Prototypical network for Few-shot learning

Jake Snell, University of Toronto
Kevin Swersky, Twitter
Richard S. Zemel, University of Toronto

Presenter: Viet Lai
Supervised learning

• MNIST: Hand-written dataset
Supervised learning

- **ImageNet**: Image database based on Wordnet
Data is new kind of electricity

- MNIST:
  - 60,000 images
- ImageNet:
  - 14,197,122 image
- English-French
  - 2,000,000 pairs of sentence
Case 1: Space vehicle launch

• Super expensive
  • $100M
• SpaceX’s budget
  • 3 launches before bankrupt
Case 2: Drug discovery

- Drug research is extremely expensive and long
- Using existing drug for new disease?
- How to confirm based on few cases?
Few-shot learning

- Supervised learning with a small amount of examples
- N-way K-shot
  - N classes
  - K samples for each class
  - K is very small
- Advantages
  - Reduce cost of annotation
  - Work in rare cases
Prototypical network

(a) Few-shot

(b) Zero-shot
Prototypical network

• Compute prototypes

\[ c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i) \]

• Compute the distribution among classes

\[ p_\phi(y = k \mid x) = \frac{\exp(-d(f_\phi(x), c_k))}{\sum_{k'} \exp(-d(f_\phi(x), c_{k'}))} \]

• Compute loss

\[ J(\phi) = -\log p_\phi(y = k \mid x) \]
As a linear model

• In case of Euclidean distance

\[-\|f_\phi(x) - c_k\|^2 = -f_\phi(x)\top f_\phi(x) + 2c_k\top f_\phi(x) - c_k\top c_k\]

\[2c_k\top f_\phi(x) - c_k\top c_k = w_k\top f_\phi(x) + b_k, \text{ where } w_k = 2c_k \text{ and } b_k = -c_k\top c_k\]
Algorithm

**Input:** Training set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, where each $y_i \in \{1, \ldots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all elements $(x_i, y_i)$ such that $y_i = k$.

**Output:** The loss $J$ for a randomly generated training episode.

```latex
V \leftarrow \text{RANDOMSAMPLE}(\{1, \ldots, K\}, N_C) \quad \triangleright \text{Select class indices for episode}
```

```latex
\textbf{for} k \textbf{in} \{1, \ldots, N_C\} \textbf{do}

```
```latex
S_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k}, N_S) \quad \triangleright \text{Select support examples}
```

```latex
Q_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k} \setminus S_k, N_Q) \quad \triangleright \text{Select query examples}
```

```latex
c_k \leftarrow \frac{1}{N_C} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i) \quad \triangleright \text{Compute prototype from support examples}
```

```latex
\textbf{end for}
```

```latex
J \leftarrow 0 \quad \triangleright \text{Initialize loss}
```

```latex
\textbf{for} k \textbf{in} \{1, \ldots, N_C\} \textbf{do}

```
```latex
\textbf{for} (x, y) \textbf{in} Q_k \textbf{do}

```
```latex
J \leftarrow J + \frac{1}{N_C N_Q} \left[ d(f_\phi(x), c_k) + \log \sum_{k'} \exp(-d(f_\phi(x), c_k)) \right] \quad \triangleright \text{Update loss}
```

```latex
\textbf{end for}
```

```latex
\textbf{end for}
```
```
Prototypical network vs Matching network

<table>
<thead>
<tr>
<th>Prototypical network</th>
<th>Matching network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean distance</td>
<td>Cosine similarity</td>
</tr>
<tr>
<td>Mean of examples</td>
<td>Weighted sum of examples based on KNN</td>
</tr>
<tr>
<td>Simple</td>
<td>Complicated</td>
</tr>
</tbody>
</table>
Omniglot dataset

- 1623 written characters
- 50 alphabets
- 20 shots

t-SNE visualization
Prototypical network vs Matching network

<table>
<thead>
<tr>
<th>Model</th>
<th>Dist.</th>
<th>Fine Tune</th>
<th>5-way Acc.</th>
<th>20-way Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
</tr>
<tr>
<td>Matching Networks [29]</td>
<td>Cosine</td>
<td>N</td>
<td>98.1%</td>
<td>98.9%</td>
</tr>
<tr>
<td>Matching Networks [29]</td>
<td>Cosine</td>
<td>Y</td>
<td>97.9%</td>
<td>98.7%</td>
</tr>
<tr>
<td>Neural Statistician [6]</td>
<td>-</td>
<td>N</td>
<td>98.1%</td>
<td>99.5%</td>
</tr>
<tr>
<td>Prototypical Networks (Ours)</td>
<td>Euclid.</td>
<td>N</td>
<td><strong>98.8%</strong></td>
<td><strong>99.7%</strong></td>
</tr>
</tbody>
</table>
miniImageNet dataset

- 60000 images
- 1000 classes
- 600 shots
Table 2: Few-shot classification accuracies on miniImageNet. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals. *Results reported by [22].

<table>
<thead>
<tr>
<th>Model</th>
<th>Dist.</th>
<th>Fine Tune</th>
<th>1-shot</th>
<th>5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Nearest Neighbors*</td>
<td>Cosine</td>
<td>N</td>
<td>28.86 ± 0.54%</td>
<td>49.79 ± 0.79%</td>
</tr>
<tr>
<td>Matching Networks [29]*</td>
<td>Cosine</td>
<td>N</td>
<td>43.40 ± 0.78%</td>
<td>51.09 ± 0.71%</td>
</tr>
<tr>
<td>Matching Networks FCE [29]*</td>
<td>Cosine</td>
<td>N</td>
<td>43.56 ± 0.84%</td>
<td>55.31 ± 0.73%</td>
</tr>
<tr>
<td>Meta-Learner LSTM [22]*</td>
<td>-</td>
<td>N</td>
<td>43.44 ± 0.77%</td>
<td>60.60 ± 0.71%</td>
</tr>
<tr>
<td>Prototypical Networks (Ours)</td>
<td>Euclid.</td>
<td>N</td>
<td><strong>49.42 ± 0.78%</strong></td>
<td><strong>68.20 ± 0.66%</strong></td>
</tr>
</tbody>
</table>
Prototypical network vs Matching network

Table 3: Zero-shot classification accuracies on CUB-200.

<table>
<thead>
<tr>
<th>Model</th>
<th>Image Features</th>
<th>50-way Acc. 0-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALE [1]</td>
<td>Fisher</td>
<td>26.9%</td>
</tr>
<tr>
<td>SJE [2]</td>
<td>AlexNet</td>
<td>40.3%</td>
</tr>
<tr>
<td>Sample Clustering [17]</td>
<td>AlexNet</td>
<td>44.3%</td>
</tr>
<tr>
<td>SJE [2]</td>
<td>GoogLeNet</td>
<td>50.1%</td>
</tr>
<tr>
<td>DS-SJE [23]</td>
<td>GoogLeNet</td>
<td>50.4%</td>
</tr>
<tr>
<td>DA-SJE [23]</td>
<td>GoogLeNet</td>
<td>50.9%</td>
</tr>
<tr>
<td>Proto. Nets (Ours)</td>
<td>GoogLeNet</td>
<td><strong>54.6%</strong></td>
</tr>
</tbody>
</table>
Question & Answer