



Article

Optimization of CNN through Novel Training Strategy for Visual Classification Problems

Sadaqat ur Rehman ¹ , Shanshan Tu ^{2,*}, Obaid ur Rehman ³, Yongfeng Huang ¹, Chathura M. Sarathchandra Magurawalage ⁴  and Chin-Chen Chang ⁵

¹ Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; z-sun15@mails.tsinghua.edu.cn (S.u.R.); yfhuang@tsinghua.edu.cn (Y.H.)

² Faculty of Information Technology, Beijing University of Technology, Beijing 100022, China

³ Department of Electrical Engineering, Sarhad University of Science and IT, Peshawar 25000, Pakistan; obaid.ee@suit.edu.pk

⁴ School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK; csarata@essex.ac.uk

⁵ Department of Information Engineering and Computer Science, Feng Chia University, Taichung City 407, Taiwan; alan3c@gmail.com

* Correspondence: sstu@bjut.edu.cn

Received: 31 January 2018; Accepted: 14 April 2018; Published: 17 April 2018



Abstract: The convolution neural network (CNN) has achieved state-of-the-art performance in many computer vision applications e.g., classification, recognition, detection, etc. However, the global optimization of CNN training is still a problem. Fast classification and training play a key role in the development of the CNN. We hypothesize that the smoother and optimized the training of a CNN goes, the more efficient the end result becomes. Therefore, in this paper, we implement a modified resilient backpropagation (MRPROP) algorithm to improve the convergence and efficiency of CNN training. Particularly, a tolerant band is introduced to avoid network overtraining, which is incorporated with the global best concept for weight updating criteria to allow the training algorithm of the CNN to optimize its weights more swiftly and precisely. For comparison, we present and analyze four different training algorithms for CNN along with MRPROP, i.e., resilient backpropagation (RPROP), Levenberg–Marquardt (LM), conjugate gradient (CG), and gradient descent with momentum (GDM). Experimental results showcase the merit of the proposed approach on a public face and skin dataset.

Keywords: CNN optimization; image classification; MRPROP; training algorithm

1. Introduction

The convolution neural network (CNN) algorithm is the most state-of-the-art algorithm due to its vast diversity of applications, which are found in many areas: classification, recognition, detection, etc. The strength of a CNN lies with its enormous capabilities of feature extraction in its layers, which are synchronized in a feed-forward manner. Furthermore, the training algorithms and output classification layer further boost up the performance. Due to the aforementioned characteristics, a deep CNN is suitable for many computer vision problems where the solutions are difficult to obtain analytically. CNN has solved many computer vision problems efficiently, e.g., handwritten digit recognition [1,2], optical character recognition [3], car detection [4], face detection [5,6], face recognition [7,8], and analysis of facial expression [9].

Previous studies have shown that many researchers have tried to improve the efficiency of the CNN through pretraining filter banks in supervised and unsupervised manners, and by adding

more number of layers (deepness) [10]. Jarrett et al. [11] indicated that multi-stage feature extraction procedures produce results with better accuracy than single-stage feature extraction procedures. Also, filter learning (supervised, unsupervised, and random) boosts up the performance efficiency. Moreover, rectification and local contrast normalization also have a positive impact on a CNN in object recognition tasks. Erhan et al. [12] and Glorot et al. [13] suggest that the initialization of the weights and training algorithm directly affects the convergence rate and efficiency of a CNN. Particularly, training algorithms, whether supervised or unsupervised, are highly responsible, because it is difficult to train a network's first few layers and weights from backpropagation due to vanishing/exploding gradients. Therefore, choosing a simple algorithm, which allows efficient training and optimized weight adjustment, is highly desirable.

Calculation of the system error function gradient is also one of the important parts of the CNN training method. Most of them include complex computations; therefore, it may be difficult to meet real-time system requirements with such methods. The error function and the system error of the control application of neural networks are correlated [14], such as the error backpropagation algorithm (EBP) [15]. The convergence of the EBP algorithm is very slow if many gulches characterize the error function. An additional variable called "momentum" is added to the weight update in order to improve the training performance of the CNN [16]. A similar approach is proposed by Orłowska-Kowalska et al. [17], in which they attached a simple proportional derivative control to the gradient processing. This technique significantly increases the algorithm efficiency, as the gradient function is not computed directly.

