ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1.
Available from http://www.i6doc.com/en/.

Neural-Network-Based Estimation of Normal Distributions in Black-Box Optimization

Jiří Tumpach[1], Jan Koza[2], Zbyněk Pitra[2] and Martin Holeňa[1,2,3] *

1 - Charles University, Prague - Czech Republic

2 - Czech Technical University, Prague - Czech Republic

3 - Czech Academy of Sciences, Prague - Czech Republic

Abstract. The paper presents a novel application of artificial neural networks (ANNs) in the context of surrogate models for black-box optimization, i.e. optimization of objective functions that are accessed through empirical evaluation. For active learning of surrogate models, a very important role plays learning of multidimensional normal distributions, for which Gaussian processes (GPs) have been traditionally used. On the other hand, the research reported in this paper evaluated the applicability of two ANN-based methods to this end: combining GPs with ANNs and learning normal distributions with evidential ANNs. After methods sketch, the paper brings their comparison on a large collection of data from surrogate-assisted black-box optimization. It shows that combining GPs using linear covariance functions with ANNs yields lower errors than the investigated methods of evidential learning.

1 Introduction

One of the key kinds of optimization is nowadays *black-box optimization*, i.e. optimization of objective functions that are accessed not through analytical description, but through empirical evaluation, e.g. measurements, simulation, experiments. Expectedly, it uses optimization methods requiring solely objective function values. They typically need a large number of function values, a serious disadvantage in view of the fact that the empirical evaluation is often expensive in terms of time and/or money. To decrease the number of such expensive evaluations, black-box optimization can be assisted by *surrogate modeling*: the true black-box objective is evaluated only in some points whereas it is predicted with a suitable regression model elsewhere. A surrogate model can be trained using *active learning* – retraining it after including those points not yet originally evaluated that are according to some criterion the most appropriate.

For criteria like expected improvement and probability of improvement, it is necessary that the surrogate model estimates the whole probability distribution of function values. There are two principally different kinds of such estimates:

*The study was supported by the Charles University, project GA UK No. 294422. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic and by the ELIXIR-CZ project (LM2018131), part of the international ELIXIR infrastructure. This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS20/208/OHK3/3T/18.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

empirical probability distributions, and parameterized distributions from an assumed family of distributions with parameters estimated from the data. For the latter kind, the Central Limit Theorem suggests to assume a family of normal distributions unless application-domain reasons indicate a different one. In the surrogate modeling context, empirical probability distributions are used with random forests, and normal distributions with Gaussian processes (GPs).

The last decade brought two approaches in which normal distributions are learned using artificial neural networks (ANNs):

(i) combining GPs with ANNs, in all layers [5], or in the last [12];

(ii) learning normal distributions with evidential ANNs [1, 7, 11].

Both approaches were developed outside the area of surrogate modeling and our research is to our knowledge a first attempt to evaluate their applicability to that area. The paper brings their comparison on a large collection of data from surrogate-assisted black-box optimization. After methods sketch in the subsequent two sections, main results of their comparison are presented in Section 4.

2 Estimation of normal distributions in Gaussian processes

A *Gaussian process* on a set $X \subset \mathbb{R}^d, d \in \mathbb{N}$ is a collection of random variables $(f(x))_{x \in X}$, any finite number of which has a joint Gaussian distribution [10]. It is completely specified by a *mean function* $\mu : X \to \mathbb{R}$, typically assumed constant, and by a *covariance function* $\kappa : X \times X \to \mathbb{R}$ such that for $x, x' \in X$, $\mathbb{E}f(x) = \mu, \mathrm{cov}(f(x), f(x')) = \kappa(x, x')$. A GP is then often denoted or $GP(\mu, \kappa)$.

The value of $f(x)$ is typically accessible only as a *noisy observation* $y = f(x) + \varepsilon$, where ε is a zero-mean Gausssian noise with a variance $\sigma_n > 0$. Then $\mathrm{cov}(y, y') = \kappa(x, x') + \sigma_n^2 \mathbb{I}(x = x')$, where $\mathbb{I}$(proposition) equals for a true proposition 1, for a false proposition 0.

