Automating the selection of preprocessing techniques for deep neural networks

Student: Marcus September

Supervisor: Francesco Sanna Passino

Department of Mathematics, Imperial College London

marcus.september22@imperial.ac.uk

1. Problem and motivation

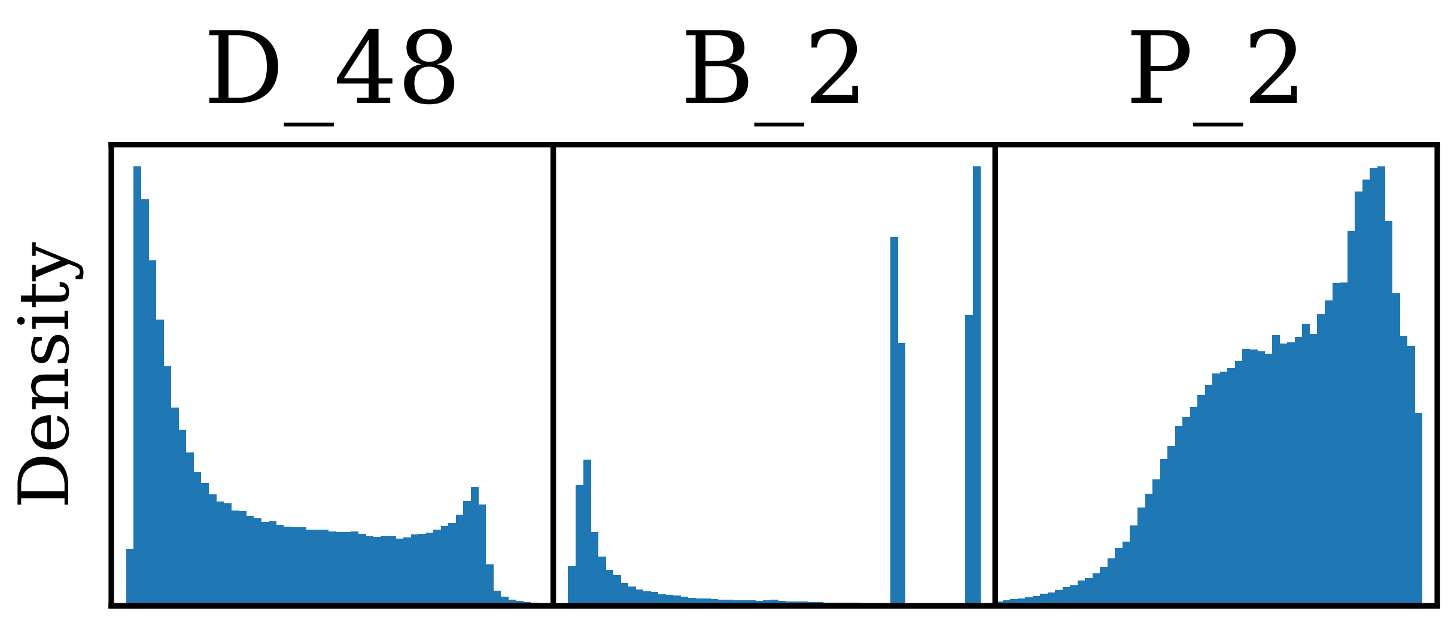
Deep learning sequence models, such as Recurrent Neural Networks (RNNs) and transformers, are sensitive to input variable distributions. Both training speed and performance can drop significantly for non-normal distributions, such as skewed distributions and those with outliers. Preprocessing includes all transformations applied to the data before feeding it into the neural network, and deciding the appropriate techniques is essential for optimising model performance. However, this is a time-consuming process. This project aims to automate this by automatically selecting the preprocessing methods to use for any given sequence dataset, increasing both model performance and training efficiency.

2. Default prediction dataset

The default prediction dataset, provided by American Express, has data from $N \approx 460\,000$ customers, each with $P = 188$ aggregated profile features recorded up to $T = 13$ different credit card statement dates. That is, the dataset has N instances of P-dimensional multivariate time-series of length T. For each multivariate time-series, the target label $y \in \{0, 1\}$ indicates whether the customer defaulted on their loan or not. The task is to predict the probability $\mathbb{P}(Y = 1)$ for each customer. Note that due to privacy concerns, the name of all the features have been anonymized. Additionally, a small amount of uniform noise has been added to all the numeric features.

