# Model Pruning Technology for Deep Neural Networks

**Kohei Yamamoto  Motoko Tachibana**
**Kurato Maeno**

Applications of deep learning, a core AI technology, are rapidly expanding in recent years. Until now, the mainstream use was for the cloud and on-premises workstations equipped with a large-scale GPU, but around 2016, implementation in edge devices and dedicated chips began to appear. At present, application is spreading to various edge devices such as in-vehicle devices, smartphones and embedded IoT devices. However, in general, high precision deep learning models are difficult to implement in edge devices due to the large amount of memory required for operation and high power consumption.

In response, OKI is researching and developing a technology that compresses the model to significantly reduce computational resources while maintaining the original accuracy (e.g. for image or speech recognition). This article introduces the current status and problems of the model compression technologies and presents OKI's own unique technology.

## Model Compression Technology

Model compression technology is a general term for methods that reduce the number of parameters and operational complexity while maintaining the accuracy of the model. Recent deep learning requires large amounts of memory and computing power to execute, therefore increasing the need for model compression technology.

Deep learning model in a narrow sense refers to a multi-layered neural network with four or more layers and has a large number of parameters as coefficients for the interlayer coupling and bias. Normally, these parameters are expressed using 16- to 32-bit floating point numbers. There are two phases in deep learning, "training" and "inference." "Training" is a process of optimizing parameters using large amounts of data, and "inference" is a process of finding an answer to unknown data utilizing parameters optimized through training.

In an execution environment with limited processing power such as edge devices, it is common to implement only the inference function, which requires less computational resources than training. Even with such a measure, it is still difficult to operate high-precision models on edge devices. The reason is that as accuracy of the model becomes higher, the required number of parameters and operational complexity grows larger. The application of model compression technology alleviates those limitations making it possible to operate the inference function of high-precision models at high speeds on edge devices.

## Model Compression Technology Status and Problems

### (1) Types of Model Compression Technologies

Various approaches have been proposed for model compression, but they can be largely classified into six types. **Table 1** shows the types along with the comparison

**Table 1. Types of Model Compression Technologies**

| Type | Description | Reduction of Memory Usage | Reduction of Operational Complexity | Ease of Combined Use | Effect on Accuracy |
|---|---|---|---|---|---|
| Low-rank Approximation | Decomposition and approximation of weight matrix into low-rank matrix | Good | Fair | Good | Good |
| Quantization | Reduce bit precision of operation | Excellent | Fair | Good | Fair |
| Distillation | Train small-scale models using trained large-scale models | Fair | Fair | Good | Fair |
| Weight Sharing | Share weighting coefficients across multiple connections | Good | Fair | Fair | Good |
| High Efficient Architecture | Replace heavy convolutional operation with combination of multiple light-load convolutional operations | Good | Good | Good | Fair |
| Pruning | Remove low-importance neurons from the model after training | Good | Good | Excellent | Excellent |

of effect on memory usage, operational complexity (number of sum-product operations), ease of combined use, and accuracy. “Memory usage” and “operational complexity” indicate the degree to which reduction can be expected for each. “Ease of combined use” indicates the compatibility of the technology when used together with other compression technologies, and “effect on accuracy” indicates the degree of reducing the accuracy degradation that occurs when the compression technology is applied. The details of each method are described below.

- **Low rank approximation:** Taking advantage of the fact that most operations in deep learning can be expressed with large matrix operations, compression is achieved by mathematically decomposing and approximating a large matrix into a small matrix. This method is mainly suitable for reducing memory usage.
- **Quantization:** Compression is achieved by replacing parameters with fixed-points or integers of 8-bits or less, but precision is degraded due to rounding errors and narrowing of the numeric expression range. In particular, it is known that the accuracy is greatly degraded when less than 4-bits are used.
- **Distillation:** In this method, a large trained “teacher” model and a small untrained “student” model are prepared. Then the student model is trained so as to minimize the difference between its output and the teacher model’s output. However, since the option for student model remains arbitrary and it is difficult to make an optimal selection, this method tends to be inferior compared with other methods in terms of the viewpoints listed in **Table 1**.
- **Weight sharing:** In this method, models are trained by sharing the models’ weighting coefficients between connections of differing neurons. Memory usage can be reduced since one coefficient is shared and used multiple times. On the other hand, there is little reduction in operational complexity.
- **High efficient architecture:** In this architecture, the convolutional operation of the convolutional neural network (CNN), which is the most frequently used network architecture in deep learning, is replaced with a combination of multiple light-load convolutional operations. One example is the parallel combination architecture where the same data is input and independently subjected to convolutional operation and then the results are integrated. Another is the serial combination architecture where a multi-dimensional convolutional operation is substituted by multiple low-dimensional convolutional operations and combined in series. Although these architectures are efficient, they are not as accurate as the large models.
- **Pruning:** In this method, after a large model is trained, low-importance neurons are removed. The idea is similar to human brain cells, which establish cognitive ability, but the cells decrease over time. However, even if some cells die, it does not affect cognitive ability. The idea is technically and actively utilized in this approach. Since this method does not significantly change the model architecture, it is highly compatible with other compression technologies.

