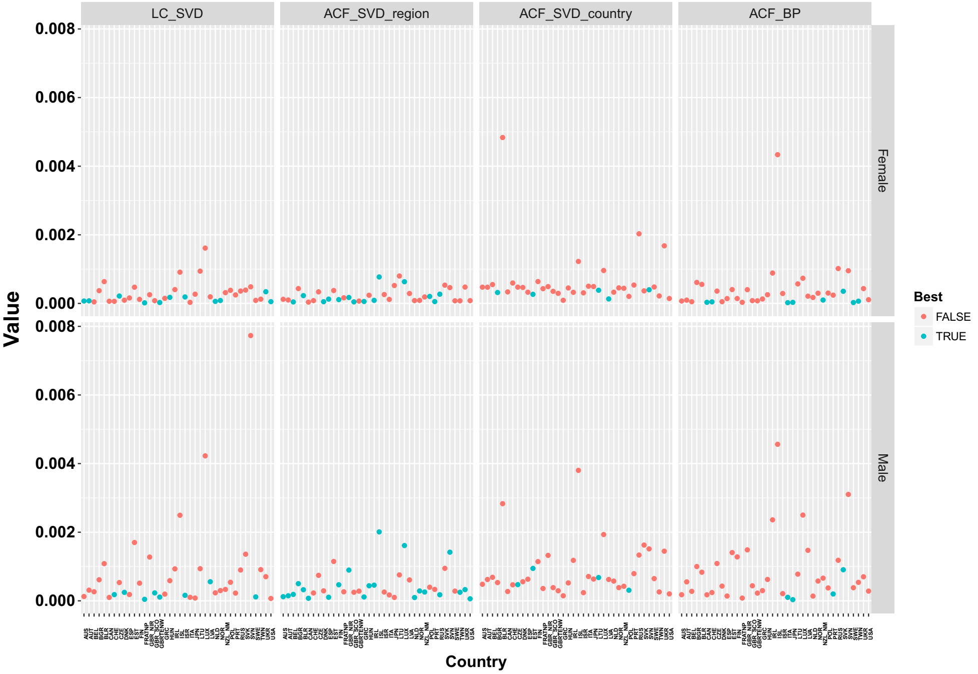

Figure 1. Out-of-sample MSE for each gender (female: top row; male: bottom row) and country in the HMD for the LC and ACF models (LC_SVD: first column; ACF_SVD_region: second column; ACF_SVD_country: third column; ACF_BP: fourth column); the dots give the corresponding errors for the countries (with "TRUE" showing the best models in blue colour).

the other models, which produced reasonable results in all circumstances. This can also be seen from the relatively high average MSE of the CAE2_SVD model in Table 3 which is being influenced by cases of very poor performance. However, we also note that the median MSE is the lowest of the models indicating that this model predicts mortality well in some instances. Unexpectedly, though, the baseline LC models beat the CAE models 33 of 76 times, indicating that this model formulation is not particularly competitive.

One might potentially challenge the modelling in this section, where random walks with drift have been used for forecasting country-specific effects, whereas the literature suggests the use of a random walk without drift or an AR(1) process, see, for example, Li & Lee (2005), who justify using these latter models on the grounds that country-specific effects should become less significant than regional effects in the long run (since it could be expected that medical technology and other advances promoting longevity should become available in all countries after several years). To investigate this, in an additional step, we have modelled the country-specific effects using a random walk without drift and found that the performance of the models discussed in this section was worse (compared to the neural network we discuss in the next section) than if a drift was included. We show these results in Appendix B. Furthermore, modelling using an AR(1) process was also attempted, but it was found that some of the time series were non-stationary and fitting

