# A Comparison of Regularization Techniques for Shallow Neural Networks Trained on Small Datasets

Jiří Tumpach[1,2], Jan Kalina[2], and Martin Holeňa[2]

[1] Charles University, Faculty of Mathematics and Physics, Prague,
[2] The Czech Academy of Sciences, Institute of Computer Science, Prague, {tumpach,kalina,martin}@cs.cas.cz

*Abstract:* Neural networks are frequently used as regression models. Their training is usually difficult when the model is subject to a small training dataset with numerous outliers.

This paper investigates the effects of various regularisation techniques that can help with this kind of problem. We analysed the effects of the model size, loss selection, L2 weight regularisation, L2 activity regularisation, Dropout, and Alpha Dropout.

We collected 30 different datasets, each of which has been split by ten-fold cross-validation. As an evaluation metric, we used cumulative distribution functions (CDFs) of L1 and L2 losses to aggregate results from different datasets without a considerable amount of distortion. Distributions of the metrics are shown, and thorough statistical tests were conducted.

Surprisingly, the results show that Dropout models are not suited for our objective. The most effective approach is the choice of model size and L2 types of regularisations.



## 1 Introduction

Neural networks are nature-inspired regression models increasingly important in machine learning. This type of model excels in predictive power but it has a poor robustness to outliers, if the training dataset has a small number of samples, the target function is complicated, or the network is over-parametrized for the problem [1, 2, 3].

On the other hand, novel theoretical analyses show different perspective on neural networks. In [4, 5] the authors investigate the effect of priors over weights for infinitely wide single layer neural network and show that a Gaussian prior results in a Gaussian process prior over its functions. The Gaussian process is a smooth non-parametric model well known for its generalisation properties, so it leads to the conjecture that there is no need to avoid overfitting of such a network. That idea was further generalised to two-layer neural networks in [6] and general deep neural networks in [7]. Experiments in [7] show that finite-width neural networks approach the infinite counterparts through increasing their width. The authors of [7] further pointed out that Dropout could be an interesting potential improvement.



In this paper we are interested in these areas where the network should struggle because we intend to use neural networks as approximation for surrogate modeling in black-box optimisation. Surrogate models are local models that estimate an unknown function in order to select better candidates for evaluation and in this way reduce the cost of optimisation.

That motivated the research reported in this paper, in which we have investigated 5 different regularisation techniques in different configurations on 30 different datasets.

## 2 Methods

### 2.1 Regularisation

Regularisation is a broad term used for methods that add some new prior belief [2, 3] to a specific machine learning method. The belief should redefine the problem to achieve a solution modified in the sense described by Occam's razor principle [8, 9] – more complex hypotheses are less likely than simple hypothesis. For example, the lasso/ridge shrinkage methods in linear regression add a new term that makes small values more suitable as the solution to a problem [2].

**The networks size regularisation** As with many other regression models, the number of free parameters has critical consequence [3]. Models with a small number of free parameters can handle only simple relationships, while large models can be more flexible. On the other hand, a large model needs more samples in order to achieve more reliable predictions for all parameters. If that requirement is not met, the regression can over-fit – the model finds some non-sensible but possible relationships in the training dataset, which not valid in general.

**Weight regularisation** One possible solution to the free parameters problem is a restriction of parameter domains. In neural networks, it is done using weight regularisation. In fact, the domains of the parameters are unchanged, but the probability of larger values is strongly reduced because of an alternation of an optimisation objective. For example, the L2 type of the weight regularisation adds new term $\mathscr{L}_{w2}$ to the loss of particular network. It is defined as

$$\mathscr{L}_{w2} = \sum_i w_i^2 \tag{1}$$

Where $w_i$ stands for the value of the i-th parameter. It is usually applied only to weights, not to biases.

**Activity regularisation** The third type of regularisation is the activity regularisation [10]. In short, it penalises big values coming from neurons. The effect may seem similar to weight regularisation, but it may have more potential in cases where the size of a layer is large enough. The reason is that the weighted activities could count up to large numbers in spite of small values of the weights. Activity regularisation is a way of making the input information denser which is a nice property that is commonly utilised in autoencoder-type neural networks [3, 10].

**Dropout** Dropout technique essentially mimics the bagging technique, which is regularly used for improving generalisations of multiple models by aggregating the results [11, 12, 3].

When Dropout is applied to a specific layer, the training and testing phases differ. When the model is in the training stage, the results are randomly dropped – replaced with zeros. Therefore the next layer is forced to adapt to this incomplete information. In the testing phase, the random sampling is replaced by a multiplicative constant in order to maintain mean values of activation for the next layer[1].

Consequently, it increases the robustness of the model and does not require any other model to train. The main difference compared with bagging is that the models in Dropout are dependent – they share weights. Such a sharing is illustrated in Figure 1.

**Alpha Dropout** Standard Dropout is suited for rectified linear units because zero is the default value of this activation [11]. Alpha Dropout is a slight modification for smoother activation functions. It deals not only with the mean, but also with the variance. It is based on maintaining a walking average of neurons' outputs and scaling them accordingly.

### 2.2 Loss functions

A crucial part of any machine learning model selection is the definition of a loss function (prediction error measure, performance measure) [13, 2]. The loss function should be fast, convex and should match the random noise that can be found in the data. Frequently, the Mean Absolute Error (MAE) and Mean Square Error (MSE) functions are selected because the corresponding noise is additive and generated by Laplacian and Gaussian distribution, respectively. In addition, also the Huber loss function (cf. [14]) is commonly used. These three loss functions are defined

[1] An alternative is to use the constant in the training phase.

