Pruning Network Based Knowledge Distillation for Offline Handwritten Chinese Character Recognition

Zhuo Li[1], Yongping Dan[2], Zongnan Zhu, and Dinggen Zhang

[1] Zhongyuan University of Technology, Zhengzhou, Henan, China
[1]lizhuo970604@gmail.com
[2] 6100@zut.edu.cn

Abstract

Recently, deep convolutional neural networks have brought great breakthrough in image classification, which provide effective solution for the handwritten Chinese character recognition problem. Researchers have experimented with various networks to increase recognition accuracy. Although good accuracy is achieved on different networks, these networks tend to be computation-intensive and memory-intensive that make them difficult to be deployed on resource-constrained devices. To solve the problem, the paper proposes an optimization to reduce the number of model parameters by using pruning network and knowledge distillation. Besides, to improve the model's ability to extract the input features, an attention mechanism is adopted in the proposal. The experimental results show that the number of parameters decreased by nearly 26%. At the same time, the recognition accuracy improves by 1.17% with the value of 96.99% compared with the original model. The optimization method presented in this paper not only improves the accuracy of handwritten Chinese characters recognition but also reduces the number of model parameters.

1 Introduction

With the continuous development of Chinese culture, handwritten Chinese character recognition (HCCR) has attracted more and more attention and has been an important research topic. HCCR has been widely used in many fields, such as automatic bill recognition, handwritten Chinese character entry, cultural heritage preservation [1], automated teaching and office work. In recent years, convolutional neural networks (CNNs) have made great progress and breakthroughs in the field of computer vision. This is mainly due to the design of different network structures. For example, AlexNet [2], VGG [3], GoogLeNet [4] and ResNet [5], which have shown excellent performance in HCCR tasks. Although these neural networks have made great success in the field of HCCR, they have large requirements for computing resources, power consumption and storage space, which make them are difficult to be deployed on embedded devices such as ARM boards and FPGAs with limited hardware resources. For the reason that CNNs has a large number of redundant computations [6]. Therefore, it has been a major research topic to reduce the number of parameters while still ensuring accuracy for HCCR.

The remainder of this paper is organized as follows: Section 2 briefly reviews the related works. Section 3 introduces the mothod of attention mechanism, model pruning, and knowledge distillation. Section 4 gives the experimental results and experimental procedure in detal. Section 5 summarizes this paper and describes the future work.

2 Related work

Attention mechanisms are widely used in deep learning to enhance the performance of CNNs. [7] terms the "Squeeze-and-Excitation" (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channel. SE blocks bring significant improvements in performance for existing state-of-the-art CNNs at slight additional computational cost. [8] proposes convolutional block attention module (CBAM), CBAM sequentially infers attention maps along two separate dimensions, channel and spatial, then the attention maps are multiplied to the input feature map for adaptive feature refinement. [9] proposes an efficient channel attention (ECA) module, which only involves a handful of parameters while bringing clear performance gain. Avoiding dimensionality reduction is important for learning channel attention, and appropriate cross-channel interaction can preserve performance while significantly decreasing model complexity. [10, 11] uses weighting and Huffman coding to minimize storage space furthermore. [12] eliminates the unimportant channles by applying L1 regularization to the scale factor of the batch normalization (BN) layer. [13] uses least absolute shrinkage and selection operator (LASSO) regression to sparse the weights and cut out unimportant channels. Then, least squares method is used to ensure that the cropping operation has little impaction on the features by using LASSO. [14] takes adopts a new Taylor expansion-based criterion for approximating the loss function change caused by pruning network parameters. This is a modern formula for achieving effective reasoning in neural networks through pruning the convolution kernel. [15] introduces a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. [16] presents matching guided distillation (MGD) as an efficient and parameter-free manner to solve the problem of adding the adaptation module in classic methods.

3 Methods

3.1 Attention mechanism

Channel attention mechanism has demonstrated to offer great potential in improving the performance of CNNs, which can be used for classification. The attention mechanism in deep learning draws on human attentional thinking to focus on the key information in an image rather than the whole image [17]. As shown in Fig. 1, the lightweight attention mechanism ECA module is used for HCCR.

