Algorithms for Accelerating CNN

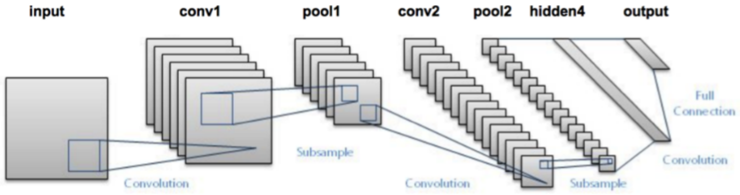
Sihan Chen

The Chinese University of Hong Kong

28th June, 2018

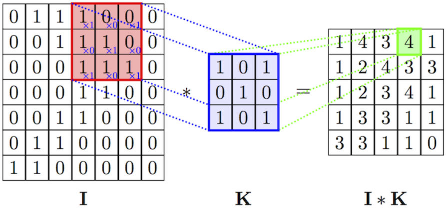
Convolutional Neural Networks

A type of neural network, mostly used for computer vision.



Convolutional Neural Networks

- Convolutional layer



- Pooling layer
- Fully connected layer

Winograd Algorithm

For simplicity, let's discuss the 2D case with the data block $d \in \mathbb{M}_{m \times m}$ and filter $g \in \mathbb{M}_{r \times r}$.

- Brutal Force:
  Needs $r \times r \times m \times m$ (here maybe a typo in the original paper) times of multiplications.
- Winograd:
  $S = A^T[[GgG^T] \odot [B^T dB]]A$
  Here $\odot$ represents element-wise product, A, G, B are constant matrices determined only by the data size.
  Needs only $(m + r - 1) \times (m + r - 1)$ times of multiplications[1].
  Really?

[1] Lavin, A; Gray, S. Fast Algorithms for Convolutional Neural Networks. *arXiv:1509.09308*, 2015

Winograd Algorithm

Write $M = [GgG^T] \odot [B^T dB]$, therefore $S = A^T MA$.

- The element-wise multiplication does involve only $(m + r - 1) \times (m + r - 1)$ times of multiplications.
- What about other matrix-matrix multiplications involved, i.e. GgG^T, $B^T dB$ and $A^T MA$? Or, is it possible to compute them in another way?

Pruning

- Motivation:
  The number of parameters in CNN is extremely large, some of them may be redundant.
  The computation of training those redundant parameters can be very expensive.
  Caring about every minor parameter may also result in over-fitting.
- Solution:
  Set parameters with small value as 0 to reduce computation without loss of accuracy[2].

[2] Li, S; Park, J; Tang, P. Enabling Sparse Winograd Convolution by Native Pruning. *arXiv:1702.08597*, 2017

Sparse Winograd Algorithm

Winograd Algorithm can also be applied to pruned CNN to accelerate the computation.

- Standard way:
  $S = A^T[[G\text{Prune(g)}G^T] \odot [B^T\text{ReLU}(d)B]]A$
  The sparse matrices (Prune(g) and ReLU(d)) become dense again when transformed from spatial domain to Winograd domain.
- Winograd-ReLU CNN[3]:
  $S = A^T[[\text{Prune}(GgG^T)] \odot [\text{ReLU}(B^T dB)]]A$
  Move the pruning and ReLU operations into Winograd domain in order to make the results sparse, without loss of accuracy.

[3] Liu, X; Pool, J; Han, S; Dally, W. Efficient Sparse-Winograd Convolutional Neural Networks. *arXiv:1802.06367*, 2018

Sparse Winograd Algorithm

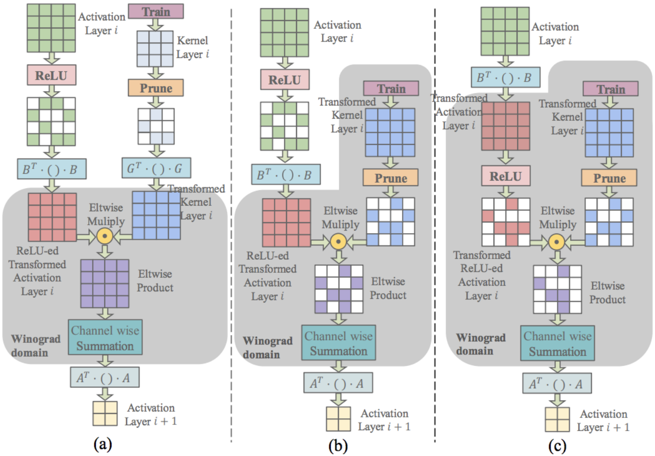


Figure 1: Combining Winograd convolution with sparse weights and activations. (a) Conventional Winograd-based convolution fills in the zeros in both the weights and activations. (b) Pruning the 4×4 transformed kernel restores sparsity to the weights. (c) Our proposed Winograd-ReLU CNN. Moving the ReLU layer after Winograd transformation also restores sparsity to the activations.

Future Works

- Do experiments with different pruning methods:
  - Resetting parameters with small value as 0 (the method mentioned above).
  - Removing filters with small Frobenius norm.
- Do some modification to Winograd Algorithm:
  - Verify the speed of computing convolution with Winograd Algorithm.
  - Implement Sparse Winograd Algorithm with MAGMA.
  - Reconsider Winograd Algorithm from the perspective of 3D convolution.