Optimal Random Layer Selection Method for Image Classification

Jongchae Lee[0009−0008−7651−4039], Kanghyun Jo[0000−0001−8317−6092]

School of Electrical Engineering,
University of Ulsan, Ulsan, South Korea,
Department of Electrical, Electronic and Computer Engineering,
University of Ulsan, Ulsan, South Korea
jabc1240@gmail.com, acejo@ulsan.ac.kr
https://islab.ulsan.ac.kr/

Abstract. Designing and creating CNN(Convolution Neural Network) structures for existing deep learning models has historically demanded substantial time and effort. This Paper was inspired by the recent paper, GNoME [5], from Google, which explores using deep learning to uncover crystal structures analogous. In this method, For the first, Creating a basic CNN structure, designated as the parent node. Child nodes are then generated from this parent node with adding random layers behind the parent node's structure. And then, Each child node trains again. After training, the child node with the highest accuracy is selected to become the new parent node and this process iterates until the desired accuracy is achieved. This method similar to tree-like structures in random forests is tailored to modify CNN architectures to better fit the dataset being used. Preliminary results indicate an approximate 2% on CIFAR-10 and 7% on CIFAR-100 increase in accuracy compared to conventional pre-trained models like ResNet [2], DenseNet [3] and VGGNet [6]. This paper introduces and explores this genetic-inspired methodology for evolving CNN structures, proposing a method for improving model accuracy.

Keywords: CNN · Random Addition · Genetic Learning · Neural Architecture Search

1 Introduction

DNN(Deep Neural Networks) have successfully addressed challenges that were difficult to solve before the era of deep learning. The structures of models that excel in various problem domains are difficultly designed by domain experts. Determining the appropriate model structure for a given problem and diverse datasets is a challenging and time-consuming task. Efforts to address this challenge and create better models have led to the emergence of various CNN structures, such as autoencoders and generative models. The key challenge in forming new CNN structures lies in how to effectively train the model to learn from data and increase accuracy. Therefore, improving the accuracy of CNN models requires the combination of convolution, activation functions, pooling, and various techniques to create a flexible architecture. For example, LeNet [4] effectively combined pooling, padding, and activation functions to form a CNN architecture, and subsequent models like VGGNet [6] and ResNet [2] have been developed. However, There have been many efforts to improve CNN structures by introducing new convolution techniques or architectures than focusing on the order of combining convolution and pooling for accuracy improvement. Nonetheless, effectively combining the structure of models to enhance accuracy is a time-consuming task that requires considerable resources. In addition to this, this paper was inspired from the methodology of identifying material crystal structures in the GNoME paper and proposes a method to randomly generate CNN structures and identify the model with the highest accuracy among them. Also draws inspiration from the structure of random forests [1] in machine learning. In the proposed method, pruning takes place from parent nodes to child nodes. The CNN structure grows by randomly generating and continuously expanding branches. Subsequently, the model accuracy of child nodes is compared each. Child node which has highest accuracy is selected and updated as new parent nodes, repeating the process. Consistently finding child node models superior to parent nodes allows for the discovering of a CNN structure achieving the desired accuracy.

2 Related Work

2.1 CNN

Convolutional Neural Networks (CNNs) have been at the forefront of image processing and pattern recognition tasks. Over the years, numerous CNN architectures have been proposed contributing to the improvement of various applications. Research on CNN architectures reflects various efforts to develop diverse neural network architectures and enhance performance. Some notable works in the realm of CNN architectures include:

1. VGGNet : It constructed deep neural networks that demonstrated outstanding performance in image recognition tasks. With a simple yet deep structure, it emphasized using small filter sizes to extract features effectively.
2. ResNet: It introduced an innovative structure by incorporating residual blocks to address the vanishing gradient problem in deep networks. This enabled effective learning in even deeper networks.

This paper aims to create diverse CNN architectures by randomly incorporating layers or techniques from existing CNN structures. The goal is to assess the performance of these architectures as different layers and methods are introduced randomly.