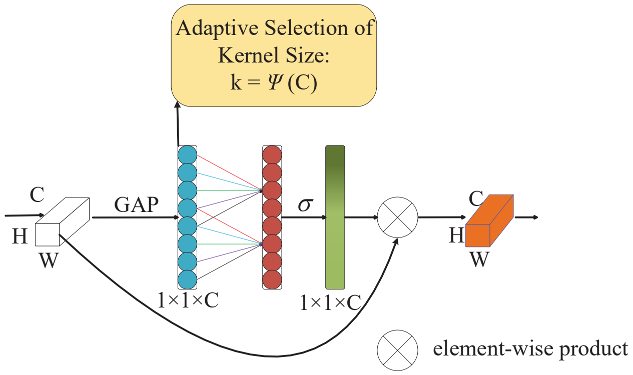

Figure 1: ECA module

Given an aggregated feature y ∈ $\mathbb{R}^C$ without dimensionality reduction, channel attention can be learned by Eq.(1).

$$\omega = \sigma(\mathrm{Wy}) \tag{1}$$

Where W is a $C \times C$ parameter matrix. In order to capture local cross-channel interaction, employing Eq.(2) to learn channel attention, aiming at guaranteeing both efficiency and effectiveness.

$$W_k = \begin{bmatrix} \omega^{1,1} & \cdots & \omega^{1,\mathrm{k}} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \omega^{2,2} & \cdots & \omega^{2,\mathrm{k}+1} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \omega^{C,C-k+1} & \cdots & \omega^{C,C} \end{bmatrix} \tag{2}$$

W_k involves $k \times C$ parameters. When all channels share the same learning parameters.

$$\omega_i = \sigma\left(\sum_{j=1}^{k} w^j y_i^j\right), y_i^j \in \Omega_i^k \tag{3}$$

Where Ω_i^k indicates the set of k adjacent channels of y_i. Eq.(3) can be readily implemented by a fast $1D$ convolution with kernel size of k.

$$\omega = \sigma(C1D_k(y)) \tag{4}$$

Where $C1D$ indicates $1D$ convolution. Here, the method in Eq.(4) is called by ECA module, which only consists of k parameters. The channel dimension C is proportional to the convolution kernel size k, as shown in Eq.(5), where k is taken as shown in Eq.(6).

$$C = \phi(k) = 2^{(\gamma * k - b)} \tag{5}$$

$$k = \psi(C) = \left|\frac{\log_2(C)}{\gamma} + \frac{b}{\gamma}\right|_{\text{odd}} \tag{6}$$

Where $|t|_{\text{odd}}$ indicates the nearest odd number of t. Setting γ and b to 2 and 1, respectively. Clearly, through the mapping ψ, highdimensional channels have longer range interaction while low-dimensional ones undergo shorter range interaction by using a non-linear mapping.

3.2 Model pruning

Shown in Fig. 2, channel pruning is a coarser-grained pruning, which is accomplished by deleting the redundant channels of feature map. A scale factor γ is added for each channel, which is then multiplied by the channel output. The network weights and these scale factors are jointly trained, and the latter is sparsely regularized. The redundant channels which are determined according to the scale factors, is pruned after training. The training objective feature is given as Eq.(7).

$$L = \sum_{(x,y)} l(f(x,w),y) + \lambda \sum_{\gamma \in \Gamma} g(\gamma) \tag{7}$$

Where $(x,\ y)$ denotes the training input and output, w denotes the trainable weight, the first term of Eq.(7) denotes the loss corresponding to regular convolutional network training, $g(\cdot)$ is a sparsity-induced penalty on the scaling factors, and λ is the balance factor of the first and second terms.

In the course of the experiment, choosing $g(s) = |s|$, which is known as L1-norm and widely used to achieve sparsity. BN has been adopted by most modern CNNs, as a standard method to achieve fast convergence and better generalization performance. Let z_{in} and z_{out} be the input and output of a BN layer, B denotes the current batchsize, BN layer performs the following transformation:

$$\hat{z} = \frac{z_{in} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, z_{out} = \gamma \hat{z} + \beta \tag{8}$$

Where μ_B and σ_B are the mean and standard deviation values of input activations over B, γ and β are trainsble affine transformation parameters which provides the possibility of linearly transforming normalized activations back to any scales.

