A modular AutoML System with optimizer selection through low-budget proxy-search-space evaluations

Jan Sander, Ojaswin Khamkar, Jude Mingay

Auto Mates

Modality 3/3

Motivation

- **Problem:** Text classification optimization faces efficiency challenges, i.e exploring vast hyperparameter spaces across diverse architectures under computational constraints leads to suboptimal models.
- **Impact:** Achieve competitive text classification with reduced computational resources while gaining insights into hyperparameter significance.

Contributions

- **DL-Architecture-Agnostic Design**: Unified interface allowing seamless integration of diverse deep learning architectures with their specific preprocessing requirements
- **Low-Resource-Proxy-Search-Space for Optimizer Selection:** Implementation of a custom HPOSuite Benchmark that is similarly structured as the final task but much faster in evaluation.

- Week 1
- Week 2
- Week 3
- Week 4
- Week 5
- Week 6
- Week 7
- Week 8
- Week 9
- Week 10
- Bonus
- Literature

Our Approach

Computational Efficiency

- **Token dropping:** A hyperparameter was implemented that controls what percentage of tokens (padding tokens prioritized) are randomly dropped to improve the training performance of LSTM and Transformer models.
- **Multi-Fidelity Optimization:** We optimized using training epochs as the fidelity dimension.
- **Fewer Data**: We only used 40% of the training dataset for the final optimization.
- **Early Stopping:** A trend-based early stopper with EMA noise filtering was implemented to end poor runs early on when the learning trend falls below a small threshold.

Optimization

- **Frameworks:** NEPS and DEHB were used as main optimization frameworks. NEPS because it was a solid starting point, and DEHB because it seemed likely that it would be able to handle complex, hierarchical and high dimensional search spaces well, as its evolutionary strategy makes it less vulnerable to irrelevant parameters.
- **HPO Optimizer Selection:** For the selection of the HPO optimizer, a more efficient but similar structured proxy search space was constructed and implemented as a HPOSuite benchmark to evaluate PriorBand [1], DEHB [2] and RandomSearch [3].

Resources Used

For development:
- 1 RTX3070 8GB GPU
- AMD Ryzen 5600X 6 Core
- Dev compute estimate: 15 GPU-h

