

# Fine-Tuning Convolutional Neural Networks Using Harmony Search

Gustavo Rosa<sup>1</sup>, João Papa<sup>1</sup> (✉), Aparecido Marana<sup>1</sup>, Walter Scheirer<sup>2</sup>,  
and David Cox<sup>2</sup>

<sup>1</sup> São Paulo State University, Bauru, São Paulo, Brazil  
{gustavo.rosa,papa,nilceu}@fc.unesp.br

<sup>2</sup> Harvard University, Cambridge, MA 02138, USA  
{wscheirer,davidcox}@fas.harvard.edu

**Abstract.** Deep learning-based approaches have been paramount in the last years, mainly due to their outstanding results in several application domains, that range from face and object recognition to handwritten digits identification. Convolutional Neural Networks (CNN) have attracted a considerable attention since they model the intrinsic and complex brain working mechanism. However, the huge amount of parameters to be set up may turn such approaches more prone to configuration errors when using a manual tuning of the parameters. Since only a few works have addressed such shortcoming by means of meta-heuristic-based optimization, in this paper we introduce the Harmony Search algorithm and some of its variants for CNN optimization, being the proposed approach validated in the context of fingerprint and handwritten digit recognition, as well as image classification.

## 1 Introduction

One of the biggest computer vision problems consists in producing good intern representations of the real world, in such way that these descriptions can allow a machine learning system to detect and classify objects in labels [3]. The problem still persists when faced with situations where there exist variations of luminosity in the environment, as well as different perspectives in the image acquisition process and problems related to rotation, translation and scale.

Traditional machine learning approaches intend to tackle the aforementioned situation by extracting feature vectors with the purpose to feed a classifier by means of a training set, and thereafter classify the remaining images. Thereby, although the feature learning problem has received great attention in the last decades, a considerable effort has been dedicated to the study of deep learning techniques [2, 5, 9]. Despite the fact that there are several deep learning techniques out there, one of the most widely used approaches is the Convolutional Neural Networks (CNN) [9]. These neural networks are composed of different stages and architectures, which are responsible for learning different kinds of information (e.g., images and signals).

The main problem related to such deep learning techniques concerns with the high amount of parameters used to adjust these neural networks to their best performance. Meta-heuristic techniques are among the most used for optimization problems, since they provide simple and elegant solutions in a wide range of applications. Nonetheless, the reader may face just a few and very recent works that handle the problem of CNN optimization by means of meta-heuristic techniques. Fedorovici et al. [6], for instance, employed Particle Swarm Optimization and Gravitational Search Algorithm to select parameter in CNN aiming to cope with optical character recognition applications.

Although some swarm- and population-based optimization algorithms have obtained very promising results in several applications, they may suffer from a high computational burden in large-scale problems, since there is a need for optimizing all agents at each iteration. Some years ago, Geem [7] proposed the Harmony Search (HS) technique, which falls in the field of meta-heuristic optimization techniques. However, as far as we know, Harmony Search and some of its variants have never been applied for CNN optimization.

Therefore, the main contributions of this paper are twofold: (i) to introduce HS and some of its variants to the context of CNN fine-tuning for handwritten digits and fingerprint recognition, as well as for image classification, and (ii) to fill the lack of research regarding CNN parameter optimization by means of meta-heuristic techniques. The remainder of this paper is organized as follows. Sections 2 and 3 present the Harmony Search background theory and the methodology, respectively. Section 4 discusses the experiments and Section 5 states conclusions and future works.

## 2 Harmony Search

Harmony Search is a meta-heuristic algorithm inspired in the improvisation process of music players. Musicians often improvise the pitches of their instruments searching for a perfect state of harmony [7]. The main idea is to use the same process adopted by musicians to create new songs to obtain a near-optimal solution according to some fitness function. Each possible solution is modeled as a harmony, and each musical instrument corresponds to one decision variable.

Let  $\phi = (\phi_1, \phi_2, \dots, \phi_N)$  be a set of harmonies that compose the so-called “Harmony Memory” (HM), such that  $\phi_i \in \mathbb{R}^M$ . The HS algorithm generates after each iteration a new harmony vector  $\hat{\phi}$  based on memory considerations, pitch adjustments, and randomization (music improvisation). Further, the new harmony vector  $\hat{\phi}$  is evaluated in order to be accepted in the harmony memory: if  $\hat{\phi}$  is better than the worst harmony, the latter is then replaced by the new harmony. Roughly speaking, HS algorithm basically rules the process of creating and evaluating new harmonies until some convergence criterion.