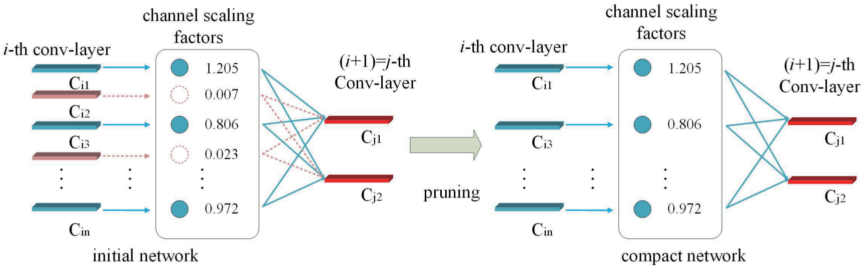

Figure 2: Channel pruning

3.3 Knowledge distillation

The goal of knowledge distillation is to use the large model's knowledge to direct the small model's training so that the small model can match the large model's output. The teacher model and the student model are described as the large and small models respectively. Fig. 3 depicts the structure. To obtain a better soft target, the temperature parameter T is quoted, as shown in Eq.(9).

$$\mathrm{q}_i = \frac{\exp(Z_i/T)}{\sum_j \exp(Z_j/T)} \tag{9}$$

Where Z_i is the probability of the i-th category in the output vector, $j \in (1, 2, ..., k)$, and k is the total number of categories. The exp is an exponential operation, and q_i is the soft target output obtained by the function. For the same input, when T is set to 1, the student network creat a hard target. Using a higher value for T produces a softer probability distribution over classes, and the teacher network and student network generate a soft target respectively. The hard target and the two soft targets are used as the input of the cross-entropy loss function to learn the weights. As a result, the objective function of the knowledge distillation can be summed up as Eq.(10).

$$L = \alpha L^{\text{soft}} + \beta L^{hard} \tag{10}$$

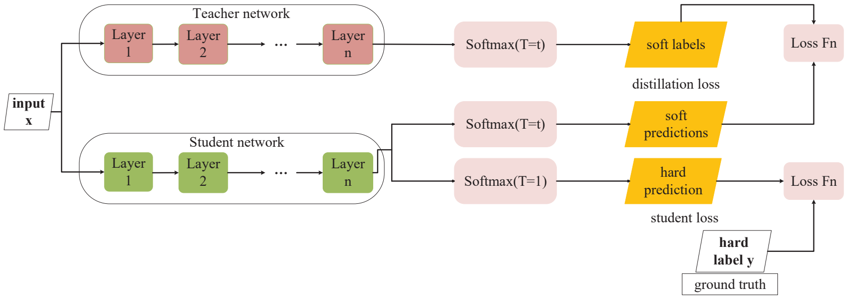

Figure 3: Knowledge distillation

4 Experiment

4.1 Experiment dataset

Shown in Fig. 4, the data on the left is from the CASIA-HWDB1.1 dataset, which is a publicly available HCCR dataset provided by the Institute of Automation of the Chinese Academy of Sciences. 16 classes are selected from the CASIA-HWDB1.1 dataset as part of the dataset in this paper. The right side is the same type of Chinese character written by different volunteers. The two parts are combined to form a new dataset, which is named MiniHWDB dataset. Shown in Table 1, MiniHWDB contains of 12,000 images. The dataset is split into two parts: the training set and test with the ratio of 8:2.

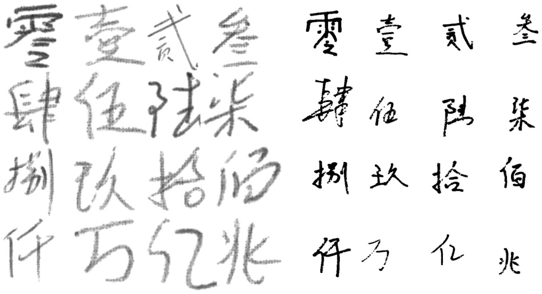

Figure 4: Offline handwritten Chinese character dataset

Dataset	Total images	Training ratio	Image size	classification
MiniHWDB	12000	0.8	256*256	16

Table 1: MiniHWDB dataset

4.2 Experimental process

In the process of training and inference, the input is resized to 224×224. The batch_size is set to be 64. The adaptive optimizer SGD is taken to optimize the loss function. All of the experiments are conducted on a computer with the 3.00 GHz Intel(R) Core(TM) i7-9700 processor, 2×8GB of RAM, and a GeForce RTX 2060 graphics card with 6GB of video memory.

At the first, ResNet18 is adopted as the original network. As shown in Fig. 4, different people have their own writing styles, and there is a lot of useless information (white area) in the input. The attention mechanism ECA is used to improve the feature extraction from the input. After that, the teacher network ECA-ResNet56 and the student network ECA-ResNet18 are obtained respectively. Next, the student network is pruned at the channel level according to the pruning ratios, which can be set 0.4 and 0.6. When the pruning rate is defined as 0.4, it means that 40% of the channels are pruned. The new student networks CS-ECA-ResNet18(0.4) and CS-ECA-ResNet18(0.6) are obtained after the pruning is completed. Finally, the teacher network ECA-ResNet56 is used to guide the pruned student network. The distilled student network is named KD-SC-ECA-ResNet18. In the course of the experiment, the parameters are set as shown in Table 2.