2.2 Random Forest Architecture

Random Forest [1] Architecture is renowned as an effective algorithm in the field of machine learning. This structure(fig 1) is composed of an ensemble of decision trees

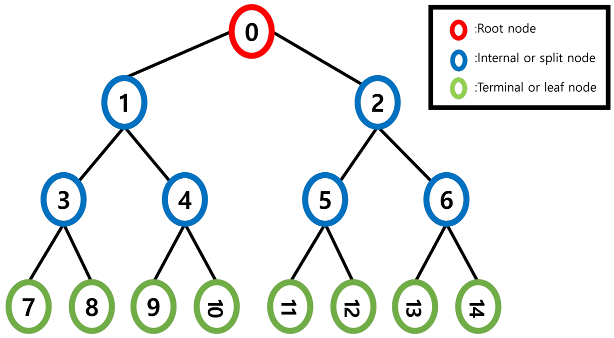

Fig. 1: Decision Tree

where each tree is responsible for learning from a specific subset of data. Random Forest automatically learns the characteristics of the data and forms a generalized prediction model based on it. This structure is highly effective in terms of stability and performance, finding applications in various fields. Paper leverages these characteristics of Random Forest to inherit the parent node's CNN structure and introduce a novel approach. By adding random layers to the child nodes during the learning process, It aims to utilize the strengths of Random Forest to create models and capitalize on the selected features to enhance the accuracy of the model. This approach involves generating diverse models and identifying the one that exhibits superior performance.

2.3 Neural Architecture Search

Neural Architecture Search (NAS) [7] is a crucial topic in the field of Automated Machine Learning (AutoML), focusing on the technical exploration of finding optimal neural network structures. This field emphasizes the automatic discovery of the best architecture for a given task, aiming to surpass manually designed neural network structures and uncover models with superior performance. In the past, the manual design of neural network structures was common, making it challenging to find the optimally tailored architecture for specific problems. However, NAS seeks to overcome these limitations. The primary methodology of NAS involves the iterative use of Recurrent Neural Network (RNN) controllers to determine each layer. These controllers attempt various combinations of layers and explore new architectures based on performance. Recent research has introduced alternative approaches, deviating from RNN controllers and incorporating other meta-models or optimization techniques. Additionally, there are studies attempting to apply traditional machine learning methods such as Random Forests, NAS for performance enhancement. The advancements in NAS contribute to the efficiency and performance improvement of deep learning models, garnering attention as a major research topic in the AutoML field.

3 Proposed Method

To find the best CNN model for data, create several models and extract the best model. The mechanism proposed by this paper is the following algorithm.

Algorithm 1 Genetic Learning Algorithm

```
1: Initialize parent node P with initial structure and weights
2: Initialize target accuracy target_accuracy
3: Initialize current accuracy current_accuracy ← 0
4: Repeat until current_accuracy ≥ target_accuracy
5:    Repeat for a specified number of iterations: for i ← 1 to 5 do
        end
        Generate 5 child nodes C
6: Duplicate parent node structure and weights to create C_i
7: Add a random layer to C_i
8: Train C_i using the training data
9: Select the child node C_best with the highest accuracy
10: Set P as the parent node of C_best
```

3.1 Tree Architecture

This paper proposes an approach inspired by the Tree Architecture of Random Forest algorithm but diverges in its methodology. Leveraging the tree structure of Random Forests, this paper initiate the process by constructing a basic and straightforward Convolution Neural Network (CNN) structure. This initial model serves as the root and also parent node, undergoing the training process. Subsequently, Create five child nodes imitating the parent node. These child nodes retain the CNN structure of the parent model and then append random layers on it. Each child node undergoes its training process, and the one which has the highest accuracy is selected to update the new parent node. And It is described on fig 3 This iterative process continues, expanding the tree and generating diverse CNN structures. Through this method, algorithm can discover the CNN structure that achieves the highest accuracy on the dataset among numerous branches. Importantly, during training, the child nodes learn with the weights inherited from the parent node model fig 2. This inheritance mechanism allows child nodes to build upon the training results of the parent model and lead to potentially superior learning outcomes. This paper refers to this process as "genetic inheritance". Without genetic inheritance, if each child node were to undergo training independently, it would require a more time to learn effectively. By utilizing genetic inheritance, this method aim to reduce training time and guide the child nodes to benefit from the parent node's weights. Through repeated iterations, the final child node demonstrates higher accuracy than any previous nodes in the tree.

