

UDC 519.226+004.852

DOI: 10.20535/1810-0546.2016.6.84115

V.V. Romanuke

Khmelnitskiy National University, Khmelnytskyi, Ukraine

TRAINING DATA EXPANSION AND BOOSTING OF CONVOLUTIONAL NEURAL NETWORKS FOR REDUCING THE MNIST DATASET ERROR RATE

Background. Due to that the preceding approaches for improving the MNIST image dataset error rate do not have a clear structure which could let repeat it in a strengthened manner, the formalization of the performance improvement is considered.

Objective. The goal is to strictly formalize a strategy of reducing the MNIST dataset error rate.

Methods. An algorithm for achieving the better performance by expanding the training data and boosting with ensembles is suggested. The algorithm uses the designed concept of the training data expansion. Coordination of the concept and the algorithm defines a strategy of the error rate reduction.

Results. In relative comparison, the single convolutional neural network performance on the MNIST dataset has been bettered almost by 30 %. With boosting, the performance is 0.21 % error rate meaning that only 21 handwritten digits from 10,000 are not recognized.

Conclusions. The training data expansion is crucial for reducing the MNIST dataset error rate. The boosting is ineffective without it. Application of the stated approach has an impressive impact for reducing the MNIST dataset error rate, using only 5 or 6 convolutional neural networks against those 35 ones in the benchmark work.

Keywords: MNIST; convolutional neural network; error rate; training data expansion; boosting.

Introduction

The MNIST (Mixed National Institute of Standards and Technology) database is widely used for training and testing in the field of machine learning [1, 2]. It consists of 70,000 images of handwritten digits along with the sets of the corresponding labels. There are only 10 image classes labeled as 1 to 10 implying digits “0” to “9”. All the MNIST gray-scale 28×28 images are represented as $28 \times 28 \times 70000$ array. The markers of data destined either for training or testing are gathered into a separate set. The first 60,000 images are for training, and the rest is used in testing.

The MNIST image dataset is a widespread benchmark dataset. A number of scientific papers have been published attempting to achieve the lowest error rate [2–4]. As of November 2016, the lowest error rate is 0.23 % [3, 5, 6]. It is achieved using a committee of 35 convolutional neural networks (CNNs) having the same architecture of 6 layers [5]. The lowest error rate achieved with a single neural network classifier is 0.35 %, where the 6-layer deep neural network (5-layer perceptron) was used [7]. Those are about a “near-human performance” on the MNIST database [5]. However, it is still aimed at to be improved for the purpose of raising standards for the modern machine vision.

Problem statement

Any of the preceding approaches for improving the MNIST image dataset error rate does not

have a clear structure which could let repeat it in a strengthened manner. Each attempt is an independent description of the results, rather than a method to achieve them. The only common idea is to apply CNN classifiers bringing the best performance and an expansion of training data. Therefore, an approach to reduce the MNIST dataset error rate further should be stated. The goal is to strictly formalize a strategy of the reduction. This goal is going to be reached after fulfilling the following tasks:

1. Construction of a single CNN whose performance on the MNIST dataset shall be better than 0.35 % error rate.
2. Construction of a series of additional CNNs having performance approximately close to the best one for a single CNN classifier.
3. Making a CNNs' ensemble whose performance shall be better than 0.23 % error rate.
4. Formalization of how to achieve the better performance by expanding the training data [5] and boosting [8, 9].

Training data expansion for a single CNN

CNNs are excellent for image classification, whatever the image size is. Construction of a CNN begins from its architecture depending on the image size. One of the known CNN architectures for the MNIST dataset consists of four convolutional layers (ConvLs), two maximum pooling layers (MPLs), a rectified linear unit layer (ReLU), and a softmax layer (SML). Denote these layers by $\{C_i\}_{i=1}^4$, $\{P_k\}_{k=1}^2$,

R and S , respectively. Then the primal architecture spoken about is

$$C_1 \rightarrow P_1 \rightarrow C_2 \rightarrow P_2 \rightarrow C_3 \rightarrow R \rightarrow C_4 \rightarrow S. \quad (1)$$