In regard to the memory consideration step, the idea is to model the process of creating songs, in which the musician can use his/her memories of good musical notes to create a new song. This process is modeled by the Harmony Memory Considering Rate (*HMC*R) parameter, which is the probability of choosing one

value from the historic values stored in the harmony memory, being  $(1 - HMCR)$  the probability of randomly choosing one feasible value<sup>1</sup>, as follows:

$$\hat{\phi}^j = \begin{cases} \phi_A^j & \text{with probability } HMCR \\ \theta \in \Phi_j & \text{with probability } (1 - HMCR), \end{cases} \tag{1}$$

where  $j \in \{1, 2, \dots, M\}$ ,  $A \sim \mathcal{U}(1, 2, \dots, N)$ , and  $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_M\}$  stands for the set of feasible values for each decision variable<sup>2</sup>.

Further, every component  $j$  of the new harmony vector  $\hat{\phi}$  is examined to determine whether it should be pitch-adjusted or not, which is controlled by the Pitch Adjusting Rate ( $PAR$ ) variable, according to Equation 2:

$$\hat{\phi}^j = \begin{cases} \hat{\phi}^j \pm \varphi_j \varrho & \text{with probability } PAR \\ \hat{\phi}^j & \text{with probability } (1 - PAR). \end{cases} \tag{2}$$

The pitch adjustment is often used to improve solutions and to escape from local optima. This mechanism concerns shifting the neighbouring values of some decision variable in the harmony, where  $\varrho$  is an arbitrary distance bandwidth, and  $\varphi_j \sim \mathcal{U}(0, 1)$ .

### 2.1 Improved Harmony Search

In the last years, several researches have attempted to develop variants based on the original HS [1] in order to enhance its accuracy and convergence rate. Some works have proposed different ways to dynamically set the HS parameters, while others suggested new improvisation schemes. Mahdavi et al. [11], for instance, proposed a new variant called Improved Harmony Search (IHS), which taps a new scheme that improves the convergence rate of the Harmony Search algorithm. In other words, the IHS algorithm differs from traditional HS as it updates dynamically its  $PAR$  and distance bandwidth values during every new improvisation step.

As stated before, the mainly difference between IHS and traditional HS algorithm is how they adjust and update their  $PAR$  and bandwidth values. In order to pursue this goal and to eradicate the handicaps that come up with fixed values of  $PAR$  and  $\varrho$ , the IHS algorithm changes their values according to the iteration number.

### 2.2 Global-Best Harmony Search

Some concepts of swarm intelligence algorithms, as the ones presented in Particle Swarm Optimization (PSO) [4, 8], have been used to enhance the Harmony

<sup>1</sup> The term “feasible value” means the value that falls in the range of a given decision variable.

<sup>2</sup> Variable  $A$  denotes a harmony index randomly chosen from the harmony memory.

Search algorithm in order to improve its effectiveness on either discrete and continuous problems. The so-called Global-best Harmony Search (GHS) [12] applies this technique by modifying the pitch-adjustment step of the IHS, so that its new harmony value is represented by the best harmony found in the Harmony Memory. Thereby, the distance bandwidth parameter  $\varrho$  is deserted off the improvisation step, so that the decision variable  $j$  of the new harmony is computed as follows:

$$\hat{\phi}^j = \phi_{best}^j, \tag{3}$$

where *best* stands for the index of the best harmony in the HM.

### 2.3 Self-Adaptive Global-Best Harmony Search

Unlikely and inspired by its predecessor (i.e., GHS), the self-adaptive Global-best Harmony Search (SGHS) algorithm [13] applies a new improvisation method and some fine-tuning adaptive parameter procedures. During the memory consideration step, in order to avoid getting trapped at a local optimum solution, Equation 1 is replaced as follows:

$$\hat{\phi}^j = \begin{cases} \phi_A^j \pm \varphi_j \varrho & \text{with probability } HMCR \\ \theta \in \Phi_j & \text{with probability } (1 - HMCR). \end{cases} \tag{4}$$

Since *HMCR* and *PAR* variables are dynamically updated during the iteration process by recording their previous values in accordance to the generated harmonies, we assume their values are drawn from normal distributions, i.e.,  $HMCR \sim \mathcal{N}(HMCR_m, 0.01)$  and  $PAR \sim \mathcal{N}(PAR_m, 0.05)$ . The variables  $HMCR_m$  and  $PAR_m$  stand for the average values of *HMCR* and *PAR*, respectively.

