

Sterblichkeitsmodellierung mithilfe von Rekurrenten Neuronalen Netzen

Nico Frisch, Wenhao Peng, Tim Vieth, Mattes Westdörp

15 01 2021

Mortalität

Unsere Daten

Exploratorische Datenanalyse

Datenvorbereitung

Erstellung der Netze

Hyperparametersuche

Projektionen

Vergleich RNN mit FNN und Lee-Carter

Mortalität

Was ist eine Mortalitätsrate

- ▶ Mortalitätsrate im Jahr t für das Alter x :

$$M_{t,x} = P(x < X < x + 1 | X > x) \in (0, 1] \quad (1)$$

- ▶ Wahrscheinlichkeit einer x Jahre alten Person das $x + 1$ -ste Lebensjahr **nicht** zu erreichen (Ereignis X)
- ▶ Mortalität deterministisch, da sie aus den Sterbetafeln hervorgeht

Unsere Daten

Wie sehen unsere Daten aus?

- ▶ Human Mortality Database (HMD)
- ▶ Mortalitätsraten unterteilt in:
 - ▶ 9 verschiedene Länder
 - ▶ Geschlecht: Weiblich und Männlich
 - ▶ Alter: 0 - 100 Jahre
 - ▶ Jahre: 1950 - 2016

Wie sehen unsere Daten aus?

```
unique(data$Country)
```

```
## [1] CHE    DEUT    DNK     ESP     FRATNP  ITA     JPN     POL     USA  
## Levels: CHE DEUT DEUTE DEUTW DNK ESP FRATNP ITA JPN POL USA
```

```
unique(data$Year)
```

```
## [1] 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963  
## [15] 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977  
## [29] 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991  
## [43] 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005  
## [57] 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016
```

```
unique(data$Gender)
```

```
## [1] Female Male  
## Levels: Female Male
```

```
unique(data$Age)
```

```
## [1] 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16  
## [18] 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33  
## [35] 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50  
## [52] 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67  
## [69] 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84  
## [86] 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
```

Struktur der Daten

```
str(data)
```

```
## Classes 'data.table' and 'data.frame':  118574 obs. of  6 variables:
## $ Country      : Factor w/ 11 levels "CHE","DEUT","DEUTE",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Year         : int  1950 1950 1950 1950 1950 1950 1950 1950 1950 1950 ...
## $ Gender       : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
## $ Age          : int   0 1 2 3 4 5 6 7 8 9 ...
## $ log_mortality: num  -3.6 -5.79 -6.39 -6.7 -6.9 ...
## $ mortality    : num   0.02729 0.00305 0.00167 0.00123 0.00101 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

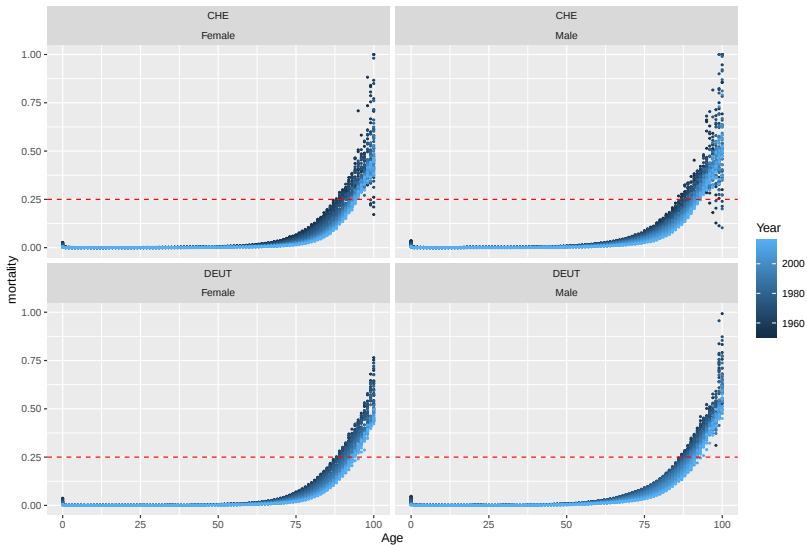
```
sample_n(data, 6)
```

