DOI 10.31891/2307-5732-2018-267-6(2)-188-192
UDC 004.93'12

P.M. RADIUK
Khmelnytskyi National University

NEUROEVOLUTION OF CONVOLUTIONAL NEURAL NETWORKS FOR THE CLASSIFICATION OF LUNG CANCER IMAGES

Convolutional neural networks demonstrate impressive results during medical imaging of lung cancer. It may be possible to make diagnoses with convolutional neural networks on conventional chest X-rays that are definitively apparent on subsequently computed tomography and biopsy. Computer vision may reduce the need for further evaluation with invasive testing or prevent errors of missed diagnoses. Using over twelve thousand images of proven lung cancer from the Prostate, Lung, Colorectal, and Ovarian dataset, we developed an algorithm to predict the presence or absence of lung cancer. The classification algorithm has achieved an accuracy of 96.09% with a positive predictive value of 99.11% and a negative predictive value of 93.25%.

Keywords: convolutional neural networks, genetic algorithms, hyperparameters, dataset, lung cancer.

П.М. РАДЮК
Хмельницький національний університет

НЕЙРОЕВОЛЮЦІЯ КОНВОЛІЦІЙНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ КЛАСИФІКАЦІЇ ЛІКУВАННЯ ЛЕГЕНЬ РАКУ

Згорткові нейронні мережі демонструють вражаючі результати під час медичної візуалізації раку легенів. Є можливість провести діагнози з використанням згорткових нейронних мереж за звичайними рентгенівськими знімками грудної клітки, які остаточно проявляються на подальшій комп'ютерній томографії та біопсії. Комп'ютерне бачення може зменшити потребу в подальшій оцінці за допомогою інвазивного тестування або запобігання помилкам пропущених діагнозів. Використовуючи понад дванадцять тисяч зображень виявленого раку легень із набору даних з раком простати, легень, колоректалу та яєчників, ми представляємо алгоритм для прогнозування наявності або відсутності раку легенів. Алгоритм класифікації досягнув точності розпізнавання 96.09% з позитивною прогностичною цінністю 99.11% та негативною прогностичною цінністю 93.25%.

Ключові слова: згорткові нейронні мережі, генетичні алгоритми, гіперпараметри, набір даних, рак легень.

Introduction

Over recent decades, the number of lung cancer patients has increased dramatically. Former or current smokers and those exposed to radiation or chemicals in the workplace have especially higher risks to the disease. According to the recent researches, over 1,465,000 people die every year from cancers, 18.2% of which is a variant of lung cancer [1]. The tumours resulting from this disease at a particular stage are visible to experienced radiologists on such mediums as chest X-rays, computed tomography scans, and positron emission tomography scans. The images of lung cancer serve as a preventative measure in several cases. Late detection of the disease leads to fatal consequences. Nowadays the level of survival from lung cancer is about 10% [2].

Owing to continued research into deep learning and convolutional neural networks (CNNs) during the past several years, image classification and object detection have shown tremendous improvements in performance [3]. Not only does the success of CNNs owe to computing power and large datasets but also the innovations into the model structure [4]. Replacement of the activation function on ReLU [5], insertion of dropout layers [6], fully connected layers, and various optimisation techniques significantly changed and improved approaches of using CNNs [7], [8].

Genetic algorithms are search heuristics that try to imitate the process of natural selection to find possible solutions to optimisation as well as search problems. They have two main components: genetic representation of the solution space and the ability to evaluate the fitness of solutions. At first, we form every possible CNN architecture through our genetic encoding scheme. Having trained the model on the training data, we evaluate the fitness of the solution. In the end, we test the trained solution on our test set, which then becomes the solution's fitness [9].

Genetic algorithms begin with the original population of genes and populations of problem-solving. Once every solution in the population is evaluated, they all are chosen based on their suitability for modification to create a new generation of solutions. In the result, the more fit the solution, the more likely it has a descendant. The population becomes better at solving the task with time. The algorithm ends when at least one individual across all generations is recognised as the best solution to the problem.

The use of neural networks in medical diagnosis

The use of data analysis on medical images is not a new approach to medical diagnosis. However, many applications of computer vision for medical applications struggle due to the medical data being noisy, inexact, sparse, or just too big. Therefore, algorithms of medical diagnosis based on CNNs have been continually improving.

In a recent project performed at the University of Bern, a group of researchers created a deep CNN architecture for the classification of lung diseases based on lung slices from computed tomography images that performed with an accuracy of 85.5% on its dataset [10]. They successfully classified lung computed tomography image patches between six different lung diseases. The dataset consisted of 2,032 different diseases. To handle such a diverse dataset, they balanced their classes using a dynamic tree-taxonomy. To eliminate the problem of small

training classes, researchers generated classes based on the number of examples rather than final diagnoses. It resulted in having 757 classes, instead of 2,032.