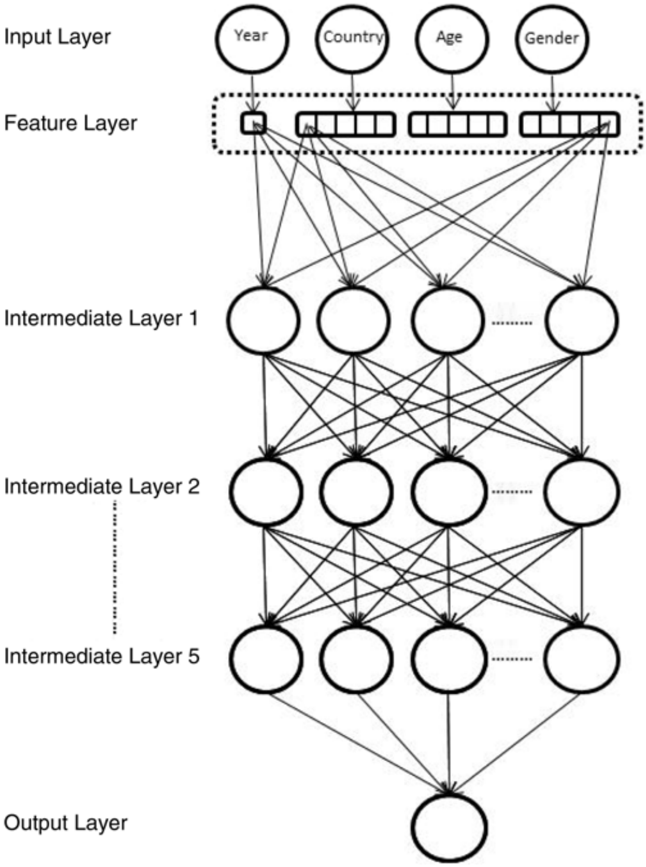

Figure 2. Five layer deep neural network depicted graphically. The feature layer consists of five dimensional embeddings for each categorical variable, and a single dimension for Year, which is the only numerical variable. Note that for clarity, only some of the connections between the feature and intermediate layers have been shown; also, dropout and batch normalisation layers are not shown.

the AR(1) process failed. Finally, we note that the random walks used in this section are uncorrelated, and since we only aim to produce best estimate forecasts (i.e. density forecasts are not addressed in this paper), we do not consider correlated random walks further.

To conclude, for comparison to the neural network models, we pick the best performing amongst the various LC, ACF and CAE models.

4.2 Deep neural network model

In this section, we describe the fitting of the neural networks in more detail. In total, six networks were fit to the HMD data. Two networks following equations (4)–(6) consist of two intermediate layers, the first of these network models using ReLU activations (DEEP1, in what follows), and the second one using tanh activations (DEEP2). The next two neural network models add three more intermediate layers to each of these first two networks, bringing the number of intermediate layers to five (respectively DEEP3 (ReLU) and DEEP4 (tanh)). We depict these deeper neural networks in Figure 2. Finally, we add a so-called "skip" connection to these deeper networks, by connecting the feature layer directly to the last intermediate layer, as well as to the first intermediate layer (respectively DEEP5 (ReLU) and DEEP6 (tanh)). Skip connections, in various configurations, have been used in the computer vision literature to train very deep neural networks successfully, see for example, He *et al.* (2015), Huang *et al.* (2016), and are thought to resolve the vanishing gradient problem that affects deeper networks (Huang *et al.* 2016) by shortening the path that the back-propagation algorithm needs to follow to get back to the first layer of the network. The code for fitting the DEEP6 network in Keras appears in the appendix in Listing 2.

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

Listing 2: Deep neural network model with tanh activations and a skip connection (DEEP6).

```
Year <- layer_input(shape = c(1), dtype = 'float32', name = 'Year')
Age <- layer_input(shape = c(1), dtype = 'int32', name = 'Age')
Country <- layer_input(shape = c(1), dtype = 'int32', name = 'Country')
Gener <- layer_input(shape = c(1), dtype = 'int32', name = 'Gender')

Age_embed = Age %>%
  layer_embedding(input_dim = 100,output_dim = 5,input_length = 1,name = 'Age_embed') %>%
  keras::layer_flatten()

Gender_embed = Gender %>%
  layer_embedding(input_dim = 2,output_dim = 5,input_length = 1,
     name = 'Gender_embed') %>%
  keras::layer_flatten()

Country_embed = Country %>%
  layer_embedding(input_dim = 41,output_dim = 5,input_length = 1,
     name = 'Country_embed') %>%
  keras::layer_flatten()

features <- layer_concatenate(list(Year, Age_embed, Gender_embed, Country_embed))

middle = features %>%
  layer_dense(units = 128, activation = 'tanh') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.05) %>%

  layer_dense(units = 128, activation = 'tanh') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.05) %>%

  layer_dense(units = 128, activation = 'tanh') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.05) %>%

  layer_dense(units = 128, activation = 'tanh') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.05)

main_output = layer_concatenate(list(features middle)) %>%
  layer_dense(units = 128, activation = 'tanh') %>%
  layer_batch_normalization() %>%
  layer_dropout(0.05) %>%
  layer_dense(units = 1, activation = 'sigmoid', name = 'main_output')

model <- keras_model(inputs = c(Year, Age, Country, Gender), outputs = c(main_output))
```

When fitting the neural networks, the Adam optimiser (Kingma & Ba 2014) was used, with the parameter values taken at the defaults. The models were each fit for 50 epochs, and the model with the best performance during these 50 epochs, as measured by the validation set, was used. A 5% random sample of the training set was used as a validation set; in other words, the network was fit on 95% of the training set data, comprising 325,090 samples, and performance was assessed on 5% of the training set, comprising 17,110 samples.

Remark 4.1. The neural network is thus fit on slightly less data than the LC, ACF and CAE models and is, therefore, not on entirely equal footing. Although the validation set could be excluded for the models fit using GNMs and back-propagation, it cannot be excluded when fitting the models using SVD (which cannot be applied in the presence of missing values), and, therefore, it was decided to include the validation set for all of the models except the neural network approach.

Since several different network architectures can be fit to the HMD data, it is necessary to choose an optimal architecture against which to test the models mentioned in the previous section; however, as described next, it is not straightforward to choose the optimal model.

