1. Introduction

Advantages of ThiNet

- We focus on filter level pruning, which do not change structure;
- We achieve acceleration and compression simultaneously;
- Pruned model can be further compressed via other methods;

Main idea

- We formally establish filter pruning as an optimization problem.
- Selection criterion: minimize the reconstruction error.
- Selection + Reconstruction

2. ThiNet Framework

Filter selection: use the information of next layer to guide the pruning of current layer.

Pruning: Remove all weak filters (channels) of current (next) layer.

Fine-tuning: Fine-tune one or two epochs to recover model accuracy.

1. Introduction

2. ThiNet Framework

Given \( m \) training examples \( \{(\mathbf{x}_i, y_i)\} \), the original channel selection problem becomes the following optimization problem:

\[
\min_{T} \sum_{i} \left( y_i - \sum_{j} w_{ij} x_{ij} \right)^2 \quad \text{s.t. } |T| = C \times (1 - r), T \subset \{1, 2, ..., C\}
\]

Final Step: After selection, we further minimize the reconstruction error:

\[
\hat{w} = \arg \min_{w} \sum_{i} \left( y_i - \sum_{j} w_{ij} x_{ij} \right)^2
\]

which can be solved using least square. Finally, we use \( \hat{w} \) to rescale the selected filter weight:

\[
w_{jk} = \hat{w}_{jk} \hat{w}_{j}
\]

3. Selection + Reconstruction

3. Selection + Reconstruction

3. Selection + Reconstruction

4. Experiments

4.1 Compared on CUB200

4.2 Prune VGG16 on ImageNet

4.3 Transfer Learning

5. Conclusion

About Me

Project Page