AUTOMATED DEEP NEURAL NETWORKS WITH GENE EXPRESSION PROGRAMMING OF CELLULAR ENCODING

Towards the Applications in Remote Sensing Image Understanding

18D5252 Clifford Broni-Bediako Prof. Masayasu Atsumi

Research scientists are gearing up for adopting deep learning methods to their respective domain problems. This work presents a novel generative encoding that combines the strengths of gene expression programming and cellular encoding for automatic architecture search of deep neural networks (AutoDNN) to develop complex modularized convolutional neural networks (CNNs). The experimental results of the proposed method are demonstrated to be extremely competitive to manually designed ones in the domain of visual perception, particularly in remote sensing image understanding tasks.

Keywords: deep learning, neural architecture search, convolutional neural networks, evolutionary algorithm, random search, remote sensing image understanding, image classification and segmentation

1 INTRODUCTION

1.1 Motivation

Deep neural networks (DNNs) such as convolutional neural networks (CNNs) have enabled remarkable progress in the application of machine learning (ML) and artificial intelligence (AI). Neural architecture search (NAS), also known as AutoDNN, aims to automate the architecture search of neural networks to enable researchers adopt DNNs with ease, and with little or no expertise in deep learning (DL).

As metaheuristic approach, NAS requires a representation scheme to encode the candidate solutions (architectures). Direct encodings of genetic algorithms (GA) and genetic programming (GP) have been widely employed in NAS methods [1]. Though easy to implement, direct encoding cannot be easily modularize and the lack of distinctive separation of genotype and phenotype spaces limits their functional complexity [2, 3]. Therefore, it may be difficult for direct encodings to evolve modules (building-blocks) with shortcut and multi-branch connections [4] which can improve training and enhance network performance [5]. Evolving CNN architectures with such modularity properties, as commonly adopted by human experts, represents one of the key motivations of this work.

The alternative to direct encoding is a generative encoding which can produce modular and regular structures [2]. The second motivation of this work is pertained to exploring the strengths of two generative encodings, gene expression programming (GEP) [3] and cellular encoding (CE) [6], with the aim of harnessing their strengths into a new encoding scheme. GEP is known for its simplicity in implementation and multi-gene chromosomes with flexible genetic modification, whereas CE has the ability to produce modular neural networks. Both GEP and CE are well established evolutionary computation methods which have experienced a lot of development and theoretical study. A large part of this previous work involves architecture search of artificial neural networks (ANNs) in a small scale, and therefore this work provides the possibility for CNNs architecture development.

1.2 Research Aims and Objectives

Based on the aforementioned motivations and from application perspective, the main aims of this work are as follows:

1. To explore the capability of GEP with CE in evolving CNNs architectures. In particular, to investigate the viability of combining the encoding schemes of GEP and CE for CNNs architecture representation.
2. To investigate whether the multi-gene chromosomes and modularity properties of GEP and CE respectively can be used to evolve network modules with shortcut connections and multi-branch connections.
3. To investigate the expressiveness and tractability of the representation space developed with GEP of CE.
4. To investigate the suitability of applying GEP of CE to evolve CNNs for visual perception tasks and its robustness when transferred to others benchmarks.

And to achieve these aims, we propose the accomplishment of the following objectives:

1. Design and development of a novel generative encoding which adopt the simplicity features of GEP with modularity properties of CE to model representation space of CNN architectures.
2. Development of optimization algorithms able to optimize the architecture search of CNNs automatically based on objective (1), given specific tasks.
3. Evaluation of the automatically evolved CNNs on various visual perception tasks in order to validate that they are competitive to, if not better than, manually designed ones. The tasks should cover a spectrum of remote sensing image understanding tasks such as scene classification and aerial image segmentation.