In order to well-balance the algorithm exploitation and exploration processes, the bandwidth parameter  $\varrho$  is computed as follows:

$$\varrho(t) = \begin{cases} \varrho_{max} - \frac{\varrho_{max} - \varrho_{min}}{T} \times 2t & \text{if } t < T/2 \\ \varrho_{min} & \text{if } t \geq T/2. \end{cases} \tag{5}$$

## 3 Methodology

### 3.1 Experimental Setup

In this work, we proposed the fine-tuning of CNN parameters using Harmony Search-based algorithms, as well as using a random initialization of the parameters (RS). We have employed three HS variants: (i) Improved Harmony Search [11], (ii) Global-best Harmony Search [12], and (iii) Self-adaptive Global-best Harmony Search [13]. In order to provide a statistical analysis by means

of Wilcoxon signed-rank test [15], we conducted a cross-validation with 10 runnings. Finally, we employed 15 harmonies over 250 iterations for convergence considering all techniques. Table 1 presents the parameter configuration for each optimization technique<sup>3</sup>.

**Table 1.** Parameter configuration.

Technique	Parameters
HS	$HMCR = 0.7, PAR = 0.5, \varrho = 0.1$
IHS	$HMCR = 0.7, PAR_{MIN} = 0.0$ $PAR_{MAX} = 1.0, \varrho_{MIN} = 0.0$ $\varrho_{MAX} = 0.1$
GHS	$HMCR = 0.7, PAR_{MIN} = 0.1$ $PAR_{MAX} = 1.0$
SGHS	$HMCR_m = 0.98, PAR_m = 0.9$ $\varrho_{MIN} = 0.0, \varrho_{MAX} = 0.1$ $LP = 100$

### 3.2 Datasets

In regard to the parameter optimization experiment, we employed two datasets, as described below:

- MNIST dataset<sup>4</sup>: it is composed of images of handwritten digits. The original version contains a training set with 60,000 images from digits ‘0’-‘9’, as well as a test set with 10,000 images<sup>5</sup>.
- CIFAR-10 dataset<sup>6</sup>: is a subset image database from the “80 million tiny images” dataset, collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Composed by 60,000 32x32 colour images in 10 classes, with 6,000 images per class. It is also divided into five training batches and one test batch, each one containing 10,000 images. Therefore we have 50,000 images for training purposes and 10,000 for testing duties.

In regard to the source-code, we used the well-known Caffe library<sup>7</sup> [16], which is developed under GPGPU (General-Purpose computing on Graphics Processor Units) platform, thus providing a more efficient implementations.

## 4 Experimental Results

In this section, we present the experimental results over MNIST and CIFAR-10 datasets. We employed the very same architecture proposed by Caffe<sup>8</sup> to handle

<sup>3</sup> Notice these values have been empirically chosen.

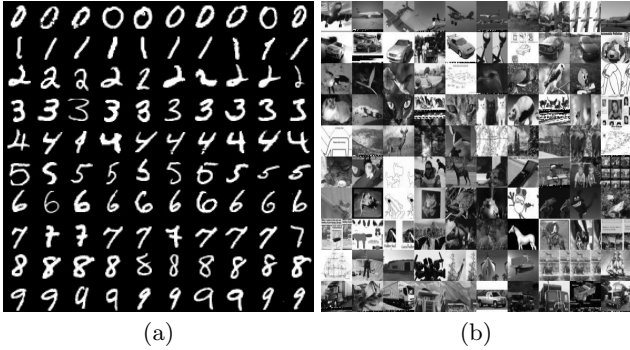
<sup>4</sup> <http://yann.lecun.com/exdb/mnist/>

<sup>5</sup> The images are originally available in gray-scale with resolution of  $28 \times 28$ .

<sup>6</sup> <http://www.cs.toronto.edu/~kriz/cifar.html>

<sup>7</sup> <http://caffe.berkeleyvision.org>

<sup>8</sup> <http://caffe.berkeleyvision.org/gathered/examples/mnist.html>



**Fig. 1.** Some training examples from (a) MNIST and (b) CIFAR-10 datasets.

MNIST dataset, which is composed of two layers with convolution and pooling operations.

Table 2 presents the mean accuracy and the standard deviation over the testing set using the best parameters found out by HS-based algorithms, random search (RS) and the set of parameters employed by Caffe library itself. According to the Wilcoxon signed-rank test, we have bolded the most accurate techniques in Table 2. Additionally, we also show the number of calls to the CNN learning procedure to give us an idea about the computational burden of each technique.