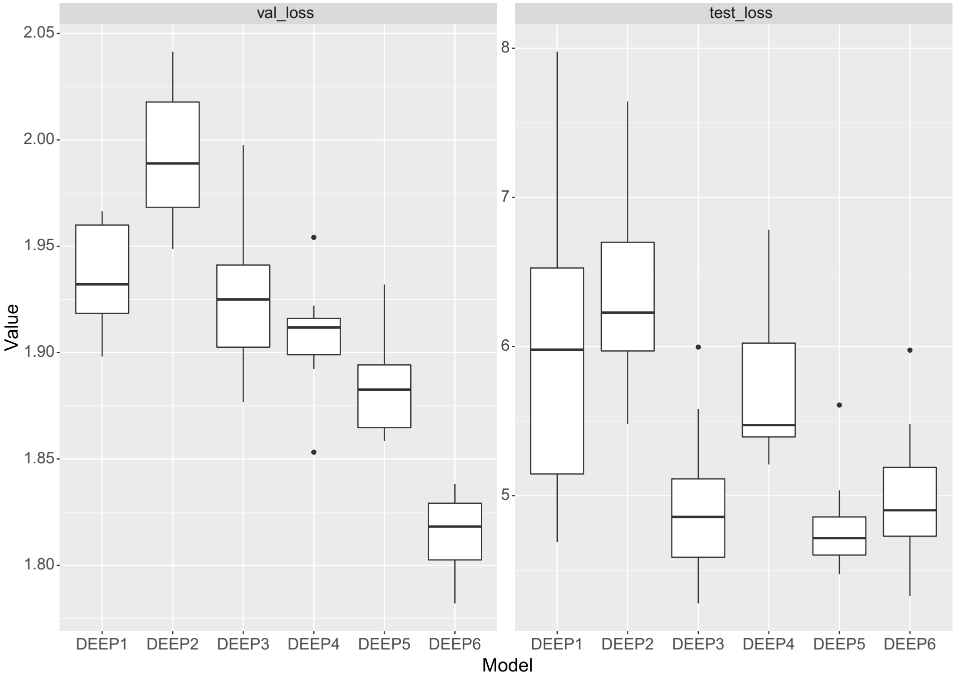

Figure 3. Boxplot of the results of round 1 of training the neural networks described in text for 10 runs on data from 1950 to 1990: left shows validation losses, right shows losses on test data from 1991 to 1999.

Firstly, we note that the results of training a neural network are somewhat variable (which is due to the initial value of the optimisation algorithm, the random selection of batches of training data to calculate the gradients used in back-propagation, as well as due to dropout, which is applied at random to the neurons of each network), and, therefore, both the in-sample and out-of-sample performance of a neural network can vary somewhat between training attempts. While we note that the reproducibility of the results could be guaranteed by setting the seed value of the random number generator used when fitting the model, there is the possibility that fitting a neural network only once will produce results that are not representative of the model's average performance (see Figure 3 for an illustration of the potential variability of the results run with 10 different seeds). Thus, in what follows, we fit each network 10 times and take the average MSE to indicate the likely performance of each architecture.

Secondly, the training, validation and test sets need to be defined. Since we seek the model that best forecasts the HMD data in the years from 2000 onwards, and we have access to the actual observations in these years, one might propose to use the mortality rates in these years as the test set. However, in a realistic forecasting scenario, the data which are being forecast have not yet been observed, and this approach will not work. Therefore, the training set must be used to determine an optimal model. We approach this by fitting the networks in two rounds, as described next.

In round 1, we split the training set (consisting of data from 1950 to 1999) into a second training set (consisting of data from 1950 to 1990) and test set (consisting of data from 1991 to 1999). We note that in this second training set, three countries have less than 10 years of data (Greece with 9 years of data and Israel and Slovenia each with 7 years of data), but the available data should be sufficient for the purpose of determining the performance of each architecture. As just mentioned, a 5% sample of the training data was taken as a validation set to track when the network has fit the training data appropriately, that is, to indicate over-fitting to the training data. All six networks were fit on the training data, and then the forecasting performance was assessed

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

Table 4. Round 1 of fitting the deep neural networks (1950–1990): validation and test set MSEs of the six deep neural network architectures described in the text is shown, averaged over ten training runs; MSE values are multiplied by 10^4

	Model	val_loss	test_loss
1	DEEP1	1.94	6.01
2	DEEP2	1.99	6.34
3	DEEP3	1.93	4.96
4	DEEP4	1.91	5.73
5	DEEP5	1.88	4.79
6	DEEP6	1.81	5.00

Table 5. Round 2 of fitting the deep neural networks (1950–1999): validation and test set MSEs of the six deep neural network architectures described in the text are shown; MSE values are multiplied by 10^4