2 RELATED WORK

2.1 Neural Architecture Search (NAS)

Automatically searching and learning neural network architectures is not a new idea in the AI community [7]. The earlier studies were mostly related to evolving neural network controllers for robots (evolutionary robotics) [8]. The work by Zoph *et al.* [9] in 2017 has attracted researchers into the field of NAS to evolve CNNs for computer vision and natural language processing tasks [10]. Most earlier studies in NAS searched for network architectures and their connection weights at a small scale [7]. However, since CNNs have millions of connection weights, recent studies searched only for the network architectures and learn their connection weights via back-propagation method [9, 11, 12]. With the growing interest in AutoDNN, various search strategies have been used to explore the space of CNN architectures, including random search, evolutionary algorithms (EA), reinforcement learning, gradient-based methods and Bayesian optimization [13]. Historically, EA were used in evolution-

ary robotics [8], and have demonstrated as a computationally feasible method for AutoDNN [1]. Random search is a simple method, easy to implement and uses less computational resources. It can achieve results that are competitive to the ones of the sophisticated search methods, if the architecture search space is not intractable and overly expansive [14, 15]. We employ EA and random search strategies and focus on developing an effective and a tractable space of network representations to find well-performing CNN architectures for visual perception tasks. We also adopt cell-based search space of architecture representation, since architectures discovered by cell-based approach are transferable, and they perform better than global-based search space [11, 12].

2.2 Remote Sensing Image Understanding

Despite the specific features of remote sensing (RS) images with respect to spectral, spatial and radiometric resolutions, the interpretation of RS images involves extracting information from the images and visual perception tasks as computer vision. Traditionally, handcrafted feature extraction methods with support vector machine [16] and artificial neural network (ANN) [17] classifiers are employed in RS image understanding. The advent of large volumes of (very) high resolution remotely-sensed imagery [18] and with the need to provide accurate interpretation for applications such as urban planning, land resource management and environmental monitoring [19] has demanded the RS community to adopt CNNs which are capable of learning automatically insightful features from large volumes of imagery data, and has shown strong generalization ability than the statistical learning methods [20]. CNNs have achieved excellent results in various RS image understanding tasks such as scene classification and aerial image segmentation [20, 21], and this work seeks evaluate the performance of automatically evolved CNNs on such tasks.

3 Proposed Method

This work is built upon the strength of two generative encodings, *gene expression programming* and *cellular encoding*, which are described in Section 3.1. Section 3.2 presents the objective function defining the problem of aforementioned research objectives. The proposed encoding scheme is presented in Section 3.3. Sections 3.4 presents preliminary experimental results.

3.1 Preliminaries

3.1.1 Gene Expression Programming (GEP)

GEP is a bio-inspired method introduced by Ferreira in 2001 [3]. Its chromosomes consist of linear fixed-length genes similar to the ones in GA, and are developed in phenotype space as expression-trees similar to the parse-trees in GP. The genes are structurally organized in a head and a tail format called Karva notation. With simple, linear and compact chromosomes and a distinct separation of genotype and phenotype spaces, GEP is more flexible and effective compared to GA and GP. Ferreira [22] has proposed that GEP can evolve ANNs via evolutionary process. Though it is easy to implement and flexible in genetic operations because of its linear structure, it can not be easily modularized. The Karva notation lacks structure-preserving representation, hence a good evolved building-block is very likely to be destroyed by genetic modification in the subsequent generations [23]. We adapt the linear representation of GEP and infuse it with modularity features to induce compact chromosomes which are able to evolve modularized CNN architectures via random search and evolutionary algorithm.

3.1.2 Cellular Encoding (CE)

Also inspired by biological development, CE is a neuron-centric encoding method introduced by Gruau in 1994 [6]. It uses graph grammar that control the division of nodes to encode ANNs. The graph grammar are represented as grammar-tree called program which facilitates the development of a modular and hierarchical networks via evolutionary process. Whilst CE has modularity property which can improve performance, it is not without weakness, as genetic modifications are applied according to GP paradigm [24] which limits its flexibility in crossover and mutation operations. We adopt the graph grammar (modularity features) of CE and embed them into linear fixed-length string of GEP, which we called *symbolic linear generative encoding* (SLGE). This enables SLGE to encode building-blocks of different shape and size with a simple linear fixed-length strings to evolve modularized CNN architectures.

