Feature Fusion for Online Mutual Knowledge Distillation

Jangho Kim, Minsung Hyun, Inseop Chung, Nojun Kwak

{kjh91, minsung.hyun, jis3613, nojunk}@snu.ac.kr

Seoul National University
Machine Intelligence Pattern Analysis Lab (MIPAL)
http://mipal.snu.ac.kr
Introduction

• Many researches on network architecture that extracts discriminative features.
  • ResNet
  • Wide-ResNet
• New approach: the feature fusion method that can combine different feature maps gained from multiple sub-networks.
• Feature fusion methods have been used in many previous deep learning studies.
Introduction

• DualNet which is feature fusion method trains independent two sub-nets with iterative training.
• This framework combine complementary two feature maps with fused classifier.

Overall process of DualNet
Introduction

• DualNet coordinated two parallel sub-networks and fused the two-stream features, directly.
• Directly combining feature maps incurs several challenges:
  • The performance of the sub-networks is significantly lower than the performance of the network that is independently trained with the same architecture. This can also influence the performance of fused classifier.
  • Because it directly combines feature maps, it is applicable only when the sub-networks have the same architecture.
Motivation & Contribution

• Motivation
  • Sub-networks can not help fused classifier with positive synergy
  • Only same architecture type can be used

• Contribution
  • Our method, Feature fusion learning (FFL) can improve the accuracy of sub-networks where gives positive synergy to a fused classifier
  • FFL can handle various architecture type
  • FFL can create meaningful feature maps used at computer vision tasks
Method

• **Fusion module**
  
  • Combining feature maps from the last layer of each sub-network with convolution operation
  
  • To reduce computational cost, FFL use Depth-wise and Point-wise convolution
  
  • Combined feature maps is named as fused feature
Method

- **Ensemble knowledge distillation (EKD)**
  - Using ensemble logits of sub-networks and knowledge distillation, fusion module can generate meaning feature map with this loss

- **Fusion knowledge distillation (FKD)**
  - Using fused logits and knowledge distillation, sub-networks can be learned with this loss

\[
\mathcal{L}_{kl}^e = \sum_{i=1}^{m} \sigma_i(z_e; T) \log \left( \frac{\sigma_i(z_e; T)}{\sigma_i(z_f; T)} \right)
\]

\[
\mathcal{L}_{kl}^f = \sum_{k=1}^{n} \sum_{i=1}^{m} \sigma_i(z_f; T) \log \left( \frac{\sigma_i(z_f; T)}{\sigma_i(z_k; T)} \right)
\]

\[
\mathcal{L}_{total} = \sum_{k=1}^{n} \mathcal{L}_{ce}^k + \mathcal{L}_{ce}^f + T^2 \times (\mathcal{L}_{kl}^e + \mathcal{L}_{kl}^f)
\]
Experiments

- Comparison with Feature Fusion Method

(a) Top-1 classification error rate of fused classifiers. DualNet outputs results from the average of classifiers and FFL uses fusion module for classification.

<table>
<thead>
<tr>
<th>(%)</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DualNet</td>
<td>FFL</td>
</tr>
<tr>
<td>ResNet-32</td>
<td>6.21±0.20</td>
<td>5.78±0.13</td>
</tr>
<tr>
<td>ResNet-56</td>
<td>5.67±0.12</td>
<td>5.26±0.17</td>
</tr>
<tr>
<td>WRN-16-2</td>
<td>5.92±0.16</td>
<td>5.97±0.13</td>
</tr>
<tr>
<td>WRN-40-2</td>
<td>4.94±0.10</td>
<td>4.6±0.13</td>
</tr>
</tbody>
</table>

(b) Top-1 classification error rate of sub-network classifiers.

<table>
<thead>
<tr>
<th>(%)</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DualNet</td>
<td>FFL</td>
</tr>
<tr>
<td>ResNet-32</td>
<td>8.23±0.31</td>
<td>6.06±0.15</td>
</tr>
<tr>
<td>ResNet-56</td>
<td>7.34±0.25</td>
<td>5.58±0.13</td>
</tr>
<tr>
<td>WRN-16-2</td>
<td>7.53±0.20</td>
<td>6.09±0.09</td>
</tr>
<tr>
<td>WRN-40-2</td>
<td>6.25±0.14</td>
<td>4.75±0.16</td>
</tr>
</tbody>
</table>

(Ablation study)
Experiments

• Comparison with Knowledge Distillation

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>vanilla</td>
<td>26.69</td>
<td>8.58</td>
</tr>
<tr>
<td>ONE</td>
<td>25.61±0.02</td>
<td>7.96±0.02</td>
</tr>
<tr>
<td>FFL-S</td>
<td>25.58±0.06</td>
<td>7.95±0.06</td>
</tr>
<tr>
<td>ONE-E</td>
<td>24.48</td>
<td>7.31</td>
</tr>
<tr>
<td>FFL</td>
<td>23.91</td>
<td>7.17</td>
</tr>
</tbody>
</table>

(DML; different arch)

(ONE; same arch)
Thank you

- The code is available at the “https://github.com/Jangho-Kim/FFL-pytorch”