	Model	val_loss	test_loss
1	DEEP1	2.33	3.29
2	DEEP2	2.31	4.73
3	DEEP3	2.29	3.30
4	DEEP4	2.26	3.07
5	DEEP5	2.23	2.79
6	DEEP6	2.16	2.64

on the test set from 1991 to 1999. These results are shown in Table 4 and Figure 3, where it can be seen that both DEEP5 and DEEP6 perform well. The optimal forecasting (i.e. out-of-sample and out-of-time) performance, as measured by the test set loss, is achieved by DEEP5, which is the five layer network with ReLU activations and a skip connection, which somewhat improves on the performance of DEEP3, which is a ReLU network without skip connection. However, the best validation performance, by a substantial margin, is the DEEP6 network which is the five layer network with tanh activations and a skip connection, the performance of which is improved dramatically by the skip connection. As already mentioned, we finally note that the results shown in Table 4 represent the average of 10 runs of a random training process, and it is possible that training only a single network will produce less optimal performance, as can be seen by the outliers in Figure 3.

In a realistic forecasting scenario, we would now refit only the best neural network architecture (DEEP5) to all of the training data up to year 2000 (unless we wished to use an ensemble of models, in which case we would use the results of more than one model); however, we wish to confirm in this work that the proposed model selection strategy is valid. Therefore, in round 2, we fit all six networks on the full training set (i.e. on the data from 1950 to 1999) and assess the forecasting performance on the test set (years 2000 onwards). These results are shown in Table 5 and Figure 4. The best model in round 2, as measured by the MSE is DEEP6, followed closely by DEEP5. Thus, the results show that the round 1 testing was roughly indicative of the optimal model architecture, leading us to select the second best model, but, unfortunately, not leading us to select DEEP6, which has slightly more optimal performance.

Table 6 shows the results of comparing the DEEP5 neural network results to the rest of the models fit in this section. Note that, for this comparison, we did not average the results of many training runs, but simply selected the results of the first training run. By a wide margin, the best performing model is the deep neural network, which is the optimal model based on the three metrics considered in this section, and produces the best out-of-time forecasts 51 of 76 times (DEEP6 would increase this to 54 of 76 times). In other words, we conclude that for the purposes

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

Table 6. Comparison of the deep neural network approach with the best models discussed in this section; MSE values are multiplied by 10^4

	Model	Average MSE	Median MSE	Best performance
1	LC_SVD	5.50	2.48	7
2	LC_ACF_region	3.46	2.50	10
3	ACF_BP	6.12	3.00	4
4	CAE_BP	5.59	3.46	4
5	DEEP	2.68	1.38	51

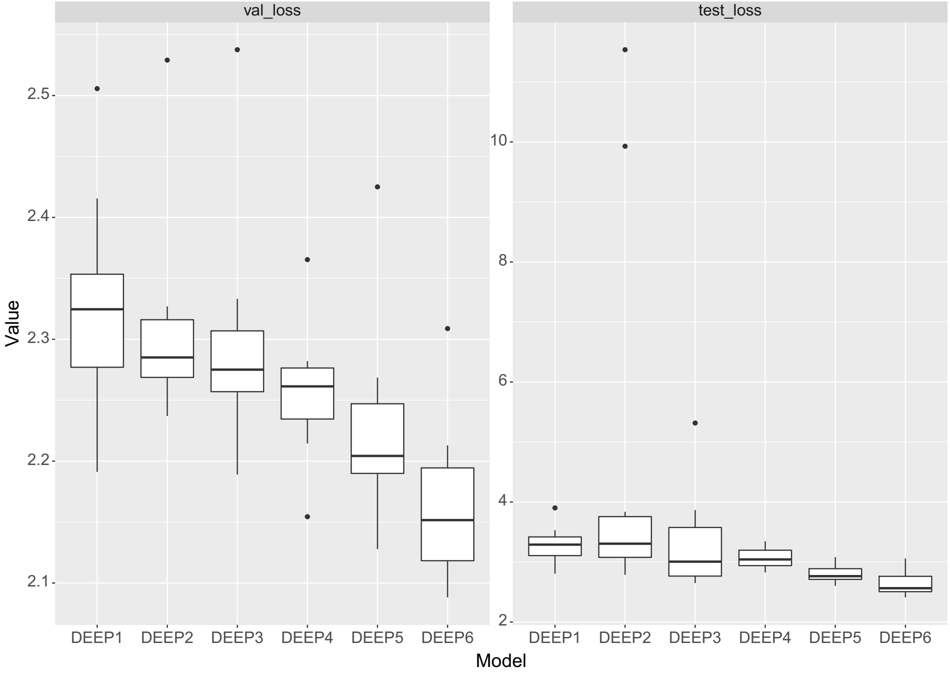

Figure 4. Boxplot of the results of round 2 of training the neural networks described in text for 10 runs on data from 1950 to 1999: left shows validation losses, right shows losses on test data from 2000 to 2016.

of large scale mortality forecasting, deep neural network architectures dramatically outperform traditional single and multi-population forecasting models.

We refer the reader to Appendix B where, as noted above, the deep neural network is compared against the ACF and CAE models, which were modified to use a random walk without a drift term to forecast the country-specific effects. There we conclude similarly that the deep neural network dramatically outperforms the other models, and, indeed, the choice of random walk for forecasting country-specific effects makes little difference to the main conclusion of this section.