A great deal of experiments proves that neither increasing number of filters in ConvLs, nor the training data expansion helps to reliably achieve the 0.4 % error rate by the architecture (1). Nevertheless, it still can be made more robust even by the same number of ConvLs and MPLs. For instance, two additional ReLUs can be inserted following the starting two ConvLs:

$$\begin{aligned} C_1 \rightarrow R_1 \rightarrow P_1 \rightarrow C_2 \rightarrow R_2 \\ \rightarrow P_2 \rightarrow C_3 \rightarrow R_3 \rightarrow C_4 \rightarrow S. \end{aligned} \quad (2)$$

Nevertheless, the architecture (2) cannot break the 0.35 % error rate. Inserting Dropout layers (DOLs) $\{D_j\}_{j=1}^3$ as

$$\begin{aligned} C_1 \rightarrow D_1 \rightarrow R_1 \rightarrow P_1 \rightarrow C_2 \rightarrow D_2 \rightarrow R_2 \\ \rightarrow P_2 \rightarrow C_3 \rightarrow D_3 \rightarrow R_3 \rightarrow C_4 \rightarrow S \end{aligned} \quad (3)$$

makes it much better [10], but it is not sufficient.

Obviously, a deeper CNN is needed to break the existing top performance. This is done with adding one more ConvL followed with a DOL and a ReLU before the two ending layers:

$$\begin{aligned} \{C_j \rightarrow D_j \rightarrow R_j \rightarrow P_j\}_{j=1}^3 \rightarrow C_4 \\ \rightarrow D_4 \rightarrow R_4 \rightarrow C_5 \rightarrow S. \end{aligned} \quad (4)$$

This 17-layer CNN itself requires the training data expansion because it contains four DOLs and three MPLs, whereas the CNN (3) containing those three and two ones cannot be sufficiently trained on 60,000 images.

The initial training data are in the set $\{\mathbf{F}_i\}_{i=1}^{60000}$, where \mathbf{F}_i is a 28×28 matrix representing an image. Each matrix is processed in order to scale, rotate, and shift the corresponding image. Firstly, the image is scaled with a scale related factor σ_{scale} via a scaling procedure s [11]:

$$\mathbf{F}_i^{(\text{Sc})} = s(\mathbf{F}_i, \sigma_{\text{scale}}). \quad (5)$$

The scaled image $\mathbf{F}_i^{(\text{Sc})}$ by (5) is then rotated with a rotation related factor σ_{rotate} via a procedure r [11]:

$$\mathbf{F}_i^{(\text{ScRt})} = r(\mathbf{F}_i^{(\text{Sc})}, \sigma_{\text{rotate}}). \quad (6)$$

Finally, the image $\mathbf{F}_i^{(\text{ScRt})}$ by (6) is shifted horizontally and vertically by numbers of pixels which are

defined with a shift related factor σ_{shift} via a procedure h [9, 11]:

$$\mathbf{F}_i^{(\text{ScRtSf})} = h(\mathbf{F}_i^{(\text{ScRt})}, \sigma_{\text{shift}}). \quad (7)$$

Although the factors

$$\sigma_{\text{scale}}, \sigma_{\text{rotate}}, \sigma_{\text{shift}} \quad (8)$$

are varied simultaneously, the procedures (5), (6), and (7) are repeated for a few times by increasing the values (8). This allows to expand the MNIST training set more than twice. If the values (8) are increased for Q times then, instead of just the image (7), it gives Q distorted images $\{\mathbf{F}_{iq}^{(\text{ScRtSf})}\}_{q=1}^Q$. Thus the initial training data set is $Q + 1$ times expanded: now it is the set

$$\{\mathbf{F}_i, \{\mathbf{F}_{iq}^{(\text{ScRtSf})}\}_{q=1}^Q\}_{i=1}^{60000}. \quad (9)$$

A useful form of training data is the average image subtraction. The set $\{\mathbf{F}_i\}_{i=1}^{60000}$ is already formed so. The average image

$$\tilde{\mathbf{F}} = \frac{\tilde{\mathbf{F}}_0 + \tilde{\mathbf{F}}^{(\text{ScRtSf})}}{2} \quad (10)$$

by

$$\tilde{\mathbf{F}}_0 = \frac{1}{60000} \sum_{i=1}^{60000} \mathbf{F}_i \quad (11)$$

and

$$\tilde{\mathbf{F}}^{(\text{ScRtSf})} = \frac{1}{60000 \cdot Q} \sum_{i=1}^{60000} \sum_{q=1}^Q \mathbf{F}_{iq}^{(\text{ScRtSf})} \quad (12)$$

is subtracted from the newly distorted images (Fig. 1) and from the initial dataset as well, and the ready expanded training data set (ETDS) is

$$\{\mathbf{F}_i - \tilde{\mathbf{F}}, \{\mathbf{F}_{iq}^{(\text{ScRtSf})} - \tilde{\mathbf{F}}^{(\text{ScRtSf})}\}_{q=1}^Q\}_{i=1}^{60000}. \quad (13)$$

