s c c h
software competence center
hagenberg

FUZZY.LOGIC.LAB.LINZ

Advances in Knowledge-Based Technologies

Proceedings of the Master and PhD Seminar Winter term 2011/12, part 2

Softwarepark Hagenberg
SCCH, Room 0/2
9 February 2012

Software Competence Center Hagenberg
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 431
Fax +43 7236 3343 434
www.flll.jku.at

Program

Session 1. Chair: Roland Richter

9:00 Gernot Stübl:

Discrepancy Norm as Fitness Function for Defect Detection on Regularly Textured Surfaces

9:30 Henrike Stephani:

Automatic Segmentation and Clustering of Spectral Terahertz Data

Session 2. Chair: Bernhard Moser

10:30 Ciprian Zavoianu:

Performance Optimization of Electrical Drives with Multi-Objective Evolutionary Algorithms and Artificial Neural Networks

11:00 Holger Schöner:

How Deep Learning Makes Sense

Discrepancy Norm as Fitness Function for Defect Detection on Regularly Textured Surfaces

Gernot Stübl[a], Jean-Luc Bouchot[b], Peter Haslinger[a] and Bernhard Moser[a]

[a]Software Competence Center Hagenberg,
Softwarepark 21, A-4232 Hagenberg, Austria
[b]Department of Knowledge-based Mathematical Systems,
Altenbergerstr. 69, A-4040 Linz, Austria

ABSTRACT

This paper addresses the problem of quality inspection of regular textured surfaces as e.g. encountered in industrial woven fabrics. The motivation for developing a novel approach is to utilize the template matching principle for defect detection in a way that does not need any particular statistical, structural or spectral features to be calculated. It is shown that in this context template matching becomes both feasible and effective by exploiting the so-called discrepancy measure as fitness function, leading to a defect detection method that shows advantages in terms of easy configurability and low maintenance efforts.

The proposed concept is a further development of the approach presented by Bouchot et al.[1] which is based on the the registration of test patches in a reference image by optimization methods. The original approach has been underpinned and refined by additional theoretical considerations, regarding estimation of an appropriate window size and the initial guess of the local optimization algorithm. While the latter was achieved by a global optimizer before, it is now performed by a faster randomized initialization step which is inspired by Randomized Sampling Consensus (RANSAC)[2] and the Patchmatch[3] algorithm.

Furthermore a new method for analysing the convergence radius of a (dis-)similarity measure is proposed which allows theoretical statements about the algorithms convergence properties and an optimal parametrization. Evaluations on real world textures complete the paper.

REFERENCES

[1] Bouchot, J.-L., Stübl, G., and Moser, B., "A template matching approach based on the discrepancy norm for defect detection on regularly textured surfaces," in [*Proceedings of the SPIE 10th International Conference on Quality Control by Artificial Vision*], SPIE (June 2011).

[2] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* **24**, 381–395 (June 1981).

[3] Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B., "Patchmatch : A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics TOG* **28**(3), 24 (2009).

Further author information: {gernot.stuebl,peter.haslinger,bernhard.moser}@scch.at, jean-luc.bouchot@jku.at

Automatic Segmentation and Clustering of Spectral Terahertz Data

Henrike Stephani

henrike.stephani@itwm.fraunhofer.de

Abstract

Due to the novelty of the technical approach there still is a big necessity to improve the analysis of hyperspectral Terahertz images. Although it would be desirable to have methods that can be applied on all spectral areas, this is impossible. Depending on the spectroscopic technique, the way the data is acquired differs as well as the characteristics that are to be detected. For these reasons, methods have to be developed or adapted to be especially suitable for the THz range and its applications. Among those are particularly the security sector and the pharmaceutical industry.

Due to the fact that in many applications the volume of spectra to be organized is high, manual data processing is difficult. Especially in hyperspectral imaging, the literature is concerned with various forms of data organization such as feature reduction and classification. In all these methods, the amount of necessary influence of the user should be minimized on the one hand and on the other hand the adaption to the specific application should be maximized.

Therefore, we aim at automatically segmenting or clustering THz-TDS data. To achieve this, we propose a course of action that makes the methods adaptable to different kinds of measurements and applications. State of the art methods will be analyzed and supplemented where necessary, improvements and new methods will be proposed. This course of action includes preprocessing methods to make the data comparable. Furthermore, feature reduction that represents chemical content in about 20 channels instead of the initial hundreds will be presented. Finally the data will be segmented by efficient hierarchical clustering schemes. Various application examples will be shown.

Further work should include a final classification of the detected segments. It is not discussed here as it strongly depends on specific applications.