5. Discussion and Improving the Neural Network Approach

The previous section has shown that the deep neural network extension of the LC model by far outperforms all of the other models considered in this paper, for the purpose of producing 15-year ahead forecasts. Figure 5 shows the average residuals produced by each of these models, for the ages 0–99 in the period 2000–2016. It can be observed that amongst the models, the deep neural network achieves the smallest residuals, with the fit for females appearing to be better than the fit for males, which is an observation that applies equally to the country and regional LC

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

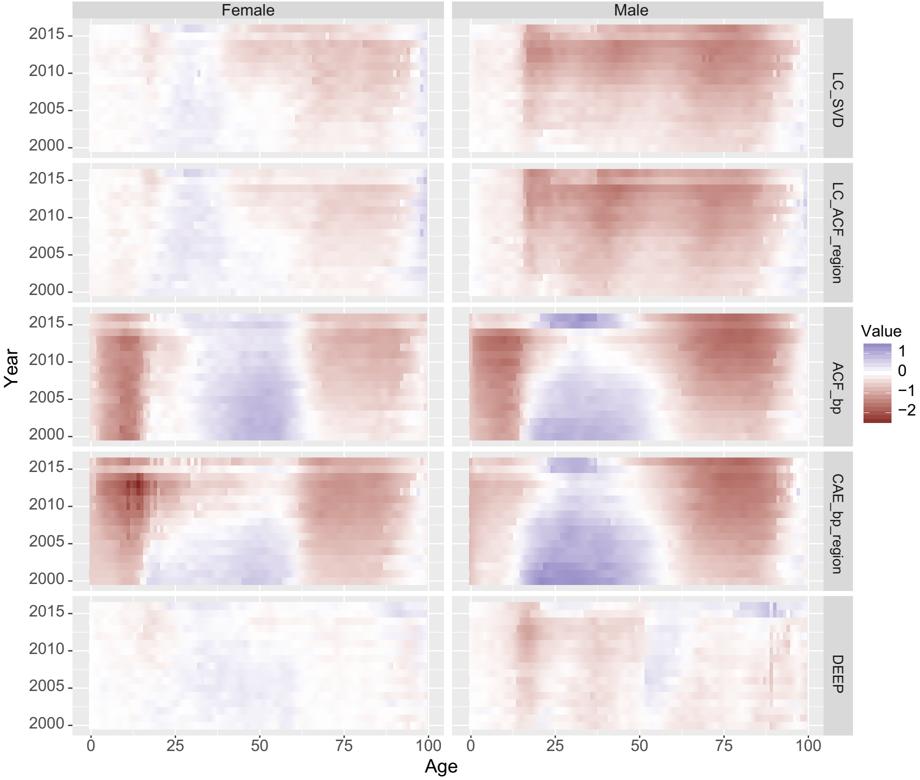

Figure 5. Residuals produced by each of the models, for each gender, year and age separately, averaged over the countries in the HMD.

models. The most significant residuals produced by the neural network are for male teenage and middle-age mortality, where the model predicts rates that are too high, and an emerging trend of predicting rates that are too low at the older ages, in the most recent years, which is in line with recent observations across many countries, see for example, Office for National Statistics (2018). The LC models appear to predict rates that are too low at all ages older than the teenage years, while the ACF and CAE models display a different pattern of residuals that suggest that the model specification has failed to capture the evolution of mortality rates appropriately.

A cohort effect can be observed in the residuals for all of the models, besides for the neural network, suggesting that part of what the neural network has learned is interactions between Year and Age, which allow the network to model the cohort effect. This observation is confirmed in Figure 6, which displays the average residual for each cohort. The residuals for the neural network are smaller than those produced by the other models and display less of a pattern, suggesting that part of the reason for the out-performance of the neural network is that cohort effects are captured automatically. Nonetheless, some patterns can be observed in these residuals, suggesting that including a cohort effect explicitly may improve the model even further; we refer the reader to the closing remarks of Section 3 for a discussion of how this might be achieved.

To examine the performance of the neural network in more detail, in Figure 7 the learned parameters of the age embedding are shown, after reducing the dimensionality of the embedding from five dimensions to two dimensions, using PCA. These values do not have an absolute interpretation, since the intermediate layers of the network shift and scale the embedding value

