Online Supplementary Material for "The use of autoencoders for training neural networks with mixed categorical and numerical features"

Łukasz Delong* Anna Kozak†

1 Details on Experiment 1

We use a NADAM optimizer with a default learning rate of 0.001 and a batch size of 1,000. To eliminate a possible increase in the reconstruction error on the training set (due to a learning rate which might be potentially too high), we apply an early stopping rule with zero delta and patience of 15 epochs on the validation set equal to the training set. See TensorFlow for R (2022) for details on implemention of early stopping - delta and patience parameters. When the autoencoder of type *MCA* is applied, the results do not depend on any hyperparameters, since the GSVD algorithm is applied to derive the representation of the categorical features. We also recall, for two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ the cosine similarity measure is defined as

$$\cos(\theta) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}|| \cdot ||\boldsymbol{b}||},$$

and it is one of the measures commonly used to measure similarity of words in text mining problems, see e.g. Blier-Wong et al. (2021). In this experiment we do not control over-fitting of the autoencoders. Over-fitting in reconstruction errors is controlled in Experiment 2 by optimizing the number of epochs used for training the autoencoders.

2 Details on Experiment 2

In Table 2.1 we present hyperparameters optimized in the experiment and their best values chosen with cross-validation. We use a NADAM optimizer and a batch size of 1,000 to train our neural networks. The set of 100,000 observations is randomly split into five cross-validation (CV) sets containing a training, a validation and a test set to the proportions 3:1:1. The training process is conducted for combinations of hyperparameters on five different CV sets. For A1, A1_CANN, A2 and A2_MCA, we train the neural networks for all possible

*SGH Warsaw School of Economics, Institute of Econometrics, lukasz.delong@sgh.waw.pl

†Warsaw University of Technology, Faculty of Mathematics and Information Science, anna.kozak@pw.edu.pl

Network A1	no. of neurons	[25, 20, 11], [35, 24, 10], [33, 32, 32]
	learning rate	$10^{-4}, 10^{-3}, 10^{-2}$
Network A2	no. of neurons	[30, 30, 30], [30, 20, 10], [20, 15, 10]
	learning rate	$10^{-4}, 10^{-3}, 10^{-2}$
1st AE for the categorical input	epochs	15, 50, 100, 200, 300
	learning rate	$5 \cdot 10^{-5}, 5 \cdot 10^{-4}, 5 \cdot 10^{-3}$
	corruption	without noise, sample, zero,
	no. of features corrupted	1, 2, 3 (out of 6)
2nd AE for the numerical input	epochs	15, 50, 100, 200, 300
	learning rate	$5 \cdot 10^{-5}, 5 \cdot 10^{-4}, 5 \cdot 10^{-3}$
	corruption	without noise, gaussian, zero
	sigma noise	0.1, 0.25, 0.5
	no. of features corrupted	1, 3, 5 (out of 11)

Table 2.1: Hyperparameters optimized in the experiment.

configurations of the hyperparameters in Table 2.1. For A2_1AE and A2_2AEs, we proceed as follows:

- We train the 1st AE for the categorical input. Next, we train the network for the supervised task by initializing the weights of the joint embedding with the weights from the encoder from the 1st AE. All other weights in the network are initialized with the Xavier initialization and the bias terms are initially set to zero. The calibrations are performed with all possible configurations of the hyperparameters for the 1st AE and the network in Table 2.1,
- We identify the best two sets of the hyperparameters for the 1st AE and the network,
- We train the 1st AE for the categorical input and the 2nd AE for the numerical input. Next, we train the network for the supervised task by initializing the parameters of the joint embedding and the first hidden layer of the sub-network with three hidden layers with the parameters from the encoders from the 1st and the 2nd AE. All other weights in the network are initialized with the Xavier initialization and the bias terms are initially set to zero. The calibrations for the 1st AE and the network are performed with the two configurations of the hyperparameters identified in the previous step. The calibrations for the 2nd AE are performed with all possible configurations of the hyperparameters in Table 2.1,
- For the set of the hyperparameters of the 1st AE and the network already chosen, we identify the best hyperparameters for the 2nd AE, and we choose the best configuration of the hyperparameters.

For each CV set (with 60,000 observations in the training set, 20,000 in the validation set and 20,000 in the test set), we train, if required, our autoencoders on the training set and apply an early stopping rule on the validation set equal to the training set (see Experiment 1). We

train our neural network for the supervised learning task on the training set with 1000 epochs (with proper initialization from the autoencoders if required), apply an early stopping rule for training the network on the validation set and, finally, we evaluate the predictive power of the trained network by calculating the Poisson loss (the Poisson deviance) on the test set. The decision about the best hyperparameters was made based on the average value of the Poisson loss on the five CV test sets. The early stopping algorithm is applied with patience equal to 15 epochs and delta equal to zero.

