Deep Subspace Networks for Few-Shot Learning

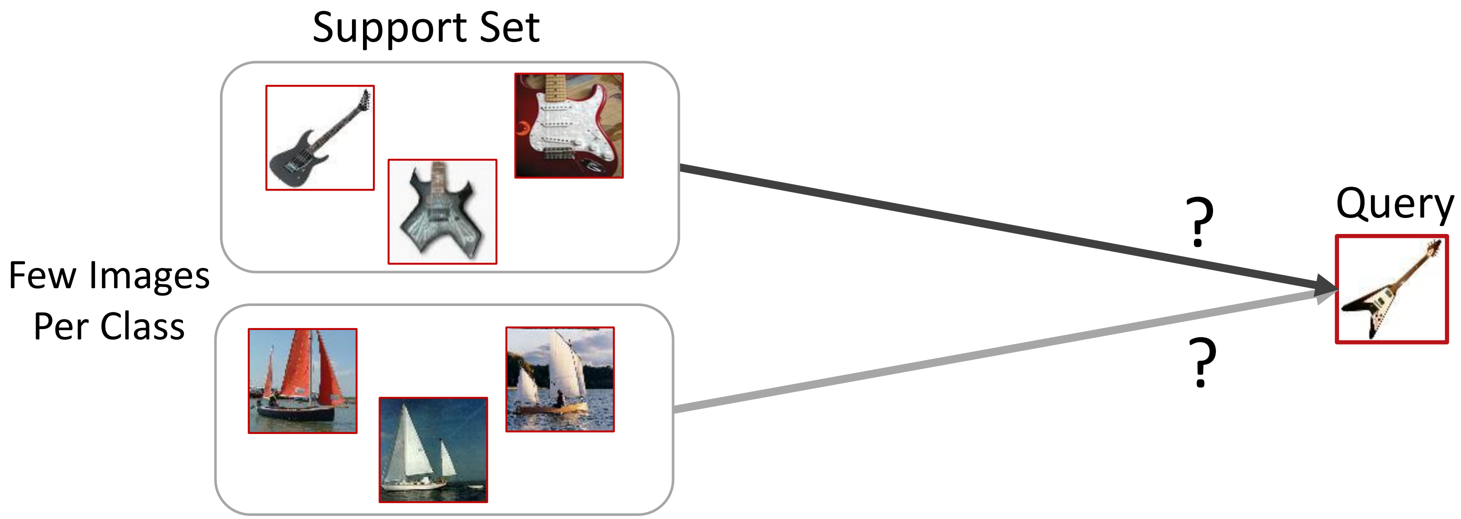
Christian Simon[†§] Piotr Koniusz[†§]

Richard Nock[†‡§] Mehrtash Harandi [#§]

† Australian National University ‡ THE UNIVERSITY OF SYDNEY # MONASH University § DATA 61 CSIRO

Problem Definition

- Given: A Support Set $S = \{(\boldsymbol{x}_{1,1}, c_{1,1}), (\boldsymbol{x}_{1,2}, c_{1,2}), \cdots, (\boldsymbol{x}_{N,K}, c_{N,K})\}; \boldsymbol{x} \in \mathbb{R}^D$
  A query $\boldsymbol{q} \in \mathbb{R}^D$
- A Support set contains *N*-way (classes) and *K*-shot (samples).
- The classes are unseen, can we classify them?.



Motivation

- 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:

$$p(c|\boldsymbol{q}) = \frac{\exp(\boldsymbol{w}_c^\top f_\Theta(\boldsymbol{q}))}{\sum_{c'} \exp(\boldsymbol{w}_{c'}^\top f_\Theta(\boldsymbol{q}))} = \frac{\exp(s_c(\boldsymbol{q}))}{\sum_{c'} \exp(s_{c'}(\boldsymbol{q}))}$$

Motivation

- The classifier needs to be updated (e.g. iterative gradient descents) using new samples if there are samples from unseen classes.