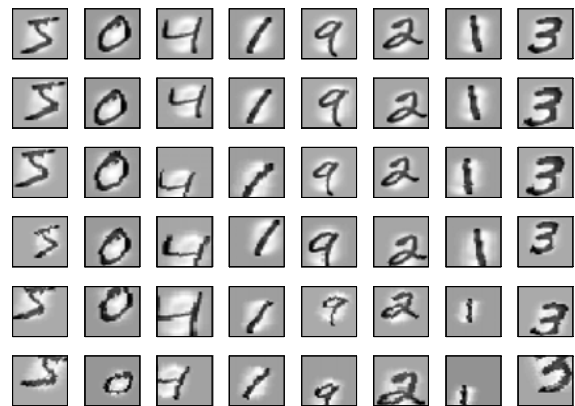


Fig. 1. A mini-subset of the initial training data images (upper line series) and the newly distorted images by the procedures (5)–(7), after the average (10) subtraction

Every time the set (13) can be produced different by changing Q or changing the maximal values of the factors (8).

Just with ETDS (13) by $Q = 5$, giving 360,000 training images, four CNNs (4) whose performance is better than 0.35 % are obtained. Their ConvLs have the same numbers of filters (50, 100, 500, 1000, and 10 before its SML), except the CNN whose ConvLs have 40, 80, 500, 1000, 2000 (before SML, although 10 filters here is theoretically sufficient) filters, respectively. The performance of this 2000-before-SML CNN achieved in 18 epochs is 0.31 % error rate. The other three CNNs with 50 filters in the starting ConvL have the following error rate: 0.33 % (in 28 epochs), 0.32 % (in 34 epochs), and 0.30 % (in 67 epochs).

Three of those four CNNs in their uniform combination (the uniform ensemble) produce 0.26 % error rate. The same holds when the combination is of all the four CNNs. Any other ensembles of those four classifiers do not perform better than at that rate. So, the training data must be expanded further.

The further expansion is fulfilled by another eight independent distortion types (Fig. 2):

- 1) linear conformal distortion (similar to the distortions described above but much severer);
- 2) affine distortion;
- 3) projective distortion;
- 4) polynomial distortion;
- 5) piecewise linear distortion;
- 6) sinusoid distortion;
- 7) barrel distortion;
- 8) radial pin cushion distortion.

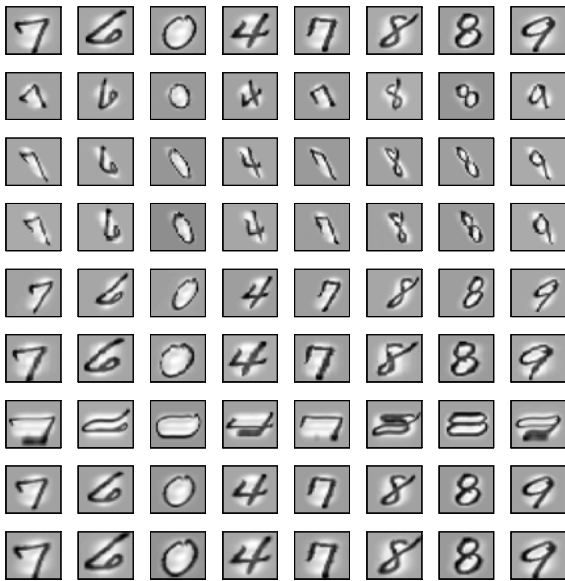


Fig. 2. A mini-subset of the initial training data images (upper line series) and the eight independent distortion types applied to them

Let the transformation t_m execute the corresponding m -th distortion type in the list above, $m = \overline{1,8}$. This transformation can be applied by varying 1 to 12 its parameters gathered in the set T . If it is executed for N times then, at the n -th version T_n of this set, a new image is

$$\mathbf{F}_{inn}^{(+)} = t_m(\mathbf{F}_i, T_n) \text{ by } n = \overline{1, N} \text{ and } m = \overline{1, 8}. \quad (14)$$