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

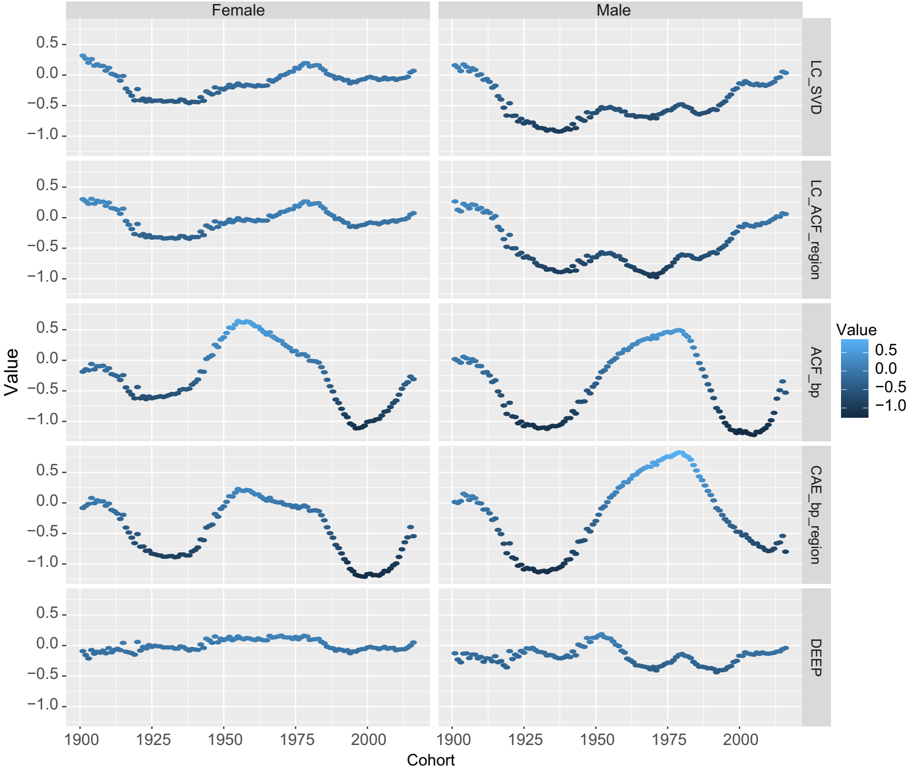

Figure 6. Residuals produced by each of the models, for each gender and cohort separately, averaged over the countries in the HMD.

(indeed, the values shown are multiplied by −1 to produce the familiar orientation of a life table); however, the values of the embeddings relative to each other are interpretable. The more significant component of the embedding has the familiar shape of a life table and is comparable to the a_x component of the LC model, indicating that the network has learned the overall relationship of mortality rates to each other from the data. The second component appears mainly to capture the relationship between middle age and old age mortality, with mortality rates increasing more quickly with age, as middle age mortality falls, as well as several other relationships between infant, teenage and old age mortality. The learned parameters of the embeddings for age and gender are less interpretable, probably because these parameters only have a meaning in the context of the deeper layers of the neural network.

6. Conclusion and Outlook

This paper has shown how the LC model can be extended to work reliably with multiple populations using deep neural networks. Rather than attempting to specify exactly how the inputs to the model determine mortality, as in the case of the original LC model and its extensions to multiple populations, the deep neural network successfully learns these relationships and projects mortality for all countries in the HMD with a high degree of accuracy.

Since many neural network architectures may be chosen, in this study we use a model selection procedure to choose an optimal architecture that eventually proves to be close to the optimal

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

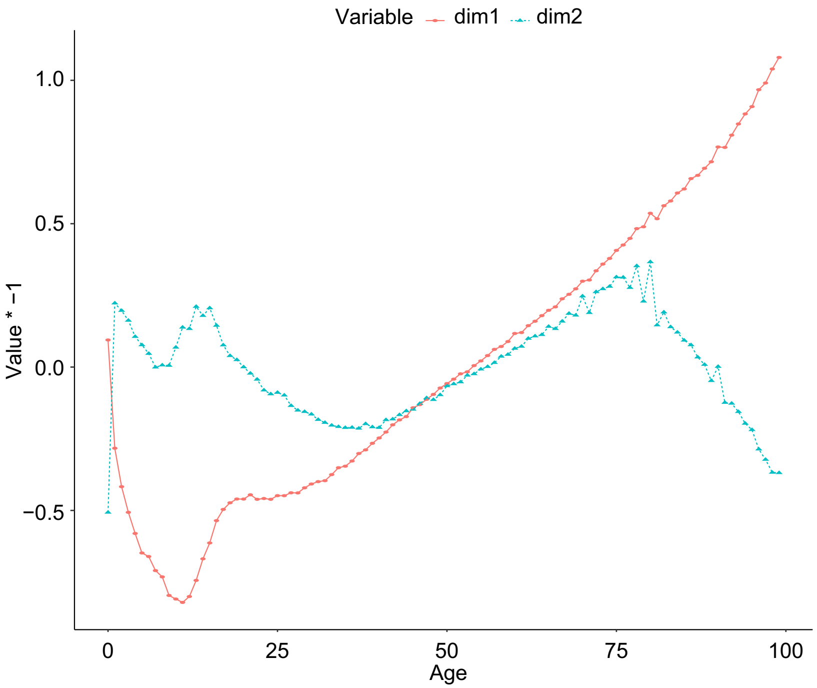

Figure 7. Parameters of the age embedding in the deep neural network, with the dimensionality reduced from 5 to 2 using Principal Components Analysis.

model, amongst those tested. Future research should examine the model selection process in more detail, and it may be the case that a less heuristic selection procedure can be proposed. We also note that an extensive search over neural architectures has not been performed in this study, and only several models were tested. A more comprehensive search over architectures, including models with more layers, difference configurations of skip connections (perhaps following more closely the architectures in He *et al.* (2015) and Huang *et al.* (2016)) and different hyperparameter settings for dropout and learning rates, may produce results that are more optimal than those presented here.