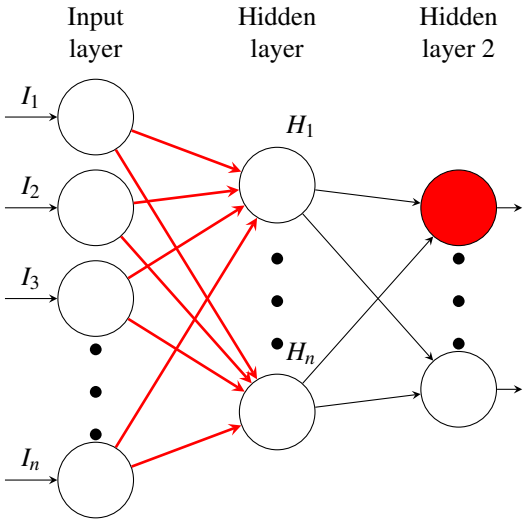



Figure 1: Dropout regularisation. The red circle depicts the neuron where Dropout regularisation causes the output to be masked – in the current iteration, the output is set to zero, and all following layers are computed as normal. This effect causes the updates of immediate incoming connections of that neuron to be zero, but other updates can still modify all other previous weights through non-dropped neurons. In this case, all red edges represent changes brought by gradients from other neurons that may influence the dropped neuron in the following iterations.

as:

$$\mathrm{MSE}(\mathscr{D}) = \frac{1}{2|\mathscr{D}|}\sum_{i=1}^{|\mathscr{D}|}(y_i-\hat{y}_i)^2 \tag{2}$$

$$\mathrm{MAE}(\mathscr{D}) = \frac{1}{|\mathscr{D}|}\sum_{i=1}^{|\mathscr{D}|}|y_i-\hat{y}_i| \tag{3}$$

$$\mathrm{Huber}(\mathscr{D}) = \frac{1}{2|\mathscr{D}|}\sum_{i=1}^{|\mathscr{D}|}\min\left((y_i-\hat{y}_i)^2, 2|y_i-\hat{y}_i|-1\right), \tag{4}$$

where $\mathscr{D}$ is the dataset on which the loss is calculated, $|\mathscr{D}|$ is its size, $y_i$ is the target value of the $i$-th sample, and $\hat{y}_i$ is its prediction.

It is common to assume that the dataset is outlier-free and normally distributed; therefore the MSE is the first choice regarding the selection of the loss function.

MAE/Huber losses are good replacement whenever the data are known to have outliers, or the MSE has not performed well for an unknown reason.

**Robust loss functions** A common way of dealing with outliers is to remove them from training data or choose an entirely different model[2] [2].

Even though the outlier removal has been thoroughly studied, the exact definition of an outlier highly depends

[2] Like k-NN or regression tree.

on the problem we want to solve. There exists definitions of an outlier relying on median absolute deviation [15], quantile and medoid [16], online Kalman filter [17] or nearest neighbour based filtering [18].

A different approach is proposed in [19] and improved in [20] where authors deal with robust linear regression by removing the most prominent residuals in the loss function. That idea was further adapted for neural networks in [21] or nonlinear regression with a known regression function in [22]. Essentially, these methods exploit the idea that neural networks can learn algorithms (hypothesis). With an assumption that more complex algorithms are harder to learn, the prior belief that reduces the probability of more complex hypotheses also serves as an outlier removal tool.

Extensions Least Trimmed Squares (LTS) and Least Trimmed Absolute Deviations (LTA) of MSE (2) and MAE (3) that follow the approach recalled in the previous paragraph and we used them in out analysis are defined in the following way

$$\mathrm{LTS}(\mathscr{D}) = \frac{1}{0.9|\mathscr{D}|} \sum_{i=1}^{|\mathscr{D}|} \rho\left((y_i - \hat{y}_i)^2\right) \tag{5}$$

$$\mathrm{LTA}(\mathscr{D}) = \frac{1}{0.9|\mathscr{D}|} \sum_{i=1}^{|\mathscr{D}|} \rho\left(|y_i - \hat{y}_i|\right), \tag{6}$$

where $\rho(x_i) = \begin{cases} x_i & \text{if less than 90 \% of residuals} \\ 0 & \text{otherwise} \end{cases}$

# 3 Methodology

## 3.1 Datasets and their preparation

We selected 30 datasets containing a relatively small number of samples. These are real-world as well as artificially generated publicly available datasets, for which a nonlinear regression model (i.e. explaining a given variable as a response against predictors under uncertainty) is a meaningful task. The list of the 30 datasets is presented in Table 1. Only datasets without missing values were selected. A ten-fold validation has been employed in order to obtain more reliable results. If the dataset had less than ten samples, we used leave-one-out cross-validation instead. Each feature was standardised according to training data in a specific fold.

## 3.2 Aggregation of results

It is not possible to visualize the results of regression methods across multiple datasets and loss functions. For example, some datasets are easier than others, and one loss function highlights outliers more, so most of the loss is made of one sample.

We tackle this problem by separating the specific dataset, fold, and function in a separate bin. In this bin, we learn the order of results creating empirical cumulative distribution function (ECDF). Every result in a specific bin was mapped by the corresponding ECDF, creating normalized order of results in a particular bin. Finally, all results are combined back together.

To compare normalized results for a specific hyperparameter, we split combined results by the value. These splits create empirical distributions of normalized results, which a violin plot can reasonably visualize.

