Fakultät
IV

Technische Universität Berlin

Prof. Dr.-Ing.
Sikora

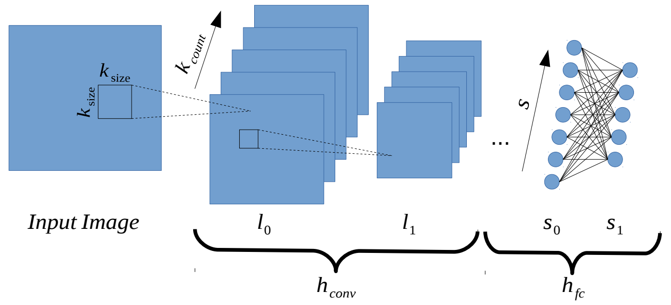
Erik Bochinski, Tobias Senst and Thomas Sikora

Hyper-Parameter Optimization for CNN Committees Based on Evolutionary Algorithms

Introduction

In a broad range of computer vision tasks, convolutional neural networks (CNNs) are one of the most prominent techniques due to their outstanding performance. Yet it is not trivial to find the best performing network structure for a specific application because it is often unclear how the network structure relates to the network accuracy. Instead of doing an "educated guess", we propose an evolutionary algorithm-based framework to automatically optimize the CNN structure by means of hyper-parameters. Furthermore, we extend our framework towards a joint optimization of a committee of CNNs to leverage specialization and cooperation among the individual networks. Experimental results show a significant improvement over the state-of-the-art on the well-established MNIST dataset for hand-written digits recognition.

Problem Definition



The goal is to find an optimal CNN network structure which will be encoded as a set of hyper-parameters $\mathbf{h} = \{\mathbf{h}_{conv}, \mathbf{h}_{fc}\} \in \mathcal{H}$ with:

- $\mathbf{h}_{conv} = \{\mathbf{l}_0, \dots, \mathbf{l}_{n-1},\}$ where $\mathbf{l}_i = (k_{count}, k_{size})$ denotes the configuration tuple of the i^{th} convolutional layer, i.e. number of kernels k_{count} and the kernel size k_{size}.
- $\mathbf{h}_{fc} = \{\mathbf{s}_0, \mathbf{s}_{n-1}\}$ with $\mathbf{s}_i$ being the number of neurons in the fully connected layers.
- no pooling layers are used

Problem:
The solution space is discrete and thus the error function is neither continuous nor differentiable, hence traditional optimization approaches such as gradient-descent are not applicable.

Hyper-parameter optimization using Evolutionary Algorithm

Evolutionary Algorithms (EA) are biologically inspired by Darwin's theory of evolution (survival of the fittest individual in a population P). Hyper-parameters $\mathbf{h}$ are treated as individuals with $\mathbf{l}$, $\mathbf{s}$ as genes.
A common $(\mu + \lambda)$ EA is employed:

- The population at time t is defined as $P^t = \{\mathbf{h}_0, \dots, \mathbf{h}_{\mu-1}\}$
- The initial population P^0 is initialized randomly in predefined boundaries
- P^{t+1} is created by selecting the μ fittest individuals of an interim population $P^t_{\mu+\lambda}$ employing a fitness function $f(\mathbf{h})$
- The interim population $P^t_{\mu+\lambda}$ is composed of P^t and λ offspring individuals

Offspring individuals are created based on randomly selected parent individuals by one-point crossover or mutation:

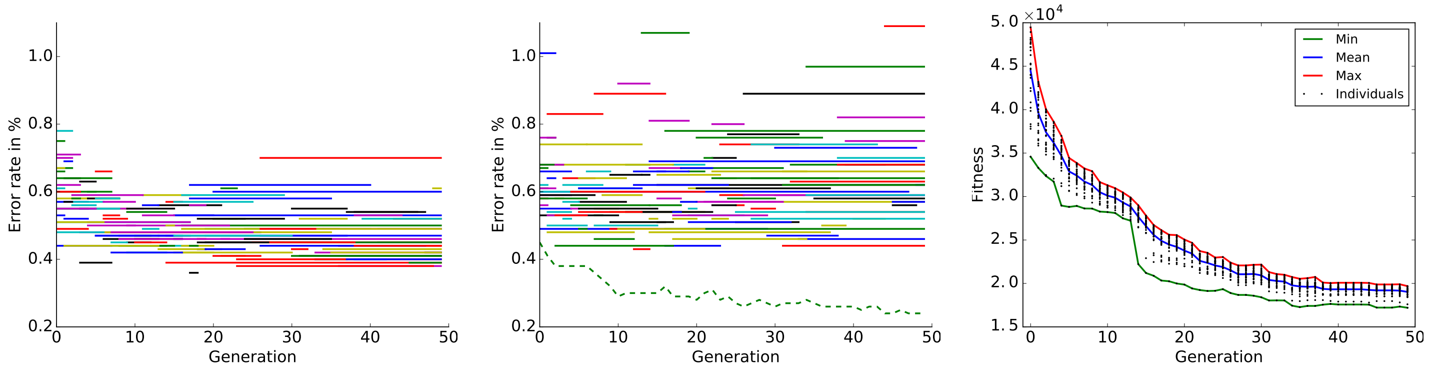
- Alternation of random genes by polynomial bounded mutation function of NSGA-II
- random creation/elimination of genes