Another avenue for improving forecasting ability is the ensembling together of several neural networks. Although we do not report these results in detail, forecasting rates as the average of the predictions of the DEEP5 and DEEP6 networks produces the best forecasts in 56 of 76 countries, which is better than the results of the DEEP5 and DEEP6 models stand alone. A similar approach would average the results over several of the same networks, which would help to reduce some of the variability in the models that we note in section 4.2 (see, for example, Guo & Berkhahn (2015) who average the results of 5 of the same neural network architecture in an application of deep networks to structured data).

Other improvements to the model are the inclusion of an explicit cohort effect, as noted in Section 5 and including regional effects within the neural model, which were shown in Section 4.1 to be an important feature within mortality forecasting models. Another useful extension is to model mortality rates in smaller geographic areas, such as states and provinces.

Although we have focused on the LC model in this study, the LC model is not fundamental to the approach that has been proposed, and in future research, one should extend this to other mortality models using neural networks. Finally, an important issue for actuaries and demographers which we have not addressed is the uncertainty of the predictions of the neural network model, and future research should consider how this may be derived.

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. & Zhang, X. (2016). TensorFlow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)* (pp. 265–283).

Allaire, J.J. & Chollet, F. (2018). R interface to Keras. *RStudio, Google*.

Bengio, Y., Courville, A. & Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **35**(8), 1798–1828.

Bengio, Y., Ducharme, R., Vincent, P. & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, **3**(2), 1137–1155.

Brouhns, N., Denuit, M. & Vermunt, J.K. (2002). A Poisson log-bilinear regression approach to the construction of projected lifetables. *Insurance: Mathematics and Economics,* **31**(3), 373–393.

Cairns, A.J.G., Blake, D. & Dowd, K. (2006). A two-factor model for stochastic mortality with parameter uncertainty: theory and calibration. *Journal of Risk & Insurance,* **73**(4), 687–718.

Chen, R.Y. & Millossovich, P. (2018). Sex-specific mortality forecasting for UK countries: a coherent approach. *European Actuarial Journal,* **8**(1), 69–95.

Chollet, F. (2015). Keras: the Python deep learning library.

Currie, I.D. (2016). On fitting generalized linear and non-linear models of mortality. *Scandinavian Actuarial Journal*, **2016**(4), 356–383.

Danesi, I.L., Haberman, S. & Millossovich, P. (2015). Forecasting mortality in sub-populations using Lee-Carter type models: a comparison. *Insurance: Mathematics and Economics,* **62**, 151–161.

Dowd, K., Cairns, A.J., Blake, D., Coughlan, G.D., Epstein, D. & Khalaf-Allah, M. (2010). Backtesting stochastic mortality models: an ex post evaluation of multiperiod-ahead density forecasts. *North American Actuarial Journal*, **14**(3), 281–298.

Enchev, V., Kleinow, T. & Cairns, A.J.G. (2017). Multi-population mortality models: fitting, forecasting and comparisons. *Scandinavian Actuarial Journal*, **2017**(4), 319–342.

Efron, B. & Hastie, T. (2016). *Computer Age Statistical Inference* (Vol. 5). Cambridge, United Kingdom, Cambridge University Press.

Guo, C. & Berkhahn, F. (2015). Entity embeddings of categorical variables. *arXiv*, arXiv:1604.06737.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*, arXiv:1207.0580.

Huang, G., Liu, Z. & Weinberger, K.Q. (2016). Densely connected convolutional networks. *CoRR*, abs/1608.06993.

Hyndman, R.J., Athanasopoulos, G., Razbash, S., Schmidt, D., Zhou, Z., Khan, Y., Bergmeir, C. & Wang, E. (2015). Forecast: forecasting functions for time series and linear models. *R package*.

Ioffe, S. & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Kingma, D.P. & Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv*, arXiv:1412.6980.

Kleinow, T. (2015). A common age effect model for the mortality of multiple populations. *Insurance: Mathematics and Economics,* **63**, 147–152.

Lee, R.D. & Carter, L.R. (1992). Modeling and forecasting US mortality. *Journal of the American Statistical Association,* **87**(419), 659–671.

Li, N. & Lee, R. (2005). Coherent mortality forecasts for a group of populations: an extension of the Lee-Carter method. *Demography,* **42**(3), 575–594.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Conference Proceedings of Neural Information Processing Systems*, Electronic Publisher. (pp. 3111–3119).

Nair, V. & Hinton, G. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning* (pp. 807–814).

Office for National Statistics, United Kingdom (2018). Changing trends in mortality: an international comparison: 2000 to 2016.