Table 1: All datasets considered in the analysis.

| Name | no. features | no. samples |
|---|---|---|
| Concrete Compressive Strength | 9 | 1030 |
| The Boston Housing | 14 | 506 |
| Auto MPG | 8 | 398 |
| Proben1 (3d reg.) | 6 | 4208 |
| Misra1a | 2 | 14 |
| Chwirut2 | 3 | 54 |
| Chwirut1 | 3 | 214 |
| Lanczos3 | 6 | 24 |
| Gauss1 | 8 | 250 |
| Gauss2 | 8 | 250 |
| DanWood | 2 | 6 |
| Kirby2 | 5 | 151 |
| Hahn1 | 7 | 236 |
| Nelson | 3 | 128 |
| MGH17 | 5 | 33 |
| Lanczos1 | 6 | 24 |
| Lanczos2 | 6 | 24 |
| Gauss3 | 8 | 250 |
| Roszman1 | 4 | 25 |
| ENSO | 9 | 168 |
| MGH09 | 4 | 11 |
| Thurber | 7 | 37 |
| BoxBOD | 2 | 6 |
| Rat42 | 3 | 9 |
| MGH10 | 3 | 16 |
| Eckerle4 | 3 | 35 |
| Rat43 | 4 | 15 |
| Bennett5 | 3 | 154 |

| Hyperparameter | Considered values |
|---|---|
| Loss | MSE, MAE, Huber, LTS, LTA |
| Size of the first layer | 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 |
| Model regularisation | No regularisation,<br>50% Dropout,<br>50% Alpha Dropout |

Table 2: Options for the first set of experiments.

| Hyperparameter | Considered values |
|---|---|
| Loss | MSE, MAE, Huber |
| Size of the first layer | 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192 |
| Model regularisation | L2-weight – 0.001, 0.01, 0.1, 1, 10 |
| | L2-activity – 0.001, 0.01, 0.1, 1, 10 |
| | Alpha dropout – 0.1, 0.2, 0.4, 0.6, 0.8 |

Table 3: Options for the second set of experiments.

### 3.3 Statistical tests

We have used only non-parametric blocking statistical tests because the results have limited values, and we wanted to utilise as much information as possible. The Friedman test was used to decide whether a particular view on some hyperparameter includes is drawn from the same distribution or not. At this point, the ECDF mapping is not needed because the test is non-parametric. All statistical tests use the usual 5% significance level.

Multiple comparison tests were done using Wilcoxon signed-rank test [23] with Holm correction [24] instead of mean-ranks post-hoc tests which can create inconsistencies and paradoxical situations in machine learning scenarios [25].

### 3.4 Architecture

We selected three-layered architecture. The first layer has T neurons, the second layer has always $T/2$ and the third layer always has one neuron. The first two layers have Scaled Exponential Linear Units (SELU) as an activation function, and the third layer has a linear function.

### 3.5 Training

We trained our models with a NAdam optimiser with a 0.001 learning rate. We use early stopping with patience = 10 and delta = $1e^{-10}$ to speed up the training. Even though this is another type of regularisation, we use it in such a manner that its effect is minuscule. The maximal number of epochs is set to 10000, and batch size is equivalent to the size of the largest dataset. In the first set of experiments, we produced 48 014 neural networks and their results. In the second set, we managed to prepare 178 092 models.

## 4 Results

### 4.1 Dropout regularisation

In the first experiment, we compared between 3 settings – no regularisation, dropout regularisation and, alpha dropout regularisation. Both dropout techniques are set to 50% probability. The results are in Figure 2, number of models that are better than the same hyperparameter counterpart can be seen in Table 2b for L1 loss and Table 2c for L2 loss. We highlighted in bold values that Wilcoxon signed-rank test with Holm correction found significantly better than the column value.

It seems that the regularisation does not help. It may be caused by the exaggerated value of the Dropout rate or a need for such models to have wider layers. We do not know the reason why Alpha Dropout performed so badly – it should be better because we used SELU as an activation function.

One possible explanation for this poor performance is the use of early stopping. In the training phase, the dropout causes the output to be stochastic, so the error is stochastic too. The stochastic error can cause accidental results, which can stop the training prematurely. Because we have one mini-batch, the variance is too significant not to be perceptible.

Though not tried in our experiments, a possible remedy could be early stopping variation where the error is exponentially smoothed.

### 4.2 Size of models

In the second and third experiments, we were interested in network size and its effect on performance. Figure 3a and Table 4 show non-regularised models and Figure 3b and Table 5 show the Dropout variants combined together. Non-regularized results are better than the Dropout variants, which are less stable and have delayed response on the increase of network size.

The stability may come from the same source as the previous problem - the early stopping could make the model undertrained. The delay may be the result of the selected dropout rate. Because we used a dropout rate of 50%, the real amount of usable information can be effectively halved in each hidden layer (given that there is no space or resources to make the information denser). Together it is a 4x delay which is not enough to explain the findings (the optimum size of the model is $2^4$ vs $2^7$). Possible other reasons could be

- the difficulty of encoding uncertain patterns
- undertraining, due to early stopping