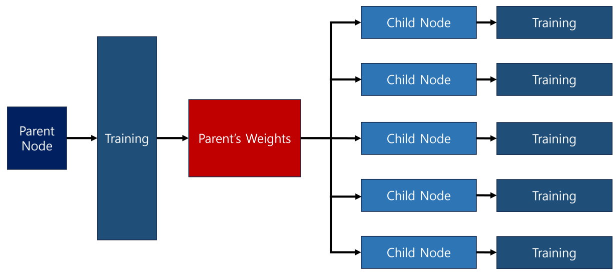

Fig. 2: Child nodes re-learn with the weight of parent nodes. This is called 'genetic Inheritance'

3.2 Randomly generated layers

To facilitate the random addition of layers to child nodes replicating the parent node's CNN structure, a list of randomly selectable layers is generated. The added layers consist of convolution, Rectified Linear Unit (ReLU) activation function, Depthwise-Separable Convolution, dropout, global average pooling, skip connections and, Squeeze-and-Excitation (SE) block. Randomly adding these layers aims to blend techniques from various deep learning models, creating a tree structure like fig 3 with diverse CNN configurations. The objective is to find the highest accuracy among these different structural CNN structures. The selection of random layers is based on equal probabilities for each layer type in the list. This process of adding random layers repeats both before training the child nodes and after training the parent node during the creation of each child node. The goal is to identify the model with the most suitable CNN structure, i.e., the model with the highest accuracy, tailored to the utilized dataset.

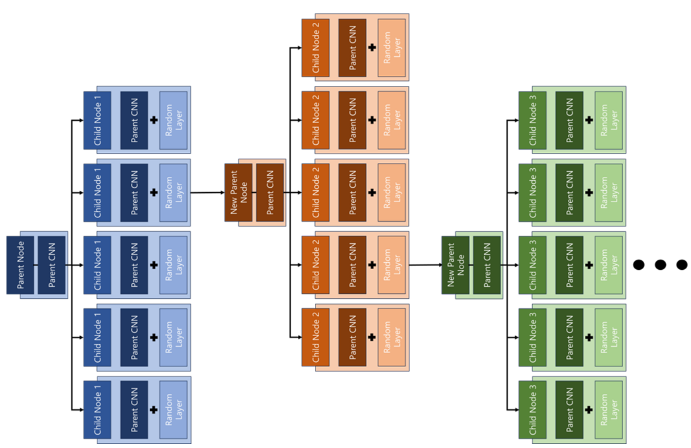

Fig. 3: Mimicking the Structure of Random Forest: Tree Architecture

4 Experiment

Create a simple CNN structure to be used as the first parent node. After training this, child nodes for the parent node are created and each child node proceeds with learning. The child node with the highest accuracy is selected among them and updated to a new parent node. After that, it is repeated until the accuracy aimed at the previous content is reached. Since performance is expected to increase as the structure of the model deepens and diversifies, the child node with the highest accuracy is selected, updated to the parent node, and repeated. For the experiment, the CNN structure obtained through the method presented in this paper is compared with the accuracy obtained through three learning models, ResNet [2] and VGGNet [6], DenseNet [3] which are representative image classification models.

4.1 Dataset

The datasets used for training are CIFAR-10, CIFAR-100. The reason for using two datasets is the expectation that the CNN structures for learning will differ across datasets with different classes. Therefore, the experiment aims to generate and examine random structures to confirm the creation of CNN architectures that are suitable for specific datasets. The goal is to verify the generation of random CNN structures tailored to each dataset through this experiment.

4.2 Implementation Setup

All experiments were conducted in the same environment, and the configuration environment was Intel Core i5-12600k, NVIDIA RTX 3080 x 1EA, 32GB memory. The training process was 50 epochs in parent node and child node, and all hyper parameters like learning rate, batch size were same.

4.3 Result

The experimental results are detailed in table 1, table 2. The models used for training include ResNet, VGGNet, DenseNet, and the proposed model from this paper. In comparison to the accuracy of each model, the proposed model shows approximately 2% higher accuracy on the CIFAR-10 dataset and about 7% higher accuracy on the CIFAR-100 dataset than the model which has the most accuracy in the comparison group.

