Extending EAGGA to Neural Networks

Simon Stürzebecher SIMON.STUERZEBECHER@CAMPUS.LMU.DE

Abstract

Keywords: tabular data, multi-objective optimization, deep learning

1 Introduction

Tabular Data - most common type of data - bad performance of deep learning models on tabular data, recent research suggests that heavy regularisation is necessary

regularisation - LM: ridge (L2) + lasso (L1) * ridge: overcome multi-collinearity by pulling coeffs close to 0 * lasso: variable selection by setting coeffs = 0 - NN: * L2 regularisation implicitly included in SGD when using weight decay (pull params towards 0) -¿ does this combat multi-collinearity as well or serve another purpose? * L1 + key claim of DL: feature selection obsolete, network finds important ones itself + still, feature usually always has some impact (back this up), no explicit feature selection + explicit feature selection simply by not inluding a feature * another classical NN regularisation technique: dropout against co-adaptation * early stopping

2 Background and Related Works

2.1 Explainable Artificial Intelligence and Interpretability

2.1.1 Why is interpretability desirable?

2.1.2 Practical implementations

(1) post-hoc methods - give examples - why are they not perfectly suited (2) model-based

2.2 Automated Machine Learning (AutoML)

2.2.1 Hyperparameter Optimization (HPO)

- grid + random search - Bayesian optimisation -¿ both unsuitable for our task + would be ok for weight decay (L2) + dropout optim + but: approaches for feature selection (L1) not feasible (a) 0-1 penalty on included features added to loss, problems: - how to choose regularisation parameter? - not all features equally important -¿ weighted 0-1 loss? how to choose weights? (b) directly optimise over binary hpyercube -¿ how to optim over non-ordinal discrete space? -¿ evolutionary algorithms can be a solution for these types of problems * CMA-ES * NSGA-II - non-dominated sorting - crowding distance

2.2.2 Neural Architecture Search (NAS)

not sure if super relevant as I don't use a specific NAS technology (e.g. one-shot, etc.) but simply optimise over # layers + # nodes as HPs

2.3 Multi-Objective Optimization

- problem: all aforementioned approaches are a-priori approaches -¿ we choose best HPs (weight decay, features used, dropout) w.r.t. performance - intro MOO

3 EAGGA

- NSGA-II inspired evolutionary / genetic algorithm - introduces group structure for more efficient optimization over the entire search space

4 Extension to neural networks

- why extend it to NNs? * NNs notoriously uninterpretable due to complex transformation of the feature space * due to (lin alg) non-linear activation functions, interactions can be modelled * no monotonicity guarantees - our approach * feature sparsity: straightforward, only train model on selected features, goes somewhat against idea of deep learning, where the model is supposed to decide on its own, which features to put importance on -¿ WHY? * feature interaction - create non-fully connected network by introducing network-groups based on the equivalence relations introduced in the EAGGA paper - show formula why the interacting features need to be grouped together when using ReLU (cf. photos) * monotonicity constraints - for groups with contraint on monotonicity: clip weights to [0, infty), bias clipping not necessary - show formula of how monotonicity is enforced with this * changes (-) + similarities (+) to EAGGA paper (in-between) + 2/3 holdout outer split for 'CV'- vs test-set - train + eval each individual via 5-fold CV with 20% of train set again reserved to determine early stopping -¿ best to visualise, * early stopping necessary (as opposed to XGB) as NN can be trained forever, whereas tree implementations used by EAGGA paper have max depth, etc. * early stopping using min-epochs 200, patience 100, i.e. train for $\geq$ 200 epochs, always safe model with lowest loss, whenever current loss ¿ mean(losses of last 100 (patience) epochs), stop early + return model with lowest loss if no early stopping, train each fold (in 5-fold CV) for max 120 secs, do entire EAGGA loop per dataset for max 8 hrs training on Sagemaker Notebook instance * initial experiments on ml.g4dn.xlarge instance (2 vCPUs, 16GiB RAM, 1 NVIDIA T4, cf. https://aws.amazon.com/de/ec2/instance-types/g4/) using cuda not much faster than on ml.t3.medium (2 Intel Xeon 8000 vCPUs, 4GiB RAM, cf. https://aws.amazon.com/de/ec2/instance-types/t3/) * thus decided for more economic + ressourcenschonend t3.medium - evaluation * AUC + NF: straightforward * NI: for each group with $\geq$ 2 features: sum from 1 to (# features - 1), i.e $\binom{\#features}{2}$, then sum over all groups like this, then divide by p(p-1)/2, i.e. $\binom{p}{2}$ * NNM: # unconstrained features / # all features feature + interaction detector: examination of decision trees not super straightforward in sklearn, hence simply use p=0.5 ? interaction detector (FAST) * train logistic model on 80% of data (only holdout train portion or all ??), evaluate on 20 * use mean accuracy (logistic metric) instead of RSS (LM metric) + monotonicity detector as in

paper, also use dec tree HPs from mlr3 encoding of group structures similar, each feature will get its individual monotonicity sign from the detector, this is then fed to the dataset, which then outputs the data according multiplied with the sign

5 Experimental Results and Discussion

6 Conclusion and Future Outlook

Here is a citation Schneider et al. (2023).

Appendix A.

In this appendix we prove the following theorem from Section 6.2:
Theorem *Let u, v, w be discrete variables such that v, w do not co-occur with u (i.e., $u \neq 0 \;\Rightarrow\; v = w = 0$ in a given dataset $\mathcal{D}$). Let N_{v0}, N_{w0} be the number of data points for which $v = 0, w = 0$ respectively, and let I_{uv}, I_{uw} be the respective empirical mutual information values based on the sample $\mathcal{D}$. Then*

$$N_{v0} \;>\; N_{w0} \;\;\Rightarrow\;\; I_{uv} \;\leq\; I_{uw}$$

with equality only if u is identically 0. ■

Appendix B.

Proof. We use the notation:

$$P_v(i) \;=\; \frac{N_v^i}{N}, \quad i \neq 0; \quad P_{v0} \;\equiv\; P_v(0) \;=\; 1 - \sum_{i \neq 0} P_v(i).$$

These values represent the (empirical) probabilities of v taking value $i \neq 0$ and 0 respectively. Entropies will be denoted by H. We aim to show that $\frac{\partial I_{uv}}{\partial P_{v0}} < 0$....

Remainder omitted in this sample. See http://www.jmlr.org/papers/ for full paper.

References

Lennart Schneider, Bernd Bischl, and Janek Thomas. Multi-objective optimization of performance and interpretability of tabular supervised machine learning models. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 538–547, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701191. doi: 10.1145/3583131.3590380. URL `https://doi.org/10.1145/3583131.3590380`.