The fitness function $f(\mathbf{h})$ equals the classification error $e(\mathbf{h})$ of the respective CNN structure $\mathbf{h}$ trained with the training data and evaluated with validation data of the respective dataset.

Joint Optimization for Committees of multiple CNNs

Committees of multiple CNNs can improve the accuracy if the individual errors are uncorrelated. A committee is a set of K trained CNNs, classification is performed by averaging all CNN decisions. We propose a novel fitness function which promotes specialization and cooperation among the networks:

$$f(\mathbf{h}_m) = \sum_{j}^{N} e_j(\mathbf{h}_m) \cdot \left(\sum_{i}^{\mu+\lambda} e_j(\mathbf{h}_i) \right)^k \quad , \ \mathbf{h}_m, \mathbf{h}_i \in P^t_{\mu+\lambda} \tag{1}$$



(a) Independent CNN optimization

(b) Joint committee optimization for k=2

Distribution of individuals during the hyper-parameter optimization. The error rate of each individual is represented as a line. The dotted line in (b) shows the accuracy of the committee. The right plot (b) shows the distribution and development of the fitness values referring to Eq.1 of the individuals in the population.

where $\mathbf{h}_m$ is the m^{th} hyper-parameter associated with the m^{th} CNN of the current interim population, $e_j(\mathbf{h}) \in \{0, 1\}$ denotes the classification error of the j^{th} sample of the validation dataset of size N and k being a penalty exponent. If the current CNN misclassifies a sample j, the penalization depends on the classification error of the whole population. With a penalization $k > 1$ this effect can be enhanced.

Experiments

Method	Test Error in %
LeNet-5	0.95
Deeply Supervised Net	0.39
Shallow CNN	0.37
Recurrent CNN	0.31
Gated Pooling CNN	0.29
IEA-CNN	0.34
CEA-CNN, k=1	0.26
CEA-CNN, k=2	**0.24**
CEA-CNN, k=3	0.28

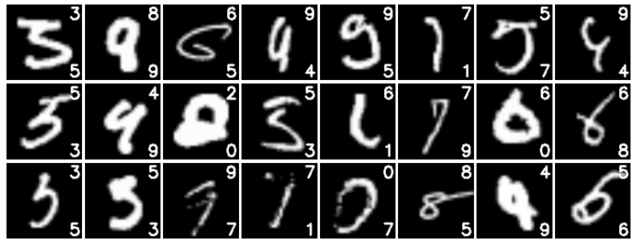
MNIST classification error: comparison of state-of-the-art methods without using any data augmentation techniques.

Experiments were performed on the well-known MNIST dataset. The train data is split into:

- 50,000 CNN training samples
- 10,000 Validation samples for fitness evaluation

Only final testing is performed on the original 10,000 test samples.
Results were obtained for the independent EA-based CNN optimization (IEA-CNN) and joint committee CNN optimization (CEA-CNN) for $k \in \{1, 2, 3\}$:

- IEA-CNN achieves state-of-the-art results but only uses convolutional and fully connected layers
- CEA-CNN outperforms the state-of-the art in all experiments, light penalization k=2 performed best



All 24 classification errors for k=2 of the 10,000 test images. The upper and bottom left numbers indicate the ground truth and classification results respectively.

Conclusion and further work

- Joint EA-based hyper-parameter optimization leverages specialization and cooperation among individual CNNs in a committee
- State-of-the-art classification error for MNIST was considerably improved
- Further work: analyze more datasets and incorporate other building blocks such as LSTMs, residual units or pooling layers

Contact

Web: www.nue.tu-berlin.de
Email: {bochinski,sikora}@nue.tu-berlin.de