$$
\begin{aligned}
\mathbb{E}[s_t] &= \mathbb{E}\left[(1-\rho_1)\sum_{i=1}^{t}\rho_1^{t-i}g_i\right] \\
&= \mathbb{E}[g_t](1-\rho_1)\sum_{i=1}^{t}\rho_1^{t-i} + \zeta \\
&= \mathbb{E}[g_t](1-\rho_1^t) + \zeta
\end{aligned}
$$

Where $\zeta = 0$ if the true first moment is stationary and otherwise ζ can be kept small because of the exponential decay [27].

Remark 5.29. *Similarly to remark 5.28 the correction term for the biased second moment can be derived.*

After the basic concepts have been explained, the last section of this chapter is dedicated to the question how well cash flows can be replicated with neural networks using different optimizer and layer structures.

5.3. Cash flow replication

For the analysis of the effectiveness of neural networks in replicating the cash flows of an insurance portfolio, a subset of an actual insurance portfolio was used. The portfolio used consists of risk insurance policies, whereby only policies of the most often sold tariff were included for the evaluations. This ensures that, if a neural network is successfully trained, a large part of the risk insurance portfolio can already be covered. In principle, all risk tariffs of the portfolio can be described with 37 different characteristics. These features include biometric data such as the sex or age of the insured person as well as actuarial parameters such as a discount label. The approximately 200,000 policies of the tariff analysed here, can be characterised by the following 17 attributes:

- Policy ID
- Age
- Premium
- Reserve
- Months since policy start
- Surpluses

- Policy status
- Policy duration
- Premium frequency
- Premium payment period
- Discount indicator
- Sex
- Main contract flag
- Sum assured
- Sales channel
- Type of sale
- Rider premium

Since most of the properties are self-explanatory, only those are briefly outlined here where there could be some ambiguity. The sales channel is a categorical variable and indicates how the policy was sold, either through exclusive distribution or through a broker. The type of sale is also a categorical variable and indicates how the contract was concluded. That is, whether it is a new business, a renewal or another type of business transaction. These 17 characteristics were then used to forecast the future portfolio development with the help of proprietary projection software called Prophet. With a future projection period of 60 years and about 60 reporting variables, this results in hundreds of millions of data points. Since this amount of data could not be handled with the available resources, especially since the learning processes of a neural networks would become too costly, there were two simplifications:

- The number of policies was reduced to approximately 25,000.
- The reporting variables were reduced to the mathematical reserve.

The policies were chosen in such a way that they form a group that is as homogeneous as possible. These are policies that:

- Are in the premium payment period with monthly payments.
- Have no discounts.
- Were sold via a broker.
- Are no renewals.

It can therefore be seen that there have been some limitations in the selection of policies in terms of discrete features. This fact must be taken into account when evaluating the results, but should not interfere with the further procedure.

The projection software was then used to forecast the reserve trajectories for the next 60 years for those 25,000 model points. To be able to check the

Table 5.1.: Average relative error between true and predicted values over a projection period of 60 years.

Configuration			
Nodes	Network Type	Average Error Year 1 - 25	Average Error Year 1 - 60
8	I	1.971%	1.995%
8	II	2.213%	2.467%
20	I	0.963%	1.600%
20	II	3.001%	3.111%
37	I	0.653%	1.341%
37	II	1.714%	2.289%
50	I	0.673%	1.466%
50	II	1.181%	2.151%

quality of the tested neural networks, the data was split into a training and a test set. For this purpose, 80% of the data points were randomly selected and assigned to the training set and the remaining 20% to the test set. The test set was never used to train the neural nets, but was only used once after training has been finished to check the predictive power. To smooth possible volatility caused by the random initialization of the weight parameters, each test configuration was tested 40 times with randomly initialized parameters and the results were averaged. The two network structures analysed are:

I) 2 hidden layers.

II) 3 hidden layers.