For HPOsuite evaluation:
- 10 GPU-h

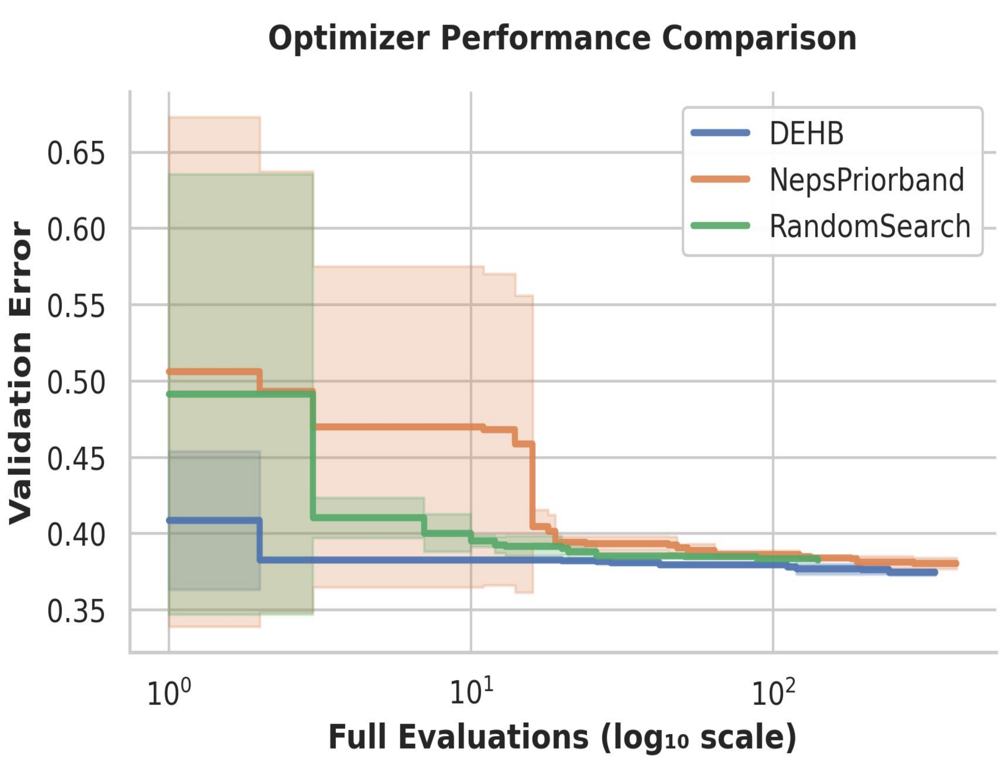
For AutoML:
- 14 GPU-h

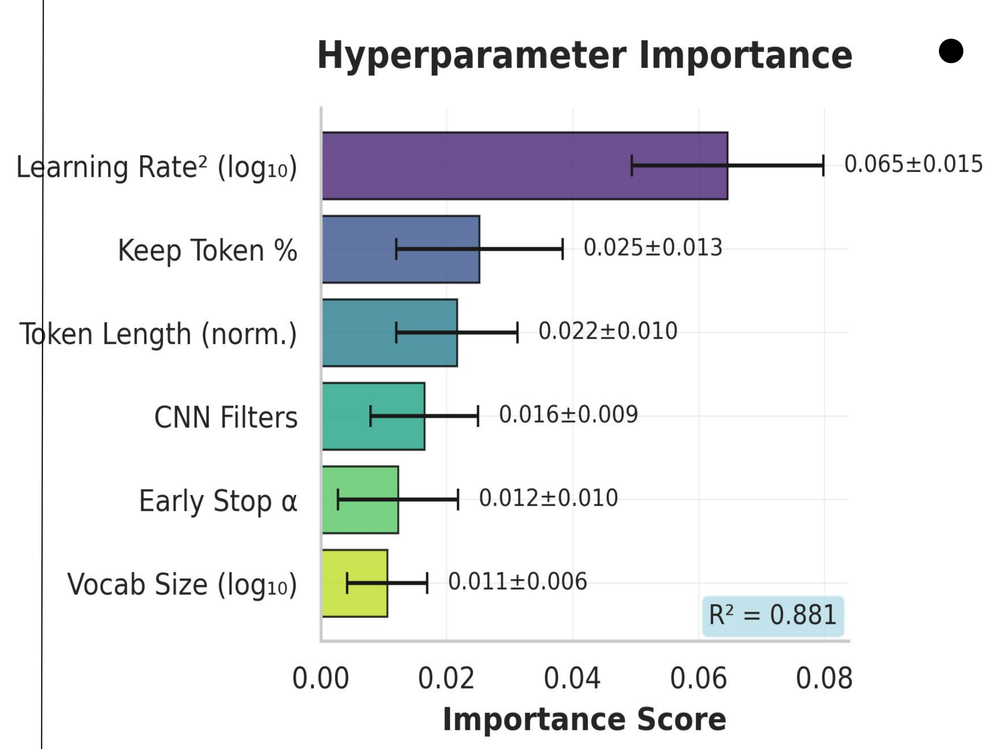
Workforce:
- 1 full week on average

Number of queries for test score generation: **2**

Empirical Results

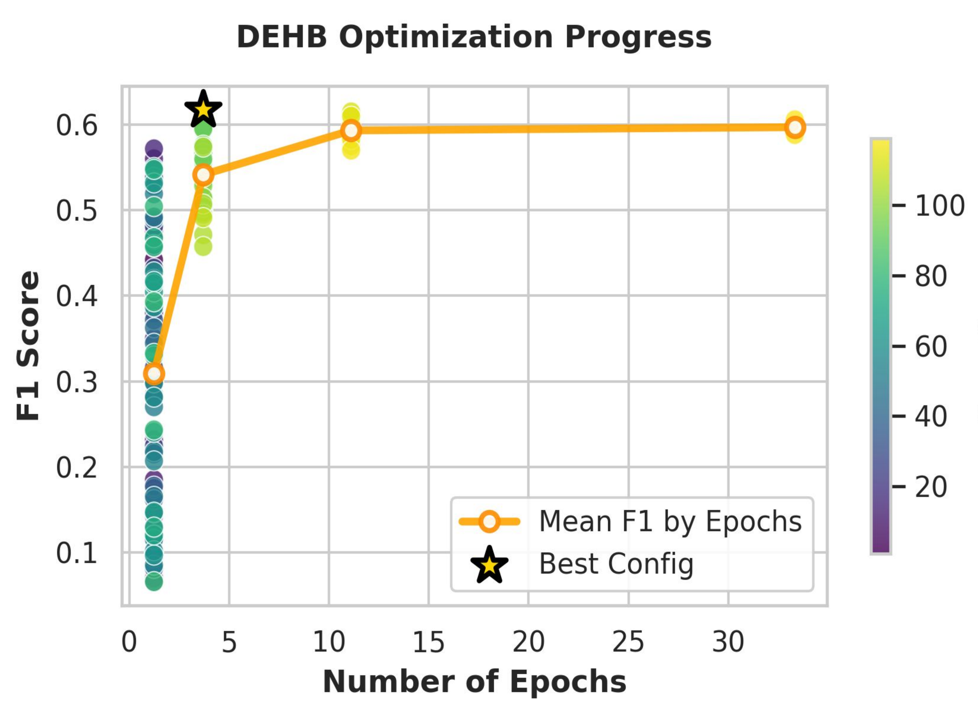
- **Search Strategy:** DEHB significantly outperforms PriorBand, especially in terms of speed, which motivated the decision to use DEHB for the final optimization run.