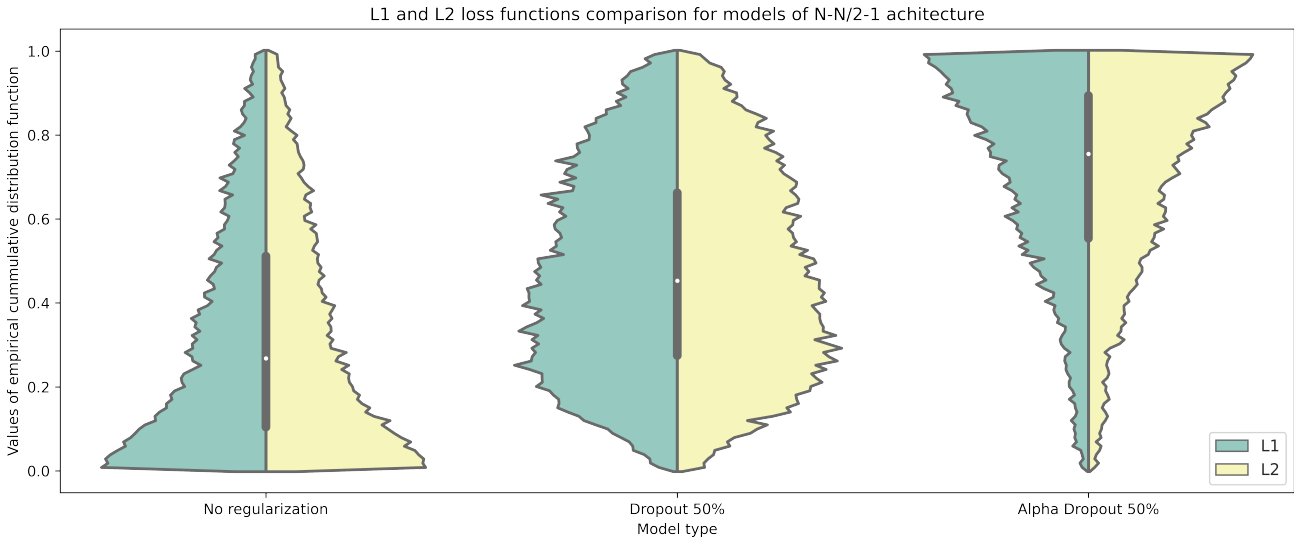
### 4.3 Loss function

In the fourth experiment, we analyzed the effect of a loss function for models without regularisation. Trimmed variants performed poorly probably because they remove some residuals (10%) and, therefore, reduce dataset size even more. In our case, Mean Squared Error (MSE) is better fitted than Mean Average Error (MAE). From the distribution in Figure 4 it seems that MSE has much worse results, but the median value (shown as the white point in the central part of the graph) of MSE is better than that of MAE. The best loss function is the Huber loss. All results can be seen in Table 6.

The Huber loss combines benefits of both worlds because its derivatives are dependent on the size of error (from MSE) while limiting the maximum value (from MAE). This effect may be responsible for the best result among the considered loss functions.

### 4.4 Weight regularisation

The weight regularisation has a prominent effect on the results, as revealed in Figure 5. Too much is certainly worse than no weight normalisation, but suitable values significantly reduce bad results.




(a) Distributions of scaled results by empirical cumulative distribution functions for each dataset and loss separately.

| | No reg. | Dropout | Alpha Dropout |
|---|---|---|---|
| No reg. | | **10898** | **14070** |
| Dropout | 4557 | | **12360** |
| Alpha Dropout | 1385 | 3095 | |

(b) Statistical tests for L1 loss function

| | No reg. | Dropout | Alpha Dropout |
|---|---|---|---|
| No reg. | | **10904** | **14020** |
| Dropout | 4551 | | **12352** |
| Alpha Dropout | 1435 | 3103 | |

(c) Statistical tests for L2 loss function

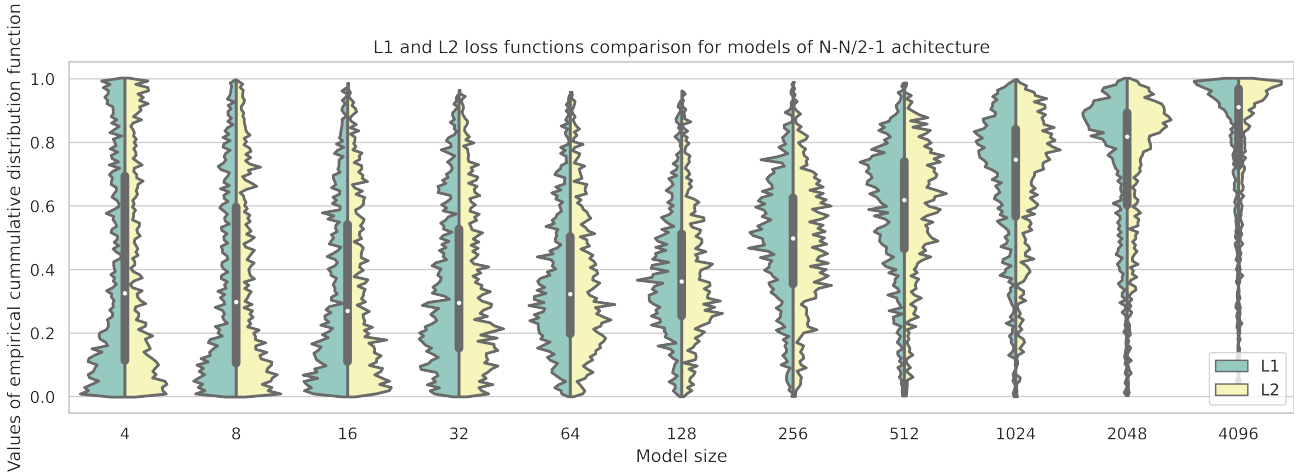
Figure 2: The first experiment compared different kinds of dropout regularizers across all different combinations of datasets and hyperparameters. The tables display the number of experiments where one model size (row) is better than the other (column). If the one-sided Wilcoxon signed-rank test with Holm correction rejected a null hypothesis (column is better than row), the value is highlighted in bold. The statistical tests clearly prefer models without dropout.

| | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | | 580 | 518 | 604 | 683 | 767 | **863** | **975** | **1047** | **1120** | **1177** |
| 8 | **825** | | 636 | 707 | 777 | **827** | **921** | **1045** | **1129** | **1213** | **1238** |
| 16 | **887** | **769** | | **797** | **859** | **871** | **978** | **1096** | **1184** | **1253** | **1275** |
| 32 | **801** | 698 | 608 | | **818** | **878** | **999** | **1122** | **1197** | **1255** | **1270** |
| 64 | **722** | 628 | 546 | 587 | | **885** | **1046** | **1148** | **1222** | **1268** | **1278** |
| 128 | 638 | 578 | 534 | 527 | 520 | | **1031** | **1164** | **1229** | **1263** | **1282** |
| 256 | 542 | 484 | 427 | 406 | 359 | 374 | | **999** | **1151** | **1189** | **1232** |
| 512 | 430 | 360 | 309 | 283 | 257 | 241 | 406 | | **1042** | **1075** | **1187** |
| 1024 | 358 | 276 | 221 | 208 | 183 | 176 | 254 | 363 | | **961** | **1072** |
| 2048 | 285 | 192 | 152 | 150 | 137 | 142 | 216 | 330 | 444 | | **973** |
| 4096 | 228 | 167 | 130 | 135 | 127 | 123 | 173 | 218 | 333 | 432 | |

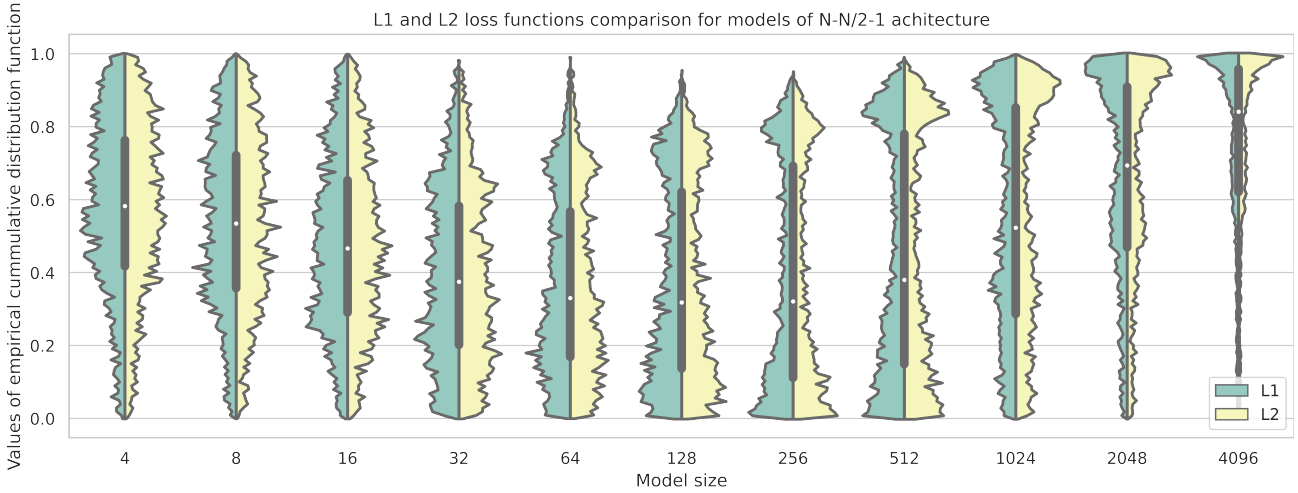
Table 4: The second experiment compares L1 loss for models of different sizes without regularization. The table displays the number of experiments where one model size (row) is better than the other (column). If the one-sided Wilcoxon signed-rank test with Holm correction rejected a null hypothesis (column is better than row), the value is highlighted in bold.

| | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | | 1170 | 948 | 737 | 717 | 795 | 870 | 1030 | 1285 | **1682** | **2087** |
| 8 | **1640** | | 1089 | 802 | 759 | 849 | 926 | 1088 | 1412 | **1855** | **2197** |
| 16 | **1862** | **1721** | | 982 | 892 | 932 | 1024 | 1210 | **1654** | **2065** | **2373** |
| 32 | **2073** | **2008** | **1828** | | 1268 | 1255 | 1247 | **1490** | **1967** | **2335** | **2542** |
| 64 | **2093** | **2051** | **1918** | **1542** | | 1347 | 1355 | **1620** | **2111** | **2438** | **2588** |
| 128 | **2015** | **1961** | **1878** | **1555** | 1463 | | **1406** | **1740** | **2193** | **2453** | **2614** |
| 256 | **1940** | **1884** | **1786** | **1563** | 1455 | 1404 | | **1792** | **2280** | **2514** | **2623** |
| 512 | **1780** | **1722** | **1600** | 1320 | 1190 | 1070 | 1018 | | **2127** | **2417** | **2585** |
| 1024 | **1525** | 1398 | 1156 | 843 | 699 | 617 | 530 | 683 | | **2083** | **2411** |
| 2048 | 1128 | 955 | 745 | 475 | 372 | 357 | 296 | 393 | 727 | | **2050** |
| 4096 | 723 | 613 | 437 | 268 | 222 | 196 | 187 | 225 | 399 | 760 | |

Table 5: The third experiment compares L1 loss for models of different sizes with Dropout or Alpha Dropout regularization. The table displays the number of experiments where one model size (row) is better than the other (column). If the one-sided Wilcoxon signed-rank test with Holm correction rejected a null hypothesis (column is better than row), the value is highlighted in bold.




(a) Distributions of test losses for non-regularized models.




(b) Distributions of test losses for Dropout and Alpha Dropout models combined together.

Figure 3: The second and third experiment analyses the effect of model size on regularized and non-regularized models. The regularisation delays over-training but does not improve the results.