Among these methods, “pruning” provides an excellent balance between the ease of combined use and the degree of effect on accuracy. However, the degree of effect on accuracy only becomes advantageous when appropriate measures are taken against the problems described in the following section. Pruning can be roughly divided into two types: neuron-wise and channel-wise methods. In the neuron-wise method, pruning is performed based on the degree of importance of each neuron, which is the basic unit of a neural network. In the channel-wise method, pruning is performed based on the degree of importance of each filter, which is a group of weighting coefficients used for CNN, or each channel, which is a set of operational results thereof. With neuron-wise pruning, it is possible to finely remove low importance neurons scattered throughout the model, and it is easy to achieve a high pruning rate while maintaining accuracy. However, in CNN, since the filter has an architecture consisting of multiple neurons, it is necessary to maintain the architecture itself even if some of the neurons are removed. This leads to problems such as frequent memory access, which makes it difficult to increase operational efficiency at the implementation level. On the other hand, with channel-wise pruning, the pruning is on a filter-by-filter basis that generates channel data. This is a great advantage in terms of both memory usage and processing speed.

**(2) Problems with Channel-wise Pruning Method**

“Channel importance indicator” and “channel pruning rate assignment” were two problems that existed with conventional channel-wise pruning technologies.

In the first problem, the indicator that measures the importance of a channel is calculated independently for each layer. With such an indicator, a possibility remains that a channel determined to be unimportant in one layer may be necessary for another layer. If so, it can be

expected that the degree of accuracy degradation after model compression will increase due to the loss of an important channel contributing to accuracy. Referring back to conventional technologies, one indicator considered that larger the absolute sum of the values constituting each filter, more important the channel[1]. Another indicator considered a channel to be less important if its pruning during inference produces only a small change in the calculation result[2), 3)]. However, those indicators are calculated using an independent method for each layer. Therefore, although good comparison can be made within a given layer, when interlayer relation is taken into account, the optimum channel is not necessarily selected and selection tends to be inefficient. Thus, an indicator that considers the relationship between all layers is desired.

The second problem is that channel pruning rate must be assigned separately for each layer. The pruning rate assigned to each layer is left up to the user, but if not properly assigned, the accuracy will be greatly lost. The reason is that each of the multiple convolutional layers constituting the CNN has different sensitivity to pruning[1]. Sensitivity is the degree of effect the channel pruning rate has on accuracy. For example, assigning a high pruning rate may have a small effect on the accuracy for one layer, but assigning an equivalent pruning rate for another layer may lead to significant deterioration in accuracy. Therefore, the user must select an appropriate pruning rate for each layer while taking sensitivity into consideration. However, that sensitivity analysis requires trial-and-error and expertise, and making an optimal choice is difficult. Furthermore, in the case of a large-scale model with numerous layers, the number of required pruning rate assignments increases which then increases the degree of difficulty dramatically. That is, it is desirable to eliminate the process of assigning channel pruning rate for each layer and instead assign one channel pruning rate for the entire model whereupon an optimal pruning rate between layers can be allocated.

## PCAS Technology

OKI possesses a unique PCAS (Pruning Channels with Attention Statistics) technology that addresses the two problems described in the previous section and optimally performs channel-wise pruning for CNN models. This technology is characterized by its ability to compress models in terms of both memory usage and operational complexity while maintaining high accuracy.

**(1) Technology Overview**

An overview of the PCAS technology is shown in **Figure 1**. A new neural network model (referred to as an attention module) is inserted between the layers of the CNN model to be compressed, and training is performed only for that module. There is a 1:1 corresponds between the number of neurons from the output layer of the module and the number of output channels from the layer of original network prior to the inserted module. The value of each neuron functions as a gate to control the propagation amount of the corresponding channel value to the next layer. The training in this configuration is the optimization of the parameters constituting the module in order to output higher values for neurons corresponding to channels that contribute to accuracy. Thus, each neuron in the module output layer will indicate the importance of the corresponding channel. Once training is completed, the module can execute inference to output the optimum importance of each data, and the average value can be used as the importance of the channel.

Unlike the indicators in conventional technologies[1), 2), 3)] relationship between layers can be considered with the channel importance indicator of the PCAS technology. This solves the first problem mentioned in the previous section. Since training is performed with each module sandwiching the original convolutional layer and everything connected together, optimization of the channel importance proceeds as gate weight while mutually affecting each other. Thus, the importance of the channels in each layer is a value optimized across all the layers. In this case, a channel determined to be unimportant in one layer is likely to be unimportant in another, meaning that each importance has a property of being less susceptible to each other. That is, it is easier to remove unimportant channels from the model as whole, and as a result, accuracy degradation is reduced.