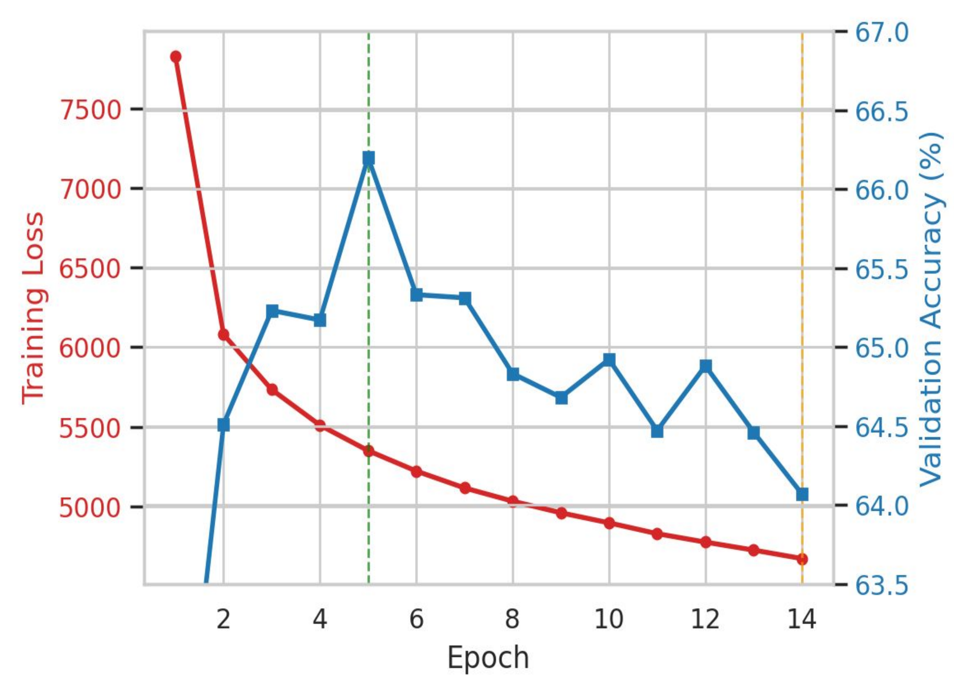




- **Hyperparameter Importance:** The learning rate's impact greatly outshone others, it was followed by the 2 token hyperparameters (Keep token percentage and Token Length)

- **DEHB Optimization:** The optimization ran for a total of 12h, finding a good configuration early on (4h), demonstrating DEHB's fast convergence.





- **Training of best configuration:** The training of the best configuration (LSTM) peaked at 0.662 Validation Accuracy.
- **Test performance**

Accuracy	0.661
Precision (Micro)	0.662
F1 Score (Micro)	0.660

[1] Mallik, N., Bergman, E., Hvarfner, C., Stoll, D., Janowski, M., Lindauer, M., ... & Hutter, F. (2023). Priorband: Practical hyperparameter optimization in the age of deep learning. Advances in Neural Information Processing Systems, 36, 7377-7391.

[2] Awad, N., Mallik, N., & Hutter, F. (2021). Dehb: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization. arXiv preprint arXiv:2105.09821.

[3] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. The journal of machine learning research, 13(1), 281-305.

AutoML.org