The resilient backpropagation (RPROP) algorithm [18] is also independent of accurate gradient calculation. Also, a high computational proficiency and quick convergence speed are essential benefits. Due to the aforementioned properties, the RPROP algorithm is chosen for CNN [19] training for the face classification task. However, in RPROP, the continuous alterations of weights did not lead to any constant value; therefore, continuous deviations are enforced. As the continuous weights change in CNN, the training process is never-ending, which leads to system instability. Therefore, in this paper, we bring forward a modified training algorithm, modified resilient backpropagation (MRPROP) for the CNN [19], which helps the system by performing better weight changes to accomplish desired outputs efficiently and swiftly. Our contribution is as follows:

- We propose the tolerant band ΔK in MRPROP to avoid CNN overtraining.
- Global best concept is introduced in weight updating criteria for optimization.
- The MRPROP allows the CNN to optimize its weights more swiftly and precisely.

2. Resilient Backpropagation Algorithm (RPROP)

Training in CNN is considered to be the most important part of this work. RPROP is an efficient training algorithm that performs weight step alterations based on local gradient data. The key aim of choosing this training algorithm in CNN is that the weight adaptation is not blurred by gradient behavior at all. In addition, RPROP plays a significant role in optimizing the weights of the CNN due to the following properties.

1. The RPROP algorithm is fast, and requires less memory compared with other training algorithms.
2. The RPROP algorithm is independent of the shape of error surfaces [19].
3. The sign of error function gradient is utilized instead of the values of the error.

Let del_{xy} be the individual update value of each weight between the x th neuron and the y th input at the " t " training step, which determine the size of the weight increase or decrease. The increase/decrease

in the size of the weight is solely determined by the local sign of error function $E(t)$, based on the below learning conditions of Equation (1):

$$\text{del}w_{xy}^t = \begin{cases} \rho^+ \times \text{del}w_{xy}^{t-1}; \text{if } \frac{\partial E^{(t-1)}}{\partial w_{xy}} \times \frac{\partial E^{(t)}}{\partial w_{xy}} > 0 \\ \rho^- \times \text{del}w_{xy}^{t-1}; \text{if } \frac{\partial E^{(t-1)}}{\partial w_{xy}} \times \frac{\partial E^{(t)}}{\partial w_{xy}} < 0 \\ \text{del}w_{xy}^{t-1}; \text{Elsewhere} \end{cases} \quad (1)$$

where $0 < \rho^- < 1$ and $\rho^+ > 1$.

Weight is adjusted in each iteration/epoch to make the actual output closer to the desired output. Therefore, in each iteration, the algorithm jumps to local minima, and the update value $\text{del}w_{xy}$ is decreased by factor ρ^- , indicating that the last update is too big due to the change of the partial derivative of the corresponding weight w_{xy} sign. However, if the derivative maintains its sign, the update value is slightly augmented to speed up the convergence, in order to escape from the local minima. Moreover, when the update value is adapted for each weight, the weight update occurs according to a very simple rule: the update value decreased the weight, in case the error is growing. Conversely, in the case of a negative derivative, the weight is increased by its update value, i.e., it is being added. This can be mathematically illustrated as:

$$\text{del}w_{xy}^t = \begin{cases} -\text{del}w_{xy}^t; \text{if } \frac{\partial E^{(t-1)}}{\partial w_{xy}} > 0 \\ +\text{del}w_{xy}^t; \text{if } \frac{\partial E^{(t-1)}}{\partial w_{xy}} < 0 \\ 0; \text{Elsewhere} \end{cases} \quad (2)$$

$$w_{xy}^{t+1} = w_{xy}^t + \text{del}w_{xy}^t \quad (3)$$

where $\text{del}w_{xy}^t$ is the weight updated between the x th neuron and the y th input, at the “ t ” training step.

Nevertheless, if the partial derivative changes sign, i.e., the minimum was missed due to a too-large previous step, the previous update of the weight is reverted. Mathematically:

$$\text{del}w_{xy}^t = -\text{del}w_{xy}^{t-1}; \text{if } \frac{\partial E^{(t-1)}}{\partial w_{xy}} \times \frac{\partial E^{(t)}}{\partial w_{xy}} < 0 \quad (4)$$

Therefore, due to this backtracking weight step, the derivative is supposed to alter its sign once more in the following step. So, to avoid this computational expense of the update value, there should be no adaptation in the succeeding step of the update value by setting $\frac{\partial E^{(t-1)}}{\partial w_{xy}} = 0$ in the $\text{del}w_{xy}$ adaptation rule above.