Performance Optimization of Electrical Drives with Multi-Objective Evolutionary Algorithms and Artificial Neural Networks

Alexandru-Ciprian Zăvoianu [a] Gerd Bramerdorfer [b]
Edwin Lughofer [a] Siegfried Silber [b] Wolfgang Amrhein [b]
Erich Peter Klement [a]

[a] *Department of Knowledge-based Mathematical Systems/Fuzzy Logic Laboratorium Linz-Hagenberg, Johannes Kepler University of Linz, Austria*

[b] *Department for Electrical Drives and Power Electronics, Johannes Kepler University of Linz, Austria*

Abstract

Performance optimization of electrical drives implies a lot of degrees of freedom in the variation of design parameters, which in turn makes the process overly complex and sometimes impossible to handle for classical analytical optimization approaches. This, and the fact that multiple non-independent design parameter have to optimized synchronously, makes a soft computing approach based on multi-objective evolutionary algorithms (MOEAs) a feasible alternative. In this paper, we are applying the MOEA algorithm NSGA-II in order to obtain high-quality Pareto fronts for three optimization scenarios. One key aspect is the usage of artificial neural networks (ANNs) in order to reduce the very huge computational effort caused by using finite element (FE) simulations in the fitness evaluation function. The ANNs are representing non-linear mappings between the design parameters and the targets to be optimized such as material costs, efficiency and torque behavior of the electrical drive. For a large portion of the optimization process, a fitness function based on the ANN mappings is used to perform very fast estimations of target parameter values for newly generated individuals. The results show that the computation time for a single optimization run can be reduced from a *few days to several hours* while achieving Pareto fronts with similar quality as those obtained when conducting FE simulations over the whole life-time of the optimization process.

Key words: electrical drives, performance optimization, design parameters, multi-objective genetic algorithms, feed-forward artificial neural networks

1 Approach

One main method aimed at improving the computational time of a multi-objective evolutionary process (**InitialOpt**) that uses a very time-intensive evaluation function is to approximate the actual function through means of *meta-models* or *surrogate functions* [1]. These should provide a very accurate estimation of the original evaluation function at a fraction of the computational effort required by the latter.

In our case, the idea is to substitute the NSGA-II time-intensive evaluation function based on finite element simulations with a very fast approximation function based on highly accurate regression models, i.e. *mappings* between the design parameters and the target values which should be estimated. As these mappings are specific for each optimization scenario considered, they will be constructed on-the-fly at each run of the evolutionary algorithm. This means that only individuals from the first few generations will be evaluated with the time-intensive FE-based evaluation function in order to construct a training set for the target mappings. For the remaining generations, the mappings will substitute the finite element simulation as the basis of the fitness function. This will yield a significant reduction in computation time, from *a few days to several hours*, as we verified during empirical tests.

At the end of the improved optimization process (**EnhancedOpt**), it is desired that all Pareto solutions found using the mappings are re-evaluated using FE calculations. The main reason for this is to assure that all the simulation solutions presented as Pareto optimal have the same approximation error (i.e., the internal estimation error of FE simulation software).

Artificial Neural Networks (ANNs) [2] are among the popular methods used for constructing mappings for surrogate functions because they possess the universal approximation capability [3] and they offer parameterization options that allow for an adequate degree of control over the complexity of the resulting approximation models. Another advantage of ANNs is the fact that they are known to perform good on non-linear and noisy data [4] and that they have already been successfully applied in evolutionary computation for designing mappings for surrogate functions on several instances [5] [6] [7].

The structures of the initial and of the enhanced multi-objective optimization processes are presented in Figure 1.

