Deep Subspace Networks for Few-Shot Learning

Christian Simon†§
Richard Nock†‡§
Piotr Koniusz†§
Mehrtash Harandi #§
Problem Definition

- Given: A Support Set $S = \{(x_{1,1}, c_{1,1}), (x_{1,2}, c_{1,2}), \cdots, (x_{N,K}, c_{N,K})\}; x \in \mathbb{R}^D$
  
  A query $q \in \mathbb{R}^D$

- A Support set contains $N$-way (classes) and $K$-shot (samples).

- The classes are unseen, can we classify them?
An approach for classification is to use a fully connected layer as a classifier following with a softmax function.

Let a function $f_\Theta : \mathbb{R}^D \to \mathbb{R}^n$, extracting a feature from an input.

Then, we can formulate the classifier and the softmax function as:

$$p(c|q) = \frac{\exp(w_c^T f_\Theta(q))}{\sum_{c'} \exp(w_{c'}^T f_\Theta(q))} = \frac{\exp(s_c(q))}{\sum_{c'} \exp(s_{c'}(q))}$$
• The classifier needs to be updated (e.g. iterative gradient descents) using new samples if there are samples from unseen classes.

\[ p(c|q) = \frac{\exp(w_c^T f_\Theta(q))}{\sum_{c'} \exp(w_{c'}^T f_\Theta(q))} \]

Should be updated
Some prior approaches use pair-wise [1], prototype [2,4], and binary classifiers [3].

We define a function $S_C$ for these classifiers.

For example:

$$p(c|q) = \frac{\exp(w_c^T f_\Theta(q))}{\sum_{c'} \exp(w_{c'}^T f_\Theta(q))} = \frac{\exp(s_c(q))}{\sum_{c'} \exp(s_{c'}(q))}$$

---

Proposed Method

• Using subspace methods as classifiers.
• Projecting each datapoint within the same class to a subspace.

Our formulation:

\[ s_c(q) = -\|f_\Theta(q) - \pi_c(q)\|_2^2, \]
\[ \pi_c(q) = W_c W_c^T f_\Theta(q) - b_c. \]

Where:

- \( W_c \) is an orthogonal basis for linear subspace spanning \( X_c = \{f_\Theta(x_i); y_i = c\} \)
- \( b_c = \mathbb{E}_{x \sim p_c}(f_\Theta(x)) \)
**Proposed Method**

**Algorithm 1** Train Deep Subspace Networks

**Input:** $S = \{(x_{1,1}, c_{1,1}), \cdots, (x_{N,K}, c_{N,K})\}$ and $q$

1: $\Theta_0 \leftarrow$ random initialization
2: for $t$ in $\{T_1, \ldots, T_T\}$ do
3:     for $c$ in $\{1, \ldots, N\}$ do
4:         Get $\tilde{X}$ examples in the support set for class $c$
5:         $[U, \Sigma, V^T] \leftarrow$ SVD($\tilde{X}_c$)
6:         $W_c \leftarrow u_{1, \ldots, n}$
7:         Project $q$ using $W_c$
8:     end for
9: end for
10: Update $\Theta$ from loss
11: end for

**Algorithm 2** Train Prototypical Networks

**Input:** $S = \{(x_{1,1}, c_{1,1}), \cdots, (x_{N,K}, c_{N,K})\}$ and $q$

1: $\Theta_0 \leftarrow$ random initialization
2: for $t$ in $\{T_1, \ldots, T_T\}$ do
3:     for $c$ in $\{1, \ldots, N\}$ do
4:         Get $\tilde{X}$ examples in the support set for class $c$
5:         $\mu_c \leftarrow \frac{1}{K} \sum_{x \in \tilde{X}} f_\Theta(x)$
6:         Calculate the similarity between $q$ and $\mu_c$
7:         Calculate log probability
8:     end for
9: end for
10: Update $\Theta$ from loss
11: end for

**Subspace Method VS Prototype Method**
Experiments

• Few-Shot Classification
  • Deep Subspace Network (DSN) compares to the state-of-the-arts

<table>
<thead>
<tr>
<th>Model</th>
<th>Backbone</th>
<th>1-shot</th>
<th>5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching Networks [1]</td>
<td>ResNet-18</td>
<td>52.91 ± 0.88</td>
<td>68.88 ± 0.69</td>
</tr>
<tr>
<td>Prototypical Networks [2]</td>
<td>ResNet-18</td>
<td>54.16 ± 0.82</td>
<td>73.68 ± 0.65</td>
</tr>
<tr>
<td>Relation Networks [3]</td>
<td>ResNet-18</td>
<td>52.48 ± 0.86</td>
<td>69.83 ± 0.68</td>
</tr>
<tr>
<td>CTM [4] (fine-tune)</td>
<td>ResNet-18</td>
<td>62.05 ± 0.55</td>
<td>78.63 ± 0.06</td>
</tr>
<tr>
<td>DSN</td>
<td>ResNet-18</td>
<td><strong>56.32 ± 0.79</strong></td>
<td><strong>75.49 ± 0.62</strong></td>
</tr>
<tr>
<td>DSN (fine-tune)</td>
<td>ResNet-18</td>
<td><strong>62.58 ± 0.80</strong></td>
<td><strong>79.62 ± 0.71</strong></td>
</tr>
</tbody>
</table>

Accuracy 5-way 1-shot and 5-way 5-shot with 95% confidence intervals on the mini-ImageNet
Experiments

• Robustness
  • There are two types of evaluation:
    • Samples come from other classes in the support set
    • Noise is appended to the input image

Accuracy on the *mini*-ImageNet
Conclusion

• Subspace method is more expressive as a classifier to capture the information from a few samples compared to prior works e.g. averaging the features.

• Subspace is also more robust compared to the prototype solution because of the denoising capability of subspaces.