Adaptive Subspaces for Few-Shot Learning

Christian Simon, Piotr Koniusz, Richard Nock, Mehrtash Harandi
CVPR 2020
Motivation

- Formulate FSL as generating dynamic classifier
  - Learning a universal feature extractor
  - Learning to generate a classifier dynamically from limited data.
- Method
  - Generate subspace from limited data
  - Singularity Vector Decomposition
FSL review

\[
p(c|q) = \frac{\exp(W_c^\top f_\theta(q))}{\sum_{c'} \exp(W_{c'}^\top f_\theta(q))} = \frac{\exp(d_c(q))}{\sum_{c'} \exp(d_{c'}(q))},
\]

Figure 2: Various classifiers for few-shot classification. (a) Matching networks create pairwise classifiers. (b) Prototypical networks create mean classifiers based on the sample in the same class. (c) Relation networks produce non-linear classifiers. (d) Our proposed method creates classifiers using subspaces.
Subspace for few-shot

A subspace $Z$ is represented by a basis. This paper try to generate the basis

$$\mathbb{R}^{D \times n} \ni \mathbf{B}_i = [b_1, \cdots, b_n]; n \leq D$$

A basis for a class can be compute by matrix decomposition (e.g. SVD)

$$\mathbf{B}_i^\top \mathbf{B}_i = \mathbf{I}_n$$
Generate subspace

Sample representation

\[ \tilde{X}_c = [f_\Theta(x_{c,1} - \mu_c, \ldots, f_\Theta(x_{c,K} - \mu_c] \]

where \[ \mu_c = \frac{1}{K} \sum_{x_i \in X_c} f_\Theta(x_i). \]

Class representation as a subspace \( P_c \) from \( X_c \) by SVD
Subspace classifier

Given a query instance \( q \), distance to subspace is

\[
d_c(q) = -\| (I - M_c)(f(q) - \mu_c) \|^2
\]

Distance distribution

\[
p_{c,q} = p(c|q) = \frac{\exp(d_c(q))}{\sum_{c'} \exp(d_{c'}(q))}
\]
Discriminative Deep Subspace Network

Maximize subspace distance using Grassmannian geometry

\[
\delta_p^2 (P_i, P_j) = \left\| P_i P_i^\top - P_j P_j^\top \right\|_F^2 = 2n - 2\| P_i^\top P_j \|_F^2.
\]

Final loss function

\[
-\frac{1}{NM} \sum_c \log(p_{c,q}) + \lambda \sum_{i \neq j} \| P_i^\top P_j \|_F^2.
\]
Mean refinement

Make use of unsupervised data

\[ \tilde{\mu}_c = \frac{K \mu_c + \sum_i m_i f_\Theta(r_i)}{K + \sum_i m_i} \]

Where

\[ m_i = \frac{\exp(-\|f_\Theta(r_i) - \mu_c\|^2)}{\sum_{c'} \exp(-\|f_\Theta(r_i) - \mu_{c'}\|^2)} \]
Algorithm 1 Train Deep Subspace Networks

Input: Each episode $\mathcal{T}_i$ with $S$ and $Q$

1: $\Theta_0 \leftarrow$ random initialization
2: for $t$ in $\{\mathcal{T}_1, \ldots, \mathcal{T}_{N_T}\}$ do
3:   for $k$ in $\{1, \ldots, N\}$ do
4:     $\tilde{X}_c \leftarrow S_c$
5:     Calculate the average of the class
6:     Calculate mean refinement (MR) using Eq. 6
7:     Subtract $\tilde{X}_c$ with an offset
8:     $[\mathcal{U}, \Sigma, \mathcal{V}^\top] \leftarrow$ Decompose($\tilde{X}_c$)
9:     $P_c \leftarrow$ Truncate $\mathcal{U}_1, \ldots, \mathcal{n}$
10:    for $q$ in $Q$ do
11:       Compute $d_c(q)$ using Eq. 2
12:    end for
13: end for
14: Compute final loss $\mathcal{L}_t$ using Eq. 5
15: Update $\Theta$ using $\nabla \mathcal{L}_t$
16: end for
## Result

<table>
<thead>
<tr>
<th>Model</th>
<th>Backbone</th>
<th>1-shot</th>
<th>5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching Nets [4]</td>
<td>Conv-4</td>
<td>43.56 ± 0.84</td>
<td>55.31 ± 0.73</td>
</tr>
<tr>
<td>MAML [7]</td>
<td>Conv-4</td>
<td>48.70 ± 1.84</td>
<td>63.11 ± 0.92</td>
</tr>
<tr>
<td>Reptile [48]</td>
<td>Conv-4</td>
<td>49.97 ± 0.32</td>
<td>65.99 ± 0.58</td>
</tr>
<tr>
<td>R2-D2 [49]</td>
<td>Conv-4</td>
<td>48.70 ± 0.60</td>
<td>65.50 ± 0.60</td>
</tr>
<tr>
<td>Prototypical Nets [20]</td>
<td>Conv-4</td>
<td>44.53 ± 0.76</td>
<td>65.77 ± 0.66</td>
</tr>
<tr>
<td>Relation Nets [14]</td>
<td>Conv-4</td>
<td>50.44 ± 0.82</td>
<td>65.32 ± 0.70</td>
</tr>
<tr>
<td>DSN</td>
<td>Conv-4</td>
<td><strong>51.78 ± 0.96</strong></td>
<td><strong>68.99 ± 0.69</strong></td>
</tr>
<tr>
<td>DSN-MR</td>
<td>Conv-4</td>
<td><strong>55.88 ± 0.90</strong></td>
<td><strong>70.50 ± 0.68</strong></td>
</tr>
</tbody>
</table>