Used Layers are as follow: Conv : Convolution + ReLU, SC: Skip Connection, BN: Batch Nomalization, Pooling: Global Average Pooling, DC: Depthwise-Separable Convolution, SE: Squeeze and excitation layer. And detailed CNN Structure of Ours is on Fig.4 and Fig.5

Table 1: Result of Training and Test on CIFAR-10

Model	ACC[%]	Model	ACC[%]	Model	ACC[%]
ResNet18	**70.72%**	VGG11	75.91%	DenseNet121	69.4%
ResNet34	70.64%	VGG13	78.11%	**DenseNet161**	**73.81%**
ResNet50	67.69%	**VGG16**	**79.43%**	DenseNet169	69.38%
ResNet101	68.31%	VGG19	78.04%	DenseNet201	69.92%
ResNet152	69.28%			**RGCNN_1(OURS)**	**81.38%**

Table 2: Result of Training and Test on CIFAR-100

Model	ACC[%]	Model	ACC[%]	Model	ACC[%]
ResNet18	37.01%	VGG11	40.42%	DenseNet121	39.17%
ResNet34	37.34%	VGG13	40.38%	**DenseNet161**	**42.74%**
ResNet50	34.98%	**VGG16**	**41.28%**	DenseNet169	39.22%
ResNet101	36.74%	VGG19	37.38%	DenseNet201	39.44%
ResNet152	**37.51%**			**RGCNN_1(OURS)**	**50.39%**

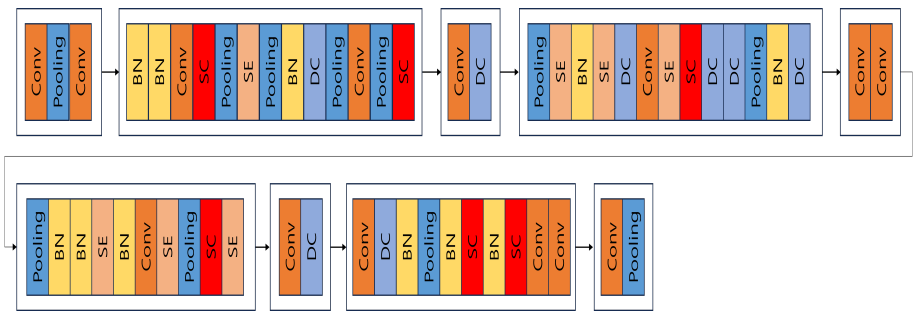

Fig. 4: Final CNN Structure on CIFAR-10 Dataset

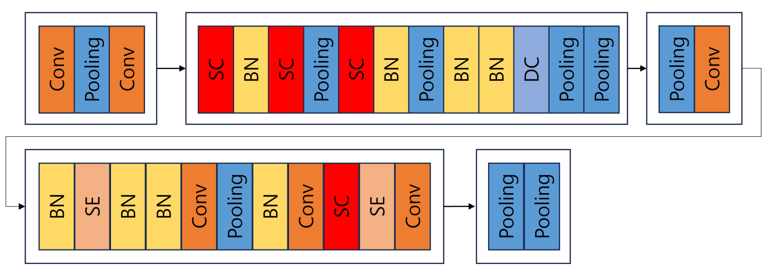

Fig. 5: Final CNN Structure on CIFAR-100 Dataset

5 Conclusion

The proposed method involves generating child nodes from parent nodes and progressively stacking random layers, creating entirely new CNN structures. Due to the random nature of the generation, child nodes with suboptimal performance may emerge. However, conversely, child nodes with superior performance are also generated. Through iterative child node generation, selecting models with superior performance, and continuously adding layers, the approach leads to better overall performance. Despite the potential for suboptimal child nodes, the strategy of favoring and expanding on models with higher accuracy results in a more effective and accurate CNN architecture.

6 Acknowledgements

This result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

References

1. Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
3. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
4. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
5. Merchant, A., Batzner, S., Schoenholz, S.S., Aykol, M., Cheon, G., Cubuk, E.D.: Scaling deep learning for materials discovery. Nature pp. 1–6 (2023)
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
7. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)