**Table 2.** Experimental results concerning MNIST dataset.

Technique	Final Accuracy (test set)	#calls
Caffe	99.07%±0.03	1
RS	98.70%±0.56	1
HS	99.23%±0.04	265
<b>IHS</b>	<b>99.24%±0.03</b>	<b>265</b>
<b>GHS</b>	<b>99.24%±0.08</b>	<b>265</b>
<b>SGHS</b>	<b>99.29%±0.06</b>	<b>265</b>

Considering the experimental results, we can drive some conclusions here: HS-based techniques seem to be very suitable for CNN optimization, since they achieved better results than a simple random search algorithm. This statement is very interesting, since most part of works employ a random search to fine-tune CNNs. Other conclusion concerns with the HS variants, for example, IHS: it seems to be slightly more important to update the *PAR* parameter dynamically regarding to the vanilla harmony search algorithm, but is still also better to consider the best harmony’s values when creating the new harmony memory, as employed by GHS and SGHS.

Currently, the best error rate we have obtained was around 0.63% with SGHS technique, being one of the best errors up to date obtained by the work of

Wan et al. [14] (0.21%). However, their work employed a different technique of the one applied in this paper. Thereafter, we are not concerned in improving the top results, but to stress we can turn the results better by using meta-heuristic techniques instead of a random search. We have shown here we can improve the results obtained by Caffe library itself by means of a proper selection of the CNN parameters.

In regard to “CIFAR-10 Dataset” experiments, we employed the CIFAR-10 quick model<sup>9</sup>, which is composed of three layers with one convolution and one pooling operation each. We bolded the most accurate techniques in Table 3 according to the Wilcoxon signed-rank test. Once again, SGHS has obtained the top results concerning CNN fine-tuning, which can be a promising indicator of the suitability of this technique to this context.

**Table 3.** Experimental results concerning CIFAR-10 dataset.

Technique	Final Accuracy (test set)	#calls
Caffe	71.51%±0.77	1
RS	66.97%±1.39	1
<b>HS</b>	<b>72.28%±0.37</b>	<b>265</b>
IHS	71.54%±0.09	265
GHS	71.86%±0.10	265
<b>SGHS</b>	<b>72.43%±0.19</b>	<b>265</b>

## 5 Conclusions

In this paper, we dealt with the problem of CNN model selection by means of meta-heuristic techniques, mainly the ones based on the Harmony Search. We conducted experiments in two public datasets for CNN fine-tuning considering HS and three variants, as well a random search and the set of parameters suggested by the open-source code we have used. The results highlighted here allow us to conclude that HS-based techniques are a suitable approach for CNN optimization, since they outperformed other techniques compared in this work.

**Acknowledgments.** The authors are grateful to FAPESP grants #2013/20387-7, #2014/09125-3, #2014/16250-9 and #2014/24491-6, and CNPq grants #303182/2011-3 and #470571/2013-6.

## References

1. Alia, O., Mandava, R.: The variants of the harmony search algorithm: an overview. *Artificial Intelligence Review* **36**, 49–68 (2011)

<sup>9</sup> <http://caffe.berkeleyvision.org/gathered/examples/cifar10.html>

2. Arel, I., Rose, D., Karnowski, T.: Deep machine learning - a new frontier in artificial intelligence research [research frontier]. *IEEE Computational Intelligence Magazine* **5**(4), 13–18 (2010)
3. Bishop, C.: *Neural networks for pattern recognition*. Oxford University Press (1995)
4. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, pp. 39–43 (1995)
5. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**, 1915–1929 (2013)
6. Fedorovici, L.O., Precup, R.E., Dragan, F., Purcaru, C.: Evolutionary optimization-based training of convolutional neural networks for ocr applications. In: *2013 17th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 207–212, October 2013
7. Geem, Z.W.: *Music-Inspired Harmony Search Algorithm: Theory and Applications*, 1st edn. Springer Publishing Company, Incorporated (2009)
8. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 1942–1948. IEEE Press (1995)
9. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11), 2278–2324 (1998)
10. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 253–256 (2010)
11. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation* **188**(2), 1567–1579 (2007)
12. Omran, M.G., Mahdavi, M.: Global-best harmony search. *Applied Mathematics and Computation* **198**(2), 643–656 (2008)
13. Pan, Q.K., Suganthan, P., Tasgetiren, M.F., Liang, J.: A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation* **216**(3), 830–848 (2010)
14. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: Dasgupta, S., Mcallester, D. (eds.) *Proceedings of the 30th International Conference on Machine Learning*. vol. 28, pp. 1058–1066. *JMLR Workshop and Conference Proceedings* (2013), <http://jmlr.org/proceedings/papers/v28/wan13.pdf>
15. Wilcoxon, F.: Individual comparisons by ranking methods. *Biometrics Bulletin* **1**(6), 80–83 (1945)
16. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding (2014). arXiv preprint [arXiv:1408.5093](https://arxiv.org/abs/1408.5093)