In both configurations, the number of neurons in the hidden layers was the same for all hidden layers in the network and was increased in steps from 8 to 50. The individual layers used relu (see equation 5.4) as activation function and initialized the weights uniformly. As batch size 32 was chosen and the number of epochs was set to 50. As loss function the mean square error was chosen and as optimization routine the Adam algorithm described in algorithm 9. In addition, a dropout rate of 10% for each layer has been implemented to prevent the model from overfitting. To ensure the reproducibility of the results, the random generator was set before each test run. Table 5.1 shows the averaged relative errors of the eight configurations tested. It is evident that the relative deviations in the first 25 years are on average lower than over the entire projection horizon of 60 years. This is due to the fact that the longer a projection goes on, the more policies have already expired. This means that towards the end of the projection there are not so many data points from which the neural network could learn and therefore the predictive power is lower. Since over the projection period the number of policies in the portfolio is decreasing, this means that the absolute values for the cash flows are also decreasing. A constant deviation in cash flows in absolute terms therefore has a much greater relative impact at the end of the projection than at the beginning. However, this phenomenon can be reduced by creating imaginary policies for the training portfolio that have a particularly long duration. This ensures that the neural network has enough data points even at the end of the projection period to learn from and can therefore make better predictions. Of course, this approach involves additional effort, since policies not in the portfolio must first be created and then projected into the future. Another finding of table 5.1 is that those models with network type I achieve better results than those with type II in all cases. This shows that those nets that have only 2 hidden layers give better results than those that consist of 3 hidden layers. The number of neurons within each layer also has an influence on the predictive power of a neural network. Nets with 8 or 20 neurons per layer perform worse than those with 37 or 50 neurons per layer as shown in table 5.1. Overall, the best performance is shown by the configuration with two hidden layers of 37 neurons each. In this neural network the number of neurons per layer is the average between the number of input neurons and output neurons. The predicted values for this configuration compared to the real values are shown in figure (5.7). A clear deviation in year 10 is striking which can be explained by an outlier. After the learning process, one of the 40 neural networks was not able to predict a proper reserve for year 10, so a value of 0 was predicted. If the hardware is designed for the training of many neural networks, such outliers can be compensated simply by increasing the number of samples from 40 to 100, for example. Apart from this outlier, the results up to year 25 are remarkably good and even after that, the deviations are within reasonable limits. It could thus be shown that surprisingly good forecasts of future reserve development are already possible with simple neural networks. Due to the limited technical possibilities, even for such simple configurations of neural networks as listed in table 5.1, no extensive tuning of the hyperparameters could be performed. Based on the configuration with the smallest deviations, an improvement of the predicted results can be achieved by adjusting the hyperparameters. Some possible adjustments would be:

- Adjusting the drop out rate for each layer.
- Adjusting the activation function for each layer.

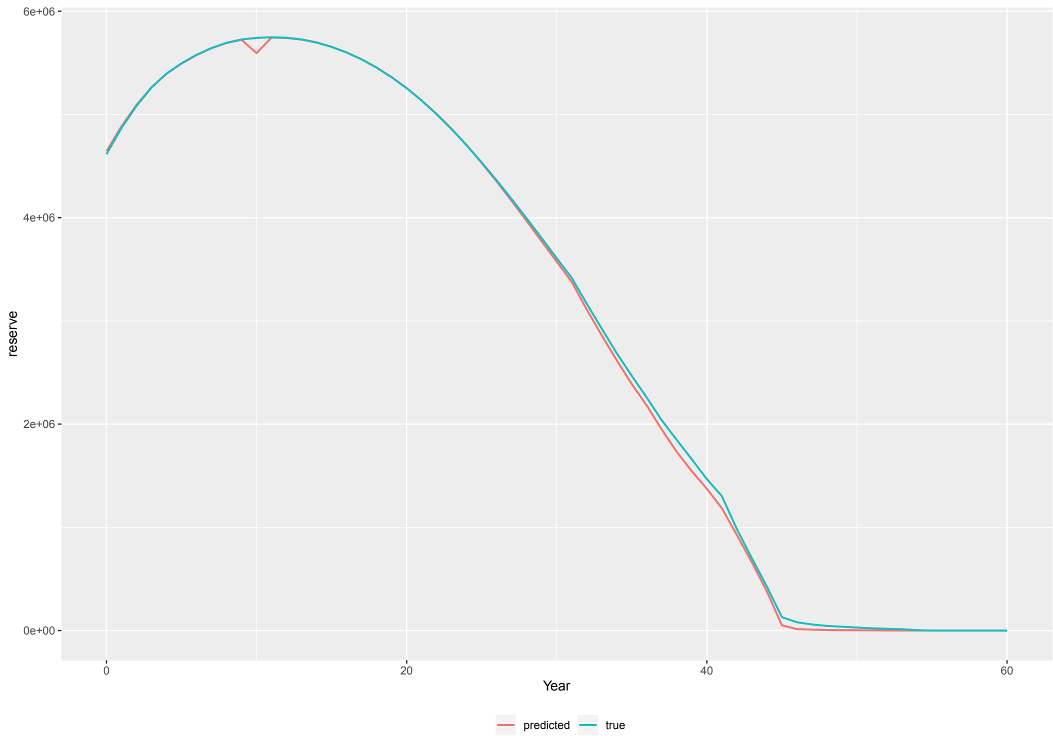

Figure 5.7.: Reserve predicted by neural network with 2 hidden layer with 37 nodes each.