```
##   Country Year Gender Age log_mortality mortality
## 1    ESP 2010   Male  51   -5.5014132  0.004081
## 2 FRATNP 1974 Female  97   -0.9490696  0.387101
## 3    CHE 1964   Male  56   -4.3785128  0.012544
## 4    DNK 1970 Female  28   -7.5889739  0.000506
## 5    POL 1968   Male   2   -6.7740989  0.001143
## 6    POL 2007 Female  44   -6.4087998  0.001647
```

Exploratorische Datenanalyse

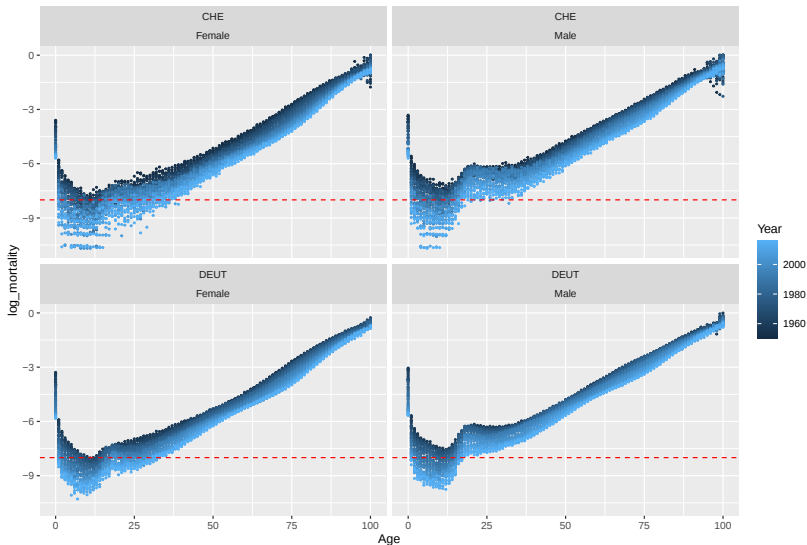
Mortalitätsraten im Wandel

Wandel der Mortalitätsraten von 1950 bis 2016 in Deutschland und Schweiz



Logarithmische Mortalitätsraten im Wandel

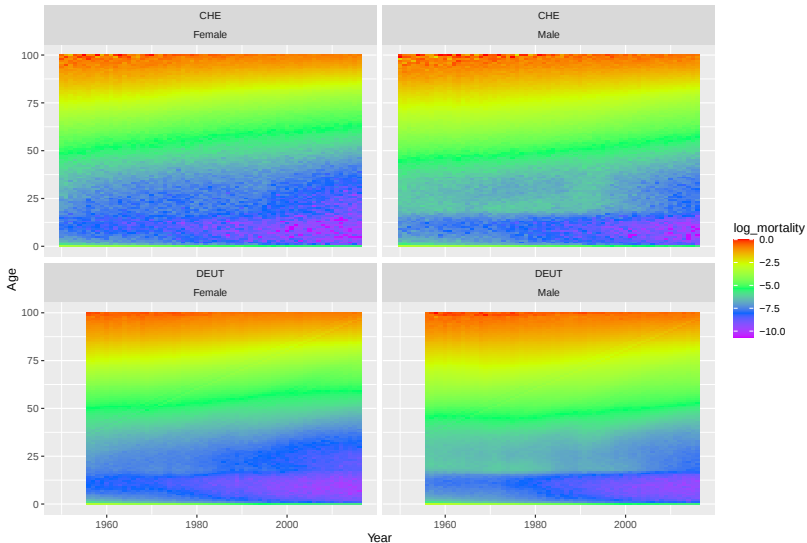
Wandel der Log-Mortalitätsraten von 1950 bis 2016 in Deutschland und Schweiz



- ▶ Frauen höhere Lebenserwartung als Männer
- ▶ Generell steigt Lebenserwartung (Mortalitätsrate sinkt)

Heatmap

Heatmap: Log-Mortalitätsraten in Deutschland & Schweiz



- ▶ Mortalität im Alter 1 - 15 inzwischen fast bei Null
- ▶ Kindersterblichkeit verschwindet

Datenvorbereitung

Datenvorbereitung

- ▶ Trainingsdaten: Jahre 1950 - 2006
- ▶ Testdaten: Jahre 2007 - 2016
