# Multilayer Perceptron and Evolutionary Algorithm for Handwritten Digit Recognition

Hanshan Zhang

Australia National University u5975228@anu.edu.au

**Abstract.** Handwritten digit recognition is an active topic in academia and industry alike. In order to improve accuracy, a lot of good works have been done, especially using the multilayer perceptron (MLP) approach. However, Setting the network structure and tuning some hyper parameters are the most difficult part of designing the network. On the other hand, an evolutionary algorithm (EA) is a stochastic search approach to deal with the optimization problem. In this paper, the evolutionary algorithm has been used to determine the structure of MLP, including setting activation functions, optimization algorithms, and number of layers and number of neurons in each layer. Our object is using EA to find a optimal structure of MLP for the given task. We test the model on the Semeion Handwritten Digit Data Set, achieving 98.11% accuracy, which is 5% higher than MetaNet, one of the baseline models.

Keywords: MLP,  $\cdot$  Evolutionary Algorithm,  $\cdot$  Handwritten digit recognition

#### 1 Introduction

In Optical character recognition (OCR) and pattern learning research area, the handwritten digit recognition is an active topic [2, 22]. In the OCR application, it is important that the computer or the device can have the intelligence to deal with printed paper work or handwritten images as input for digit interpretation. As there are many scenarios involving a large number of handwritten digit files such as cheques in the banks and the letter (mail) splitting in the post office [9, 22, 24], we need a system that can help us automatically process the handwritten documents. The handwritten recognition task might be a good example for the performance evaluation of classification models.

Following the publishing of MNIST dataset [19], many studies have tried to use complex MLP model to achieve the higher Accuracy. [29] showed that only one hidden layer MLP achieved 0.70% testing error rate with 800 hidden units, and Convolutional Neural Network (CNN) achieved a 0.40% error rate with elastic training image deformations. Later, [16] tried to combine neural network with support vector machine, achieving 0.54% error rate. According to the MNIST dataset website, it seems that the more complex model listed on the websites seems to achieve a higher accuracy (lower error rate). Besides the high accuracy MLP achieved, [20] also found that compared with traditional machine learning algorithms like k-nearest neighbor (KNN), the fully connect multi-layer neural network actually was faster in prediction section. This finding shows MLP can be ideal for automatic processing of a massive handwritten dataset.

However, how to design the structure of MLP is still a question. For example, the number of hidden units in hidden layers is a critical design issue. If the hidden units are too large, the model will face the over-fitting problem where it can only poorly on testing dataset while perform well on the training dataset but[17]. On the other hand, with fewer hidden units, the result generally will have an under-fitting problem. Another important factor of performance is the activation function. Activation function is used to introduce non-linearity into the MLP model. Without activation function, no matter how deep your model is , it can be seen as one hidden layer network. The optimization algorithm is thought to be another important factor, because it helps MLP model to learn the appropriate parameters. This is why we need to use an appropriate optimization algorithm to adjust the parameters of the model.

An evolutionary algorithm (EA) is often used to find the solution for the optimization problem. EA is inspired mechanisms of biological evolution, such as reproduction, selection, mutation and crossover. The combination of EA and artificial neural network has a long history [28, 30]. Previously weights and connection between neurons in neutral network were optimized together by EA, but recently, only architectures are optimized by EA, and Stochastic Gradient Descent(SGD) and its variants are used to optimize the weights [31].

In this paper, we propose a simple but effective method to find the optimal architectures of MLP on Semeion Handwritten Digit Dataset. We choose this dataset because it provides enough data instances for training, and has no missing value. Furthermore, using this dataset can also show the generative ability of MLP model. Once the EA algorithm starts, there is no need to require human participation and its output is a fully trained model. The final model achieved 98.11% of accuracy comparing to MetaNet model [4], the accuracy of which is 93.09%.

#### 2 Hanshan Zhang

In the rest of the paper, section 2 will discuss related literature, and the details of the dataset and method used in this paper will be covered in section 3. Section 4 will show the result of our model. In the final section, future research directions will be discussed.