3.2 Objective Function for Architecture Search

We formulated the NAS problem as follows. Given the problem space $\psi = \{\mathcal{A}, \mathcal{S}, \mathcal{P}, t\mathcal{D}, v\mathcal{D}\}$, where $\mathcal{A}$ is the architecture search space, $\mathcal{S}$ represents the search strategy, $\mathcal{P}$ denotes the performance measure, and $t\mathcal{D}$ and $v\mathcal{D}$ are the training and validation datasets respectively, the objective is to find a small CNN architecture $a^* \in \mathcal{A}$ via random search or evolutionary search strategy $\mathcal{S}$, which maximizes the performance measure $\mathcal{P}$ of accuracy on the validation dataset $v\mathcal{D}$ after training it on the training dataset $t\mathcal{D}$. Small architecture here means a CNN model has the number of parameters θ less than or equal to the target network size. Mathematically, the objective function $\mathcal{F}$ for the automatic architecture search of CNNs can be formulated as:

$$\begin{aligned} \mathcal{F}(\psi) = \max_{\theta,\ a} \ \mathcal{P}(\mathcal{L}(a(\theta), t\mathcal{D} \mid \mathcal{S}, a \in \mathcal{A}), v\mathcal{D}) \\ s.t. \text{ the number of parameters } \theta \leq \mathcal{T}_{params} \end{aligned} \tag{1}$$

where $\mathcal{L}$ represents the training of the model parameters θ with the loss function and $\mathcal{T}_{params}$ denotes the target number of parameters.

3.3 Symbolic Linear Generative Encoding (SLGE)

This section presents the proposed SLGE which can encode cells (building-blocks) with shortcut (skip) and multi-branch connections to evolve modularized CNN architectures.

3.3.1 Genotype Representation

The chromosome is a simple program to grow a cell. The program is a linear fixed-length structured string which consists of multiple genes with head and tail similar to GEP representation (see Fig. 1). The head composes of a CE graph grammar and the tail is made up of common CNN convolution operations. Given the length of a gene head h, the tail length t is a function of h expressed as: $t = h + 1$, thus, the length of a gene is $2h + 1$. The CE graph grammar adopted are described as follows (see illustration in Fig. 2):

- SEQuential division (*SEQ*): it splits current node into two and connects them in serial.

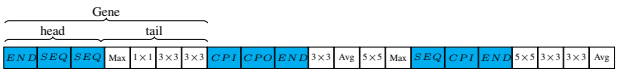

Figure 1: Schematic representation of SLGE genotype

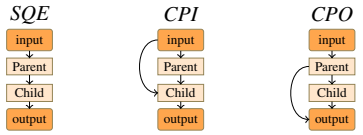

Figure 2: Illustration of CE graph grammar

- CoPy Input division (*CPI*): it performs *SEQ*, then shares the same inputs with parent and child nodes.
- CoPy Output division (*CPO*): it performs *SEQ*, then shares the same outputs with parent and child nodes.
- END program (*END*): it stops the developing process.

In Fig. 1, 1×1, 3×3 and 5×5 represent Conv1×1, Conv3×3 and Conv5×5 regular convolution operations respectively, and Avg and Max represent average pooling and max pooling operations respectively, which are introduced to achieve flexibility in the representation space to develop modules similar to human experts designed ones like Inception-ResNet-Blocks [4].

3.3.2 *Phenotype Representation*

The phenotype of a candidate solution is a cell, that is a directed acyclic graph: $G = (V, E)$, where V is a set of nodes and E is a set of connections. The input and output nodes are input and output tensors respectively, and the other nodes represent various convolution operations. The convolution operations without successor are depthwise concatenated to produce the output tensor, and if an operation has more than one predecessor, the feature maps of the predecessors are added together. The connections are latent information flow direction in the architecture (see Fig. 3).

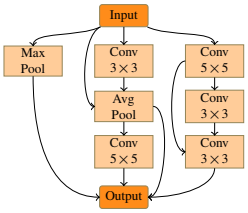

Figure 3: Schematic representation of SLGE phenotype. This is the cell representation of the chromosome in Fig. 1

3.3.3 *Genotype-Phenotype Mapping*

The development of each phenotype starts from an initial cell G_τ with the input and the output nodes. Then each gene in the phenotype's chromosome is developed as a subgraph G_i with the first convolution operation in its tail as hidden node connected to input and output nodes. The subgraph G_i is developed by applying the head program of the gene to its tail part. The final cell G_τ is the merging of the subgraphs $G_{i:n}$ at the input and output nodes, that is $G_\tau = \Gamma(G_{i:n})$, where Γ is the function to merge the subgraphs $G_{i:n}$. The evolved cells are repeatedly stacked, for a predefined number of times, to build modularized CNN architectures as candidate solutions.