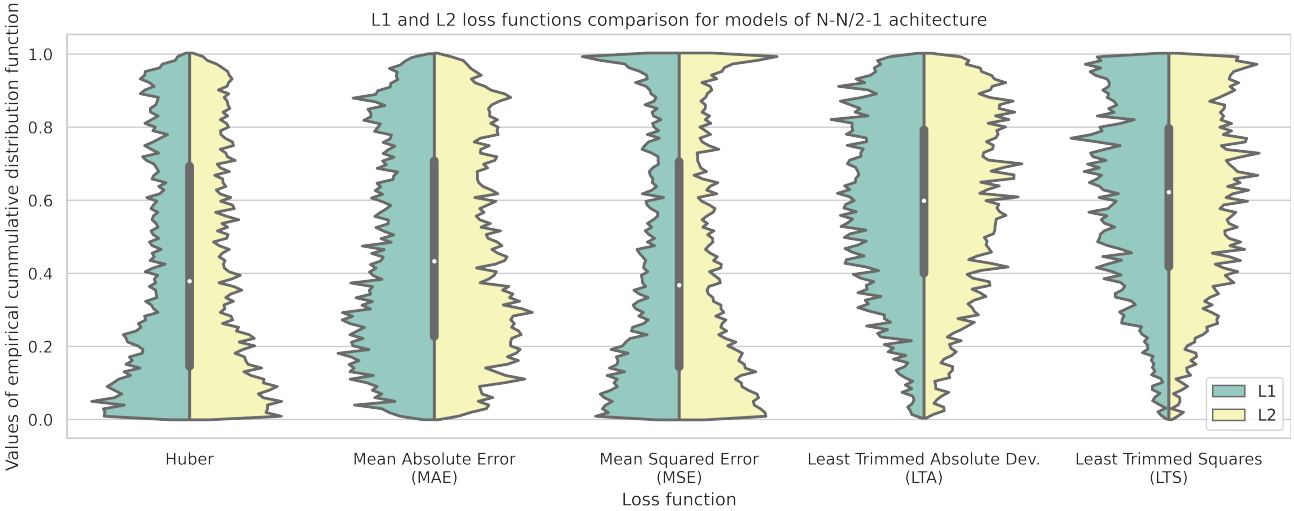



Figure 4: The results of the fourth experiment – comparison between loss functions for non-regularized models. MSE has a lot of good and bad results; Huber seems to be the best; trimmed losses are the worst.

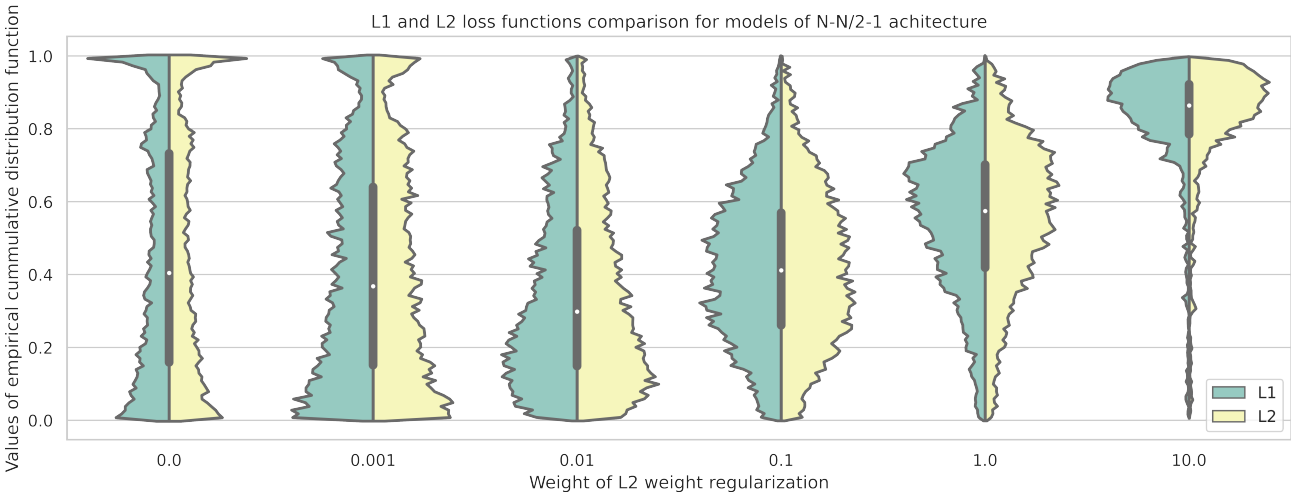



Figure 5: The fifth experiment exposes the effect of L2 normalisation weight.