# 2 Related Work

The neural network [3] has been widely used on handwritten digit and character recolonization tasks. We will talk about some previous studies using neural network model and combining neural network with EA to address the handwritten digit or character recognition problem.

In [21], LeCun tested a neural network model with only one hidden layer on the MNIST test set. Their architecture was 400-300-10, getting a 1.6 % error rate.

The MLP has also been tested on the recognition problem for English characters [25, 26]. In [25], the author using the information from boundary tracing and their Fourier Descriptors of handwritten characters as inputs of a two-layer neural network. The recognition accuracy was 94% reported in paper. And later, in [26], they proposed a two hidden layers' model with the architecture of four-layer MLP model (54(69)-100-100-26). They used inputs from diagonal feature extraction scheme and set the distribution function of mean squared errors as the loss function, and trained the model with Gradient descent, adaptive learning and momentum. The final result was accuracy of 98.5% for 69 features and 97.8% for 54 features.

One of the baseline models used in this paper is from [4], they proposed a new neural network structure, named MetaNet. According to their experiment, the model can achieve a 93.09% accuracy rate on the whole ten digits.

[35] tried to optimize the inputs and the architecture of MLP with EA, achieving 90.77% accuracy, while [1] used EA to find the optimal architecture and initial weights of MLP.

## 3 Dataset and Method

In this section, details of data and preprocessing of data will be first discussed, followed by discussions on the EA assisting model. The final part will show the baseline models we proposed apart from the Metanet model from [4]

#### 3.1 Dataset

The dataset has 1593 digit image with  $16 \times 16$  pixel image size and it can be downloaded from [33]. The original pixel value was 256 grays scale. The pixel initial value greater than 127 was set to be 1 and the value less than 127 was set to be 0. The example image from the dataset can be seen in Fig. 1



Fig. 1: The number 3 in dataset

In order to increase the model's robust ability, we randomly rotate images with degrees from -10 to 10, when feeding the data into the model. With this setting, we also increase the size of the dataset , which can also help train a good model. And we split the whole dataset into balanced train dataset and test dataset by 50-50.

#### 3.2 EA assisting model

**MLP** A perceptron can be defined as a binary linear classifier. It takes the input vector x, multiply the weight vector w and add bias b to get the output y : y = w \* x + b. Multilayer perceptron[18, 27, 34] is a deep, neural network inspired by the human brain function, which has more than one perceptron. MLP uses the error backpropagation algorithm to train the model.

EA The following summarizes how the proposed EA algorithm works.

- 1. Load dataset and setting of population size (20), converge threshold (1e-4), and crossover rate (0.8) and mutation rate (0.01).
- 2. Create the population with discrete-valued representations (representation details will be covered later)
- 3. Use each chromosome in population to create the network and use optimization algorithm in chromosome to train the network. Set cross entropy loss as object function.
- 4. Use test dataset to get the test accuracy  $a_i$  of each individual i and use accuracy  $a_i$  as fitness for individual i. Keep track the best network from each generation.
- 5. Use selection, crossover and mutation operations to generate the next population.
- 6. Repeat 3-5, until the converge threshold has been reached.
- 7. Output the best model in the current population.

Table 1. Domain of each position in enrollosomes			
Column	Possible Value		
1 (activation functions)	$\{ReLU, ELU, Sigmoid, Tanh\}$		
2 (optimization algorithm)	$\{SGD, Adam, Adamax, RMSprop\}$		
3 (number of hidden layers)	$\{1, 2, 3, 4, 5, 6\}$		
4-N (number of hidden units each layer)	$\{64, 128, 256, 512, 1024\}$		

Table 1: Domain of each	position in chromosomes
-------------------------	-------------------------