3. Modified Resilient Backpropagation Algorithm

In this work, the convergence speed and performance of the RPROP algorithm is enhanced by a novel approach. Some modifications are made in MRPROP, so that it will reach and achieve global solutions faster than the basic RPROP algorithm.

The overall error is defined as the mean square error (mse) between the network outputs and the desired outputs; mathematically, it can be written as:

$$E_{mse} = \frac{1}{U \times NL} \sum_{u=1}^U \sum_{n=1}^{NL} |y_n^u - d_n^u|^2 \quad (5)$$

where the training set has a U input data matrix and U desired output representations; X^u represents the u th training image; d^u is our corresponding preferred output vector; and y_n^u represents the actual network output. This shows the function of the overall network parameters i.e., weights and biases.

It is obvious that the error generated during the training process influences the output error function. This error is calculated at each step of the training process for a presumed set of weights. Taking this into consideration, the gradient of the CNN error function can be calculated as follows:

$$f_{xy}(t) = \frac{\Delta E(t)}{\Delta W_{xy}(t)} = \frac{E(t) - E(t-1)}{W_{xy}(t) - W_{xy}(t-1) + \Delta z_i} \quad (6)$$

where $E(t)$ and $E(t-1)$ represent the error in training steps t and $t-1$, respectively; and $W_{xy}(t)$ and $W_{xy}(t-1)$ are the weights between the x th neuron and the y th input in training steps t and $t-1$. Whereas, Δz_i is introduced to avoid division by zero, in case the weight changes halted, and for describing the sign of the slope, regardless of the inaccurate value of $f_{xy}(t)$.

When the analysis of the parameter ρ factor fluctuates in the RPROP algorithm, as determined by Equation (1), it can be observed that the condition of keeping a factor at a constant value is achieved only if:

$$f_{xy}(t) \cdot f_{xy}(t-1) = 0 \quad (7)$$

This state practically does not exist in real case scenarios by any system processing physical measurement data. Therefore, the continuous alterations of the del_{xy} factor do not lead to any constant value, but continuous deviations are enforced. Also, due to the continuous alteration in the CNN weights ($delw_{xy}$), it is impossible to achieve the state described by Equation (7). This leads to a never-ending training process. Due to this, CNN overtraining can be noticed, which leads to system instability. In order to avoid such a situation, a modification of the RPROP algorithm is proposed. We make two major modifications in the MRPROP algorithm. Firstly, we introduce a tolerant band ΔK into the training conditions of Equation (1). Secondly, a global best concept is incorporated into the weight updating criteria, in order to allow the training algorithm of CNN to optimize its weights more swiftly and precisely to find a good solution, as shown in Equations (8) and (9).

The introduction of ΔK helps to protect the system against overtraining, which in return affects the overall efficiency of CNN in a positive way. A properly selected range of the tolerant band ΔK is found through using the trial and error method after comprehensive tests on a wealth of case studies with various operating points, assuring the stable operation of the CNN network, and limiting the excessive increase in the MRPROP weights.

To obtain more reliable solutions for CNN weight updating and achieve an optimized solution swiftly, we propose the “global best” concept in weight updating criteria. In RPROP, the change in weight Δw depends on whether the updated value del_{xy} increased or decreased according to error, in order to reach a better solution. However, the previously updated values are neglected after every iteration; this means that all of the best values previously achieved in the weight changes would not be referred back. Hence, there is no information sharing between the optimized values that have been achieved at the previous iterations and the current result. Therefore, by using the term “global best” concept in MRPROP, the information of previous weight change is the only source for the accurate result. Therefore, the past optimized value is randomly selected from all of the updated values of the previous weight change, and is used to update the process. This variable is called global best, or “*gbst*”. For a minimization problem, the global best selection procedure is given as follows:

- First, select two updated values randomly from all of the past change in weights Δw .
- Compare these two values against the optimized solution, and choose the better one as *gbst*.