Combining Gaussian processes and neural networks (NN-GP) The approach integrating a GP into an ANN as its output layer was proposed in [12]. It relies on the following two assumptions:

1. ANN with n_I input neurons computes a *mapping* net from $\mathbb{R}^{n_I}$ into the set X on which is the GP. The number of output neurons is the dimension d, and the ANN maps an input v into a point $x = \mathrm{net}(v) \in X$, corresponding to an observation $f(x + \varepsilon)$ governed by GP. From the point of view of the ANN inputs, the GP is now $GP(\mu(\mathrm{net}(v)), \kappa(\mathrm{net}(v), \mathrm{net}(v')))$.

2. The GP mean μ *is assumed to be a known constant*, thus not contributing to the GP hyperparameters and independent of net.

Therefore, the trainable parameters of this combined model are parameters of the GP θ^κ and the network weights θ^W. Consider now n inputs to the neural network, $v_1, \ldots, v_n$, mapped to the inputs $x_1 = \mathrm{net}(v_1), \ldots, x_n = \mathrm{net}(v_n)$ of the GP, corresponding to observations $y = (y_1, \ldots, y_n)^\top$. Then the log-likelihood of θ is $L(\theta) = \ln p(y; \mu, \kappa, \sigma_n^2)$ where μ is the constant assumed in Assumption 2., and $(K)_{i,j} = \kappa(\mathrm{net}(v_i), \mathrm{net}(v_j))$. This allows the model

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

parameters to be trained together using smooth optimization such as gradient descent.

3 Estimation of normal distributions in evidential neural networks

Evidential neural networks learn the parameters of a prior distribution on a set of probabilistic models. A crucial property of evidential ANNs is that they follow the basic principle of the *Dempster-Shafer theory of evidence*, to fall back onto a prior belief for unfamiliar (out-of-distribution) data. A number of evidential ANNs have been proposed (cf. [11] for a survey), the probably best known being *prior networks*, in which the probabilistic models are multidimensional normal distributions $\mathcal{N}(\mu, \Sigma)$ and the network learns the parameters of a normal-inverse-Wishart distribution $\mathcal{NW}^{-1}(m, S, \kappa, \nu)$, which is a conjugate prior to normal distributions.

Density Networks (DN) Neural networks are trainable non-linear functions. Density networks (DN) predict probability density function instead of one value. The networks must be trained according to selected densities. There are several ways to train density networks, the most common is to maximize the likelihood of training samples. In our case, the network predicts normal distribution, so the output consists of two values – the mean value function and the standard deviation. **Ensembles of DN (Ens)** are useful tool to improve overall robustness to out-of-domain inputs [9, 6].

Distillation from the Ens (EnD) An ANN ϕ learning parameters of a normal distribution is said to destil from an ensemble $\theta_1, \ldots, \theta_M$ of DNs if it is most similar to that ensemble in the sense of minimizing the expected KL divergence of the distributions parametrized by the outputs of $\theta_1, \ldots, \theta_M$ from the output of the distribution parametrized by the output of ϕ [9, 8]:

$$\frac{1}{NM} \sum_{i=1}^{N} \sum_{m=1}^{M} \mathrm{KL}\left[\mathrm{p}\left(\boldsymbol{y} \mid \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(m)}\right) \| \mathrm{p}\left(\boldsymbol{y} \mid \boldsymbol{x}^{(i)}, \boldsymbol{\phi}\right)\right] \tag{1}$$

where $\hat{\mathrm{P}}(\boldsymbol{x})$ is the empirical distribution of the ensemble if the input is x, N is the size of training dataset.

Distribution Distillation from the Ens (EnD^2) In this case, the trained network ϕ is above mentioned prior network learning parameters of $\mathcal{NW}^{-1}$. It is trained by minimizing the expected negative log-likelihood of the ensemble's distribution:

$$\mathbb{E}_{\hat{p}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{x})}[-\ln \mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \boldsymbol{x}; \boldsymbol{\phi})] = \mathbb{E}_{\hat{p}(\boldsymbol{x})}[\mathrm{KL}[\hat{\mathrm{p}}(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \boldsymbol{x}) \| \mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \boldsymbol{x}; \boldsymbol{\phi})]] + Z \tag{2}$$

where Z is a constant.