The first column stores the selected activation functions. Rectified linear unit (ReLU)[13, 23] is defined as  $ReLU(X) = \max(0, x)$ . Exponential Linear Units (ELU) [5] is defined as  $ELU(X) = \max(0, x) + \min(0, \alpha * (\exp(x) - 1))$ . Sigmoid activation function is defined as  $sgmoid(x) = \frac{1}{1 + \exp(-x)}$ , and Tanh is defined as  $tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$ . The second column stores the optimization algorithms from SGD[32], Adam[14] and Adamax [14], and RMSprop [10]

**Representation** We use discrete-valued representations to encode the model and store it in chromosomes. The domain of each position in chromosomes can be found in Table.1 Each chromosome has 4 columns at least. First 3 columns store the activation function, optimization algorithm and number of hidden layers information respectively. Starting from the fourth column, it stores the number of hidden units for each layer. Hence the total length of each chromosome depends on the value of 3rd column.

**Create Network** When using chromosomes to create network, we add the final output layer with 10 units for 10 classes. After each hidden layer, we add the activation function layer as encoded in chromosomes. We also set the optimization algorithm based on the chromosome.

Selection First, we calculated the fitness value for each chromosome. The fitness for chromosome i is defined as Equation.1. After that, the chromosomes are sorted from highest fitness value to the lowest. We just remove the half of lower ranked chromosome out of the population. We apply proportional selection approach, selecting two chromosomes according to their fitness from the remaining population as parents to perform sexual crossover operation.

$$f_i = a_i \tag{1}$$

**Crossover** We use uniform crossover operation on the first two columns from parents to set the two offspring's first two columns. Starting from 4th columns, we apply one-point crossover operator to get the offspring's network structure. The position 4 of chromosomes will be automatically set according to the network structure. Finally for every offspring, we apply mutation operator on it.

**Mutation** For the mutation operator, we randomly change the value of columns except the third column according to the mutation rate.

#### 3.3 Baseline

In this section, two baseline models which are forward network classifier and forward network classifier with autoencoder will be discussed. The details of model structure can be directly seen from Figures (see Fig.2 and Fig.3)

Autoencoder Autoencoder was first proposed in the 1980s by [27] to deal with the problem "backpropagation without a teacher". And the autoencoder recently has been stacked and trained with bottom-up ways, and then fine-tuned with the supervised learning signal [11, 12].

In this paper, the autoencoder architectures have 5 layers with 256, 128, 64, 128, 256 units respectively. If the two layers have the same color, it indicates that these two layer share the same weight. The weight-sharing is following [6, 7] in order to reduce the free parameters and stable training. For example, the weight from  $I_1$  to  $H_1^1$  is as same as weight from  $H_1^3$  to  $O_1$ , shown in 2. Instead of using stack and bottom-up unsupervised methods to train the autoencoder, we directly train the whole layers in the model at the same time. The object function for the autoencoder model is mean squared error as the loss function.



Fig. 2: Structure of AutoEncoder Shared Weights(ASW) AutoEncoder Shared Weights (ASW) have 5 layers with 256-128-64-128-256. The same-colored layers share their weight. For example, the weight from  $I_1$  to  $H_1^1$  is as same as weight from  $H_1^3$  to  $O_1$ .

We propose two models, Classifier(CS) as in Fig.3a and Classifier with Autoencoder shared weight (CSASW) as two baselines. Our image input has 256 pixels ( $16 \times 16$ ), our first two classification models' architectures were designed as 4-layer forward networks with 256 units as input neurons, 128 hidden units as the first hidden layer, and 64 hidden units in the second hidden layer and 10 units in the final output layer standing for 10 digit classes (See Fig.3). Following suggestions from [8], the nonlinear activation function, rectified linear unit (ReLU) will be used in our model. The ReLU is the activation defined as  $h(z) = \max(0, f(x))$ . One special setting is the second classification model (Fig.2) set to use the encode layers from AutoEncoder model. It can be seen that the first 3 layers of classification model in the Fig.3b and AutoEncoder models in the Fig.2 have the same color. The object function for classification models will be cross entropy loss.

