Performance of Different Graph Neural Networks on Graph Classification

Zikang Zhan
Dame Allan’s School, Newcastle upon Tyne, NE49YJ, UK
*Corresponding author’s e-mail: vivian.wang@cas-harbour.org

Abstract. In recent years, machine learning has become a common approach for solving problems of artificial intelligence, which is developing at a high speed. Convolutional Neural Network (CNN) is popular on solving image problems for its believable accuracy. However, it does not mean that other neural networks are useless on these problems. In this paper, performance of Convolutional Neural Network (CNN), Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN) in the same database are presented. Performance of different types of networks are compared, and a conclusion is given that recurrent neural network might be able to classify pure black and white images at high accuracy, and that convolutional neural network performs better than multilayer perceptron on graph classification problems.

1. Introduction
Nowadays, neural networks are successful on machine learning such as in the aspects of graph classification, object recognition, language translation and speaking recognition[4, 7]. To solve the problems of graph classification, Convolutional Neural Networks (CNN) is the most popular method due to its high accuracy rate. However, few people follow up performances of other types of neural networks, so in this paper the same datasets are used to train different types of neural networks and their performance will also be demonstrated. Using proper network models to solve specific problems is definitely more efficient and effective. It is believed that these results could be helpful on the choice of the model training in the future.

Through analysis, the neural networks picked in this paper are CNN, MLP and RNN because in terms of applications, they represent different areas respectively, such as natural language processing and graph classification. The author trained these three kinds of network models via two common datasets for graph classification, which are Modified National Institute of Standards and Technology (MNIST) and Canadian Institute For Advanced Research (CIFAR)-10, and showed their results of evaluation and the graphs of training curves to vividly watch the process. Eventually, this paper listed some possible uses of CNN, MLP and RNN and presented a research direction about RNN.

2. Overview of Graph Neural Network
In computer science, a graph is a data structure used to represent objects (nodes) and the connections (edges) between them. In reality, a large number of systems can be represented as graphs such as social networks, structure of protein and physical system[1]. Neural networks were firstly applied on directed acyclic graphs in 1997. Based on the study of that, Graph Neural Network (GNN) was born in 2005 and was further researched in 2009[4].
One of the features of GNN is that it represents original input as dense vectors. For instance, it can transform words into word vector, transform articles into article vector and transform graphs into graph vector to classify.

GNN is a powerful tool for modeling structural data. However, the original GNN still has some disadvantages. To illustrate, iteratively updating hidden layers is not efficient. Moreover, GNN usually uses the same parameters in the iteration but actually, what parameters the popular networks use are various in different layers[1]. In addition, some features on the edges cannot be accurately represented on original GNN.

Based on that above, a variety of neural networks for particular purposes occurred. An overview can be seen in figure 1.

Figure 1. An overview of variants of graph neural networks[1].
3. Analysis

3.1. Principles of the networks and databases

3.1.1. Graph. In mathematics, graph is a structure, which consists of edges and vertices. Each vertex represents an object, and each edge represents the relationship between two objects. Graph could be presented as the set of vertices and edges, \( G=(V, E) \), where \( V \) is the set of vertices and \( E \) is the set of edges. The following figure 2 is an example of graph, where \( V=\{v1,v2,v3,v4,v5\} \), and \( E=\{(v1,v5),(v1,v4), (v1,v2), (v2,v3)\} \).

![Figure 2. An example of the graph.](image)

3.1.2. Graph Neural Network (GNN). With the development of neurosciences and cognitive science, it has been already known that the smart behaviors of human beings have something to do with the activities of brains. Scientists were inspired to create a model to simulate neural system, which was called neural network. In the area of machine learning, neural network consists of numerous neurons, and the connections among neurons are the weights which can be adjusted. Every neuron is composed of input signal, linear combination and non-linear activation function. And the network contains multiple layers, input layer, hidden layers and output layer, which consists of multiple neurons. This is the basic neural network. Data structure studied in many scientific areas is a non-Euclidean space, such as social network in social sciences and regulatory networks in genetics[6], so GNN is a great approach to deal with them. GNN is a big class, which can be mainly divided into two directions, Convolutional Neural Network (CNN) and Recurrent Neural Network(RNN). CNN processes image data and RNN processes time series data.

