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 against overfitting when using iterative method

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 -¿ give intuition why non-dom-sort (asc) + crowding dist (desc) makes sense for selecting individuals

©WS 2024/25 Simon Stürzebecher.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/.

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 (general algorithm) * EAGGA algorithm implemented just as described in the paper * HP init - NN total layers $\in \{3, ..., 10\}$, i.e. hidden layers $\in \{1, ..., 8\}$, init from trunc geom with p=0.5 - NN nodes per hidden layer (for each group) $\in \{3, ..., 20\}$, init from trunc geom with p=0.5 - NN dropout % $\in [0, 1]$, init from trunc gamma with shape=2, scale=0.15 * group structure init - feature detector * as described in paper * but instead of fitting 10 trees + taking rel. # of features used as p for trunc geom, we simply use 0.5 (sklearn dectree examination not straightforward) * also in preliminary experiments we found the sampled # of features to be used occassionally to be ¿ # of non-0 values in normalised information gain filter (e.g. if former is = total # of features and one feature is indep. from target) in these edge cases we only use the features with non-0 filter values - interaction detector * as described in paper * same as for feat detect: use p=0.5 to sample # of interactions used instead of 10 dec trees * also for FAST algorithm don't use RSS (lin reg metric) but mean accuracy, as we fit log reg - monotonicity detector * as described in paper * use default HPs (max depth 30, minsplie = 20) of mlr3 classification tree implementation, as this is the library the paper uses -¿ both interaction + monotonicity detectors fit their models on 80% of train split from holdout + eval on remaining 20% (implementation details) * group structures + datasets - group structures implemented as described in the paper with additional list encoding sign of monotonicity of the individual features as detected by the monotonicity detector - included features are passed to dataset, dataset outputs included features, multiplied by the features' individual monotonicity (-1 / 1) - entire group structure is passed to NN, where architecture is built accordingly * neural network - "instead of XGB as in the original paper, we apply the method to NNs to examine whether this type of regularisation can achieve interpretability on NNs while outperforming EAGGA on XGB (original paper), ..., this requires special architecture, etc. pp." - NN * comprises of "sub-NNs", one for each equivalence relation -¿ basically non-fully connected NN * hidden layers use ReLU activation and dropout afterwards, then a shared output layer with concatenated sub-NNs' outputs as input and sigmoid activation (implicit in the loss function for better numerical stability, NN itself outputs ”logits”, which refers to pre-activation output in pytorch) - feature sparsity thus achieved by only training on included feature groups -¿ goes somewhat against of deep learning where model is supposed to decide itself, which feature to ”use” / put importance on -¿ ELABORATE - feature interaction achieved by grouping, max-operation in ReLU induces interaction effect (different kind of interaction than e.g. in LM with multiplication) -¿ equation why the interacting features need to be grouped together when using ReLU, cf. photos - monotonicty constraint achieved by clipping weights to [0, infty) for restricted equivalence relations (monotonic decrease achieved via dataset object multiplying features with their individual signs), bias clipping not necessary (constant additive term) -¿ equation how monotonicity is enforced with this * evaluation, holdout, cv, early stopping - evaluation via dominated hypervolume along AUC-ROC, NF, NI, NNM as defined in original paper * NF simply rel. # of included features * NI = sum of all possible pairwise interactions in each group over all possible pairwise interactions among all features = $\frac{\sum_g^G \binom{p_g}{2}}{\binom{p}{2}}$ * NNM = rel. # unconstrained features - outer holdout split: 2/3 train, 1/3 test (as in paper) * run EAGGA on holdout train split, i.e. train + select best $\mu = 100$ individuals based on non-dom-sorting (ascending) + crowding distance (descending) as tie breaker - inner CV split on outer train portion: 5-fold (as in paper) * in each fold fit model on 80% of CV-train portion, use remaining 20% of CV-train for early stopping - early stopping criterion * always train for min 200 epochs, keep track of model with lowest loss * after that, use patience of 100: if current model's loss is ¿ mean of last 100 epochs' losses * if no early stopping, train each model for max 10 minutes * then stop training, return model with lowest loss -¿ add graphic of entire dataset split - then evaluate on last fold from CV + keep best μ individuals + generate $\lambda = 10$ offspring for new generation + repeat * hardware: 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 + ressourcen-schonend t3.medium

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.