#### 3.4 Evaluation Method

The evaluation method will use test accuracy following [4]'s setting. The accuracy is the measurement of closeness to a standard or known value.

#### 3.5 Training Process

In this section, the process of training model will be discussed. The hyper parameters for classifier and classifier with autoencoder can be seen in Table.2. Weight decay technology can improve the generalization of the model according to [15], so we also set the weight decay rate for each model.





5

(b) Classifier With AutoEncoder Shared Weight (CSASW, 256-128-64-10)

Fig. 3: Classifier Structure

As in Fig.3a and Fig.3b, CS and CSASW's main structures are the same, having 4 layer (256-128-64-10), including input layer. In Fig.3b, the first 3 layers have the same weight as in the AutoEncoder model (same color in figures).

Table 2: The summary of hyper parameters for CS and CSASW								
Model	Optimization Algorithm	learning rate	Epochs for	Classification	Epochs for	Autoencoder	Weight	decay
CS	Adam	0.001	200		NA		0.0001	
CSASW	Adam	0.001	200		300		0.0001	

CS is our simple 3-layer forward network with 256-128-64-10 structure, where 256 is input size and 10 is output size, and 128 and 64 are the number of hidden units of two hidden layers respectively. And CSASW is the classifier with autoencoder shared weight. It has a similar structure with CS.

**EA** Assisting Model For the EA Assisting model, we directly run the EA algorithm according to the previous summaries. The details of the final selected model will be discussed in section 4.

Classifier (256-128-64-10) and Classifier with autoencoder For the classifier, we directly train the model on the train dataset and report the test accuracy. For the classifier with autoencoder, we first train the autoencoder for feature extraction or (can also be thought as dimension reduction) with 300 epochs and then replace the first 3 layers of the classification model with the first 3 layer from autoencoder models, as in Fig.2. When we train the classification model with autoencoder, we fine-tune the whole model, also adjusting the learned weights of the first 3 layers borrowed from the autoencoder model. We also apply the weight decay technology in Adam optimization steps as Table.1 shows.

#### **Results and Discussion** 4

	J
Model	Final Test Accuracy
MetaNet [4]	93.09%
CS (Adam)	95.30%
CSASW (Adam)	96.55%
Best Classifier from EA (ReLU-Adamax-3-1024-1024	-64) 98.84%
Second Best Classifier from EA (ReLU-Adamax-2-51	2-64) 98.12%

Table 3: Final Test Accuracy

#### 6 Hanshan Zhang

The final test accuracy can be found in Table.3. It is clear that the model proposed in this paper outperforms the initial baseline model, MetaNet from [4]. Comparing to the MetaNet's network structure and training process, Our models are much simpler and more efficient to train. When we compare CS and CSASW, adding autoencoder as the feature extractor also help the classifier to get higher performances. The most likely reason is that the learned feature vector captures the major information of different digits in the true data distribution, just like the finding from [11]. After 9 generations and the creation of 79 chromosomes, the optimal structure of classifier model has been identified, which has 3 hidden layers with 1024,1024,and 64 hidden units and ReLU as activation function, and Adamax as the optimization algorithm. The second best structure has 2 hidden layers with 512 and 64 hidden units, and ReLU as activation function, and Adamax as optimization algorithm. EA algorithm continues to boost the performance to 98.84% and 98.12%. The selected activation function of these two models also supports the finding in [8]. It is clear from the final test accuracy table that using EA algorithm can help us find the optimal structure of MLP, in order to get higher performances.

# 5 Conclusion and Future Work

#### 5.1 Conclusion

This paper concludes that the optimal structure of MLP model can be identified by EA algorithm without human supervision. From two baselines proposed in this paper, the test accuracy result also support that the autoencoder is an important feature extractor which can be used in the classification task.

### 5.2 Future Work

One clear future work direction is to apply EA algorithm to more complex neural network components, like Recurrent Neural Networks (RNN) and their varieties and Convolutional Neural Networks (CNN), to create models for other tasks, such as image reconstruction and natural language processing.