- ▶ Idee: Ermittle $\log(M_{t,x})$ mithilfe der logarithmischen Mortalitätsraten der T vorherigen Jahre für Altersklassen x und deren Nachbarn ($x - 1, x + 1, \dots$)
- ▶ wie viele Nachbarn hängt vom Wert τ_0 ab:
 - ▶ $\tau_0 = 5 \rightarrow$ Betrachte x sowie $x - 1, x + 1, x - 2, x + 2$ (insgesamt 5 Altersklassen)
- ▶ Im Folgenden: $T = 10$ und $\tau_0 = 5$
- ▶ Beispiel: $\log(M_{1960,50})$ wird approximiert mithilfe von $\log(M_{1950,48:52}), \log(M_{1951,48:52}), \dots, \log(M_{1959,48:52})$

Trainingsdaten

- ▶ Vollständiges Trainingsset:

$$\mathcal{T} = \{(\mathbf{x}_{t-T,x}, \dots, \mathbf{x}_{t-1,x}, Y_{t,x})\}$$

mit $0 \leq x \leq 100$ und $1950 + T \leq t \leq 2006$, wobei

$$\mathbf{x}_{t,x} = \left(\log(M_{t,(x-2) \vee 0}), \log(M_{t,(x-1) \vee 0}), \log(M_{t,x}), \right. \\ \left. \log(M_{t,(x+1) \wedge 100}), \log(M_{t,(x+2) \wedge 100}) \right) \in \mathbb{R}^5$$

die Inputs des Netzes und

$$Y_{t,x} = \log(M_{t,x})$$

die Sample-Outputs

- ▶ Insgesamt $|\mathcal{T}| = 101 \cdot (57 - 10) = 4747$ Trainingsamples (bei $T = 10$)

Implementierung: Trainingsdaten