The average image

$$\tilde{\mathbf{F}} = \frac{\tilde{\mathbf{F}}_0 + \tilde{\mathbf{F}}^{(+)}}{2} \quad (15)$$

by (11) and

$$\tilde{\mathbf{F}}^{(+)} = \frac{1}{480000 \cdot N} \sum_{i=1}^{60000} \sum_{m=1}^8 \sum_{n=1}^N \mathbf{F}_{inn}^{(+)} \quad (16)$$

is subtracted from every image (14). Thus an alternative ETDS is

$$\{\mathbf{F}_i - \tilde{\mathbf{F}}, \{\{\mathbf{F}_{inn}^{(+)} - \tilde{\mathbf{F}}\}_{n=1}^N\}_{m=1}^8\}_{i=1}^{60000}. \quad (17)$$

Even if the eight transformations are executed once, i. e. $N = 1$ the ETDS (17) has 540,000 images.

Before trying the set (17), a resuming-training technique can be applied [12]. This implies to continue training a CNN on a differently produced ETDS (13) or ETDS (17). With ETDS (13), three additional CNNs perform at the following error rate: 0.29 % (in 68 epochs), 0.30 % (in 69 epochs), and 0.27 % (in 74 epochs). Their uniform ensemble performs at 0.24 % error rate. Any other ensembles included the four previously obtained CNNs do not perform better. Hence, we need the set (17), anyway (Fig. 3).

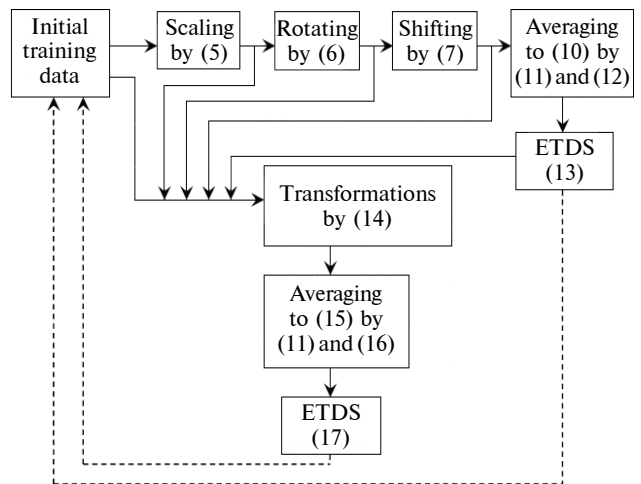


Fig. 3. A concept of the training data expansion

It is not necessary to involve the whole set (17) into training. Just involving only 180,000 entries, another two promising CNNs are obtained performing at the following error rate: 0.29 % (in 77 epochs) and 0.27 % (in 82 epochs). Thus, the training data expansion allows as to get a better error rate, as well as to train more good-performing CNNs by switching between different ETDSs and the subsequent resuming-training.

Ensembles of CNNs

The uniform ensemble of those nine CNNs (see their tags in Table 1) performs at 0.26 % error rate. Amazingly enough, but such an ensemble performs even worse than the mini-ensemble of just three CNNs [13]. A solution is to try any uniform ensembles of those nine CNNs. That is, to search through all possible ensembles of 2, 3, ..., 9 CNNs. Eventually, the best uniform ensemble of CNNs #1, #7, and #9 gives 0.22 % error rate.

Table 1. Tags of CNNs and their performance on the MNIST testing dataset

CNN (epochs of training)	#	Error rate, %
74	1	0.27
82	2	0.27
77	3	0.29
68	4	0.29
67	5	0.30
69	6	0.30
18	7	0.31
34	8	0.32
28	9	0.33

When trying non-uniform ensembles, their weights are searched with the step 0.1, which is acceptable. Finally, four ensembles are found performing at 0.21 % error rate (Table 2). The ensembles #2, #3, and #4 include the same CNNs, although with slightly different weights.

Table 2. Tags and weights of CNNs whose ensemble performs at 0.21 % error rate on the MNIST testing dataset

Ensemble #1		Ensemble #2		Ensemble #3		Ensemble #4	
CNN #	Weight	CNN #	Weight	CNN #	Weight	CNN #	Weight
1	0.3	1	0.3	1	0.2	1	0.2
2	0.3	4	0.1	4	0.2	4	0.1
5	0.2	5	0.2	5	0.2	5	0.2
6	0.1	6	0.1	6	0.1	6	0.1
9	0.1	7	0.1	7	0.1	7	0.2
		9	0.2	9	0.2	9	0.2

It is not guaranteed that this error rate can be improved further just by adding new CNNs. This is because properties of the used ETDSs bettering the generalization may be exhausted at the stage.