Another direction is to improve the speed of EA algorithm. Currently, since all the steps of EA algorithm are not paralleled, in order to find the optimal architecture of neural network, it will consume a lot of computational power, while it is clearly better than brute force.

# References

- Ahmad, F., Mat-Isa, N.A., Hussain, Z., Boudville, R., Osman, M.K.: Genetic Algorithm Artificial Neural Network (GA-ANN) hybrid intelligence for cancer diagnosis. Proceedings 2nd International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN 2010 pp. 78–83 (2010). https://doi.org/10.1109/CICSyN.2010.46
- Babu, U.R., Venkateswarlu, Y., Chintha, A.K.: Handwritten digit recognition using k-nearest neighbour classifier. Proceedings - 2014 World Congress on Computing and Communication Technologies, WCCCT 2014 pp. 60–65 (2014). https://doi.org/10.1109/WCCCT.2014.7
- [3] Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press (1995)
- Buscema, M.: Metanet\*: The theory of independent judges. Substance Use & Misuse 33(2), 439–461 (1998). https://doi.org/10.3109/10826089809115875, https://doi.org/10.3109/10826089809115875
- [5] Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) pp. 1–14 (nov 2015). https://doi.org/10.3233/978-1-61499-672-9-1760, http://arxiv.org/abs/1511.07289
- [6] Gedeon, T.D., Catalan, J.A., Jin, J.: Image Compression using Shared Weights and Bidirectional Networks. Proceedings 2nd International ICSC Symposium on Soft Computing (SOCO'97) pp. 374–381 (1997)
- [7] Gedeon, T.: Stochastic bidirectional training. In: SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218). vol. 2, pp. 1968–1971. IEEE (1997). https://doi.org/10.1109/ICSMC.1998.728185, http://ieeexplore.ieee.org/document/728185/
- [8] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
- [9] Gorgevik, D., Cakmakov, D.: Handwritten Digit Recognition by Combining SVM Classifiers. EURO-CON 2005 - The International Conference on Computer as a Tool" 2(February), 1393–1396 (2005). https://doi.org/10.1109/EURCON.2005.1630221
- [10] Graves, A.: Generating Sequences With Recurrent Neural Networks pp. 1–43 (2013). https://doi.org/10.1145/2661829.2661935, http://arxiv.org/abs/1308.0850
- [11] Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006). https://doi.org/10.1126/science.1127647
- [12] Hinton, G.E., Osindero, S., Teh, Y.W.: A Fast Learning Algorithm for Deep Belief Nets. Neural Computation 18(7), 1527–1554 (2006). https://doi.org/10.1162/neco.2006.18.7.1527, http://www.mitpressjournals.org/doi/10.1162/neco.2006.18.7.1527
- [13] Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? BT - Computer Vision, 2009 IEEE 12th International Conference on. Computer Vision, 2009 ... pp. 2146–2153 (2009), http://ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=5459469
- [14] Kingma, D.P., Ba. J.: Adam: А Method for Stochastic Optimization 1pp. (2014).https://doi.org/http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503, 15http://arxiv.org/abs/1412.6980
- [15] Krouk, G., Mirowski, P., LeCun, Y., Shasha, D., Coruzzi, G.: Predictive network modeling of the high-resolution dynamic plant transcriptome in response to nitrate. Genome Biology 11(R123) (December 2010). https://doi.org/doi:10.1186/gb-2010-11-12-r123, http://genomebiology.com/2010/11/12/R123/abstract, ja href='http://dx.doi.org/10.1186/gb-2010-11-12-r123'¿DOI (open access)j/a;
- [16] Lauer, F., Suen, C.Y.: A trainable feature extractor for handwritten digit recognition To cite this version : A trainable feature extractor for handwritten digit recognition. Pattern Recognition 40(6), 1816–1824 (2007). https://doi.org/10.1016/j.patcog.2006.10.011¿, https://hal.archives-ouvertes.fr/hal-00018426
- [17] Lawrence, S., Giles, C., Tsoi, A.C.: Lessons in neural network training: overfitting may be harder than expected. 14th National Conference on Articial Intelligence pp. 540–545 (1997)
- [18] LeCun, Y.: Une procédure d'apprentissage pour réseau a seuil asymmetrique (a learning scheme for asymmetric threshold networks). In: Proceedings of Cognitiva 85. pp. 599–604. Paris, France (1985)
- [19] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (November 1998)
- [20] LeCun, Y., Jackel, L.D., Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Muller, U.a., Sackinger, E., Simard, P., Vapnik, V.: Learning algorithms for classification: A comparison on handwritten digit recognition. Neural networks: the statistical mechanics perspective 2, 261–276 (1995)
- [21] LeCun, Y., Jackel, L.D., Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Muller, U.A., Sackinger, E., Simard, P., Vapnik, V.: Learning algorithms for classification: A comparison on handwritten digit recognition.