```
# function that outputs training data set (  $x_-(t,x)$ ,  $Y_-(t,x)$  )
data_preprocessing <- function(data, gender, country, timesteps, age_range, last_observed_year = 2006) {

  mort_rates <- data[which((data$Gender == gender) & (data$Country == country)),
    c("Year", "Age", "log_mortality")]
  mort_rates <- dcast(mort_rates, Year ~ Age, value.var = "log_mortality")

  # selecting data
  train_rates <- as.matrix(mort_rates[which(mort_rates$Year <= last_observed_year),
    ])
  # adding padding at the border
  (delta0 <- (age_range - 1)/2)
  if (delta0 > 0) {
    for (i in 1:delta0) {
      train_rates <- as.matrix(cbind(train_rates[, 1], train_rates[, 2], train_rates[,
        -1], train_rates[, ncol(train_rates)]))
    }
  }
  train_rates <- train_rates[, -1]
  n_years <- nrow(train_rates) - (timesteps - 1) - 1
  n_ages <- ncol(train_rates) - (age_range - 1)
  n_train <- n_years * n_ages # number of training samples
  xt_train <- array(NA, c(n_train, timesteps, age_range))
  yt_train <- array(NA, c(n_train))

  for (t0 in (1:n_years)) {
    for (a0 in (1:n_ages)) {
      xt_train[(t0 - 1) * n_ages + a0, , ] <- train_rates[t0:(t0 + timesteps -
        1), a0:(a0 + age_range - 1)]
      yt_train[(t0 - 1) * n_ages + a0] <- train_rates[t0 + timesteps, a0 +
        delta0]
    }
  }
  list(xt_train, yt_train)
}
```

Zu beachten

- ▶ Beachte: Input $\mathbf{x}_{t,x}$ sind logarithmische Mortalitätsraten, d.h. das Modell enthält keine kausalen Faktoren wie z.B. Jahr, Alter, Geschlecht etc.
- ▶ Zusätzliche Faktoren werden dem Netz extern in Form von zusätzlichen Inputs, welche in einfachen Dense Layern trainiert werden, beigefügt
- ▶ Hier zunächst nur Geschlecht und Land als zusätzliche Faktoren, weitere Faktoren können aber noch hinzugefügt werden

Testdaten

- ▶ Testdaten $\mathcal{V}$ müssen rekursiv generiert werden:

$$\mathcal{V} = \{(\mathbf{x}_{t-T,x}, \dots, \mathbf{x}_{2006,x}, \hat{\mathbf{x}}_{2007,x}, \dots, \hat{\mathbf{x}}_{t-1,x}, Y_{t,x})\}$$

mit $0 \leq x \leq 100$ und $2007 \leq t \leq 2016$ wobei

$$\hat{\mathbf{x}}_{s,x} = \left(\log(\hat{M}_{s,(x-2) \vee 0}), \log(\hat{M}_{s,(x-1) \vee 0}), \log(\hat{M}_{s,x}), \right. \\ \left. \log(\hat{M}_{s,(x+1) \wedge 100}), \log(\hat{M}_{s,(x+2) \wedge 100}) \right) \in \mathbb{R}^5$$

für $2007 \leq s \leq 2015$.

Erstellung der Netze

Netzarchitekturen

- ▶ wir nutzen LSTM oder GRU Architektur
- ▶ LSTM als exemplarisches Beispiel (GRU analog)

Implementierung

```
library(keras)

create_lstm_model <- function(input_shape, unit_sizes, activation, recurrent_activation,
                               output_activation, average_label) {

  input <- layer_input(shape = input_shape, dtype = "float32")
  input_gender <- layer_input(shape = c(1), dtype = "float32")
  input_country <- layer_input(shape = c(1), dtype = "float32")

  current_output <- input

  if (length(unit_sizes) > 1) {
    for (layer_number in 1:(length(unit_sizes) - 1)) {
      current_output <- layer_lstm(units = unit_sizes[layer_number], activation = activation,
                                   recurrent_activation = recurrent_activation, return_sequences = TRUE)(current_output)
    }
  }

  # For the last LSTM-layer the output is not return as sequences.
  current_output <- layer_lstm(units = unit_sizes[length(unit_sizes)], activation = activation,
                                recurrent_activation = recurrent_activation, return_sequences = FALSE)(current_output)

  # Concatenate LSTM result and gender indicator.
  current_output <- layer_concatenate(list(current_output, input_gender))
  current_output <- layer_concatenate(list(current_output, input_country))
  if (output_activation == "NULL")
    output_activation <- NULL
  output <- layer_dense(units = 1, activation = output_activation, weights = list(array(0,
                                                dim = c(unit_sizes[length(unit_sizes)] + 2, 1)), array(log(average_label),
                                                dim = c(1))))(current_output)

  model <- keras_model(inputs = list(input, input_gender, input_country), outputs = c(output))
}
```

Kompilierung des Modells

```
# compile model  
optimizer <- optimizer_adam()  
model %>% compile(optimizer = optimizer, loss = "mse", metrics = list("mae"))
```

Callbacks

```
# early stopping
early_stop <- callback_early_stopping(monitor = "val_loss", patience = 35)

# adaptive learning rate
lr_reducer <- callback_reduce_lr_on_plateau(monitor = "val_loss", factor = 0.1, patience = 25,
      verbose = 0, mode = "min", min_delta = 0.001, cooldown = 0, min_lr = 0)

# use best model only
use_best_model <- callback_model_checkpoint("best_model.h5", monitor = "val_loss",
      verbose = 1, save_best_only = TRUE, save_weights_only = FALSE)
```

Fitting-Prozess