Richman, R. (2018). AI in actuarial science. *SSRN Manuscript* ID 3218082. Version July 24, 2018.

Rumelhart, D., Hinton, G. & Williams, R. (1986). Learning representations by back-propagating errors. *Nature,* **323**(6088), 533.

Turner, H. & Firth, D. (2007). Generalized nonlinear models in R: an overview of the gnm package. *R package*.

Villegas, A.M., Haberman, S., Kaishev, V.K. & Millossovich, P. (2017). A comparative study of two-population models for the assessment of basis risk in longevity hedges. *ASTIN Bulletin*, **47**(3), 631–679.

Wilmoth, J.R. & Shkolnikov, V. (2010). Human Mortality Database. University of California.

Wüthrich, M.V. & Buser, C. (2016). Data analytics for non-life insurance pricing. *SSRN Manuscript* ID 2870308. Version February 5, 2019.

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

Appendix A. Allocation of Countries to Regions

Table A.1. Allocation of the countries in the HMD to regions

	Region	Country
1	America and Australia	AUS
2	Western Europe	AUT
3	Western Europe	BEL
4	Eastern Europe	BGR
5	Eastern Europe	BLR
6	America and Australia	CAN
7	Western Europe	CHE
8	America and Australia	CHL
9	Eastern Europe	CZE
10	Western Europe	DEUTNP
11	Scandinavia	DNK
12	Southern Europe	ESP
13	Eastern Europe	EST
14	Scandinavia	FIN
15	Western Europe	FRATNP
16	Great Britain	GBRTENW
17	Great Britain	GBR_NIR
18	Great Britain	GBR_SCO
19	Southern Europe	GRC
20	Southern Europe	HRV
21	Eastern Europe	HUN
22	Great Britain	IRL
23	Scandinavia	ISL
24	Southern Europe	ISR
25	Southern Europe	ITA
26	Japan and Taiwan	JPN
27	Eastern Europe	LTU
28	Western Europe	LUX
29	Eastern Europe	LVA
30	Western Europe	NLD
31	Scandinavia	NOR
32	America and Australia	NZL_NM
33	Eastern Europe	POL
34	Southern Europe	PRT
35	Eastern Europe	RUS
36	Eastern Europe	SVK
37	Eastern Europe	SVN
38	Scandinavia	SWE
39	Japan and Taiwan	TWN
40	Eastern Europe	UKR
41	America and Australia	USA

https://doi.org/10.1017/S1748499519000071 Published online by Cambridge University Press

Appendix B. Forecasts using a Random Walk without Drift

In this appendix, we show the results of forecasting the country-specific time effects of the ACF and CAE models using a random walk without a drift term.

As shown in Table B.1, we find that of the ACF models, the ACF_SVD_country model performs better than when a drift term was allowed for; therefore, we select this model instead of the ACF_BP for comparison to the neural network, but we note firstly that the best of these models is still the ACF_SVD_region model (i.e. the model that does not allow for country-specific effects), and secondly that the ACF_BP model performs slightly worse with this choice.

Table B.2 shows that amongst the CAE models, the performance of the CAE2_BP model is enhanced slightly by not allowing for a drift term, but the CA_SVD model performs slightly worse.

Finally, comparing these models to the deep neural network in Table B.3, we see that the conclusion reached above, that the deep neural network outperforms all of the other models, remains the same, and, indeed, the CAE model performs much worse in this comparison than above. This is also shown by the median MSE of this model, which has worsened compared to when a drift term was allowed for, see Table 6 above.

Table B.1. Out-of-sample performance of the Augmented Common Factor (ACF) model where country-specific effects have been fit using a random walk without drift; MSE values are multiplied by 10^4

	Model	Average MSE	Median MSE	Best performance
1	LC_SVD	5.50	2.48	20
2	ACF_SVD_region	3.46	2.50	33
3	ACF_SVD_country	6.24	4.64	12
4	ACF_BP	6.10	3.37	11

Table B.2. Out-of-sample performance of the Common Age Effect (CAE) model where country-specific effects have been fit using a random walk without drift; MSE values are multiplied by 10^4

	Model	Average MSE	Median MSE	Best performance
1	LC_SVD	5.50	2.48	34
2	CAE_SVD	4.93	3.10	11
3	CAE2_SVD	10.84	2.51	14
4	CAE2_BP	5.41	3.61	17

Table B.3. Comparison of the deep neural network approach with the best models discussed in this Appendix; MSE values are multiplied by 10^4

	Model	Average MSE	Median MSE	Best performance
1	LC_SVD	5.50	2.48	8
2	LC_ACF_region	3.46	2.50	11
3	ACF_SVD_country	6.24	4.64	5
4	CAE_BP	5.41	3.61	0
5	DEEP	2.68	1.38	52

Cite this article: Richman R and Wüthrich MV (2021). A neural network extension of the Lee–Carter model to multiple populations. *Annals of Actuarial Science* **15**, 346–366. https://doi.org/10.1017/S1748499519000071