3.1.3. Convolutional Neural Network (CNN). CNN is a deep feed-forward neural network, which is successful on visualizations tasks such as graph classification and target detection. In CNN, convolution and pooling are the most important operations. Each pixel in an image is treated as a node, and its neighbors are judged via the size of filters like 3*3 and 4*4[1]. The size of the filter is usually smaller than that of the image. For example, to get a node in a hidden layer, a good approach is to take the weighted average value of feature value of the nodes in the filter. Convolution uses the filter to scan the image for several times until the whole image finishes and generates a set of new data. Pooling is an operation of reducing calculation and parameters, including two common ones, max-pooling and average-pooling.
3.1.4. Multilayer Perceptron (MLP). MLP is also called fully-connected neural network or feed-forward neural network. It contains multiple layers, input layer, hidden layers and output layer, which consists of multiple neurons. Weights are connections among layers, and each neuron in a layer is connected with each neuron in the next layer.

3.1.5. Recurrent Neural Network (RNN). RNN is different from the feed-forward neural networks. The nodes with connections form recurrent directed graphs, showing dynamic changes over time. Therefore it suits the tasks like audio recognition and handwriting recognition.

3.1.6. Modified National Institute of Standards and Technology (MNIST). MNIST database is a classical database of handwritten digits, commonly used for training a variety of image processing systems. The MNIST database contains 60,000 training images and 10,000 testing images in ten different classes, where the black and white images were normalized to fit into a 28*28 pixel bounding box and anti-aliased. The ten classes represent zero, one, two, three, four, five, six, seven, eight and nine.
3.1.7. Canadian Institute For Advanced Research (CIFAR)-10. CIFAR-10 database is a classical database of collection of colourful images that commonly used for training a variety of image processing systems. The CIFAR-10 database contains 60,000 32*32 images in ten different classes, which are 50,000 training images and 10,000 testing images. The ten classes are airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships and trucks.

3.2. Process of analysis

In this analysis, MNIST and CIFAR-10 were run via CNN, MLP and RNN respectively. MNIST database is a classical database of handwritten digits, commonly used for training a variety of image
processing systems. The MNIST database contains 60,000 training images and 10,000 testing images in ten different classes, where the black and white images were normalized to fit into a 28*28 pixel bounding box and anti-aliased. The ten classes represent zero, one, two, three, four, five, six, seven, eight and nine.

Here are the structures of network used for MNIST:

Table 1. Structures of networks used in MNIST.

| CNN | Layers(type) | Output Shape |
|-----|--------------|--------------|
|     | Conv2D       | (24,24,32)   |
|     | Conv2D       | (20,20,64)   |
|     | MaxPooling2D | (10,10,64)   |
|     | Flatten      | (6400)       |
|     | Dropout      | (6400)       |
|     | Dense        | (128)        |
|     | Dropout      | (128)        |
|     | Dense        | (10)         |

| MLP | Layers(type) | Output Shape |
|-----|--------------|--------------|
|     | Dense        | (512)        |
|     | Dropout      | (512)        |
|     | Dense        | (512)        |
|     | Dropout      | (512)        |
|     | Dense        | (10)         |

| RNN | Layers(type) | Output Shape |
|-----|--------------|--------------|
|     | SimpleRNN    | (50)         |
|     | Dense        | (10)         |
|     | Activation   | (10)         |

CIFAR-10 database is a classical database of collection of colourful images, commonly used for training a variety of image processing systems. The CIFAR-10 database contains 60,000 32*32 images in ten different classes, which are 50,000 training images and 10,000 testing images. The ten classes are airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships and trucks.

Here are structures of network used for CIFAR-10:

Table 2. Structures of networks used in CIFAR-10.

| CNN | Layers(type) | Output Shape |
|-----|--------------|--------------|
|     | Conv2D       | (32,32,32)   |
To make most of parameters same, the number of epochs was 20, and the batch size was 32[2], and the used database was same.

**3.3. Results of analysis**

**MNIST:**

| Type of network | Accuracy | Loss |
|-----------------|----------|------|
| CNN             | 0.9085   | 0.2761|
| MLP             | 0.8897   | 0.3423|
| RNN             | 0.8366   | 0.4502|
Figure 8. The graphs of training curves in MNIST.