```
# fit model  
history <- model %>% fit(x = x_train, y = y_train, validation_split = 0.1, batch_size = 100,  
  epochs = 200, verbose = 1, callbacks = list(early_stop, lr_reducer, use_best_model))
```

Hyperparametersuche

Hyperparametersuche

- ▶ Wahl der verschiedenen Parameter (timesteps, age_range, Anzahl Layer, Anzahl Units, etc.) schwierig
- ▶ Zufällige Wahl verschiedener Kombinationen
- ▶ Ergebnis: Tabelle mit best fitting models
- ▶ Wähle bestes Modell oder ein Ensemble der besten Modelle

Hyperparametersuche

```
parameters <- list(  
  model_type = c("LSTM"),  
  timesteps = c(5, 10, 15), #  
  age_range = c(5,7,9),  
  layers = c(1,2,3,4),  
  feature_dimension0 = c(20, 30, 40),  
  feature_dimension1 = c(15, 20),  
  feature_dimension2 = c(10, 15),  
  feature_dimension3 = c(5, 10),  
  batch_size = c(100, 200, 400),  
  activation = c("tanh"),  
  recurrent_activation = c("tanh", "sigmoid"),  
  output_activation = c("exponential", "linear")  
)
```

Ergebnisse der Hyperparametersuche

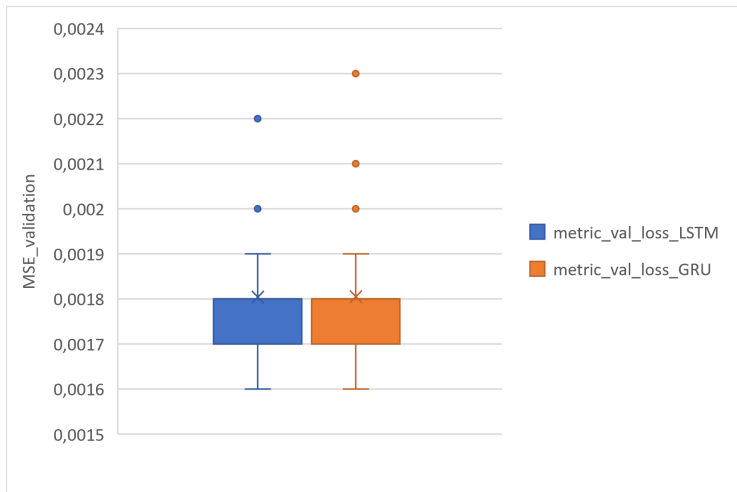