Hence, the new proposed selection strategy will ensure that the diversity of the update value is preserved to avoid being trapped into a local optimum. The “*gbst*” is selected on the optimized value of current population $delw$, as shown in Equations (8) and (9):

$$del_{xy}^t = \begin{cases} \min(\rho^+ \times del_{xy}^{t-1}, gbst); & \text{if } f_{xy}(t) \cdot f_{xy}(t-1) > \Delta K \\ \max(\rho^- \times del_{xy}^{t-1}, gbst); & \text{if } f_{xy}(t) \cdot f_{xy}(t-1) < -\Delta K \\ del_{xy}^{t-1}; & \text{if } -\Delta K \leq f_{xy}(t) \cdot f_{xy}(t-1) \leq \Delta K \end{cases} \quad (8)$$

$$delw_{xy}^t = \begin{cases} -del_{xy}^t \cdot sign(f_{xy}(t)); & \text{if } f_{xy}(t) \cdot f_{xy}(t-1) > \Delta K \\ del_{xy}^t \cdot sign(f_{xy}(t)); & \text{if } f_{xy}(t) \cdot f_{xy}(t-1) < -\Delta K \\ 0; & \text{if } -\Delta K \leq f_{xy}(t) \cdot f_{xy}(t-1) \leq \Delta K \end{cases} \quad (9)$$

A number of satisfactory experimental tests are shown in Table 1, which prove the appropriate operation of the CNN with the proposed approach. The block diagram of the MRPROP for the CNN is shown in Figure 1. It is clear that the introduction of the tolerant band and the variable *gbst* cause not only the optimized weights that are achieved, but also the suboptimal weights. Similarly, the proposed strategy helps the system optimize its weight more swiftly and precisely, which directs the CNN into its best position and rapidly converges the training algorithm to its global optimum solution.