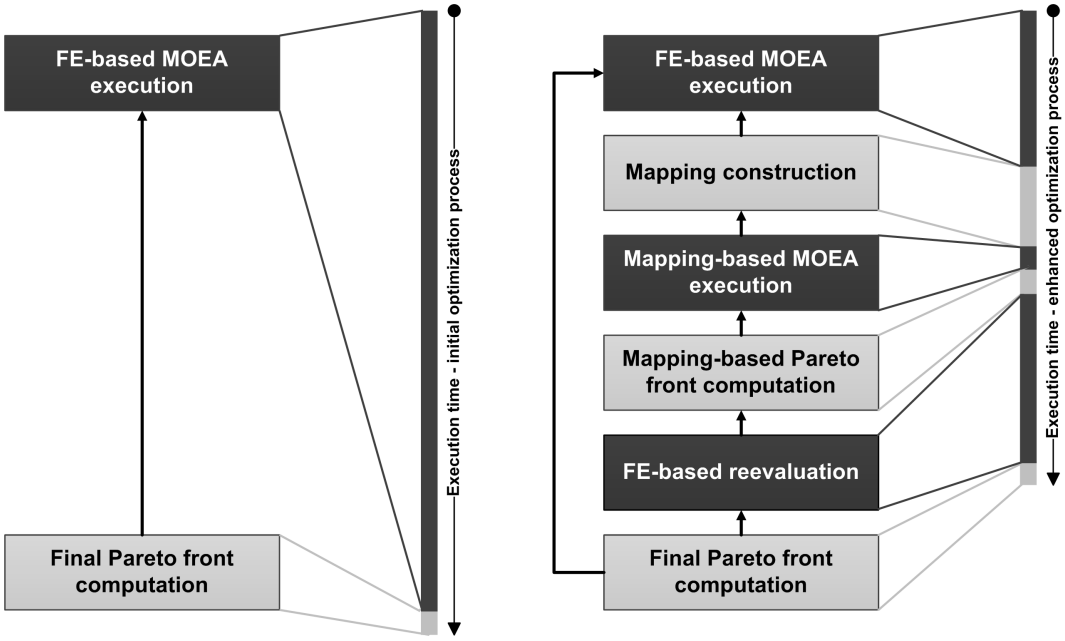

Fig. 1. Diagram of the initial optimization process - InitialOpt (left side) and of the enhanced optimization process - EnahncedOpt (right side)

2 Results

Using 3 fairly complicated optimization scenarios, we compared the performance of InitialOpt and EnhancedOpt after 100 generation runs with a population of 50 individuals. The performance of our method is very good for two scenarios as the resulting final Pareto fronts have slightly better hypervolumes [8], better spreads [9] and were computed $\approx 65\%$ and $\approx 72\%$ faster than their counterparts.

On the third, highly constrained, scenario, the enhanced optimization process performs a little bit worse (the hypervolume is smaller by $\approx 7\%$). The main reason for this is that the hard contraints determine a high ratio of geometrically invalid individuals to be generated during the mapping based evaluation stage (in our optimization processes, geometric validity is carried out during FE simulations). However, the computation time could still be reduced by $\approx 46\%$.

References

[1] L. V. Santana-Quintero, A. A. Montao, C. A. C. Coello, A review of techniques for handling expensive functions in evolutionary multi-objective optimization, in: Y. Tenne, C.-K. Goh, L. M. Hiot, Y. S. Ong (Eds.), Computational Intelligence

in Expensive Optimization Problems, Vol. 2 of Adaptation, Learning, and Optimization, Springer Berlin Heidelberg, 2010, pp. 29–59.

[2] S. Haykin, Neural Networks: A Comprehensive Foundation (2nd Edition), Prentice Hall Inc., Upper Saddle River, New Jersey, 1999.

[3] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (5) (1989) 359 – 366.

[4] M. Paliwal, U. A. Kumar, Neural networks and statistical techniques: A review of applications, Expert Systems with Applications 36 (1) (2009) 2 – 17.

[5] Y. Jin, M. Hüsken, M. Olhofer, B. Sendhoff, Neural networks for fitness approximation in evolutionary optimization, in: Y. Jin (Ed.), Knowledge Incorporation in Evolutionary Computation, Springer Berlin, 2004, pp. 281–305.

[6] M. Papadrakakis, N. D. Lagaros, Y. Tsompanakis, Optimization of large-scale 3-D trusses using evolution strategies and neural networks, International Journal of Space Structures 14 (3) (1999) 211–223.

[7] Y.-S. Hong, H. Lee, M.-J. Tahk, Acceleration of the convergence speed of evolutionary algorithms using multi-layer neural networks, Engineering Optimization 35 (1) (2003) 91–102.

[8] E. Zitzler, Evolutionary algorithms for multiobjective optimization: Methods and applications, Ph.D. thesis, Swiss Federal Institute of Technology (1999).

[9] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang, Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion, in: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, 2006, pp. 892 –899.

s c c h
software competence center
hagenberg

09.02.2012

How Deep Learning Makes Sense

Neural Networks, Feature Learning, and the Strive for Robustness
A presentation for the AdvKBT Seminaöf SCCH and FLLL

Holger Schöner
Software Competence Center Hagenberg
email *holger.schoener@scch.at*

COMET
Competence Centers for
Excellent Technologies

Funded by the Austrian Government
the state of Upper Austria
the Johannes Kepler University

1 How Deep Learning Makes Sense

by Holger Schöner, Software Competence Center Hagenberg

After the initial enthusiasm when first learning Neural Networks (NNs) and their flexibility, there has been quite some potential for disillusion when practically applying them. Beside parameter adaptations and local minima, there has always been the problem of taking advantage of their flexibility in deep architectures. Trained by standard (or even accelerated) backpropagation, the lower layers tend to hardly learn at all. On the other hand, there have been a few architectures reported to perform well [6]. But only in recent years, there have been quite a number of diverse reports about state of the art performance of deep Neural Network architectures in several application domains, especially computer vision.