CIFAR-10:

Table 4. Evaluation results of CIFAR-10.

| Type of network | Accuracy | Loss  |
|-----------------|----------|-------|
| CNN             | 0.7320   | 0.7699|
| MLP             | 0.3738   | 1.7712|
4. Discussion

Obviously, training accuracy of three networks mentioned above all rose, which means CNN, MLP and RNN can all be trained for graph classification. However, there were differences among their efficiencies. In MNIST, the accuracy evaluated of CNN and MLP were both close to 0.9000, but that of RNN was lower than theirs, and loss evaluated of RNN was always approximately to 0.5000 later. This result showed that RNN had the ability to classify different types of graphs but not so well as CNN and MLP. The images in CIFAR-10 are colourful while the images in MNIST are black and white. The difference is that images in CIFAR-10 have three channels but that of MNIST only has one channel, which results in the failure of RNN in CIFAR-10. It is possible to use RNN to classify pure black and white images at a high precision, so how to improve the performance of RNN on graph classification could become an interesting research direction in the future. Actually, the majority of RNN is applied to the area of natural language processing such as language translation, poetry generation and speech recognition, rather than graph classification[5].

In CIFAR-10, RNN was not considered temporarily for two reasons. One reason is that the performance of RNN in MNIST is failed, so it is abandoned to continue by the author. The other reason is that images in CIFAR-10 have three channels, and simple RNN cannot be trained by that, based on the features of RNN.

The accuracy of CNN was 0.7320, followed by that of MLP with 0.3738. The loss of MLP did no longer continue to drop or roughly kept unchanged when the epoch approximately reached 5. By analyzing the results of MNIST and CIFAR-10, CNN and MLP performed better than RNN on graph classification obviously. The performances of CNN and MLP in MNIST were similar, so it was difficult to judge which one was more perfect. However, when looking at the performances in CIFAR-10, the performance of MLP was not as well as expected and was even more awful than that of CNN. CNN almost performed much better than MLP. Additionally, at the start of CNN and MLP, the absolute value of gradient of curve of CNN was larger than that of MLP, which means CNN could learn more efficiently than MLP. The gradient can show the rate of learning which is vital for a large project. Whether it is on accuracy or efficiency, the MLP lost, compared with the CNN, so CNN can do better on processing of graphs. MLP is fully connected, and due to this feature, MLP can be applied in almost every area and lots of neural networks can be transformed from MLP based on the specific purposes. Then basic MLP could be used as a benchmark to judge if the new neural network is successful.

5. Conclusion

In this paper, CNN, MLP and RNN were selected to be trained via MNIST database and CIFAR-10 database respectively. Through the results of analysis, the performance of RNN is the worst in the used three kinds of neural networks, but its accuracy on MNIST proves that RNN is able to be used to solve some graph classification problems, such as handwriting images. Therefore, improving the performance of RNN on classification of black and white images is a worthy research direction. To
sum up, CNN and MLP can both perform well on graph classification, but for most of the databases, CNN is more efficient and more precise. Fully-connected MLP can be transformed to other networks based on certain aims, so MLP could be used as a good benchmark.

In addition, the author has only tested two common datasets, not containing every training dataset for graph classification. More datasets will be added in for future testing. Furthermore, it is a worthwhile research direction for improving RNN to try to recognize three-channel images.

Acknowledgement
My deepest gratitude goes first and foremost to Prof. Petro Lio in University of Cambridge for academic guidance. Prof. Petro Lio is the first tutor for me on artificial intelligence area and taught me knowledge of artificial intelligence and correct experimental design concepts. I also want to express my appreciation to my parents and friends for their support.

References
[1] J. Zhou, G. Q. Cui, Z. Y. Zhang, C. Yang, Z. Y. Liu, L. F. Wang, C. C. Li and M. S. Sun. Graph Neural Networks: A Review of Methods and Applications. arXiv preprint arXiv:1812.08434.
[2] Dominic Masters and Carlo Luschi. Revisiting Small Batch Training For Deep Neural Networks. arXiv preprint arXiv:1804.07612.
[3] Keyulu Xu, Weihua Hu, Jure Leskovec and Stefanie Jegelka. (2019) How Powerful Are Graph Neural Networks? In International Conference on Learning Representations (ICLR).
[4] Z. H. Wu, S. R. Pan, F. W. Chen, G. D. Long, C. Q. Zhang and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. arXiv:1901.00596.
[5] Tsung-Hsien Wen, Steve Young. (2020) Recurrent neural network language generation for spoken dialogue systems[J]. Computer Speech & Language, 63.
[6] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam and Pierre Vandergheynst. Geometric deep learning: going beyond Euclidean data. arXiv:1611.08097.
[7] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Thomas Laurent, Yoshua Bengio and Xavier Bresson. Benchmarking Graph Neural Networks. arXiv:2003.00982.