4 Experimental comparison

To assess the performance of the ANN-based surrogate models, we compared their results on an offline dataset extracted from many black-box optimization runs. For the evaluation of the models summarized in previous sections, we used their original Python implementations by the authors of those models?

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

4.1 Employed data

We decided to utilize the large amount of data collected during our previous research using DTS-CMA-ES [2], a surrogate-assisted version of the state-of-the-art black-box optimizer CMA-ES [3]. This allowed us to effectively evaluate the surrogate model on its own without having to repeat the time-consuming optimization. Those data were collected on the *COCO* platform [4], from which we used 23 of the 24 noiseless functions, skipping the easy linear slope in dimensions 2, 3, 5, and 10. We had available altogether more than 1.6 million iterations (CMA-ES generations) each of which apart from the first can be used for evaluating surrogate models, trained on samples from the previous generations. In spite of that large amount of available data, there is typically only a small number of relevant training points in the current search area of DTS-CMA-ES, computed by the original objective function.

4.2 Experiment setup

We compare five different surrogate models. The first is the combined NN-GP outlined in 2 using linear covariance function. The next four models are different variations of the Prior networks described in 3.

For the ANN-GP combination, the above mentioned small amount of available training data in the area searched by CMA-ES incited us to use a multilayer perceptron with a single hidden layer, containing at most 5 neurons. As the activation function for both the hidden and the layer, we chose the logistic sigmoid. We trained the weights and biases of the neural network together with the parameters of the Gaussian process.

The Density Networks have 5 layers of 200 neurons each with ELU activation function and 5% dropout rate. Small ($\sigma = 0.05$) Gaussian perturbation was applied on the input. We use 75-25 train-test split and mini-batches of size 5, 600 warm-up steps with learning rate of 0.001 final learning rate. All submodels in the ensemble were trained in the same way with different initialization. The EnDis trained using a smoothed KL divergence where a temperature is continuously lowered, in order to bring the means closer together. Similarly to the EnD, the EnD^{2}is also trained using temperature. Firstly, it is focused on the mean of the distribution, and later it optimizes all parameters based on the loss function.

4.3 Comparison results

The metric we used to assess the performance of each model is the *ranking difference error* (RDE), which only reflects the ordering of values, due to the invariance of CMA-ES with respect to monotone transformations. Because the CMA-ES algorithm uses the surrogate model to select the most promising candidates for true evaluation, the metric considers only k best samples. The range of the RDE metric is $[0, 1]$, it equals 0 for the exact ordering of the first k smallest values and 1 for the reverse order. The complete formula and description can be found in [2].

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

We aggregated the resulting RDE computed for every CMA-ES generation, with enough training samples in Table 1. In which is clearly visible that the NN-GP model with linear kernel performs better than other examined models. We verified the results using Friedman test, subsequently Wilcoxon signed-rank test with Holm correction was performed for all pairs of tested models. The test confirmed that all differences between models are statistically significant.

Table 1: Aggregated results of the models for each dimension and type of function. SEP – separable functions; MOD – low or moderate conditioning; HC – high conditioning and unimodal; MMA – multi-modal with adequate global structure; MMW – multi-modal with weak global structure.