An important question in this context is, when to use deep architectures. In principle, two-layer NNs are universal approximators, given enough hidden neurons. But certain problems, which are in principle perfectly learnable by a small deep network, can require exponential numbers of hidden neurons for shallow NNs [1]. In consequence it is reasonable to use any domain knowledge available to design good features, and use these in a shallow network, if the problem is made simple enough by the features. In cases, where knowledge about the domain is too limited to design good features, deep learning might be an option to learn good features from the structure of the data. This is done unsupervised and is not necessarily appropriate for any supervised task one wants to perform using these learned features; but often, these features seem to be general enough to support a wide range of learning tasks.

Learning these representations is performed in an unsupervised greedy layerwise manner: using only the input data, a single layer representation is learned using some appropriate unsupervised algorithm. Candidates for such algorithms are Auto-Encoders (AEs), Restricted Boltzman Machine (RBM) layers, or Sparse Predictive Coding (SPC) layers. Conceptually especially simple are AE architectures, with denoising Auto-Encoders and Contractive Auto-Encoders [4] being the best known. Both take the approach of regularizing the representation learned by the encoder to become insensitive to changes in the input; of course, directions present in the training data will lead to variations in the representation anyway (given appropriate weighting of the regularization), because otherwise the inputs could not be reconstructed well. Denoising AEs take a stochastic approach by feeding corrupted (noisy) inputs to the encoder, while requiring the decoder to obtain the original inputs. Contractive AEs enforce the insensitivity in the cost function by requiring the Jacobian of the hidden layer representation wrt. the inputs (corresponding to the sensitivity of the representation wrt. the inputs) to be small.

By training one such layer after the other, with each new layer using as input the output of the previous completely trained layer, one arrives at a deep architecture, with usually more and more high level features in each new layer. Finally, a simple supervised method such as Logistic Regression or a linear NN layer using mean squared cost function can be put on top of these layers, and be trained in a supervised fashion using classical backpropagation or variants, also still slightly adapting the hidden layer representations.

Other design principles leading to well trainable deep architectures include receptive fields, weight sharing, and pooling layers. When using receptive fields, each hidden neu-

ron is not completely connected to all previous layer neurons/inputs, but only locally to such corresponding to its receptive field, leading to a great reduction in parameters to train. This can be well combined with weight sharing, in the sense, that all neurons (for a given feature) in a hidden layer share the weights of the connections to the previous layer, and vary just by their receptive field. This reduces the number of trainable parameters greatly again. Max pooling layers reduce the number of neurons in the next layer, by combining the outputs of neurons in their (not necessarily overlapping) receptive field by some aggregation function (usually the max function), without introducing new trainable parameters.

Deep architectures usually are implemented using Stochastic Gradient Descent, to achieve acceptable learning times despite the usually huge number of parameters [2]. Some examples for applications using approaches based on deep learning are given in [3], [5].

Literatur

[1] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009. ISSN 1935-8237. doi: 10.1561/2200000006. URL `http://www.iro.umontreal.ca/~lisa/publications2/index.php/publications/show/239`.

[2] James Bergstra, Frédéric Bastien, Olivier Breuleux, Pascal Lamblin, Razvan Pascanu, Olivier Delalleau, Guillaume Desjardins, David Warde-Farley, Ian Goodfellow, Arnaud Bergeron, and Yoshua Bengio. Theano: Deep learning on GPUs with python. In *NIPS 2011, BigLearning Workshop*, Granada, Spain, December 2011.

[3] Salah Rifai, Yann N. Dauphin, Pascal Vincent, Yoshua Bengio, and Xavier Muller. The manifold tangent classifier. In *Neural Information Processing Systems (NIPS 25)*, Granada, Spain, December 2011.

[4] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive Auto-Encoders: explicit invariance during feature extraction. In *28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

[5] Ruslan Salakhutdinov, Josh Tenenbaum, and Antonio Torralba. Learning to learn with compound Hierarchical-Deep models. In *Neural Information Processing Systems (NIPS 25)*, Granada, Spain, December 2011.

[6] A. Waibel, T. Hanazawa, Geoffrey E. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time delay neural networks. *IEEE Transactions on Accoustics, Speech and Signal Processing*, 37:328–339, 1989.