Figure 1: Boxplot zur Veranschaulichung der Größenordnung der Validation-Losses der LSTM- und GRU-Modelle

Erste Ergebnisse

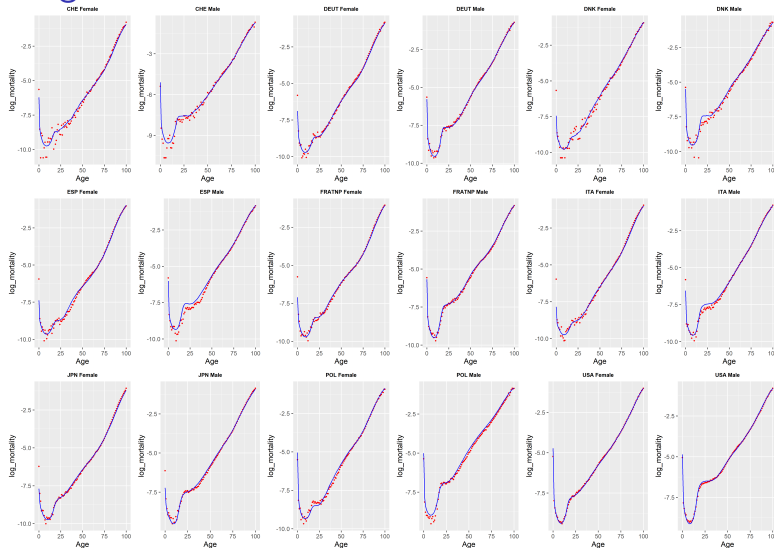


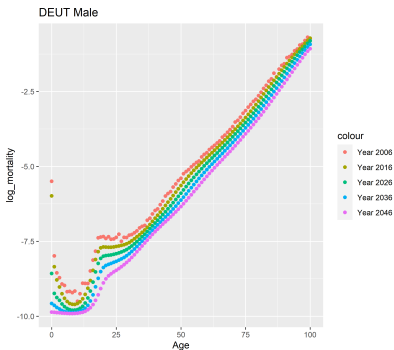
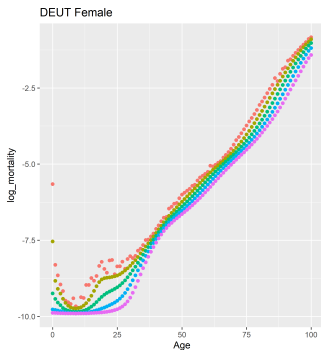
Figure 2: Beobachtete vs. mit bestem LSTM-Modell geschätzte Raten für das Jahr 2014.

Projektionen

Projektionen

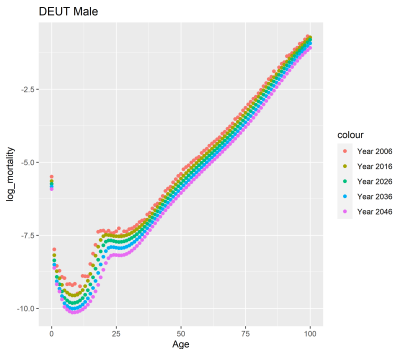
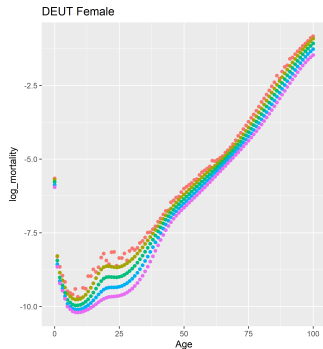
- ▶ Betrachte die Jahre 2006, 2016 (vorhanden), 2026, 2036, und 2046 (projeziert)
- ▶ für Jahre, die in der Zukunft liegen, muss rekursiv predictet werden

Projektionen LSTM



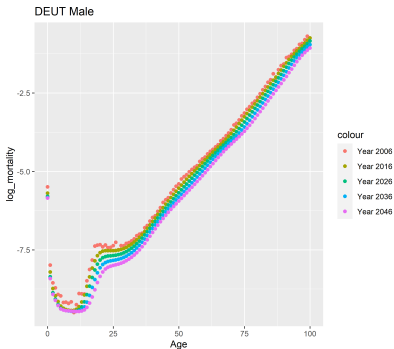
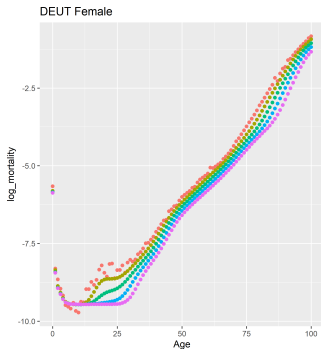
Projektionen der nächsten 40 Jahre gegeben durch das beste LSTM-Modell der Hyperparametersuche.

Projektionen LSTM



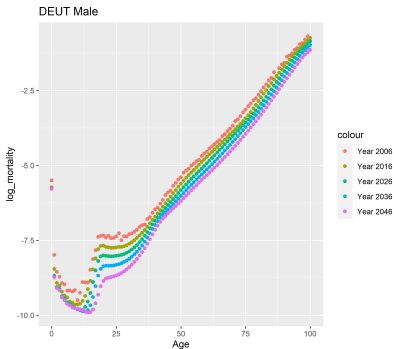
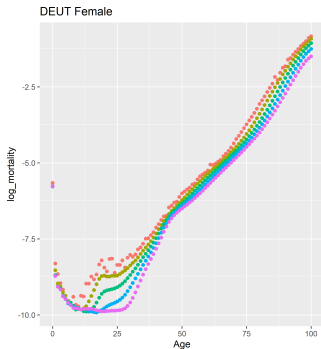
Projektionen der nächsten 40 Jahre gegeben durch das zweitbeste LSTM-Modell der Hyperparametersuche.

Projektionen LSTM



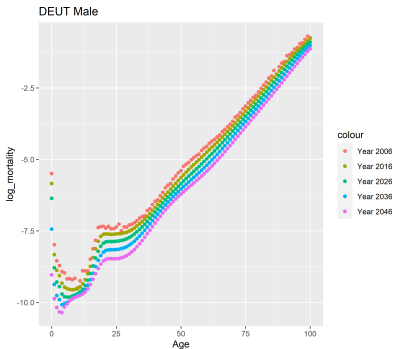
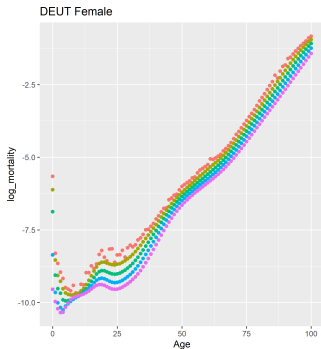
