Original

Recognition of Image One Feature Point Using Convolutional Neural Networks

Miki Hori, Makoto Jincho, Tadasuke Hori, Hironao Sekine, Akiko Kato, Atsuko Ueno and Tatsushi Kawai

1 Department of Dental Materials, School of Dentistry, Aichi Gakuin University, Nagoya, Japan
2 Center for Advanced Oral Science, School of Dentistry, Aichi Gakuin University, Nagoya, Japan
3 Department of Oral Anatomy, School of Dentistry, Aichi Gakuin University, Nagoya, Japan
4 Department of Gerodontology and Home Care Dentistry, Division of Implant Dentistry, School of Dentistry, Aichi Gakuin University, Nagoya, Japan

(Accepted for publication, February 16, 2021)

Abstract: Most studies of artificial intelligence in the medical field involve classification problems, but few consider recognition of one characteristic point in images or regression analysis such as data recognition. In this research, we constructed a fundamental convolutional neural network framework for regression analysis. Images of the handwritten digit “3” from the MNIST dataset were used as training data, with the protruding middle point as an image feature point. Input images and training data (x1, y1) were connected to 6 convolutional layers and then run through 2 affine layers to produce the output data (x2, y2). The loss function was the mean radial error (MRE) between the training and output data. After machine learning, the error converged to 0.75 pixels on average. We expect that this algorithm can be clinically applied to points having certain characteristics in images, such as locating hard tissue lesions or recognizing measurement points in cephalograms.

Key words: Artificial intelligence (AI), Convolutional neural network (CNN), Feature point recognition, Image analysis, Regression problem

Introduction

Over 60 years has passed since the term “artificial intelligence (AI)” was first proposed in 1956 at a Dartmouth University workshop (1,2). There have been several AI “booms” since then. The first is now called “good old-fashioned AI” (3), referring to the time when neural network algorithms that can perform inference and computer searches were being developed. The high-level language FORTRAN was first released around that time (4), and there have been many other achievements, such as the perceptron (5,6). The second boom focused on giving computers information for learning, that is, “knowledge,” in order to construct higher-level recognition systems. Backpropagation of error, which speeds up calculations, was developed in this period (7), and learning speeds dramatically improved. However, AI systems make predictions by “learning” characteristic features through “calculations,” and because these calculations require large amounts of memory and computing power, AI is highly reliant on hardware performance. In both eras, the performance and speed of AI were far from expectations for practical use, so these efforts were abandoned in less than 10 years.

The third boom started around 2000 with new advances in hardware, and the AI field has been explosively progressing ever since (8). The programming language Python, which is now used for most AI development, was created during this era (9). This language allows the use of many modules, such as Keras, for AI development. In recent years, there has been increasing research into the development of base frameworks in Python for the advanced training of convolutional neural networks (CNNs), which can be applied in fields such as natural language processing (10), speech recognition (11), and data analysis (12). In medical imaging in particular, CNNs have demonstrated superior performance in various applications such as pathological analysis (13-15). Such projects can be generally divided into “classification” problems for grouping data by type and “regression” problems for predicting the output of continuous quantities. Most current AI projects using medical images are classification problems (16-18), and studies addressing regression problems are rarely seen. Our goal is the application of AI to recognize, for example, feature points in dental X-ray images, such as positions of lesions, areas of tooth decay, locations of tooth cusps, and measurement points in cephalograms. As a first step, we aimed to build a fundamental system that solves regression problems to recognize a certain feature point in a simple image.

Materials and Methods

We extracted 6,133 images of the handwritten digit “3” from among 70,000 images in the MNIST (Modified National Institute of Standards and Technology) dataset (19)(Fig. 1). All images had dimensions of 28 × 28 pixels. The protruding point in the center of a “3” was set as a feature point. The following describes in detail the method by which we created training data, the fundamental CNN framework, and the verification method.

Training dataset

AI-based learning requires large amounts of training data. We developed a custom program in Python to automatically input one characteristic point (x1, y1) from each image in CSV format, and 2 workers created training data from all 6,133 images.
CNN architecture and training

We developed the CNN program with reference to Saito\textsuperscript{19}, using Python version 3.6.3 with various extensions, such as modules for numerical and image processing. Specifically, we used Numpy, the Python Image Library for image processing, and matplotlib to graph the learning progress.

For machine learning, the use of various types of images, rather than similar ones, is recommended in order to avoid overfitting\textsuperscript{20}, which refers to learning results that are compatible with training data but do not provide correct answers for unknown data. In other words, if training data coordinates \((x_1, y_1)\) in this project are concentrated only around the image center, learning will suggest that correct answers must be near the image center, making it difficult to recognize the target feature points elsewhere. Therefore, we cropped images fed to the CNN to \(42 \times 42\) pixels, with the training images randomly placed therein. This dataset did not include image transformations such as rotation or enlargement.