The better performance by ETDS and boosting

Fig. 4 explains an algorithmic way for achieving the better performance by expanding the training data and boosting with ensembles. The performance of a single CNN shall always be worse than that of an ensemble. Similarly, the performance of a uniform ensemble is expected to be improved by boosting with non-uniform weights.

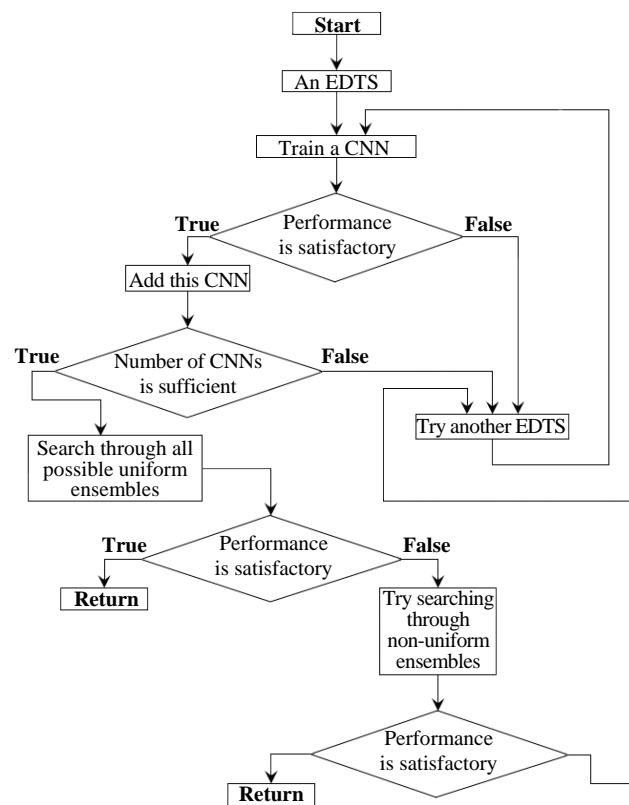


Fig. 4. Boosting an ensemble of CNNs trained on the expanded training data

The algorithm in Fig. 4 fits any image classification problem. According to Fig. 3, the training data expansion can be fulfilled as many times as needed. So the subsequent efficient boosting is realizable.

Discussion

In relative comparison, the single CNN performance on the

MNIST dataset has been bettered almost by 30 %. Nevertheless, the improvement is just 0.08 % by the absolute reckoning. The absolute improvement on the ensembles from Table 2 is even smaller – just 0.02 %, although it is about 9.5 % by the ratio of 0.23 % to 0.21 % error rate. However, the performance at 0.21 % error rate means that only 21 handwritten digits from 10,000 are not recognized. Practically, this is really the human performance.

Conclusions

The stated approach for reducing the MNIST dataset error rate relies on two principal compo-

nents. The first is the crucial one – the training data expansion. Without it, the boosting which is the second component, is ineffective. Coordination of these components formalized in Figs. 3 and 4 is a strategy of the error rate reduction. Application of the stated approach has a pretty impressive impact for reducing the MNIST dataset error rate. Besides, the efficient boosting ensemble includes only 5 or 6 CNNs (Table 2) against those 35 CNNs in the work [5]. This also shortens the time of classification. In this way, further work is going to be focused on acceleration of larger image classification.