Projektionen der nächsten 40 Jahre gegeben durch das schlechteste LSTM-Modell der Hyperparametersuche.

Projektionen GRU



Projektionen der nächsten 40 Jahre gegeben durch das beste GRU-Modell der Hyperparametersuche.

Projektionen GRU



Projektionen der nächsten 40 Jahre gegeben durch das zweitbeste GRU-Modell der Hyperparametersuche.

Vergleich RNN mit FNN und Lee-Carter

Mean-Squared-Error im Vergleich

	MSE LSTM	MSE GRU	MSE FF	MSE LC
CHE Female	1,06E-04	1,10E-04	1,47E-04	8,06E-05
CHE Male	1,42E-04	1,43E-04	1,86E-04	1,82E-04
DEUT Female	7,66E-05	7,81E-05	1,29E-04	8,32E-05
DEUT Male	6,60E-05	8,40E-05	1,07E-04	1,12E-04
DNK Female	3,53E-05	3,62E-05	6,29E-05	8,57E-05
DNK Male	3,14E-04	3,12E-04	3,54E-04	3,83E-04
ESP Female	1,42E-05	1,36E-05	2,24E-05	6,92E-05
ESP Male	3,01E-05	3,00E-05	4,09E-05	2,03E-04
FRATNP Female	3,51E-05	3,78E-05	2,93E-05	4,28E-05
FRATNP Male	3,01E-05	3,96E-05	5,46E-05	7,96E-05
ITA Female	3,26E-05	3,29E-05	2,43E-05	1,17E-05
ITA Male	2,34E-05	3,05E-05	3,62E-05	4,94E-05
JPN Female	5,12E-05	7,12E-05	1,08E-04	4,71E-05