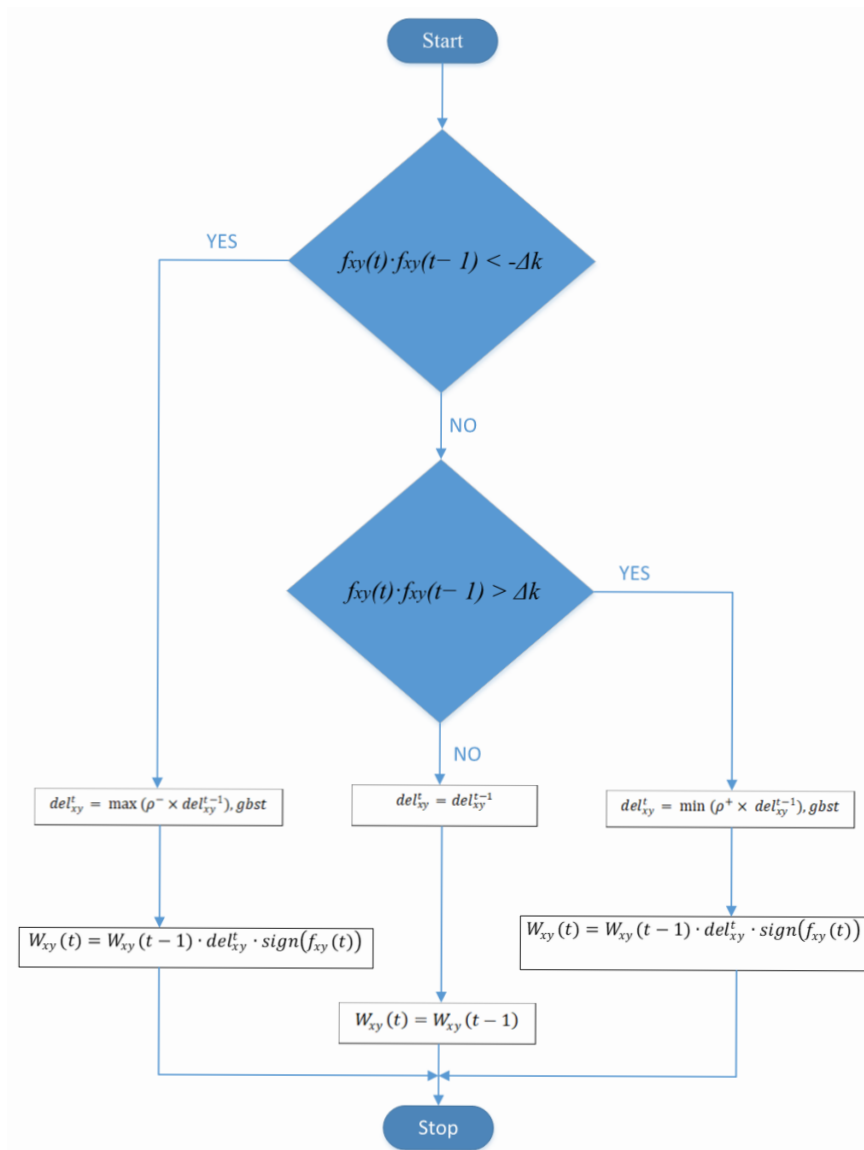


Figure 1. Flowchart of the proposed modified convolution neural network (CNN) training procedure through modified resilient backpropagation (MRPROP).

Table 1. Classification Results of Different Training Algorithms on Face and Skin Dataset. GDM: gradient descent with momentum; RPROP: resilient backpropagation; CG: conjugate gradient; LM: Levenberg–Marquardt.

Training Algorithm	Efficiency on Training Dataset	Efficiency on Test Dataset
GDM	97.1%	96.2%
RPROP	98.4%	97.1%
CG	98.0%	96.9%
LM	98.6%	97.3%
MRPROP	99.2%	97.8%

4. Experimental Results and Discussion

In this section, we analyze the convergence speed, output evaluations, training time, and number of training epochs based on the weight optimization criteria of the five different training algorithms. For comparison, in this article, we focus on five representative training algorithms, namely: resilient backpropagation (RPROP) [20], Levenberg–Marquardt (LM) [21], gradient descent with momentum (GDM) [22], conjugate gradient (CG) [23], and the proposed training algorithm (MRPROP). Our main objective is to determine a training algorithm for CNN that is fast, robust, and capable of handling large training datasets independent of the weights initialization strategy (supervised, unsupervised, random). The convergence speed of the training algorithm is highly affected by weight initialization and the choice of parameters that are used when training. For example, a slow decay in the mean square error (mse) occurs through a small learning rate α in the training algorithm; however, a large value of α may lead the training of the network to divergence. Subsequently, it is not possible to calculate all of the choices of the training parameters for CNN; we have determined the parameters by using the trial and error method after comprehensive mathematical experiments on a wealth of case studies. Yet, the optimized weight selection approach and training parameters in this paper reveal the basic tendencies in the convergence speed of the corresponding training algorithms for CNN.

The four different training algorithms, along with the proposed training algorithm for a CNN, are assessed on a face/non-face classification problem. The dataset contains 20,000 images with 10,000 face images and 10,000 non-face images, which are manually cropped web images and randomly extracted scenery photos, respectively. A sample of the dataset is shown in Figure 2, and its statistics are depicted in Table 2.

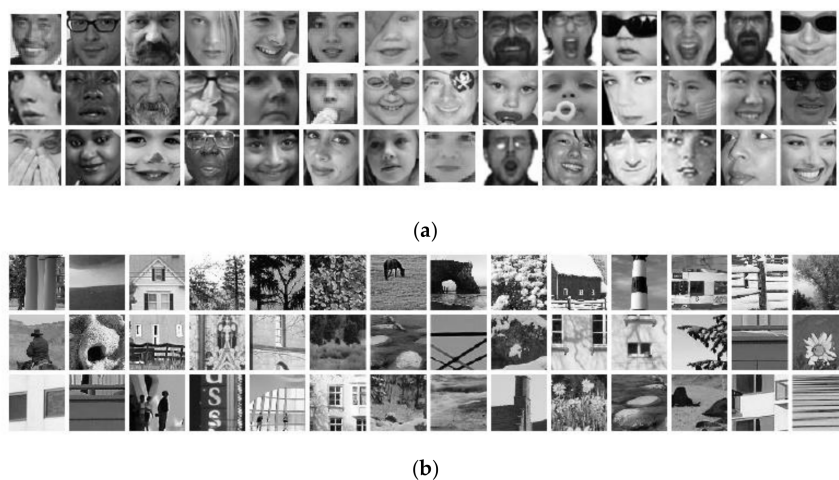


Figure 2. Sample images from Phung et al. [24] for evaluation of different training algorithms through CNN. (a) Face images; (b) Non-face images.