Fig. 2 shows a flowchart of the CNN. The network consisted of 6 convolutional layers and 3 max pooling layers. The filter size and number of filters for each layer are specified in Fig. 2. Each convolutional layer is followed by a rectified linear unit (ReLU). In the last layers, all the units are fully connected to output probabilities for 2 scalar values \((x_2, y_2)\) using the softmax function. The loss function is defined as the absolute distance between \((x_1, y_1)\) and \((x_2, y_2)\), as described in Evaluation. We used Adam as the optimizer, with a learning rate of 0.001. The number of training epochs was set to 400.

The development environment was an Apple MacBook Pro (Apple Inc., Cupertino, CA, USA) with a 3.1 GHz Dual Core Intel Core i7 processor and 16 GB of 1867 MHz DDR3 RAM. The operating system was MacOS Catalina 10.15.4.

**Evaluation**

The radial error \(R\) is formulated as \(R = \sqrt{\Delta x^2 + \Delta y^2}\), where \(\Delta x\) and \(\Delta y\) are the absolute distance in the \(x\)- and \(y\)-directions, respectively, between the obtained output data and the training data. The mean radial error (MRE) is defined as \(\text{MRE} = \frac{\sum_i R_i}{N}\).

**Results**

About 18 hours was required to perform machine learning. Fig. 3 shows the learning curve of validation for each epoch. The learning curve had mostly converged after 40 epochs, and MRE decreased with each epoch. For example, the deviation was 0.920 pixels after 200 epochs, but finally converged to about 0.747 pixels. Fig. 4 shows examples of automatically detected points.
Using the CNN, one characteristic point could be recognized in images with high accuracy. We devised a calculation of the loss function to solve the regression problem. In the classification problem, the item indicating the most probable value in the final output layer is calculated as “correct” or “incorrect,” and its weight is adjusted. In this study, this was solved by making adjustments such that the x- and y-coordinate point values were directly output in the final layer, the MRE with the training data \((x_1, y_1)\) was used as the accuracy, and the weight was adjusted to decrease this value.

We built a CNN from scratch, along with a system that recognizes a specific point in a simple image. AI quantifies the incoming information and extracts parts where values such as edges change significantly. This extracted part becomes the convolutional layer and the pooling layer. This approach imitates recognition in the human brain\(^{21}\). Humans divide information from the optic nerve using 6 layers and gradually emphasize characteristic information before passing it to cognitive regions. It is difficult for humans to recognize the entirety of huge amounts of data, and the same is true for AI, so important parts are extracted while reducing the amount of data. To recognize one target point from this extracted feature point, the accuracy is improved while reducing distance from the training data. In AI learning, a random point is presented first, and when the distance from the correct answer (the derivative gradient) is large, it converges so that the gradient becomes smaller. Finally, the point is found where the differential coefficient is smallest (the distance from the correct answer is closest to 0).

There are various parameters in this learning process. For example, the optimizer, which is a gradient method for deep learning, includes Adam\(^{22}\), as well as stochastic gradient descent (SGD) and AdaGrad. SGD is the earliest of these algorithms and is considered suitable for simple cases such as when there is a clear convergence point. AdaGrad\(^{23}\) is advantageous in that it automatically updates the learning rate. Adam combines a gradient descent method like Momentum that imitates motion of inertia with AdaGrad, and is currently considered the state of the art. Accuracy in this study was most improved when Adam was selected as a result of verification, but other cases may result in different results. Thus, it is essential to improve the accuracy of all parameter settings through trial and error.

When using AI, learning can seemingly be done in almost the same way, by adjusting how training data is created, selecting various parameters for the CNN layer structure, and evaluation (setting the loss function). Following this basic structure, we plan to develop applications such as point detection or range recognition in various medical images, which is our final goal.

**Acknowledgements**

We are grateful to Mrs. Yumiko Uchiyama and the deceased Mr. Shunsuke Yoshimitsu for help in creating the training data set. This research was supported by the Nitto Foundation (2018).

**Conflict of Interest**

The authors have declared that no COI exist.

**References**

1. Kline R. Cybernetics, automata studies, and the Dartmouth conference on artificial intelligence. IEEE Ann History Computing 33: 5-16, 2011
2. Park WJ and Park JB. History and application of artificial neural networks in dentistry. Eur J Dent 12(4): 594-601, 2018
3. Bobrow DG. Natural language input for a computer problem solving system. MIT libraries, http://hdl.handle.net/1721.1/6903, 1964 (accessed: February 8th, 2021)
4. Backus JW, Beeber RJ, Best S, Goldberg R, Haight LM, Herrick HL, Nelson RA, Sayre D, Sheridan PB, Stern H, Ziller I, Hughes RA and Nurr T. The FORTRAN automatic coding system. IRE-AIEE-ACM 1957: doi: org/10.1145/1455567.1455599
5. Rosenblatt F. Principles of neurodynamics, perceptrons and the theory of brain mechanisms. Spartan books, 1962.
6. Widrow B and Lehr MA. 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. Proceedings of the IEEE 78(9): 1415-1442, 1990
7. Rumelhart DE, Hinton GE and Williams RJ. Learning internal representation by error propagation. MIT Press 18(8): 319-362, 1985
8. Presutto M. Current A.I. Trends: Hardware and Software Accelera-