JPN Male	3,84E-05	4,41E-05	1,61E-04	2,50E-05
POL Female	4,02E-05	3,64E-05	3,36E-05	3,24E-04
POL Male	1,08E-04	9,86E-05	1,00E-04	3,56E-04
USA Female	3,46E-06	3,59E-06	8,85E-06	1,04E-05
USA Male	1,27E-05	1,14E-05	2,19E-05	7,23E-05

Figure 3: MSE Vergleich der besten LSTM- und GRU-Modelle mit Feedforward-Netz und Lee-Carter-Modell.

Mean-Squared-Error im Vergleich

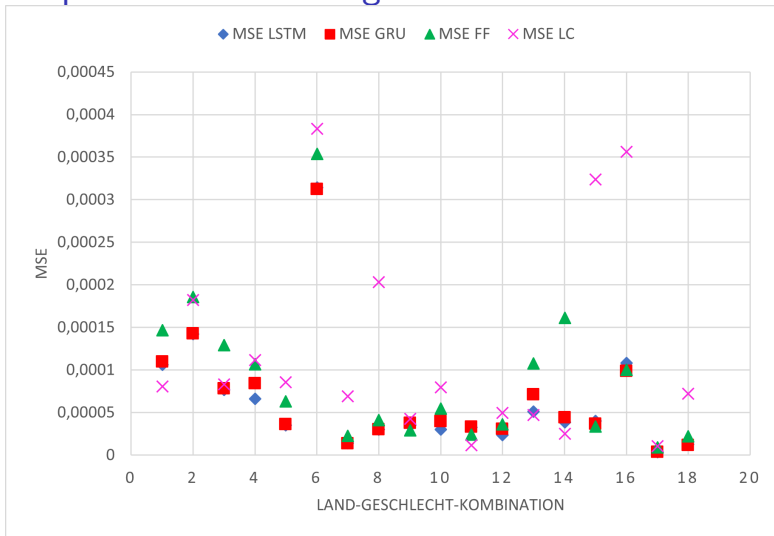


Figure 4: Out-of-sample-loss für verschiedene Land-Geschlecht-Kombinationen für das beste LSTM- und GRU-Modell im Vergleich mit Feedforward Netzen und Lee-Carter-Modell.

Mean-Squared-Error im Vergleich

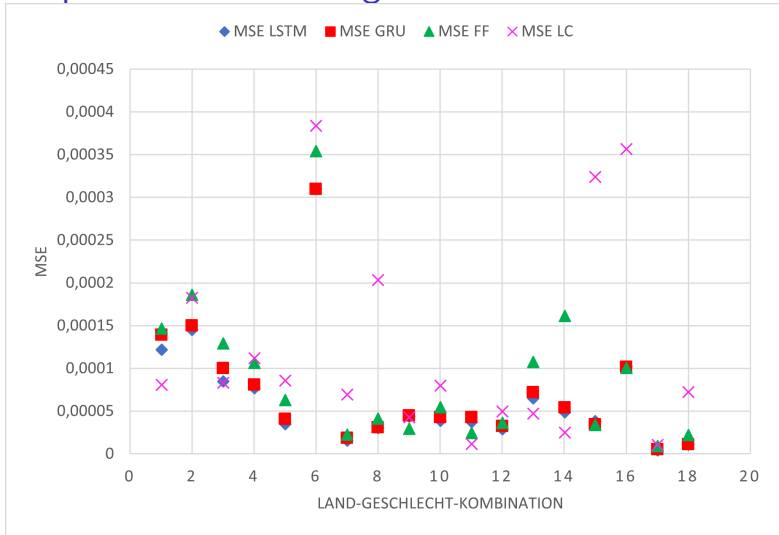


Figure 5: Out-of-sample-loss für verschiedene Land-Geschlecht-Kombinationen für die besten fünf LSTM- und GRU-Modell im Vergleich mit Feedforward Netzen und Lee-Carter-Modell.

Mean-Squared-Error im Vergleich

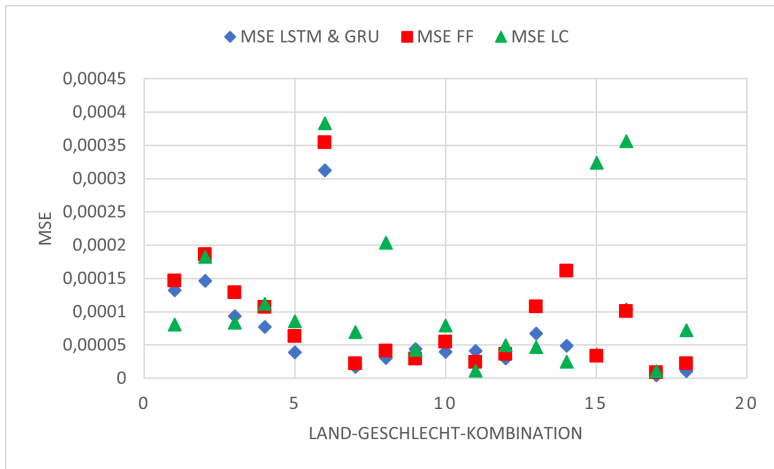


Figure 6: Out-of-sample-loss für verschiedene Land-Geschlecht-Kombinationen für die besten fünf LSTM- bzw. GRU-Modelle im Vergleich mit Feedforward Netzen und Lee-Carter-Modell.