TransOpt: Transformer-based Representation Learning for Optimization Problem Classification

1st Gjorgjina Cenikj
Computer Systems Department
Jožef Stefan Institute
Jožef Stefan International
Postgraduate School
Ljubljana, Slovenia
gjorgjina.cenikj@ijs.si

2nd Gašper Petelin
Computer Systems Department
Jožef Stefan Institute
Jožef Stefan International
Postgraduate School
Ljubljana, Slovenia
gasper.petelin@ijs.si

3rd Tome Eftimov
Computer Systems Department
Jožef Stefan Institute
Ljubljana, Slovenia
tome.eftimov@ijs.si

Abstract—In this work, we propose a novel representation of optimization problem instances using a transformer-based neural network architecture trained for the task of problem classification of the 24 problem classes from the Black-box Optimization Benchmarking (BBOB) benchmark. We show that transformer-based methods can be trained to recognize problem classes with accuracies in the range of 70%-80% for different problem dimensions, suggesting the possible application of transformer architectures in acquiring representations for black-box optimization problems.

Index Terms—single-objective continuous optimization, representation learning, problem landscape features

I. Introduction

The representation of optimization problems in terms of numerical features, often referred to as problem landscape features, is essential for the Machine Learning based analyses of the similarity and representativeness of problem instances [1]–[3], automated algorithm selection and configuration. Several types of features capturing properties of single-objective continuous optimization problems have been proposed, which can be broadly categorized as explicitly defined features (such as those based on Fitness Landscape Analysis (FLA) [4], Exploratory Landscape Analysis (ELA) [5] and Topological Landscape Analysis (TLA) [6]), and feature-free approaches [7], [8]. However, FLA features require human effort to be computed, while the ELA features can be computationally expensive to compute for high dimensional problems, have been shown to be sensitive to the sample size and sampling method [9], and are not invariant to transformations such as scaling and shifting of the optimization problem [1].

Motivated by the success of transformer-based models [10] in various fields and applications, in this paper, we explore the possibility of generating vectorized representations of optimization problems through the use of transformer models applied to samples of the optimization problem. In particular, we train a transformer-based model architecture for classifying optimization problem instances from the Black-box Optimization Benchmarking (BBOB) [11], [12] suite into one of the 24 problem classes.

II. Methodology

The Black-box Optimization Benchmarking (BBOB) [11], [12] suite contains 24 single-objective optimization problem classes. Each problem class can further have multiple problem instances, which are a transformation of the original problem class. We use the first 999 instances of each problem class to train the transformer model to predict one of the 24 problem classes. We use problem classes of dimensions 3 and 20.

The first step of the proposed methodology involves the generation of samples from which the problem representations are calculated. The samples are obtained using Latin Hypercube Sampling, and we explore sample sizes of $50d$ and $100d$, where d is the problem dimensionality. The objective values of the samples (y-values) are scaled to be in the range [0,1], while the candidate solutions (x-values) are kept in their original range of [-5,5], since this range is fixed in the BBOB.

Figure 1 shows the model architecture. The input to the model are the samples from the optimization problems. A single training instance of the ML model is a matrix of shape $[s, d+1]$, where s is the number of problem instance samples, and d is the dimensionality of the problem. In this case, the second dimension of the input matrix is $d+1$, since the value of the objective function is also included. The encoder part of the transformer model produces an embedding of size e, which is a specified model parameter, for each of the given samples from the optimization problem, i.e. it outputs a matrix of the shape $[s, e]$. In order to obtain a single, flat representation of the problem, we calculate several descriptive statistics on the matrix obtained from the encoder. In particular, we calculate the minimum, maximum, mean, and standard deviation of the representations of the samples produced by the encoder. Concatenating the vectors obtained using each descriptive statistic, we obtain the representation of the problem, which is then fed into a classification head, which produces the class of the problem. The classification head contains a linear layer with a Rectified Linear Unit activation, a dropout layer, and a linear layer which performs the classification into 24 problem classes.