List of literature

1. *Gradient-based learning applied to document recognition* / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // *Proc. IEEE*. – 1998. – **86**, iss. 11. – P. 2278–2324. doi: 10.1109/5.726791
2. *Kussul E., Baidyk T. Improved method of handwritten digit recognition tested on MNIST database* // *Image and Vision Computing*. – 2004. – **22**, iss. 12. – P. 971–981. doi: 10.1016/j.imavis.2004.03.008
3. *Flexible, high performance convolutional neural networks for image classification* / D. Cirean, U. Meier, J. Masci et al. // *Proc. 22nd Int. Joint Conf. Artificial Intelligence*. – 2011. – **2**. – P. 1237–1242.
4. *Efficient learning of sparse representations with an energy-based model* / M. Ranzato, C. Poultney, S. Chopra, Y. LeCun // *Advances in Neural Information Processing Systems*. – 2006. – **19**. – P. 1137–1144.
5. *Ciresan D., Meier U., Schmidhuber J. Multi-column deep neural networks for image classification* // *2012 IEEE Conf. Computer Vision and Pattern Recognition*. – 2012. – P. 3642–3649. doi: 10.1109/CVPR.2012.6248110
6. *Date P., Hendlar J.A., Carothers C.D. Design index for deep neural networks* // *Procedia Computer Sci*. – 2016. – **88**. – P. 131–138. doi: 10.1016/j.procs.2016.07.416
7. *Deep big simple neural nets excel on handwritten digit recognition* / D. Cirean, U. Meier, L.M. Gambardella, J. Schmidhuber // *Neural Computation*. – 2010. – **22**, № 12. – P. 3207–3220. doi: 10.1162/NECO_a_00052
8. *Kégl B., Busa-Fekete R. Boosting products of base classifiers* // *Proc. 26th Annual Int. Conf. Machine Learning*. – 2009. – P. 497–504. doi: 10.1145/1553374.1553439
9. *Romanuke V.V. Boosting ensembles of heavy two-layer perceptrons for increasing classification accuracy in recognizing shifted-turned-scaled flat images with binary features* // *J. Inform. Organiz. Sci*. – 2015. – **39**, № 1. – P. 75–84.
10. *Dropout: A simple way to prevent neural networks from overfitting* / N. Srivastava, G. Hinton, A. Krizhevsky et al. // *J. Machine Learning Res*. – 2014. – **15**, iss. 1. – P. 1929–1958.
11. *Romanuke V.V. Two-layer perceptron for classifying flat scaled-turned-shifted objects by additional feature distortions in training* // *J. Uncertain Systems*. – 2015. – **9**, № 4. – P. 286–305.
12. *Romanuke V.V. A method of resume-training of discontinuous wear state trackers for composing boosting high-accurate ensembles needed to regard statistical data inaccuracies and shifts* // *Problems of Tribology*. – 2015. – № 3. – P. 19–22.
13. *Romanuke V.V. Accuracy improvement in wear state discontinuous tracking model regarding statistical data inaccuracies and shifts with boosting mini-ensemble of two-layer perceptrons* // *Problems of Tribology*. – 2014. – № 4. – P. 55–58.

References

1. Y. LeCun *et al.*, “Gradient-based learning applied to document recognition”, *Proc. IEEE*, vol. 86, iss. 11, pp. 2278–2324, 1998. doi: 10.1109/5.726791
2. E. Kussul and T. Baidyk, “Improved method of handwritten digit recognition tested on MNIST database”, *Image and Vision Computing*, vol. 22, iss. 12, pp. 971–981, 2004. doi: 10.1016/j.imavis.2004.03.008
3. D. Cirean *et al.*, “Flexible, high performance convolutional neural networks for image classification”, in *Proc. 22nd Int. Joint Conf. Artificial Intelligence*, vol. 2, pp. 1237–1242, 2011.
4. M. Ranzato *et al.*, “Efficient learning of sparse representations with an energy-based model”, *Advances in Neural Information Processing Systems*, vol. 19, pp. 1137–1144, 2006.
5. D. Cirean *et al.*, “Multi-column deep neural networks for image classification”, in *Proc. 2012 IEEE Conf. Computer Vision and Pattern Recognition*, pp. 3642–3649, 2012. doi: 10.1109/CVPR.2012.6248110