3. Exploratory data analysis

Figure 1: ***Histogram of 3 of the 188 variables from the default prediction dataset.***



The default prediction dataset exhibits many traits of real-world datasets, such as very skewed distributions, multiple modes, unusual peaks and extreme values. Across the whole dataset, 8.50% of the numeric data points are missing. The 5 most incomplete variables are missing between 91.52% and 92.62% of the data points. Only 138 out of the 177 numeric variables have less than 1% missing values. All this makes the dataset ideal for evaluating how effective the proposed preprocessing techniques are on real-world data.

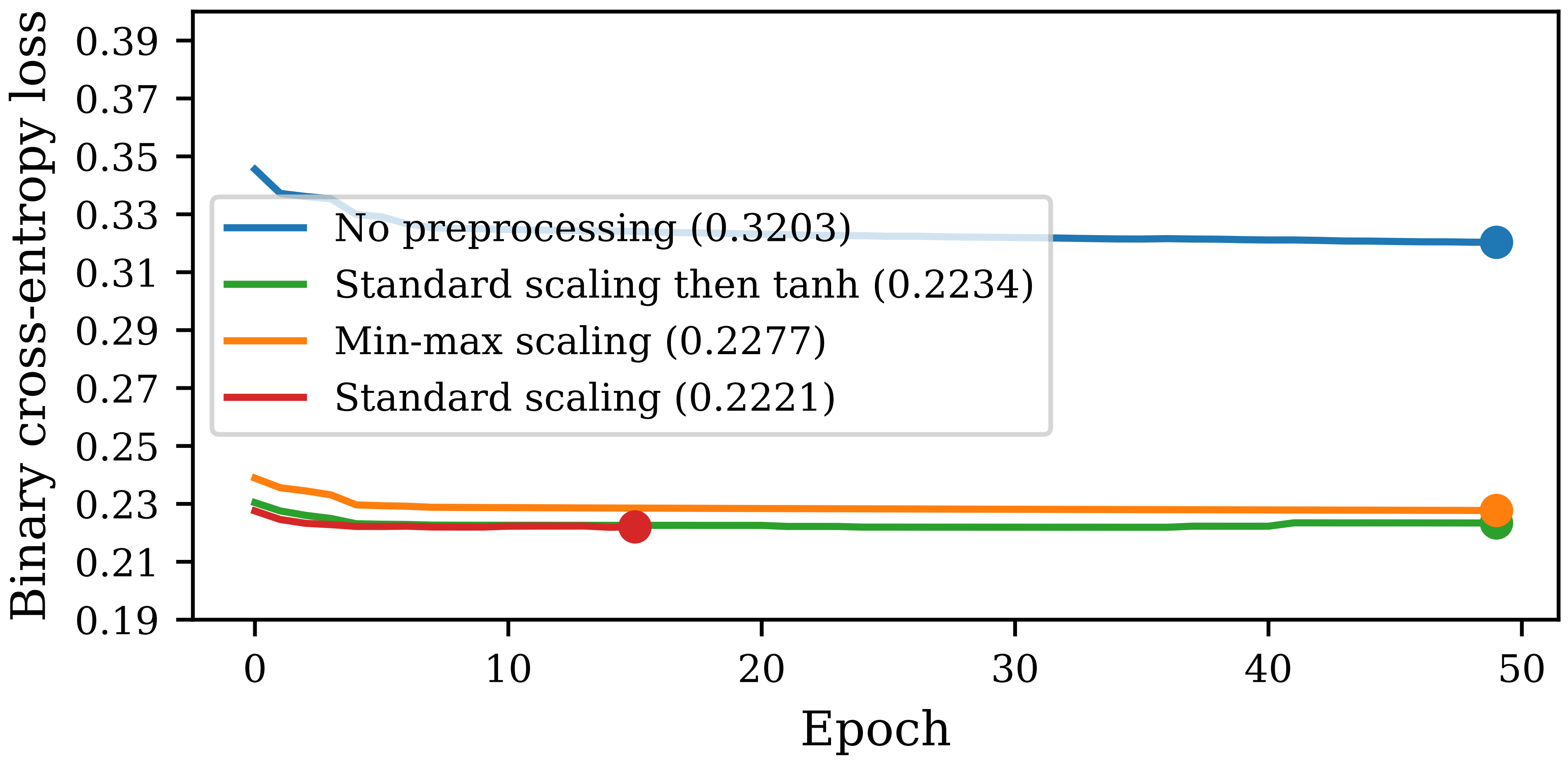
4. Synthetic data

With a custom-made synthetic data generation procedure, I can generate new labelled multivariate time-series where the variables follow arbitrary distributions based on the unnormalized PDFs I provide. This allows synthesizing data with real-world-like distributions in a controlled matter, which makes it easier and more efficiency to experiment and learn when each preprocessing method works best. This insight will then be used to automate the preprocessing step.

5. Preliminary result I: Preprocessing on real data