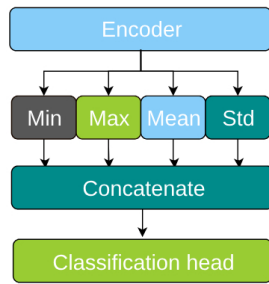The model is trained with a cross-entropy loss, the Adam
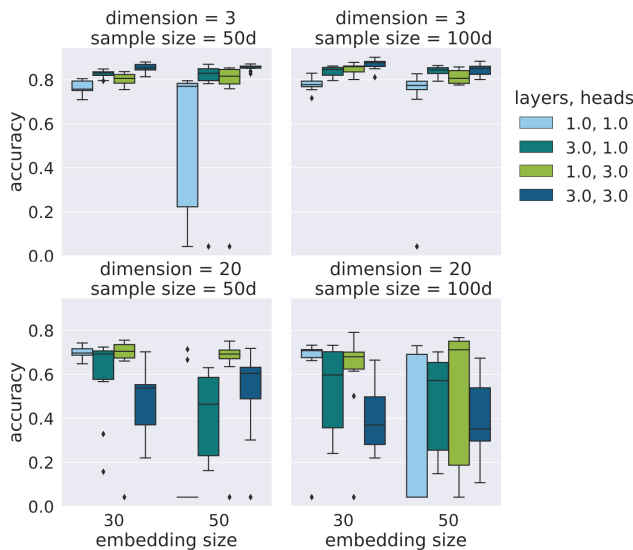
Fig. 1. Model architecture

Fig. 2. Accuracy obtained with different model parameters

optimizer and a maximal learning rate of 0.001. The training is executed for a maximum of 200 epochs, with an early stopping mechanism that prevents overfitting by terminating the training process if the validation loss does not observe a decrease of at least 0.001 for five epochs. We use stratified 10-fold cross-validation to evaluate the model.

III. RESULTS

In order to find a set of reasonable model parameters for further analysis of the architecture, we first ran an initial analysis of the impact of different model parameters on the obtained problem classification accuracy. Figure 2 shows the accuracy obtained in the problem classification task with different parameters for problems of dimension 3 and 20. In particular, we tune the number of heads and layers in the transformer architecture, as well as the dimension of the generated embeddings. The results indicate that increasing the number of heads and layers has a positive impact on the classification accuracy for lower dimensions, however, the same does not always hold for higher dimensions. Increasing the sample size does not seem to substantially impact the classification accuracy for problems of lower dimensions, however, for larger dimensions, the model training is not successful with a sample size of $100d$. In general, smaller

models with lower values of the investigated parameters seem to provide satisfactory results for all dimensions. The best performing configuration is (sample size = $50d$, embedding size = 30, heads = 1, layers = 1), providing accuracies in the range 70%-80% for problems of different dimensions.

IV. CONCLUSION

We propose a novel representation for single objective optimization problem instances, which uses a transformer model trained for the task of problem classification of samples of the problem instances. The proposed model achieves accuracies around 70%-80% for different problem dimensions, indicating its potential for representing high-dimensional problems and possible utilization as end-to-end algorithm selection models.

V. ACKNOWLEDGEMENTS

REFERENCES

[1] U. Škvorc, T. Eftimov, and P. Korošec, "Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis," *Applied Soft Computing*, vol. 90, p. 106138, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620300788

[2] G. Cenikj, R. Dieter Lang, A. Petrus Engelbrecht, C. Doerr, P. Korošec, and T. Eftimov, "SELECTOR: Selecting a Representative Benchmark Suite for Reproducible Statistical Comparison," in *Proceedings of The Genetic and Evolutionary Computation Conference*, 2022, in Press.

[3] M. A. Muñoz and K. Smith-Miles, "Generating new space-filling test instances for continuous black-box optimization," *Evolutionary Computation*, vol. 28, pp. 379–404, 09 2020.

[4] K. M. Malan and A. P. Engelbrecht, "A survey of techniques for characterising fitness landscapes and some possible ways forward," *Information Sciences*, vol. 241, pp. 148–163, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025513003125

[5] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, "Exploratory landscape analysis," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 829–836.

[6] G. Petelin, G. Cenikj, and T. Eftimov, "Tla: Topological landscape analysis for single-objective continuous optimization problem instances," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022, pp. 1698–1705.

[7] M. V. Seiler, R. P. Prager, P. Kerschke, and H. Trautmann, "A collection of deep learning-based feature-free approaches for characterizing single-objective continuous fitness landscapes," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 657–665. [Online]. Available: https://doi.org/10.1145/3512290.3528834

[8] N. van Stein, F. X. Long, M. Frenzel, P. Krause, M. Gitterle, and T. Bäck, "Doe2vec: Deep-learning based features for exploratory landscape analysis," 03 2023.

[9] Q. Renau, C. Doerr, J. Dreo, and B. Doerr, *Exploratory Landscape Analysis is Strongly Sensitive to the Sampling Strategy*, 09 2020, pp. 139–153.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[11] N. Hansen, S. Finck, R. Ros, and A. Auger, "Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions," INRIA, Research Report RR-6829, 2009. [Online]. Available: https://hal.inria.fr/inria-00362633

[12] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff, "Coco: A platform for comparing continuous optimizers in a black-box setting," *Optimization Methods and Software*, pp. 1–31, 2020.