In another project performed at the Federal University of Parana, a CNN performed to classify images of cell slides of breast cancer patients [11]. Using the BreaKHis dataset [12], researchers used AlexNet to classify microscopic biopsy images of benign and malignant breast tumours. Each slide of breast tissue contained four images, each with different levels of magnification. To handle the high-resolution nature of their dataset, they invoked a few techniques. The first was the use of sliding windows with 50% overlap, and the second was random crops of the raw image with no overlap.

The goal of the article and tasks to fulfil

For this paper, we used the modified NEAT algorithm, dubbed DeepNEAT [13], to evolve the architecture of CNN. We inserted convolution and pooling layers with pseudorandom hyperparameters into a minimal architecture and then optimised the weights through backpropagation on the training set. The fitness of a model is the final accuracy on the test set after two epochs of training.

During neural networks deployment, dozens of parameters require optimisation. Optimising parameters through any search algorithm is impractical, especially one based on chance.

We claim the following tasks that must be fulfilled:

1. The encoding had to be able to encode directed acyclic graphs of variable size;
2. Be able to track topological innovations over time,
3. Be able to create a new, coherent, individual from the genes of two parent individuals;
4. The encoding scheme must inherently allow for an efficient search for optimal network architectures.

Few encoding schemes satisfy all these requirements. We chose direct encoding in the form of graph encoding. More specifically, we used Schiffman encoding [14]. Its basic structure is a list of neurons with their connectivity information. We program our own rules for mutations, so mutations do not result in illegal phenotypes. Each vertex in the graph represented a layer in a CNN, and also stored hyper-parameter information for the construction node.

The NEAT algorithm was slightly modified in order to evolve a CNN. First of all, we defined primary mutations. Inject Node injects a random node (convolution, pool, or ReLU) with pseudo-random hyperparameters into the genome's network between a pre-existing connection. Before injecting the new node, we checked to ensure that it would produce a valid network. If it did, the injection occurred, if it did not, we changed the hyperparameters to values that would result in a valid network. This approach solves the problem of convolving the image to zero dimensions, as is guaranteed to occur as the number of injects increases. Inject Segment injects a pair of convolution with a ReLU, as well as a pool layer in a preexisting connection. Point Mutate changes the essential hyperparameters of a node.

Dataset

The dataset used for training and testing was compiled from the Prostate, Lung, Colorectal, and Ovarian Cancer Screening Trial (PLCO) dataset [15]. The dataset contains images of randomised and controlled trials to determine whether specific screening exams reduce the mortality of prostate, lung, colorectal, and ovarian cancer. Approximately 155,000 participants took part in the screening portion of the trial from 1993 to 2006. If a participant developed cancer at any point during the screening phase, all CXRs preceding the diagnosis were considered to be cancerous and marked as positives, which resulted in over 9,200 positives in the dataset. The typical instance of the PLCO dataset is presented in figure 1.

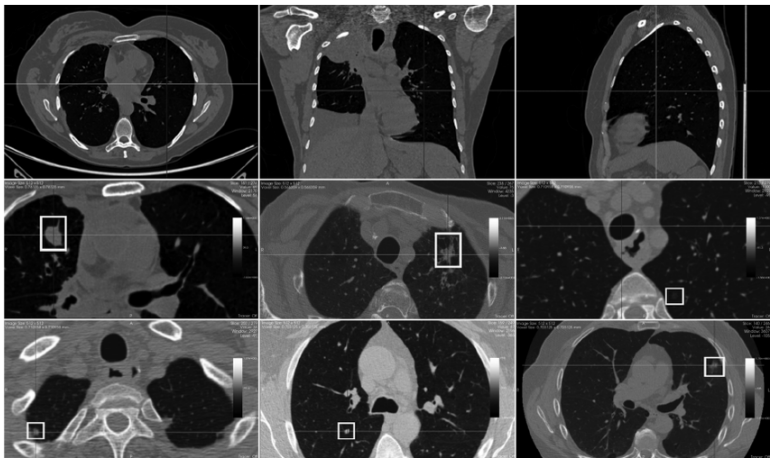

Fig. 1. From the PLCO dataset to lung cancer diagnosis

The original PLCO image dataset occupied 2.2TB in TIF format with individual images of chest radiographs having an approximate size of 2000x3000 pixels. In [16], the dataset was improved through down-scaling and then cropped. The current dataset was uniformly downscaled to 256 x 256 pixels and stored in PNG format. The dataset was randomly split into 70% and 30% for training and testing, respectively. In order to train the DeepNEAT algorithm, we selected several CNN models and applied them using the machine learning framework TensorFlow v. 1.10.0 [17].

