Project Summary

Name: DANG ANH CHUONG
Supervisor: Prof. Okatani
Assist. Prof. Suganuma

Title: Exploring Attention Convolutional Neural Networks via Neural Architecture Search

I. Introduction

Attention mechanism has been playing an important role in achieving state-of-the-art results in many computer vision tasks such as image recognition and vision-language tasks. However, researchers have not paid attention to architecture design of an attention unit despite the fact that architecture affects performance of neural networks. In this paper, we propose a method to design good attention units for Convolutional Neural Networks (CNNs) and reveal the impacts of an attention unit architecture on networks performance. To do this, we employ a neural architecture search method which can automatically design neural architectures based on gradient descent. The experimental results show that the proposed method works better than existing methods by a good margin on image recognition tasks using CIFAR-10/100 datasets.

II. Methodology

1) *Backbone architecture*

In this work, we use Deep Residual Network (ResNet)[3], as a backbone architecture following Squeeze-and-Excitation Networks (SENet)[2].

2) *Attention Type*

Attention is an approach providing convolutional neural networks a way to focus on a certain region of an image. While other type of attention is generally based on single aspect such as: spatial or channel relationship, mixed attention allows network to attend on both spatial and channel regions. In this work, we used the mixed attention.

3) *Attention Unit architecture*

An attention block that we used in this work is based on a directed acyclic graph (DAG) comprising of ordered sequence of N nodes {x_0, x_1, …, x_{N-1}}. Each intermediate node is a latent representation, a feature map in convolutional network, and each directed edge is, during search stage, associated with mixed candidate operations, which will be then replaced by a single optimal operation in evaluating stage. The block has two input nodes, which are duplicates of a single output from previous layer within residual block, and a single output node. At the output node, mixed attention score for every unit of feature mapping is calculated by summing outputs of all intermediate nodes.

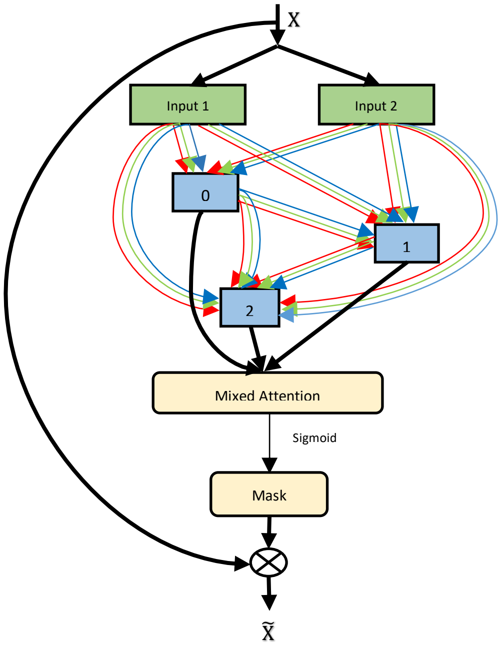

Figure 1: Attention unit architecture, multi-colored directed lines represent paths with candidate mathematical operations, black bold lines are identity paths (e.g. for summation)

4) *DARTS algorithm*

To search for the optimal architectural attention units, we employed DARTS[4], a differentiable gradient-based neural architecture search (NAS) algorithm, DARTS relaxes its categorical choice of operations into continuous search space using Softmax over every possible operations:

$$\bar{o}^{(i,j)}(x) = \sum_{o\in O} \frac{\exp\left(\alpha_o^{(i,j)}\right)}{\sum_{o'\epsilon O}\exp\left(\alpha_{o'}^{(i,j)}\right)} o(x)$$

Where, O is the set of candidate operations, $o(.)$ is an operation within set O and vector $\alpha^{(i,j)}$ of dimension $|O|$ parameterize the operation mixing weights for a pair of node(i,j).

After relaxation, the goal is to optimize a bi-level optimization problem, in another word, searching for an architecture α^* that minimizes the validation loss $\mathcal{L}_{val}(\omega^*,\alpha^*)$, where the weights ω^* associated with the architecture obtained by minimizing the training loss $\omega^* = \text{argmin}_{\omega}\mathcal{L}_{train}(\omega,\alpha^*)$:

$$\min_{\alpha} \ \mathcal{L}_{val}(\omega^*(\alpha),\ \alpha)$$

$$\text{s.t} \quad \omega^*(\alpha) = \text{argmin}_\omega \ \ \mathcal{L}_{train}(\omega, \alpha)$$

5) *Search process:*

We adopted all of DARTS candidate operations (e.g. separable convolution, max pooling, skip connection, *zero* …). Besides above operations, we added the following three new operations that are believed to be beneficial in creating attention masking.