In Figure 2.1 we present the average Poisson loss values on the five CV training and test sets for all neural networks trained in this step of the experiment. We can see that there are many A2_AE networks, which is the notation used for all A2_1AE and A2_2AEs, with many possible configurations of the hyperparameters, which lead to a lower loss on the test set than the loss which could be achieved with the other architectures and training processes of the network, in particular, better than A1. Of course, the number of A2_AE trained in this experiment is much higher than the number of other networks due to a larger number of hyperparameters for A2_AE and a much higher number of possible configurations of the hyperparameters. The purpose of Figure 2.1 is to show that our new architecture of a neural network pre-trained with (denoising) autoencoders should have better generalization properties if only reasonable, not necessarily optimal, hyperparameters can be identified.

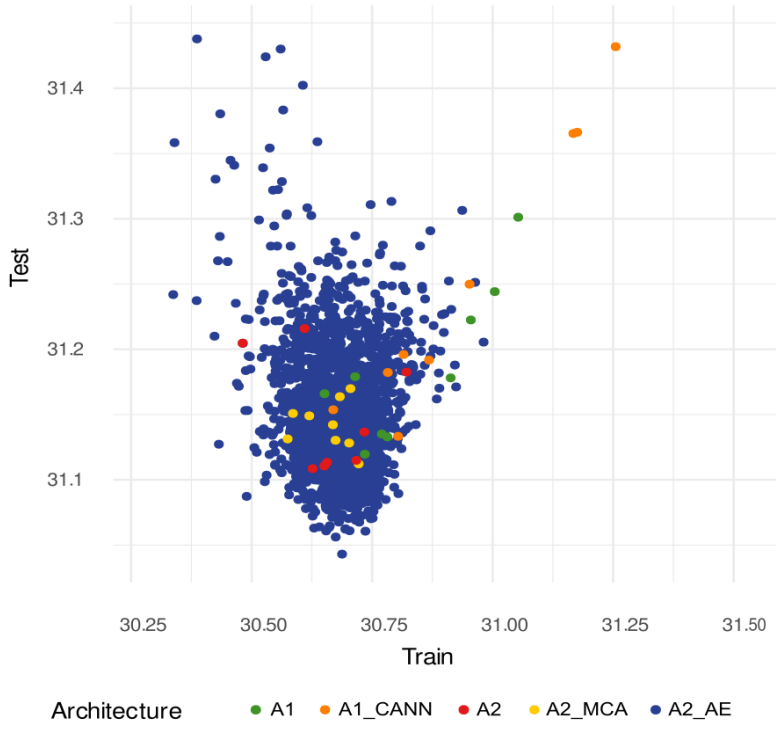

Figure 2.1: Average Poisson loss values on the CV training and test sets for all networks trained in the experiment.

The best hyperparameters we selected are presented in Tables 2.2-2.3 below.

Architecture	No. of neurons	Learning rate	Loss
A1	[25, 20, 11]	10^{-3}	31.133
A1_CANN	[35, 24, 10]	10^{-3}	31.134
A2	[20, 15, 10]	10^{-4}	31.111
A2_MCA	[30, 20, 10]	10^{-4}	31.113

Table 2.2: Best hyperparameters and average Poisson loss values on the CV test sets.

No. of neurons A2	Learning rate A2	AE	Noise type AE	Noise level AE	Learning rate AE	Epochs AE	Loss
[20, 15, 10]	10^{-4}	1st	zero	2	$5 \cdot 10^{-4}$	100	31.069
[20, 15, 10]	10^{-4}	2nd	—	—	$5 \cdot 10^{-3}$	100	31.061

Table 2.3: Best hyperparameters for A2_1AE and A2_2AEs and average Poisson loss values on the CV test sets.

3 Bias of A1 and A2

As discussed in Wüthrich (2020), bias in predictions at a portfolio level may be an issue when using neural networks for actuarial pricing. For each calibration from 100 calibrations from the second step of Experiment 2, we predict the claim frequencies for all observations in the test set and calculate the mean predicted claim frequency on the test set weighted with the exposures. The mean prediction should match the sample mean claim frequency on the test set. The results are presented in Figure 3.1. We can observe that the architecture A1 has almost no bias, and the architecture A2 yields slight upward bias in the predictions on the test set. We can also observe that the standard deviation of the mean predicted claim frequency is smaller for A2 compared to A1, in particular, it is equal to 0.2096 for A1 and 0.0653 for A2_2AEs, which illustrates that we also gain stability in the predictions when we apply the autoencoders for pre-training the neural network.