Training

During the training, we used a gene pool of fifty individuals, throughout ten generations. The DeepNEAT algorithm received mutation rate parameters, which determined the frequency of each mutation. Each mutation received specific rate (table 1).

Table 1

The mutation rate of the genetic algorithm

Mutation	Rate, %
Inject Convolution	50
Inject Pooling	50
Add ReLU	30
Point Mutate	45
Inject Segment	15

At the first stage of training, the genetic algorithm graphically coded the network. After that, we export the created graph to the TensorFlow model. Each model received an identical set of hyperparameters. The names of hyperparameters and their values are listed in table 2.

Table 2

Parameters of the CNN

Hyperparameter	Value
Optimisation function	SGD
Epochs	5
Learning rate policy	INV
Learning rate	0.01
Momentum	0.9
Weight Decay	0.0005

Results

We conducted several experiments with different sizes of generations of the population. According to the training results, fifty individuals for ten generations were the most efficient in producing a fit population, as well as producing very fit top models. We trained five hundred models and outperformed the state-of-the-art classification models by 4%. The recognition accuracy increases with the increase in the number of generations of the DeepNEAT algorithm (fig. 2).

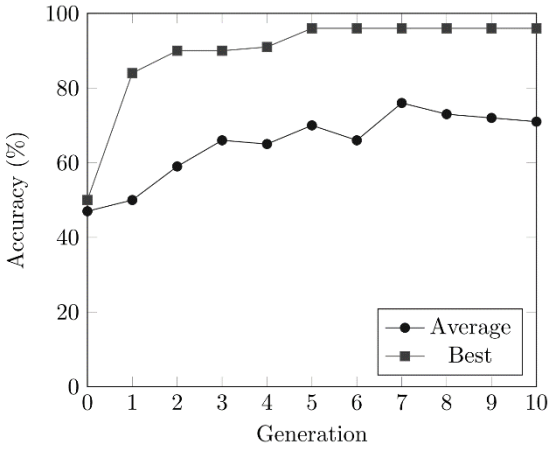

Fig. 2. Improvement of accuracy over generations

To test the effectiveness of DeepNEAT, we compared it to the best models from the past 5 years. As shown in table 3, the accuracy of DeepNEAT exceeds every model. DeepNEAT executed in its entirety in 4 hours, whereas AlexNet, ResNet-151 and GoogLeNet took over 21, 40 and 35 hours to train, respectively.

Table 3

Testing accuracy for applied models

Model	Test accuracy, %	Time training, hours
AlexNet	79.88	21.06
GoogleNet	89.34	40.19
ResNet-151	92.03	35.56
DeepNEAT	96.09	4.89

For binary classifiers, such as the DeepNEAT model, we calculated several statistics regarding the performance of the algorithm. In table 4 we introduce the contingency statistical data, which present the frequency of the real condition variable and the predicted condition variable.

Table 4

Binary classifier evaluation contingency table

	True positive	False negative	False positive	True negative
Population, numbers	1224	94	11	1298
Rate, %	92.87	7.13	0.84	99.16

From table 4 many ratios can be derived, most notably the rates of false positives and negatives among the population. The statistical ratios were calculated from these inputs and presented in table 5. To calculate the contingency rates, the network, based on the DeepNEAT model, has been calculated on all 1,884. Statistical coefficients were calculated from these data and are presented in table 5.

Table 5

Statistical rations of predicted and confirmed population samples

Positive predictive rate, %	99.11
False discovery rate, %	0.89
False omission rate, %	6.75
Negative predictive value, %	93.25
Positive likelihood ratio, %	51.29
Negative likelihood ratio, %	7.19
Accuracy, %	96.09

The best model generated by the DeepNEAT algorithm achieved an accuracy of 96.09% with a positive predictive rate of 99.11% and a negative predictive rate of 93.25%.

Conclusion

The final accuracy of the network over a set check of 96.09% indicates that the model has succeeded in learning the functions associated with the presence of different types of confirmed lung cancer in these images. Given that the human radiologist must spend a significant amount of time in each image to make the correct prediction, indications for many types of cancer are not often seen early, causing diagnoses often arriving during the late stages of these diseases. This model can handle images at a speed of 3.41 milliseconds each; the potential for using such a model as the previous step of screening can save many lives from early detection and misdiagnosis. The automation provided by this tool may reduce costs as well as increase the speed and accuracy of diagnoses.