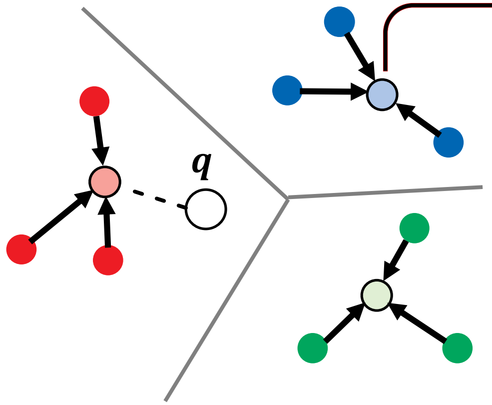
Should be updated

$$p(c|\boldsymbol{q}) = \frac{\exp(\boldsymbol{w}_c^\top f_\Theta(\boldsymbol{q}))}{\sum_{c'} \exp(\boldsymbol{w}_{c'}^\top f_\Theta(\boldsymbol{q}))}$$

Motivation

- 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:

Prototype Method



A prototype as a classifier $\boldsymbol{w}_c = \frac{1}{K}\sum_i f_\Theta(\boldsymbol{x}_i)$

$$p(c|\boldsymbol{q}) = \frac{\exp(\boldsymbol{w}_c^\top f_\Theta(\boldsymbol{q}))}{\sum_{c'} \exp(\boldsymbol{w}_{c'}^\top f_\Theta(\boldsymbol{q}))} = \frac{\exp(s_c(\boldsymbol{q}))}{\sum_{c'} \exp(s_{c'}(\boldsymbol{q}))}$$

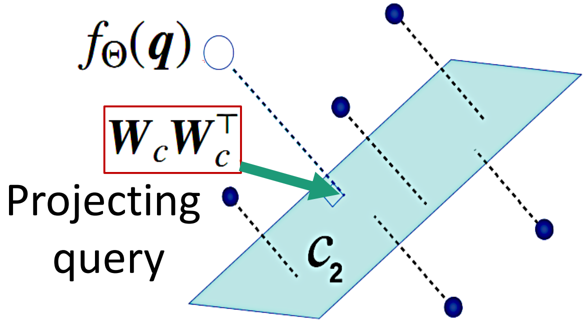
[1] Vinyals et al.,"Matching networks for one-shot learning," NIPS, 2016.
[2] Snell et al., "Prototypical networks for few-shot learning," NIPS, 2017.
[3] Sung et al., "Learning to compare: relation network for few-shot learning," CVPR, 2018
[4] Gidaris and Komodakis,"Learning without forgetting," CVPR, 2018.

Proposed Method

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

Our formulation:

$$\boxed{\begin{aligned} s_c(\boldsymbol{q}) &= -\left\| f_\Theta(\boldsymbol{q}) - \pi_c(\boldsymbol{q}) \right\|_2^2, \\ \pi_c(\boldsymbol{q}) &= \boldsymbol{W}_c \boldsymbol{W}_c^\top f_\Theta(\boldsymbol{q}) - \boldsymbol{b}_c. \end{aligned}}$$



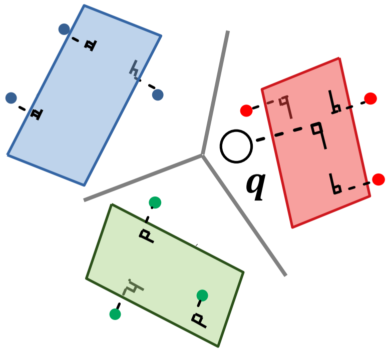
Where:

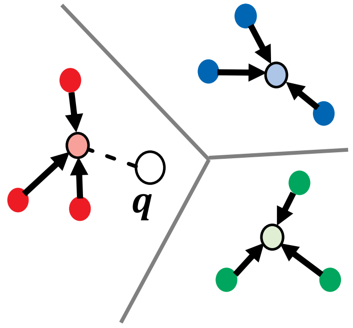
$\boldsymbol{W}_c$ is an orthogonal basis for linear subspace spanning $\mathbb{X}_c = \{f_\Theta(\boldsymbol{x}_i);\ y_i = c\}$

$\boldsymbol{b}_c = \mathbb{E}_{\boldsymbol{x}\sim p_c}\big(f_\Theta(\boldsymbol{x})\big)$

Proposed Method

Subspace Method VS Prototype Method





Algorithm 1 Train Deep Subspace Networks

Input: $S = \{(\boldsymbol{x}_{1,1}, c_{1,1}), \cdots, (\boldsymbol{x}_{N,K}, c_{N,K})\}$ and $\boldsymbol{q}$