It is advised in the actuarial literature that predictions from neural networks should be corrected for bias with autocalibrated predictors, see e.g. Ciatto et al. (2022). The authors suggest to autocalibrate predictors on a validation set. Such autocalibration of a predictor removes the bias of the predictor on the validation set but the predictor is still biased on a test set if the sample means in the validation and the test sets are different (as is the case here). We autocalibrate the predictors learned with A1 and A2_2AEs and denote the autocalibrated predictors with A1_auto and A2_2AEs_auto. The autocalibrated predictors have no bias on the validation set and very similar bias on the test set, see Figure 3.2. By comparing A2_2AEs_auto with A1_auto, we can compare the predictive power of our predictors with the same bias. The results are presented in Figure 3.2. We conclude that A2_2AEs_auto is superior compared to A1_auto.

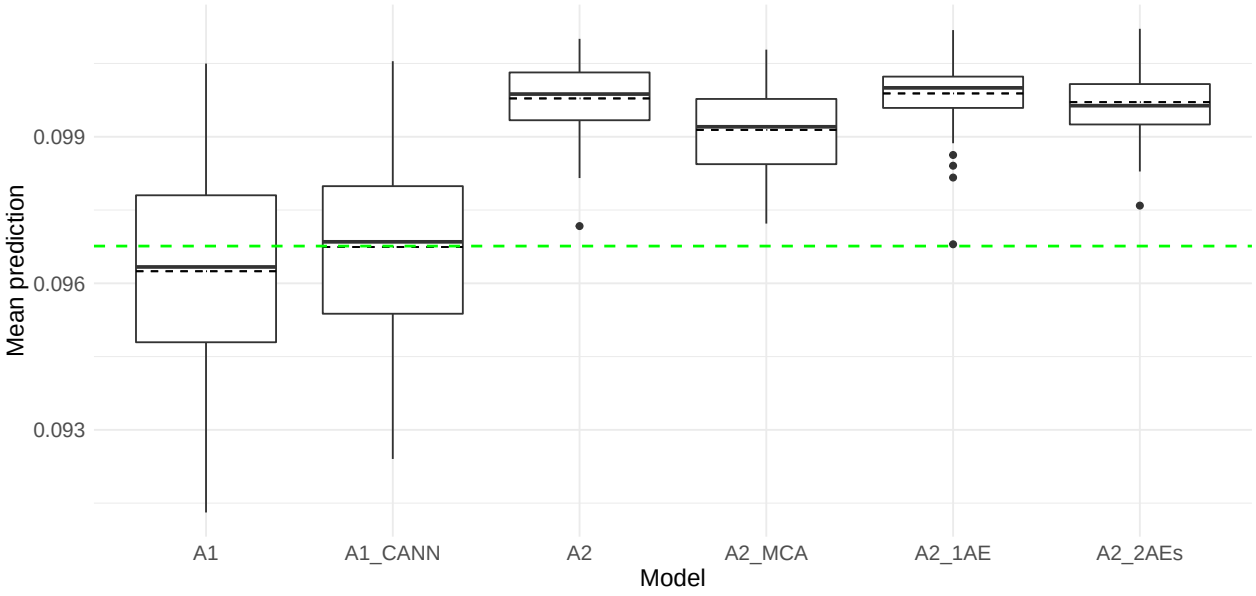

Figure 3.1: Distributions of the mean predicted claim frequency on the test set (for each network the dotted line represents the average of the mean predicted claim frequency in 100 calibrations). The green dotted line represents the sample mean claim frequency.