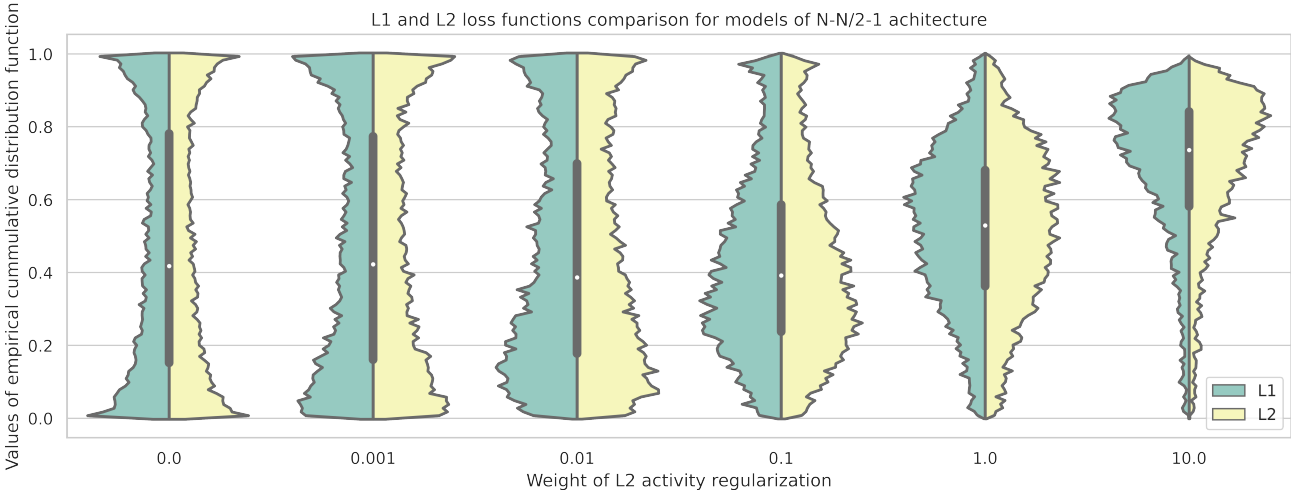



Figure 6: The sixth experiment exposes the effect of L2 activity normalisation weight.

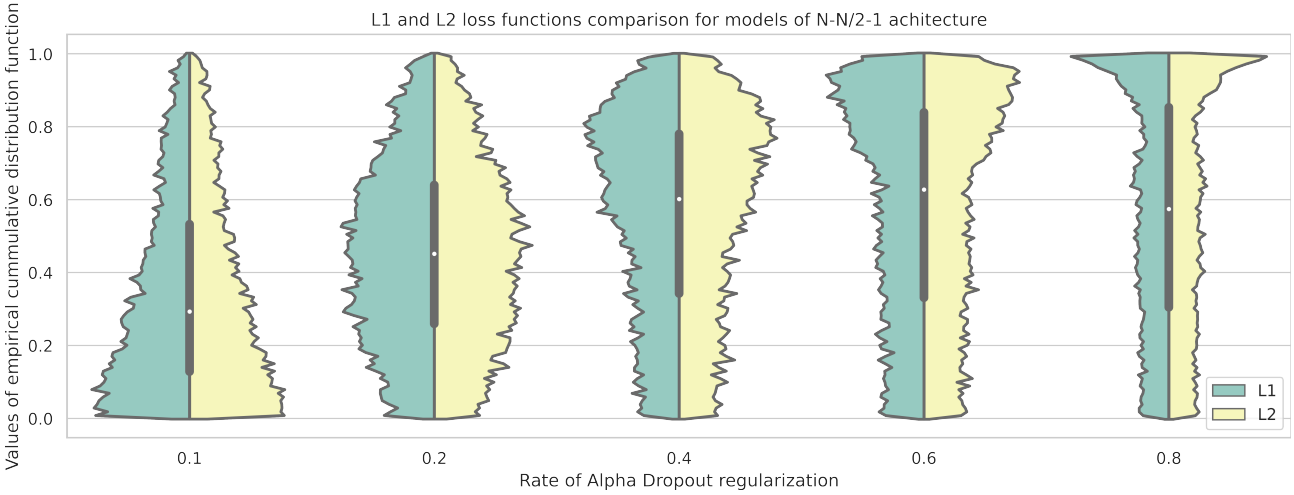



Figure 7: The seventh experiment exposes the effect of Alpha Dropout rate (probability).

| | Huber | MAE | MSE | LTA | LTS |
|---|---|---|---|---|---|
| Huber | | **1854** | **1614** | **2111** | **2184** |
| MAE | 1237 | | 1341 | **2125** | **2171** |
| MSE | 1477 | **1750** | | **2006** | **2085** |
| LTA | 980 | 966 | 1085 | | **1750** |
| LTS | 907 | 920 | 1006 | 1341 | |

Table 6: The fourth experiment compares L1 losses for models trained by optimization of different loss functions without regularization. The table displays the number of experiments where one model size (row) is better than the other (column). If the one-sided Wilcoxon signed-rank test with Holm correction rejected a null hypothesis (column is better than row), the value is highlighted in bold.

| | 0.0 | 0.001 | 0.01 | 0.1 | 1.0 | 10.0 |
|---|---|---|---|---|---|---|
| 0.0 | | 4430 | 3705 | 5053 | **6260** | **8348** |
| 0.001 | **5686** | | 4364 | **5735** | **6987** | **8967** |
| 0.01 | **6411** | **5752** | | **7086** | **8278** | **9601** |
| 0.1 | **5063** | 4381 | 3030 | | **8325** | **9686** |
| 1.0 | 3856 | 3129 | 1838 | 1790 | | **9449** |
| 10.0 | 1768 | 1149 | 515 | 430 | 647 | |

Table 7: The fifth experiment exposes the effect of L2 normalisation weight. The table displays the number of experiments where one weight of the regularization (row) is better than the other (column). If the one-sided Wilcoxon signed-rank test with Holm correction rejected a null hypothesis (column is better than row), the value is highlighted in bold.

| | 0.0 | 0.001 | 0.01 | 0.1 | 1.0 | 10.0 |
|---|---|---|---|---|---|---|
| 0.0 | | **5197** | 4895 | 4813 | **5787** | **7044** |
| 0.001 | 4919 | | 4921 | 4901 | **5833** | **7115** |
| 0.01 | **5221** | **5195** | | 5237 | **6352** | **7654** |
| 0.1 | **5303** | **5215** | 4879 | | **7313** | **8509** |
| 1.0 | 4329 | 4283 | 3764 | 2803 | | **8631** |
| 10.0 | 3072 | 3001 | 2462 | 1607 | 1485 | |