3.4 Preliminary Experiments

The preliminary experiments aimed at verifying the viability and expressiveness of the proposed encoding method for discovering cells to evolve modularized CNN architectures. Because remotely-sensed imagery datasets are mostly large, for a fast preliminary experiments, two relatively small general purpose image classification datasets (CIFAR-10 and CIFAR-100) were used. Using an evolutionary search, we ran eight experiments, two each of four different configurations of chromosomes and a random search as baseline. The search was conducted on CIFAR-10, and the best discovered architecture was transferred to CIFAR-100. The best network obtained 3.74% error rate on CIFAR-10 and 22.95% error rate on CIFAR-100, which is a competitive performance with the state-of-the-art networks.

4 Experiments on RS Image Understanding

4.1 Evaluating SLGE on RS Scene Classification

4.1.1 *Experiments*

Using SLGE-based architecture representation, we explored random search with early-stopping strategy as search technique to automatically evolve modularized CNNs architectures for classification of RS image scenes. Two types of multi-class image scene classification tasks were performed: single-label scene classification and multi-label scene classification, using four different remotely-sensed imagery datasets.

4.1.2 *Datasets*

The experiments were performed on two RGB aerial image benchmarks (NWPU-RESISC45 and AID) and two multispectral satellite image benchmarks (EuroSAT and BigEarthNet). NWPU-RESISC45, AID and EuroSAT are single-label tasks, and BigEarthNet is a multi-label task.

4.1.3 *Results*

The experimental results show the automatically discovered networks demonstrate the promising capability in classifying multispectral satellite image scenes compared with fine-tuned pre-trained CNN models. Using fewer parameters with 0.56B FLOPS, the best network achieves 96.56% and 96.10% accuracy rate on NWPU-RESISC45 single-label and AID single-label RGB aerial image datasets respectively, and 99.76% and 93.89% accuracy rate on EuroSAT single-label and BigEarthNet multi-label multispectral satellite image datasets respectively. The results position our approach amongst the best of the state-of-the-art, which shows that the search space is expressive and not intractable.

4.2 Evaluating SLGE on RS Image Segmentation

4.2.1 *Experiments*

The SLGE was extended to construct a two-separate search space representation: normal cell and atrous spatial pyramid pooling (ASPP) cell. Using evolutionary algorithm with genetic operators: uniform mutation, two-point crossover and gene crossover, we joint search for a normal cell and an ASPP cell as a pair of cells to build a modularized encoder-decoder CNN architecture called SLGENet for solving RS image semantic segmentation problem. Three ISPRS benchmarks were used to verify the performance of the proposed SLGENet on RS image segmentation tasks.

4.2.2 *Datasets*

The three ISPRS benchmarks used in the experiments are: Vaihingen, Potsdam and UAVid datasets. The Vaihingen

and Potsdam consist of high resolution aerial images over Vaihingen and Potsdam cities respectively, and the UAVid is a of high-resolution UAV images focusing on street scenes.

4.2.3 Results

On Vaihingen and Potsdam datasets, the proposed SLGENet achieved performance gains in the overall accuracy by 1.0% and 1.4% respectively, compared with the state-of-the-art models. And on UAVid dataset, SLGENet significantly improved the stat-of-the-art by 18.9% mean IoU. In addition, the SLGENet uses fewer parameters and reasonable computational resources of 2.5 GPU days. This demonstrates the effectiveness of the proposed SLGENet on three challenging ISPRS semantic segmentation benchmarks.

5 Conclusion

5.1 Contributions

The following are the scientific contributions of this dissertation:

1. We introduced a novel encoding scheme, SLGE, which extends GEP to AutoDNN by injecting the modularity features of CE into the linear representation of GEP to evolve modularized CNN architectures.
2. We demonstrated that SLGE can discover modules with shortcut and multi-branch connections commonly adopted by human experts and develop modularized CNN architectures of arbitrary complexity with fewer parameters.
3. We achieved results that are competitive to, or even exceed, human experts designed networks in various RS image understanding tasks. For each of the tasks used in the evaluation, the results of the best automatically discovered CNNs architecture contributed to the state-of-the-art.
4. By evolving and evaluating CNN architectures via random search policy with early-stopping and evolutionary algorithm on remotely-sensed imagery data, we have extended AutoDNN approach to the field of RS image understanding.