- **Channel score:** to score how important a unit of feature representation is, according to channel relationship. Specifically, global pooling is used to squeeze input into a vector of channel representations, then following multilayer perceptron (MLP) to create channel's scores.
- **Spatial score:** to give feature mappings the score based on spatial relationship. Pooling along channel dimension is used to flatten tensor input into spatial representations, then convolution is used to create spatial score.
- **Bottom-up top-down:** to capture the importance of each spatial unit by dramatically increasing the convolutional receptive field. From input, max pooling is performed to increase the receptive field of the next convolutional layer. Down-sampled feature is linear interpolation up sample and summed with the input, followed by a 1x1 convolution.

III. Experiment and Results:

1) *Dataset*

We carried out architecture search on CIFAR-10 dataset and evaluation on CIFAR-10/100 dataset. In searching stage, we hold out half of the CIFAR-10 training data as the validation set. On the other hand, in evaluating stage, we used 50,000 images for training and 10,000 for evaluation.

2) *Architecture search process*

Our candidate operation set O includes: 3x3 separable convolution, 3x3 dilated separable convolution, 3x3 max pooling, 3x3 average pooling, identity, zero, 1x1 ReLU-Conv-BN, channel score, spatial score, and bottom-up top-down operations. Each attention block consists of three intermediate nodes (i.e. N=3). We carried out search based on ResNet-20 backbone. We trained our search model for 50 epochs.

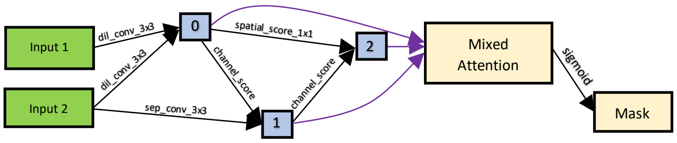

Figure 2: Attention unit obtained by search on CIFAR-10 with ResNet20 backbone. Purple line denotes identity paths (i.e. for summation)

3) *Architecture evaluation*

Table 1: Classification error of ResNet20 variation models on CIFAR-10 test set.

Model	Top-1 Error (%)
ResNet20 (from original paper)	8.75
SE-ResNet20 (replicated)	7.93
ApNAS_20 (ours)	**7.16**

Table 2: Classification error of ResNet56 variation models on CIFAR-10 test set.

Model	Top-1 Error (%)
ResNet56 (from original paper)	6.97
SE-ResNet56 (replicated)	6.29
ApNAS_56 (ours)	**5.55**

Table 3: Classification error of ResNet20 variation models on CIFAR-100 test set.

Model	Top-1 Error (%)
SE-ResNet20 (replicated)	32.68
ApNAS_20 (ours)	**28.39**

After architecture search completed, we retrain our model (ApNAS) from scratch for 180 epochs to evaluate them. We first evaluated ResNet20 variation models, then increased the network depth into ResNet56 variation and also evaluated them. Finally, we transfer our model to CIFAR-100 dataset. Results of experiments are listed in Table1, 2 and 3 respectively.

It is seen that our method works better than the baselines on all cases. This indicates that architecture search on attention units improves the performance of neural networks.

IV. Conclusion

In this work, we investigated the importance of attention mechanism in an image classification task. Moreover, we also proposed a method to search for a good architecture design of an attention unit for deep neural networks. Experimental results show that our model can outperform its counter-parts, which implies that there is still much room for exploration of search spaces of attention units. We believe that this work will facilitate more directions to further investigate NAS and attention mechanism.

V. References

[1] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. CVPR, 2017.

[2] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. CVPR, 2018.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. CVPR, 2016.

[4] H. Liu, K. Simonyan, and Y. Yang. DARTS: Differentiable architecture search. ICLR, 2019.

[5] S.Woo, J.Park, J.Lee, I.Kweon. CBAM: Convolutional Block Attention Module. ECCV, 2018.