Table 8: The sixth experiment exposes the effect of L2 activity normalisation weight. The table displays the number of experiments where one weight of the regularization (row) is better than the other (column). If the one-sided Wilcoxon signed-rank test with Holm correction rejected a null hypothesis (column is better than row), the value is highlighted in bold.

| | 0.1 | 0.2 | 0.4 | 0.6 | 0.8 |
|---|---|---|---|---|---|
| 0.1 | | **7226** | **7654** | **7614** | **7275** |
| 0.2 | 2890 | | **7075** | **6813** | **6335** |
| 0.4 | 2462 | 3041 | | **6137** | **5684** |
| 0.6 | 2502 | 3303 | 3979 | | 5449 |
| 0.8 | 2841 | 3781 | 4432 | 4667 | |

Table 9: The seventh experiment exposes the effect of Alpha Dropout rate (probability). The table displays the number of experiments where one rate of the regularization (row) is better than the other (column). If the one-sided Wilcoxon signed-rank test with Holm correction rejected a null hypothesis (column is better than row), the value is highlighted in bold.

If the regularisation is too high, the loss is effectively replaced only with the term that reduces weights on the network's connections. If it is too low, the network can lack regularisation – creating potentially volatile responses.

### 4.5 Activity regularisation

In our case, the effect of activity regularisation is similar but smaller than the weight penalty. The difference in weight and activity regularisation effectiveness can be explained by the specific activation used in training. The results are in Figure 6 and in Table 8.

### 4.6 Alpha Dropout rate

In Figure 7 and Table 9 the effects of Alpha Dropout rate can be seen. It may be good to investigate smaller values more because the 0.1 rate is the best. The preference for not having this regularisation can be explained equally as in the subsection 4.1.

## 5 Conclusion

In this paper, we analyzed several types of regularisation techniques on databases where effective hyperparameter optimization is not possible due to the lack of samples or the existence of outliers in the database. We showed that Dropout techniques in these scenarios are not a good choice because their results are not stable enough to compete with models without regularisation. The model's size is an essential aspect, and it seems that the optimum has a far bigger number of free parameters than the theoretical number computed using the average across our training databases. Huber loss function is the best because it does not suffer from inconsistencies of MAE or MSE losses. Trimmed variants of loss functions [21] performed poorly here, but they may be better if a particular dataset has more samples than we had. The third best hyperparameter to look for is the weight normalization – small weight dramatically reduces the frequency of bad results while keeping the median of results low.

## 6 Acknowledgement

The research reported in this paper has been supported by SVV project number 260 575 and partially supported by the Czech Science Foundation (GA ČR) projects 18-18080S and 19-05704S.

Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

## References

[1] Steve Lawrence, Clyde Lee Giles, and Ah Chung Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, Institute for Advanced Computer Studies University of Maryla, 1996.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[4] Radford M. Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

[5] Christopher K. I. Williams. Computing with infinite networks. *Advances in neural information processing systems*, pages 295–301, 1997. morgan kaufmann publishers.

[6] Tamir Hazan and Tommi Jaakkola. Steps toward deep kernel methods from infinite neural networks. *arXiv preprint arXiv:1508.05133*, 2015.

[7] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes, 2017. _eprint: 1711.00165.

[8] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987. Publisher: Elsevier.

[9] Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. *Advances in neural information processing systems*, pages 294–300, 2001. Publisher: MIT; 1998.

[10] Jason Brownlee. *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery, 2018.

[11] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-Normalizing Neural Networks. *arXiv:1706.02515 [cs, stat]*, September 2017. arXiv: 1706.02515.

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, June 2014.

[13] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[14] P. J. Huber. *Robust statistics*. Wiley, New York, 2nd edition, 2009.

[15] Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974. Publisher: Taylor & Francis.

[16] Irad Ben-Gal. Outlier detection. In *Data mining and knowledge discovery handbook*, pages 131–146. Springer, 2005.

[17] Hancong Liu, Sirish Shah, and Wei Jiang. On-line outlier detection and data cleaning. *Computers & Chemical Engineering*, 28(9):1635–1647, 2004.

[18] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient Algorithms for Mining Outliers from Large Data Sets. In *Proceedings of the 2000 ACM SIGMOD Inter-*

*national Conference on Management of Data*, SIGMOD '00, pages 427–438, New York, NY, USA, 2000. Association for Computing Machinery. event-place: Dallas, Texas, USA.

[19] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984. Publisher: Taylor & Francis.

[20] Peter Rousseeuw and Katrien Van Driessen. Computing LTS regression for large data sets. *Data mining and knowledge discovery*, 12(1):29–45, 2006. Publisher: Springer.

[21] J. Kalina and P. Vidnerová. Robust Multilayer Perceptrons: Robust Loss Functions and Their Derivatives. In L. Iliadis, P. Angelov, C. Jayne, and E. Pimenidis, editors, *Proceedings of the 21st EANN (Engineering Applications of Neural Networks) 2020 Conference*, pages 546–557, Cham, 2020. Springer. event-place: Cham.

[22] J. Kalina, A. Neoral, and P. Vidnerová. Effective automatic method selection for nonlinear regression modelling. *International Journal of Neural Systems*, 2021.

[23] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945. Publisher: JSTOR.

[24] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979. Publisher: JSTOR.

[25] Alessio Benavoli, Giorgio Corani, and Francesca Mangili. Should We Really Use Post-Hoc Tests Based on Mean-Ranks? *Journal of Machine Learning Research*, 17(5):1–10, 2016.