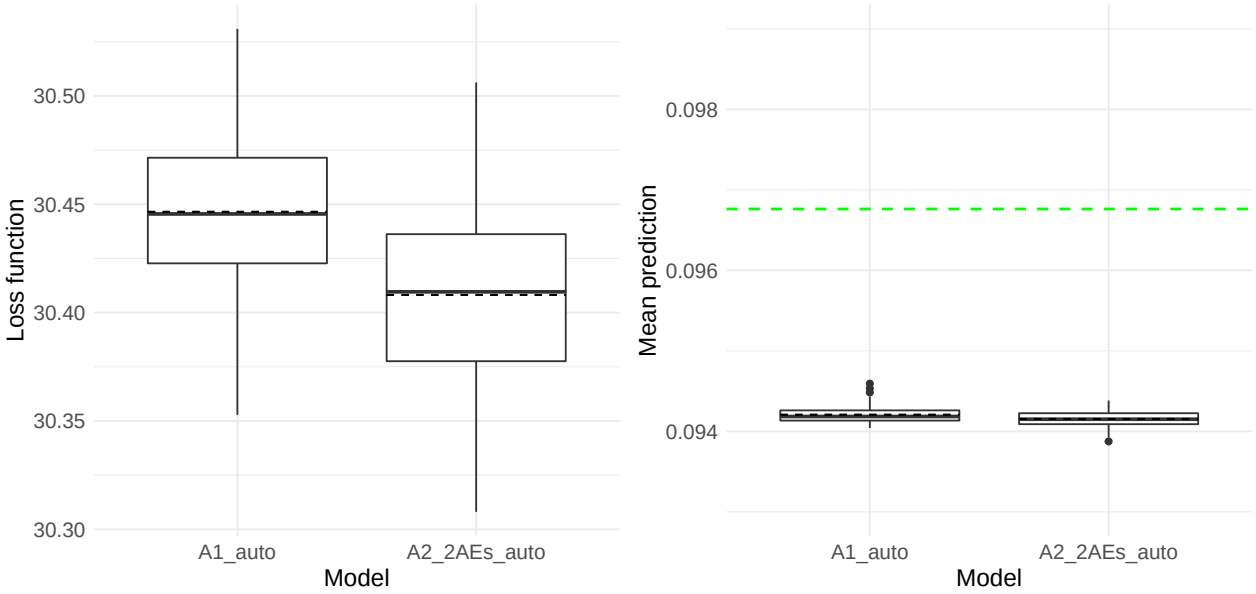

Figure 3.2: Distributions of the Poisson loss and distributions of the mean predicted claim frequency on the test set (for each network the dotted line represents the average value in 100 calibrations). The green dotted line represents the sample mean claim frequency.

4 Optimal dimension of the joint embedding in A2_2AEs

We study a range of possible dimensions of the joint embedding for the six categorical features and evaluate their performance with cross-validation, as in the first step of Experiment 2.

The average values of the Poisson loss on the five CV test sets are presented in Figure 4.1. The average Poisson loss values are similar for dimensions in the range from 4 to 12. We focus on the dimension of the numerical representation of the categorical features in this range and we repeat the calibrations 100 times for each dimension, as in the second part of Experiment 2. We observe that the average Poisson loss values on the test set in 100 calibrations are very similar for dimensions 6-10 (around 30.34).

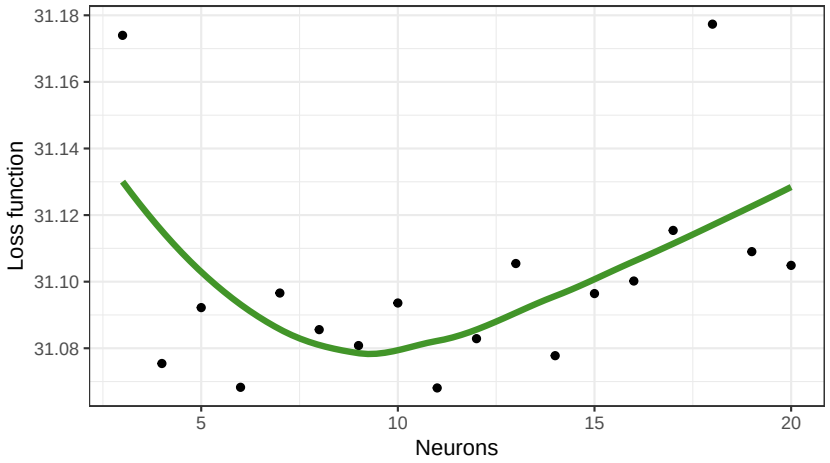

Figure 4.1: Average Poisson loss values on the CV test sets.

References

Blier-Wong, C., Baillargeon, J.-T., Cossette, H., Lamontagne, L., and Marceau, E. (2021). Rethinking representations in P&C actuarial science with deep neural networks. https://arxiv.org/abs/2102.05784.

Ciatto, N., Verelst, H., Trufin, J., and Denuit, M. (2022). Does autocalibration improve goodness of lift? *European Actuarial Journal*. In press.

TensorFlow for R (2022). Online tutorial. https://tensorflow.rstudio.com/.

Wüthrich, M. (2020). Bias regularization in neural network models for general insurance pricing. *European Actuarial Journal*, 10:179–202.