Description	Value
Adaptive selection of kernel size k	5
Temperature	5
Batch_size	64
Minimun number of epochs	30
Maximun number of epochs	100

Table 2: Parameter Setting

4.3 Results and analysis

The accuracy of the original network ResNet18 reached 94.40%. By introducing the attention mechanism ECA, the accuracy is improved by 1.42%, while the number of parameters only increases by 4.5%. Since ECA is a lightweight module, it can be seen that the parameters increase by introducing the attention mechanism is negligible. Channel pruning reduce the number of parameters by removing unimportant channels, but the result is loss of accuracy. To improve the loss of accuracy due to pruning, the method of teacher network is taken to guide the pruned network. Teacher networks usually to be deep networks. Although the increasing in depth of the network improves the accuracy, it also brings significant increase of parameters. For example, the teacher network is much deeper than the student network, but only 3.39% accuracy improvement. However, the number of parameters is 2.19 times than the student network. Although parameters and accuracy are difficult to balance in the task of HCCR, the parameters are given priority. Because these networks are mostly deployed on devices like mobile phones that do not have large storage.

With the increase in the number of parameters, the model is hard to be deployed on embedded devices. So channel pruning is adopted to reduce the number of parameters, this results in a loss of accuracy. Therefore, knowledge distillation is used to improve the accuracy of the pruned network. When the pruning rate is 0.4. The accuracy of the KD-SC-ECA-ResNet18(0.4) is improved 1.71%, and the number of parameters is reduced 16.7%, compared to the ECA-ResNet18. When the pruning rate is 0.6. The accuracy of the student network is improved 1.17%, and the number of parameters is reduced 25.6%, compared to before the pruning and knowledge distillation. The pruning rate is over 0.6, it is tough to obtain a good result even after knowledge distillation. The results of different models are shown in Table 3.

Model	Accuracy	Params
ResNet18	94.40	11.19M
ECA-ResNet18	95.82	11.69M
ECA-ResNet56	99.21	25.56M
KD-SC-ECA-ResNet18(0.4)	97.53	9.74M
KD-SC-ECA-ResNet18(0.6)	96.99	8.70M

Table 3: Experiment accuracy and model parameter number

5 Conclusion and future work

This paper focuses on images classification for offline handwritten Chinese character recognition. The method by using attention mechanism, channel pruning and knowledge distillation, not only obtains higher recognition accuracy, but also has a lower number of parameters than original network. In this paper, the attention mechanism is used to improve the network's ability to extract features, channel pruning effectively reduces the number of parameters, and the knowledge distillation improves the accuracy. It is beneficial for the model to be deployed on the resource-canstrained devices. In future work, the model can be compressed with other methods to further reduce model size. It is very useful for the development of artificial intelligence, especially for the field of computer vision.

References

[1] Lin Meng, Bing Lyu, Zhiyu Zhang, C. V. Aravinda, Naoto Kamitoku, and Katsuhiro Yamazaki. Oracle bone inscription detector based on ssd. *ICIAP2019*, pages 126–136, 2019.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pages 1097–1105, 2012.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1904–1916, 2015.

[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CVPR2015*, page 1–9, 2015.

[5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR2015*, 2015.

[6] Hengyi Li, Zhichen Wang, Xuebin Yue, Wenwen Wang, Tomiyama Hiroyuki, and Lin Meng. A comprehensive analysis of low-impact computations in deep learning workloads. *Proceedings of the 2021 on Great Lakes Symposium on VLSI*, 2021.

[7] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2011–2023, 2019.

[8] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. *Springer, Cham*, 2018.

[9] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. *CVPR2020*, 2020.

[10] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. *MIT Press*, 2015.

[11] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. *arXiv:1608.04493*, 2016.

[12] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. *ICCV2017*, 2017.

[13] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. *ICCV2017*, 2017.

[14] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv:1611.06440*, 2017.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *Computer Science*, pages 38–39, 2015.

[16] Kaiyu Yue, Jiangfan Deng, and Feng Zhou. Matching guided distillation. *arXiv:2008.09958*, 2020.

[17] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *Advances in Neural Information Processing Systems*, 2014.