Table 2. Statistics of Face and Skin Dataset [24].

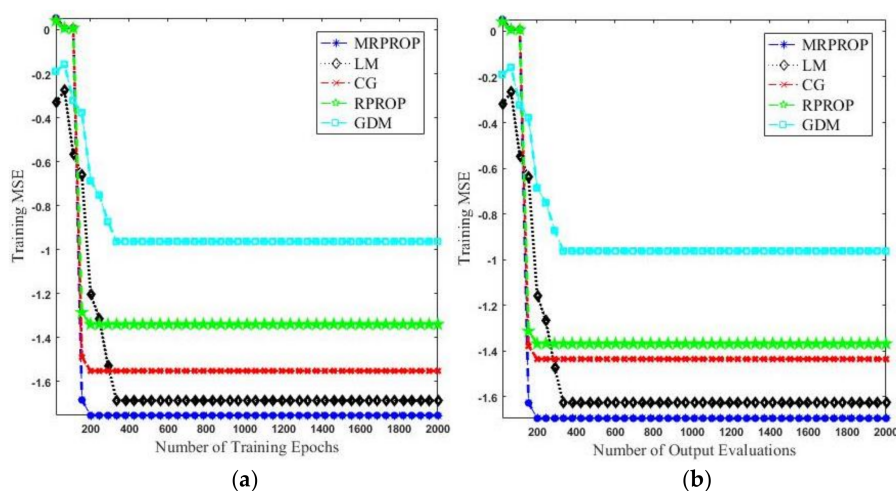
Skin Type and Lightening Condition	Images
Whitish, pinkish	1665
Dark brown, reddish	965
Yellowish, light brown	1402
Other Skin type	102
Indoor lightening conditions	1931
Outdoor lightening conditions	1855
Other lightening conditions	214

The size of the input images was 20×20 pixels, which is similar to the image sizes that have been used by numerous authors for face/non-face classification [25–27]. The network was trained on an individual training algorithm for 2000 epochs. To compare the performance, we utilize ten-fold cross-validation of different indicators i.e., mse training, training time, the number of output evaluations, and the number of training epochs.

Figure 3 indicates the comparison results of CNN using five different training algorithms. It is clear from Figure 3a that for each epoch count, the proposed MRPROP attained the lowest mse among all of the other training algorithms of CNN, followed by LM. Meanwhile, GDM achieved the highest mse, followed by RPROP. Figure 3b shows that MRPROP took a smaller number of output evaluations to acquire a lower mse, compared to LM, CG, RPROP, and GDM. The CG and RPROP algorithms had a very similar performance, with a slight difference.

Figure 3c reveals the number of gradient evaluations against the mse error function. It is clear from Figure 3c that our proposed algorithm takes a smaller number of gradient evaluations to achieve a lower mse. For example, the MRPROP algorithm uses 86 output evaluations to reach a mse of 0.35, whereas the LM and CG algorithms need 452 and 512 gradient evaluations, respectively. The RPROP algorithm has a better performance than the GDM algorithm when the number of output evaluations is above 200.

Figure 3d shows the training time of the individual training algorithms. In order to obtain a reliable comparison of the training speed, we measured the training time in terms of the MRPROP epoch time unit. We define one MRPROP epoch time unit as: the average time taken to perform one MRPROP training epoch on a fixed training set and a fixed-size network; it remains stable during the MRPROP training process. All four training algorithms—MRPROP, LM, CG, and RPROP—converged faster than the GDM algorithm. MRPROP was the fastest training algorithm among all of them due to its better weight optimization strategy compared to the other algorithms, which had no such strategy for weight regulations.