---

**Discussion**

Using the CNN, one characteristic point could be recognized in images with high accuracy. We devised a calculation of the loss function to solve the regression problem. In the classification problem, the item indicating the most probable value in the final output layer is calculated as “correct” or “incorrect,” and its weight is adjusted. In this study, this was solved by making adjustments such that the x- and y-coordinate point values were directly output in the final layer, the MRE with the training data \((x_1, y_1)\) was used as the accuracy, and the weight was adjusted to decrease this value.

We built a CNN from scratch, along with a system that recognizes a specific point in a simple image. AI quantifies the incoming information and extracts parts where values such as edges change significantly. This extracted part becomes the convolutional layer and the pooling layer. This approach imitates recognition in the human brain\(^{21}\). Humans divide information from the optic nerve using 6 layers and gradually emphasize characteristic information before passing it to cognitive regions. It is difficult for humans to recognize the entirety of huge amounts of data, and the same is true for AI, so important parts are extracted while reducing the amount of data. To recognize one target point from this extracted feature point, the accuracy is improved while reducing distance from the training data. In AI learning, a random point is presented first, and when the distance from the correct answer (the derivative gradient) is large, it converges so that the gradient becomes smaller. Finally, the point is found where the differential coefficient is smallest (the distance from the correct answer is closest to 0).

There are various parameters in this learning process. For example, the optimizer, which is a gradient method for deep learning, includes Adam\(^{22}\), as well as stochastic gradient descent (SGD) and AdaGrad. SGD is the earliest of these algorithms and is considered suitable for simple cases such as when there is a clear convergence point. AdaGrad\(^{23}\) is advantageous in that it automatically updates the learning rate. Adam combines a gradient descent method like Momentum that imitates motion of inertia with AdaGrad, and is currently considered the state of the art. Accuracy in this study was most improved when Adam was selected as a result of verification, but other cases may result in different results. Thus, it is essential to improve the accuracy of all parameter settings through trial and error.

When using AI, learning can seemingly be done in almost the same way, by adjusting how training data is created, selecting various parameters for the CNN layer structure, and evaluation (setting the loss function). Following this basic structure, we plan to develop applications such as point detection or range recognition in various medical images, which is our final goal.
9. Rossum GV. Python (programming language). https://thereaderwiki.com/en/Python_(programming_language). (accessed: October 22nd, 2020)

10. Liddy ED. Natural Language Processing. In: Encyclopedia of Library and Information Science, 2nd ed, Marcel Decker, Inc., New York, 2001.

11. Hannun A, Case C, Casper J, Catanzaro B, Diamos G, Elsen E, Prenger R, Satheesh S, Sengupta S, Coates A and Ng AY. Deep speech: scaling up end-to-end speech recognition. arXiv:1412.5567, 2014

12. Xindong Wu. Data mining: artificial intelligence in data analysis. Proceedings. IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004: doi: 10.1109/IAT.2004.1342916

13. Hemanth DJ, Vijila CKS and Anitha J. A survey on artificial intelligence based brain pathology identification techniques in magnetic resonance images. Int J Rev Comput. 2076-331X, 2009

14. Hosny A, Parmar C, Quackenbush J, Schwartz LH and Aerts HJWL. Artificial intelligence in radiology. Nat Rev Cancer 18(8): 500-510, 2018

15. Niazi MKK, Parwani AV and Gurcan MN. Digital pathology and artificial intelligence. Lancet Oncol 20(5): e253-e261, 2019

16. Bera K, Schalper KA, Rimm DL, Velcheti V and Madabhushi A. Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology. Nat Rev Clin Oncol 16: 703-715, 2019

17. Hung K, Montalvao C, Tanaka R, Kawai T and Bornstein MM. The use and performance of artificial intelligence applications in dental and maxillofacial radiology: a systematic review. Dentomaxillofac Radiol 2020: doi: 10.1259/dmfr.20190107

18. Yann LeCun. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/ (accessed: October 22nd, 2020)

19. Saito K. Zero kara manabu Deep Learning -Python de manabu deep learning no riron to jissou. O’Reilly Japan. 2016. (Japanese)

20. Lawrence S and Giles CL. Overfitting and neural networks: Conjugate gradient and backpropagation. Proc IEEE 2000: doi: 10.1109/IJCNN.2000.857823

21. Fong RC, Scheirer WJ and Cox DD. Using human brain activity to guide machine learning. Sci Rep 8: 5397, 2018

22. Kingma DP and Ba JL. ADAM: A method for stochastic optimization. 3rd ICLR, San Diego, 2015

23. Duchi J, Hazan E and Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12: 2121-2159, 2011

164