6. P. Date *et al.*, "Design index for deep neural networks", *Procedia Computer Sci.*, vol. 88, pp. 131–138, 2016. doi: 10.1016/j.procs.2016.07.416
7. D. Cireşan *et al.*, "Deep big simple neural nets excel on handwritten digit recognition", *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010. doi: 10.1162/NECO_a_00052
8. B. Kégl and R. Busa-Fekete, "Boosting products of base classifiers", in *Proc. 26th Annual Int. Conf. Machine Learning*, pp. 497–504, 2009. doi: 10.1145/1553374.1553439
9. V.V. Romanuke, "Boosting ensembles of heavy two-layer perceptrons for increasing classification accuracy in recognizing shifted-turned-scaled flat images with binary features", *J. Inform. Organiz. Sci.*, vol. 39, no. 1, pp. 75–84, 2015.
10. N. Srivastava *et al.*, "Dropout: A simple way to prevent neural networks from overfitting", *J. Machine Learning Res.*, vol. 15, iss. 1, pp. 1929–1958, 2014.
11. V.V. Romanuke, "Two-layer perceptron for classifying flat scaled-turned-shifted objects by additional feature distortions in training", *J. Uncertain Systems*, vol. 9, no. 4, pp. 286–305, 2015.
12. V.V. Romanuke, "A method of resume-training of discontinuous wear state trackers for composing boosting high-accurate ensembles needed to regard statistical data inaccuracies and shifts", *Problems of Tribology*, no. 3, pp. 19–22, 2015.
13. V.V. Romanuke, "Accuracy improvement in wear state discontinuous tracking model regarding statistical data inaccuracies and shifts with boosting mini-ensemble of two-layer perceptrons", *Problems of Tribology*, no. 4, pp. 55–58, 2014.

В.В. Романюк

ЗБІЛЬШЕННЯ ОБСЯГУ ДАНИХ НАВЧАННЯ ТА БУСТИНГ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗНИЖЕННЯ РІВНЯ ПОМИЛОК БАНКУ ДАНИХ MNIST

Проблематика. Оскільки попередні підходи до покращення рівня помилок банку зображень MNIST не мають чіткої структури, яка могла би бути чітко відтворена, розглядається формалізація покращення продуктивності.

Мета дослідження. Метою роботи є строга формалізація стратегії зниження рівня помилок банку даних MNIST.

Методика реалізації. Пропонується алгоритм досягнення кращої продуктивності за допомогою збільшення обсягу даних навчання та бустингу. Алгоритм використовує розроблену концепцію збільшення обсягу даних навчання. Координація цієї концепції та алгоритму визначає стратегію зниження рівня помилок.

Результати дослідження. У відносному порівнянні продуктивність однієї згорткової нейронної мережі на банку даних MNIST покращена майже на 30 %. За допомогою бустингу продуктивність становить 0,21 % – не розпізнається лише 21 рукописна цифра з 10000.

Висновки. Збільшення обсягу даних навчання є визначальним для зниження рівня помилок банку даних MNIST. Без цього бустинг неефективний. Застосування викладеного підходу має виразний вплив на зниження рівня помилок банку даних MNIST при використанні лише 5 або 6 згорткових нейронних мереж проти 35 в еталонній роботі.

Ключові слова: MNIST; згорткова нейронна мережа; рівень помилок; збільшення обсягу даних навчання; бустинг.

В.В. Романюк

УВЕЛИЧЕНИЕ ОБЪЕМА ДАННЫХ ОБУЧЕНИЯ И БУСТИНГ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ СНИЖЕНИЯ УРОВНЯ ОШИБОК НАБОРА ДАННЫХ MNIST

Проблематика. Поскольку предыдущие подходы к улучшению уровня ошибок набора данных MNIST не имеют четкой структуры, которая могла бы быть четко воссоздана, рассматривается формализация улучшения производительности.

Цель исследования. Целью работы является строгая формализация стратегии снижения уровня ошибок набора данных MNIST.

Методика реализации. Предлагается алгоритм достижения лучшей производительности с помощью увеличения объема данных обучения и бустинга. Алгоритм использует разработанную концепцию увеличения объема данных обучения. Координация этой концепции и алгоритма определяет стратегию снижения уровня ошибок.

Результаты исследования. В относительном сравнении производительность одной сверточной нейронной сети на наборе данных MNIST улучшена почти на 30 %. С помощью бустинга производительность составляет 0,21 % – не распознается лишь 21 рукописная цифра из 10000.

Выводы. Увеличение объема данных обучения является определяющим для снижения уровня ошибок. Без этого бустинг неэффективен. Применение изложенного подхода оказывает выразительное влияние на снижение уровня ошибок набора данных MNIST при использовании лишь 5 или 6 сверточных нейронных сетей против 35 в эталонной работе.

Ключевые слова: MNIST; сверточная нейронная сеть; уровень ошибок; увеличение объема данных обучения; бустинг.

Рекомендована Радою
факультету прикладної математики
НТУУ "КПІ ім. І. Сікорського"

Надійшла до редакції
23 листопада 2016 року