**Figure 3.** Cont.

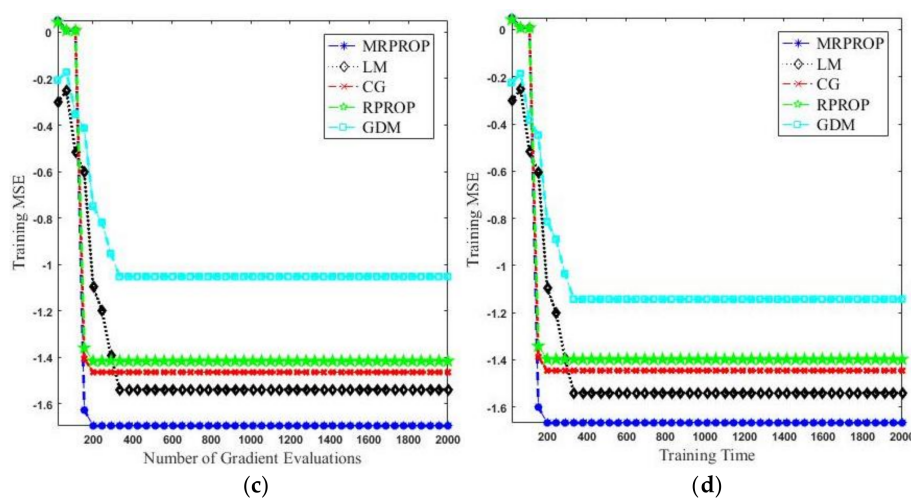


Figure 3. Comparison on face and skin dataset of five different training algorithms for CNN. mse training versus (a) total number of training epochs, (b) total number of output evaluations, (c) total number of gradient evaluations, and (d) total training time.

Table 1 depicts the classification accuracy of the CNN, with different training algorithms proposed for experimental purposes. It demonstrates that the CG and RPROP networks had similar classification rates, of 98% and 98.4% respectively, on the training dataset, and 96.9% and 97.1% on the testing dataset (the overall difference was no more than 0.4%), respectively, whereas the GDM achieved the lowest classification rates (97.1% and 96.2%). The highest classification rates were achieved by the proposed method, which were 99.2% and 97.8% on the training and testing dataset, respectively. These relative performances are consistent with the training speed comparison that was discussed earlier. CNN training with slower algorithms, such as GDM, requires more training time to find an optimal solution, as shown in Figure 3.

5. Conclusions

A new CNN training algorithm for visual classification problems, called MRPROP, has been presented. Features extraction at CNN layers is determined entirely through training. Our analysis of the five different training algorithms reveals that the RPROP and CG algorithms have reasonable convergence speeds and require small memory storage, whereas the LM algorithm is fast, but requires significantly more memory. However, the proposed MRPROP outperforms all of the other training algorithms, both in term of convergence speed and small memory usage. When evaluated on the face and skin dataset, CNN achieved classification rates of 99.2% and 97.8% with MRPROP on the training and testing datasets, respectively, which is significantly better than the same CNN with other training algorithms i.e., LM, CG, RPROP, and GDM.

In future, we plan to apply this concept on a range of CNN problems to explore its competency in diverse application domains. We also intend to study possible extensions of the pretraining algorithms and their connected theoretical guarantees with CNN convergence, optimality, and efficiency.

Acknowledgments: This work is supported in part by the National Key Research and Development Program of China under Grant No. 2016YFB0800402 and in part by the National Natural Science Foundation of China (Grant No. U1705261, U1536201, U1536207, U1636113).