	f \ κ	NN-GP	DN	Ens	EnD	EnD^2
Dim. 2	SEP	0.252	0.492	0.493	0.497	0.532
	MOD	0.341	0.512	0.511	0.513	0.534
	HC	0.204	0.509	0.508	0.512	0.535
	MMA	0.416	0.512	0.511	0.514	0.535
	MMW	0.443	0.507	0.506	0.510	0.531
	all	0.334	0.502	0.499	0.506	0.533
Dim. 3	SEP	0.194	0.460	0.458	0.466	0.521
	MOD	0.262	0.478	0.477	0.483	0.519
	HC	0.174	0.488	0.487	0.492	0.524
	MMA	0.353	0.495	0.493	0.497	0.522
	MMW	0.387	0.499	0.497	0.503	0.531
	all	0.278	0.474	0.472	0.479	0.524
Dim. 5	SEP	0.227	0.446	0.443	0.452	0.522
	MOD	0.260	0.443	0.440	0.447	0.502
	HC	0.202	0.471	0.468	0.475	0.516
	MMA	0.373	0.480	0.477	0.482	0.511
	MMW	0.363	0.479	0.476	0.485	0.531
	all	0.289	0.453	0.450	0.458	0.517
Dim. 10	SEP	0.278	0.458	0.452	0.463	0.507
	MOD	0.285	0.426	0.422	0.430	0.487
	HC	0.206	0.464	0.460	0.467	0.503
	MMA	0.422	0.500	0.496	0.499	0.518
	MMW	0.363	0.475	0.471	0.480	0.524
	all	0.313	0.449	0.445	0.452	0.497

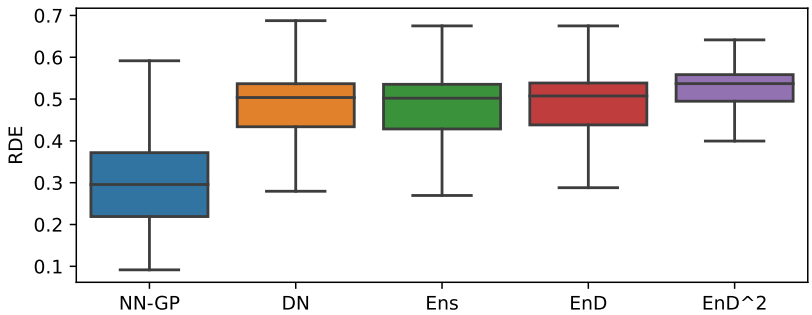

Fig. 1: Overall results of the evaluation. NN-GP model was significantly better than the rest of the models. EnD^2 is the worst but has the narrowest confidence interval. The Ens improve DN only by small insignificant amount.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

5 Conclusion

We examined various ANN based models for normal distribution estimation in the black-box optimization setting. We outlined five different models and evaluated them on large dataset collected in previous research.

The pure deep neural network-based models were significantly worse then the stacking of shallow neural networks on the Gaussian process. It can be explained by the dependence of the deep networks on the performance of density networks (DN). Because DN does not perform well, the ensemble and distillation methods can hardly be better. The main focus should be to optimize the architecture and learning process of the DN or replace it altogether. Needless to say, before any of the investigated approaches can be used in real-world black-box optimization, they have to be compared also with the state-of-the art approach based on GPs alone.

References

[1] A. Amini, W. Schwarting, A. Soleimany, and D. Rus. Deep evidential regression. In *NIPS*, pages 1–11, 2020.

[2] L. Bajer, Z. Pitra, J. Repický, and M. Holeňa. Gaussian process surrogate models for the CMA evolution strategy. *Evolutionary Computation*, 27:665–697, 2019.

[3] N. Hansen. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*, pages 75–102. Springer, 2006.

[4] N. Hansen, A. Auger, R. Ros, O. Merseman, T. Tušar, and D. Brockhoff. COCO: a platform for comparing continuous optimizers in a black box setting. *Optimization Methods and Software*, 35, 2020.

[5] A. Hebbal, L. Brevault, M. Balesdent, E.G. Taibi, and N. Melab. Bayesian optimization using deep Gaussian processes with applications to aerospace system design. *Optimization and Engineering*, 22:321–361, 2021.

[6] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. 12 2016.

[7] A. Malinin, S. Chervontsev, I. Povilkov, and M. Gales. Regression prior networks. ArXiv:2006.11590v2, 2020.

[8] Andrey Malinin, Sergey Chervontsev, Ivan Provilkov, and Mark Gales. Regression prior networks, 2020.

[9] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation, 2019.

[10] E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, 2006.

[11] D. Ulmer. A survey on evidential deep learning for single-pass uncertainty estimation. ArXiv:2110.03051v2, 2021.

[12] A.G. Wilson, Z. Hu, R. Salakhutdinov, and E.P. Xing. Deep kernel learning. In *ICAIS*, pages 370–378, 2016.