In: Oh, J.H., Kwon, C., Cho, S. (eds.) Neural Networks: The Statistical Mechanics Perspective. pp. 261–276. World Scientific (1995)

- [22] Liu, C.L., Nakashima, Κ., Sako, Н., Fujisawa, H.: Handwritten digit recognition: benchmarking state-of-the-art techniques. Pattern of Recognition 36(10). (2003).https://doi.org/https://doi.org/10.1016/S0031-3203(03)00085-2, 22712285http://www.sciencedirect.com/science/article/pii/S0031320303000852
- [23] Nair, V., Hinton, G.E.: Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on Machine Learning (3), 807–814 (2010). https://doi.org/10.1.1.165.6419
- [24] Oliveira, A.L., Mello, C.A., Silva, E.R., Alves, V.M.: Optical digit recognition for images of handwritten historical documents. Proceedings of the Ninth Brazilian Symposium on Neural Networks, SBRN'06 (2006). https://doi.org/10.1109/SBRN.2006.34
- [25] Pal, A., Singh, D.: Handwritten english character recognition using neural network. International Journal of Computer Science & ... 1(2), 141–144 (2010), http://csjournals.com/IJCSC/PDF1-2/30..pdf
- [26] Pradeep, J., Srinivasan, E., Himavathi, S.: Diagonal based feature extraction for handwritten character recognition system using neural network. 2011 3rd International Conference on Electronics Computer Technology pp. 364–368 (2011). https://doi.org/10.1109/ICECTECH.2011.5941921, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5941921
- [27] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chap. Learning Internal Representations by Error Propagation, pp. 318–362. MIT Press, Cambridge, MA, USA (1986), http://dl.acm.org/citation.cfm?id=104279.104293
- [28] Schaffer, J., Whitley, D., Eshelman, L.: Combinations of genetic algorithms and neural networks: a survey of the state of the art. [Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks pp. 1–37 (1992). https://doi.org/10.1109/COGANN.1992.273950, http://ieeexplore.ieee.org/document/273950
- [29] Simard, P., Steinkraus, D., Platt, J.: Best practices for convolutional neural networks applied to visual document analysis. Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. 1(Icdar), 958–963 (2003). https://doi.org/10.1109/ICDAR.2003.1227801, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1227801
- [31] Suganuma, M., Ozay, M., Okatani, T.: Exploiting the Potential of Standard Convolutional Autoencoders for Image Restoration by Evolutionary Search (2018), http://arxiv.org/abs/1803.00370
- [32] Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings (2010), 8609–8613 (2013). https://doi.org/10.1109/ICASSP.2013.6639346
- [33] Tactile Srl, B.: Semeion Handwritten Digit Data Set (1994), http://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+
- [34] Werbos, P.J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. thesis, Harvard University (1974)
- [35] Yeremia, H., Yuwono, N.A., Raymond, P., Budiharto, W.: Genetic algorithm and neural network for optical character recognition. Journal of Computer Science 9(11), 1435–1442 (2013). https://doi.org/10.3844/jcssp.2013.1435.1442