Author Contributions: Sadaqat ur Rehman and Shanshan Tu conceived and designed the experiments; Sadaqat ur Rehman performed the experiments; Sadaqat ur Rehman, Yongfeng Huang, Obaid ur Rehman and Chathura M. Sarathchandra Magurawalage analyzed the data; Sadaqat ur Rehman, Yongfeng Huang and chin-chen chang interpreted the results; Sadaqat ur Rehman wrote the paper; Obaid ur Rehman, Shanshan Tu and Yongfeng Huang revised the manuscript and helped in most of the corrections.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, C.; Shih, P.; Liang, W. Integration of orthogonal feature detectors in parameter learning of artificial neural networks to improve robustness and the evaluation on hand-written digit recognition tasks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 2354–2358.
2. Lopes, G.; Silva, D.; Rodrigues, A.; Filho, P. Recognition of handwritten digits using the signature features and Optimum-Path Forest Classifier. *IEEE Latin Am. Trans.* **2016**, *14*, 2455–2460. [[CrossRef](#)]
3. Bulan, O.; Kozitsky, V.; Ramesh, P.; Shreve, M. Segmentation- and Annotation-Free License Plate Recognition with Deep Localization and Failure Identification. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2351–2363. [[CrossRef](#)]
4. Hu, Q.; Wang, H.; Li, T.; Shen, C. Deep CNNs with Spatially Weighted Pooling for Fine-Grained Car Recognition. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3147–3156. [[CrossRef](#)]
5. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [[CrossRef](#)]
6. Souza, G.; Santos, D.; Pires, R.; Marana, A.; Papa, J. Deep Texture Features for Robust Face Spoofing Detection. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *64*, 1397–1401. [[CrossRef](#)]
7. Ding, C.; Tao, D. Robust Face Recognition via Multimodal Deep Face Representation. *IEEE Trans. Multimed.* **2015**, *17*, 2049–2058. [[CrossRef](#)]
8. Low, C.; Teoh, A.; Toh, K. Stacking PCANet+: An Overly Simplified ConvNets Baseline for Face Recognition. *IEEE Signal Process. Lett.* **2017**, *24*, 1581–1585. [[CrossRef](#)]
9. Xie, S.; Hu, H. Facial expression recognition with FRR-CNN. *Electron. Lett.* **2017**, *53*, 235–237. [[CrossRef](#)]
10. Rehman, S.; Tu, S.; Huang, Y.; Yang, Z. Face recognition: A novel un-supervised convolutional neural network method. In Proceedings of the IEEE International Conference of Online Analysis and Computing Science (ICOACS), Chongqing, China, 28–29 May 2016; pp. 139–144.
11. Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.; LeCun, Y. What is the best multi-stage architecture for object recognition? In Proceedings of the IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2146–2153.
12. Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.; Vincent, P.; Bengio, S. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* **2010**, *11*, 625–660.
13. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res.* **2010**, *9*, 249–256.
14. Pajchrowski, T.; Zawirski, K.; Nowopolski, K. Neural Speed Controller Trained Online by Means of Modified RPROP Algorithm. *IEEE Trans. Ind. Inform.* **2015**, *11*, 560–568. [[CrossRef](#)]
15. Zhang, Y.; Guo, D.; Li, Z. Common Nature of Learning between Back-Propagation and Hopfield-Type Neural Networks for Generalized Matrix Inversion with Simplified Models. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 579–592. [[CrossRef](#)] [[PubMed](#)]
16. Qiu, G.; Varley, M.; Terrell, T. Accelerated training of backpropagation networks by using adaptive momentum step. *IEEE Electron. Lett.* **1992**, *28*, 377–379. [[CrossRef](#)]
17. Orłowska-Kowalska, T.; Szabat, K. Control of the drive system with stiff and elastic coupling using adaptive neuro-fuzzy approach. *IEEE Trans. Ind. Electron.* **2007**, *54*, 228–240. [[CrossRef](#)]
18. Riedmiller, M.; Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993; Volume 1, pp. 586–591.
19. Phung, S.; Bouzerdoum, A. A pyramidal neural network for visual pattern recognition. *IEEE Trans. Neural Netw.* **2007**, *18*, 329–343. [[CrossRef](#)] [[PubMed](#)]
20. Riedmiller, M.; Braun, H. Neural Speed Controller Trained Online by Means of Modified RPROP Algorithm. *IEEE Trans. Ind. Inform.* **2015**, *11*, 586–591.
21. Hagan, M.; Menhaj, M. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993. [[CrossRef](#)] [[PubMed](#)]
22. Demuth, H.; Beale, M.; Hagan, M.; Jesús, O. *Neural Network Design*, 2nd ed.; Martin Hagan: Boston, MA, USA, 2014.
23. Chong, E.; Zak, S. *An Introduction to Optimization*, 4th ed.; Wiley: New York, NY, USA, 2013.

24. Phung, S.; Bouzerdoum, A.; Chai, D. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 148–154. [[CrossRef](#)] [[PubMed](#)]
25. Sung, K.; Poggio, T. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 39–51. [[CrossRef](#)]
26. Yang, M.; Roth, D.; Ahuja, N. A SNoW-Based Face Detector. In Proceedings of the Advances in Neural Information Processing Systems, Cambridge, MA, USA, 27 November– 2 December 2000; Volume 12, pp. 855–861.
27. Li, S.; Jain, A. *Handbook of Face Recognition*, 2nd ed.; Springer: New York, NY, USA, 2011.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).