$\Theta_0 \leftarrow$ random initialization
for t in $\{\mathcal{T}_1, ..., \mathcal{T}_{N_\mathcal{T}}\}$ **do**
  for c in $\{1, ..., N\}$ **do**
    Get $\tilde{\boldsymbol{X}}$ examples in the support set for class c
    $[\mathcal{U}, \Sigma, \mathcal{V}^\top] \leftarrow \mathrm{SVD}(\tilde{\boldsymbol{X}}_c)$
    $\boldsymbol{W}_c \leftarrow \mathcal{U}_{1,...,n}$
    Project $\boldsymbol{q}$ using $\boldsymbol{W}_c$
    Calculate log probability
  end for
  Update Θ from loss
end for

Algorithm 2 Train Prototypical Networks

Input: $S = \{(\boldsymbol{x}_{1,1}, c_{1,1}), \cdots, (\boldsymbol{x}_{N,K}, c_{N,K})\}$ and $\boldsymbol{q}$

$\Theta_0 \leftarrow$ random initialization
for t in $\{\mathcal{T}_1, ..., \mathcal{T}_{N_\mathcal{T}}\}$ **do**
  for c in $\{1, ..., N\}$ **do**
    Get $\tilde{\boldsymbol{X}}$ examples in the support set for class c
    $\boldsymbol{\mu}_c \leftarrow \frac{1}{K} \sum_{x \in \boldsymbol{X}} f_\Theta(\boldsymbol{x})$
    Calculate the similarity between $\boldsymbol{q}$ and $\boldsymbol{\mu}_c$
    Calculate log probability
  end for
  Update Θ from loss
end for

Experiments

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

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

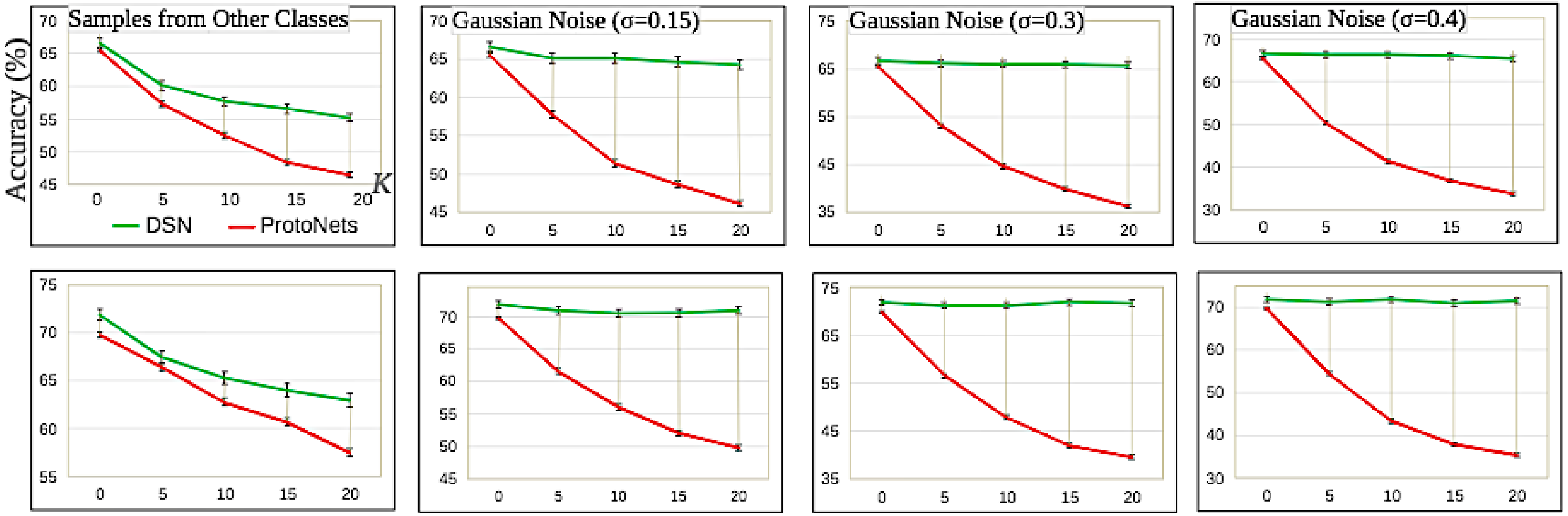
Accuracy 5-way 1-shot and 5-way 5-shot with 95% confidence intervals on the *mini*-ImageNet

Experiments

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

5-way
5-Shot

5-way
10-Shot



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.