The channel pruning rate that the PCAS technology needs for model compression is one for the entire model, not for each layer. This solves the second problem mentioned previously. Specifically, using the fact that different layers can be evaluated on the same basis, the channel importance indicator of the PCAS technology removes low important channels based on the importance of all channels in all layers until the pruning rate of the entire model is achieved, thus making it possible to remove different number of channels from each layer. Afterwards, re-training (fine-tuning) is performed with reduced model network to complete compression. Since the inserted module is removed after estimating the channel

importance, the increase in the operational complexity due to this process will not affect inference.

As can be seen from above, PCAS technology makes it unnecessary to assign channel pruning rate for each layer, the distribution of the channel pruning rate is optimal since there is no human intervention, and significant reduction in memory usage and operational complexity can be expected while maintaining accuracy.

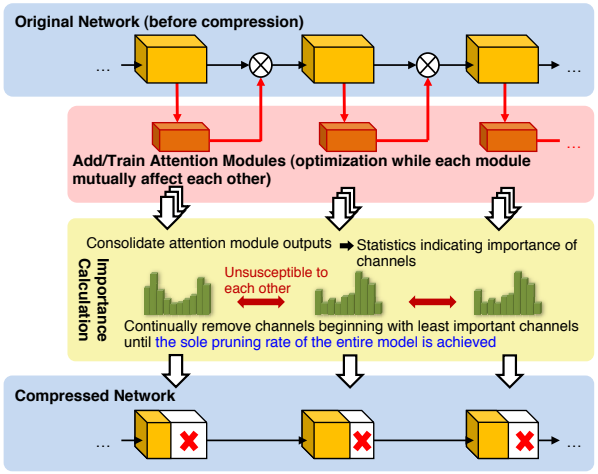


**Figure 1. Conceptual Diagram of PCAS Application**

### (2) Evaluation Results

The effectiveness of the PCAS technology was confirmed using a 50-layer model and dataset generally used in deep learning benchmarks. The results are shown in **Figure 2**. The left axis is the number of parameters and operations expressed as ratios with 100% representing the state before channel-wise pruning is applied. The right axis represents accuracy.

As for the results of PCAS technology, the number of parameters and operations were reduced to less than half without any deterioration in accuracy from before the compression. Furthermore, even under the same conditions as the benchmarks of conventional technologies[2), 3), 4)] presented recently at leading international conferences, parameter and operation reduction rates were improved 13 points and 12 points, respectively, confirming that an efficient model can be realized in terms of both memory usage and operational complexity.

These results were obtained based on a single channel pruning rate for the entire model. Therefore, it also shows that excellent results can be obtained despite the fact that neither pruning rate determination nor sensitivity analysis was performed for the individual layers.

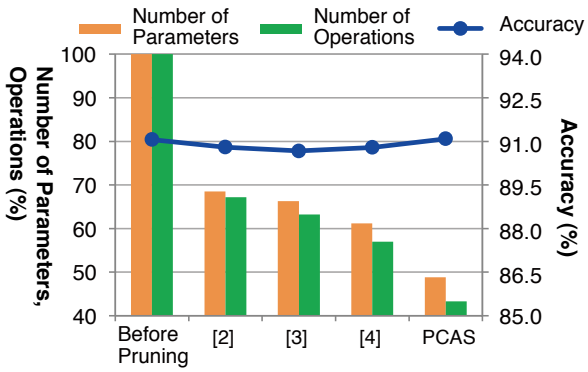


**Figure 2. Evaluation Results**

## Future Prospects

This article introduced PCAS, OKI's own model compression technology. At present, work is proceeding for combined use with quantization in order to further enhance the model compression effect and improve affinity for hardware implementation. This resource-saving, high-precision deep learning model is expected to greatly accelerate the spread of AI implementation in the edge domain, and development is being advanced to apply this technology to OKI's various AI edge solutions.

## Acknowledgement

Part of the results was obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). ◆◆

## References

1) Hao Li, Asim Kadav, Igor Durdanovic and Hanan Samet, Hans Peter Graf: Pruning Filters for Efficient ConvNets, International Conference on Learning Representations (ICLR), 2017.
2) Yihui He, Xiangyu Zhang and Jian Sun: Channel Pruning for Accelerating Very Deep Neural Networks, International Conference on Computer Vision (ICCV), 2017.
3) Jian-Hao Luo, Jianxin Wu and Weiyao Lin: ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression, International Conference on Computer Vision (ICCV), 2017.
4) Z. Huang and N. Wang : Data-Driven Sparse Structure Selection for Deep Neural Networks, European Conference on Computer Vision (ECCV), 2018.

## Authors

**Kohei Yamamoto**, Innovation Promotion Department, Corporate Research & Development Center, Corporate Infrastructure Group

**Motoko Tachibana**, Innovation Promotion Department, Corporate Research & Development Center, Corporate Infrastructure Group

**Kurato Maeno**, Innovation Promotion Department, Corporate Research & Development Center, Corporate Infrastructure Group


## TIPS【Glossary】

**Neuron**
Basic unit that make up a neural network. It has a large number of inputs and has a structure that performs operations such as activation functions on the linear combinations of those inputs and weights then outputs the results.