References

1. Ambrosini V. PET/CT imaging in different types of lung cancer: an overview / V. Ambrosini, S. Nicolini, P. Caroli // Eur J Radiol. – 2012. – Volume 81, Issue 5. – P. 988–1001. – URL : https://doi.org/10.1016/j.ejrad.2011.03.020

2. Yang P. Clinical Features of 5,628 Primary Lung Cancer Patients: Experience at Mayo Clinic from 1997-2003 / P. Yang, M. Allen, M. Aubry, J. Wampfler // Chest. – 2005. – Volume 128, Issue 1. – P. 452–462. – URL : https://doi.org/10.1378/chest.128.1.452

3. Simonyan K. Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman // arXiv preprint arXiv:1409.1556. – 2014.

4. Krizhevsky A. ImageNet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G. Hinton // Advances in neural information processing systems. – 2012. – P. 1097–1105. – URL : https://doi.org/10.1145/3065386

5. Romanuke V. V. Appropriate number and allocation of ReLUs in convolutional neural networks / V. V. Romanuke // Research Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute". – 2017. – No. 1. – P. 69 – 78. – URL : https://doi.org/10.20535/1810-0546.2017.1.88156

6. Romanuke V. V. Appropriateness of DropOut layers and allocation of their 0.5 rates across convolutional neural networks for CIFAR-10, EEACL26, and NORB datasets / V. V. Romanuke // Applied Computer Systems. – 2017. – Volume 22. – P. 54 – 63. – URL : https://doi.org/10.1515/acss-2017-0018

7. Romanuke V. V. Optimal training parameters and hidden layer neuron number of two-layer perceptron for generalised scaled object classification problem / V. V. Romanuke // Information Technology and Management Science. – 2015. – Volume 18. – P. 42 – 48. – URL : https://doi.org/10.1515/itms-2015-0007

8. Radiuk P. M. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets / P. M. Radiuk // Information Technology and Management Science. – 2017. – Volume 20. – P. 20 – 24. – URL : https://doi.org/10.1515/itms-2017-0003

9. Angeline P.J. An Evolutionary Algorithm that Constructs Recurrent Neural Networks / P.J. Angeline, G.M. Saunders, J.B. Pollack // IEEE Transactions on neural networks. – 1994. – Volume 5, Issue 1. – P. 54–65. – URL : https://doi.org/10.1109/72.265960

10. Anthimopoulos M. Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network / M. Anthimopoulos, S. Christodoulidis, L. Ebner // IEEE Transactions on Medical

Imaging. – 2016. – Volume 35, Issue 5. – P. 1207–1216. – URL : https://doi.org/10.1109/TMI.2016.2535865

11. F.A. Spanhol Breast cancer histopathological image classification using Convolutional Neural Networks / F.A. Spanhol, L.S. Oliveira, C. Petitjean, L. Heutte // International Joint Conference on Neural Networks. – 2016. – ISSN: 2161–4407. – URL : https://doi.org/10.1109/IJCNN.2016.7727519

12. F.A. Spanhol A Dataset for Breast Cancer Histopathological Image Classification / F.A. Spanhol, L.S. Oliveira, C. Petitjean, L. Heutte // IEEE Transactions on Biomedical Engineering. – 2015. – Volume 63, Issue 7. – P. 1455–1462. – URL : https://doi.org/10.1109/TBME.2015.2496264

13. Miikkulainen R. Evolving Deep Neural Networks / R. Miikkulainen, J. Liang, E. Meyerson // arXiv:1703.00548. – 2017.

14. Schiffmann W. Performance evaluation of evolutionarily created neural network topologies / W. Schiffmann, J. Merten, R. Werner // Parallel Problem Solving from Nature. – 1990. – № 496. – P. 274–283. – URL : https://doi.org/10.1007/BFb0029764

15. The Prostate, Lung, Colorectal and Ovarian Cancer Screening Trial [Electronic resource] // Cancer Data Access System. – URL : https://biometry.nci.nih.gov/cdas/datasets/plco/21/

16. Hafemann L. Forest Species Recognition Using Deep Convolutional Neural Networks / L.G. Hafemann, L.S. Oliveira, P. Cavalin // 22nd International Conference on Pattern Recognition. – 2014. – ISBN: 978–1–4799–5209–0. – URL : https://doi.org/10.1109/ICPR.2014.199

17. Abadi M. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems / M. Abadi, A. Agarwal, P. Barham // OSDI'16 Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation. – 2016. – P. 265–283. – ISBN: 978-1-931971-33-1.

Рецензія/Peer review : 7.11.2018 р. Надрукована/Printed :19.12.2018 р.
Рецензент: д.т.н., проф. Романюк В.В.