5.2 Future Work

Within the current framework, related visual perception tasks such as instance segmentation and object detection might be plausible. Moreover, expanding the current cell-based search space to include a global search space might be beneficial to develop completely automatic architecture search of CNNs. Another work is implementing real-world AutoDNN system with SLGE to show its practical and commercial viability to interpret remotely-sensed imagery or any other computer vision related problem.

5.3 Concluding Remarks

With the aforementioned contributions, we hope the work in this dissertation to have positive impact in future research in the field of NAS. The results presented were influenced by the world of remote sensing. Hopefully, this might also suggest something inspiring and opens new research path in such interdisciplinary approach, as well as practical applications in NAS methods in real-life situations, to enable the research scientists to use DNNs with little or without expertise in deep learning. This can improve the user experience and productivity in designing an AI-enabled product.

References

[1] E. Galván and P. Mooney, "Neuroevolution in deep neural networks: Current trends and future challenges," *IEEE TAI*, 2021.

[2] J. Fekiač, I. Zelinka, and J. C. Burguillo, "A review of methods for encoding neural network topologies in evolutionary computation," in *Proceedings 25th ECMS*, 2011, pp. 410–416.

[3] C. Ferreira, "Gene expression programming: a new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, p. 4278–4284.

[5] T. Liu, M. Chen, M. Zhou, S. S. Du, E. Zhou, and T. Zhao, "Towards understanding the importance of shortcut connections in residual networks," in *Advances in NeurIPS*, vol. 32, 2019.

[6] F. Gruau, "Neural network synthesis using cellular encoding and the genetic algorithm." Doctoral Dissertation, L'universite Claude Bernard-lyon I, 1994.

[7] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[8] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 24–35, 2019.

[9] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proceedings of 5th ICLR*, 2017.

[10] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, "A comprehensive survey of neural architecture search: Challenges and solutions," *ACM Comp. Surveys*, vol. 54, no. 4, pp. 1–34, 2021.

[11] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *IEEE Conference on CVPR*, 2018, pp. 8697–8710.

[12] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *Proceedings of the 35th ICML*, vol. 80, 2018, p. 4095–4104.

[13] M. Wistuba, A. Rawat, and T. Pedapati, "A survey on neural architecture search," *arXiv:1905.01392*, 2019.

[14] L. Li and A. Talwalkar, "Random search and reproducibility for neural architecture search," in *Proceedings of Conf. on UAI*, 2019.

[15] K. Yu, C. Sciuto, M. Jaggi, C. Musat, and M. Salzmann, "Evaluating the search phase of neural architecture search," in *Proceedings of ICLR*, 2020.

[16] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 3, pp. 247–259, 2011.

[17] P. M. Atkinson and A. R. L. Tatnall, "Introduction neural networks in remote sensing," *International Journal of Remote Sensing*, vol. 18, no. 4, pp. 699–709, 1997.

[18] S. S. Durbha, K. R. Kurte, and U. Bhangale, "Semantics and High Performance Computing Driven Approaches for Enhanced Exploitation of Earth Observation (EO) Data: State of the Art," *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, vol. 87, no. 4, pp. 519–539, 2017.

[19] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[20] J. Song, S. Gao, Y. Zhu, and C. Ma, "A survey of remote sensing image classification based on CNNs," *Big Earth Data*, vol. 3, no. 3, pp. 232–254, 2019.

[21] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS Journal of Photo. and RS*, vol. 152, pp. 166–177, 2019.

[22] C. Ferreira, "Designing neural networks using gene expression programming," in *Applied Soft Computing Technologies: The Challenge of Complexity*, 2006, pp. 517–535.

[23] X. Li, C. Zhou, W. Xiao, and P. C. Nelson, "Prefix gene expression programming," in *Proceedings of the GECCO*, 2005, p. 25–31.

[24] W. Banzhaf, F. D. Francone, R. E. Keller, and P. Nordin, *Genetic Programming: An Introduction*. San Francisco: Morgan Kaufmann, 1998.