- Applying appropriate preprocessing can improve both the predictive performance and the training efficiency of deep sequence models (see figure 1a).
- Non-normalized data reduces the predictive power of deep sequence models (see figure 1b/table)

Figure 2: ***Validation loss for different preprocessing techniques***



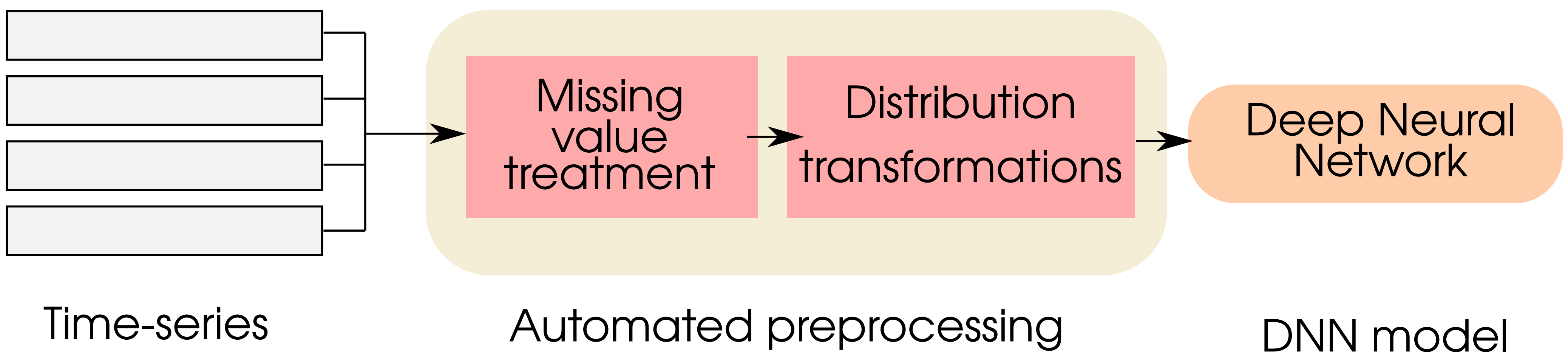
Some text

6. Preliminary result II: Using synthetic data

TODO: also briefly present how this experiment conducted, then present conclusion that having a preprocessing step that has the capability to undo these skewed or non-normal/non-uniform distributions can increase performance. TODO jk: Talk next here about synthetic data results, and not able to learn to undo $F_1^{-1}, \ldots, F_P^{-1}$ transformations.

Dataset	Validation accuracy
Synthetic data with non-normal distributions	95.65%
Synthetic data with uniform distributions	98.65%

7. Automating the preprocessing procedure



To automate the selection of preprocessing techniques, heuristics based on each variable's distribution can be used. Another option is parameterizing the distribution transformations and incorporating it as part of the forward- and back-propagation of the training procedure. This ensures that the optimal transformations are learned automatically.

References

- TODO