Boulder

Computer Vision; Image Classification; Few-shot Learning

Maziar Raissi

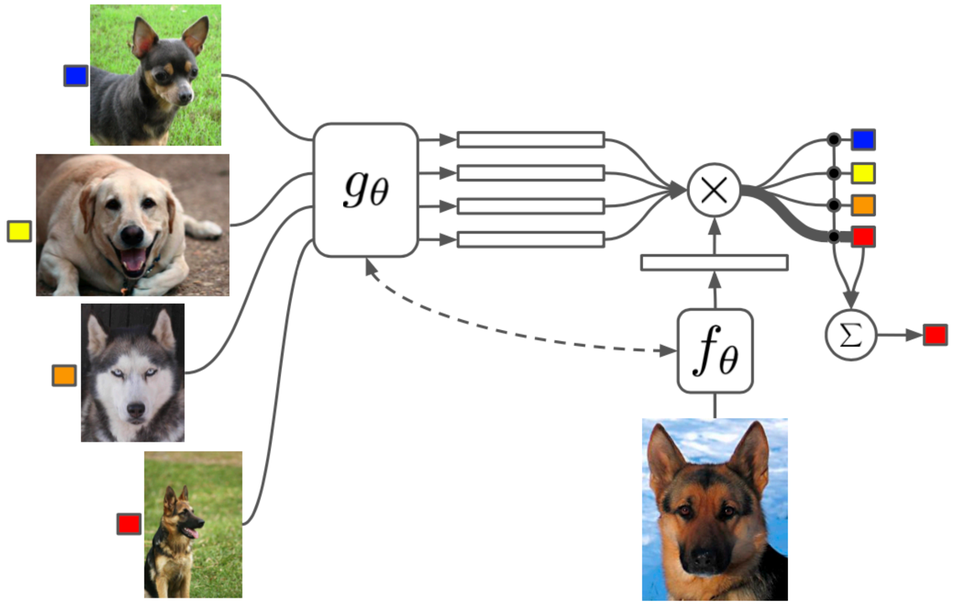
Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

Matching Networks for One Shot Learning



$S = \{(x_i, y_i)\}_{i=1}^{m} \rightarrow$ a (small) support set of m examples

$$P(\hat{y}|\hat{x}, S) = \sum_{i=1}^{m} a(\hat{x}, x_i) y_i$$

$$a(\hat{x}, x_i) = \frac{\exp(c(f(\hat{x}), g(x_i)))}{\sum_{j=1}^{m} \exp(c(f(\hat{x}), g(x_j)))}$$

$a \rightarrow$ attention mechanism
$c \rightarrow$ cosine similarity distance
f & $g \rightarrow$ neural networks

Training

$$\theta = \arg\max_{\theta} E_{L\sim T}\left[E_{S\sim L, B\sim L}\left[\sum_{(x,y)\in B} \log P_\theta\left(y|x, S\right)\right]\right]$$

$T \rightarrow$ task
$L \sim T \rightarrow$ pick N classes
$S \sim L \rightarrow$ provide the model with k examples per each class
$B \sim L \rightarrow$ provide the model with k examples per each class
$B \rightarrow$ Batch
$S \rightarrow$ Support Set
N-way k-shot learning task

Full Context Embeddings (FCE)

$$f(\hat{x}, S) = \text{attLSTM}(f'(\hat{x}), g(S), K)$$

$f' \rightarrow$ a neural network (e.g., VGG or Inception)
$K \rightarrow$ number of processing steps

The state after k processing steps is as follows:

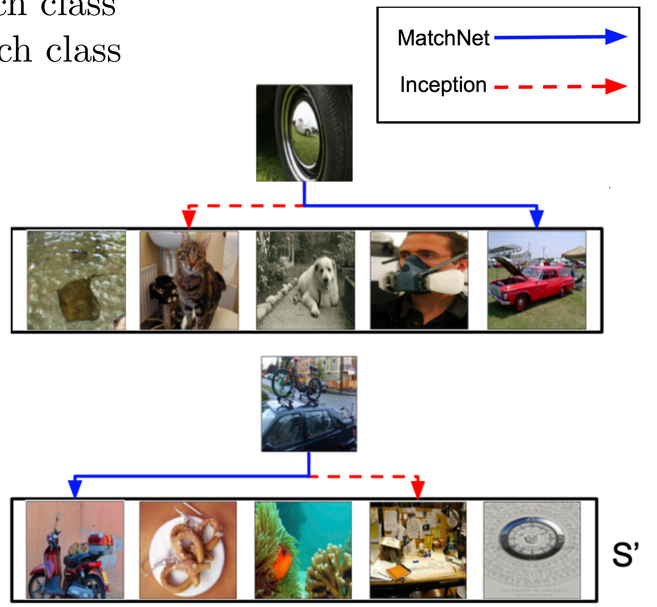
$$\begin{aligned} \hat{h}_k, c_k &= \text{LSTM}(f'(\hat{x}), [h_{k-1}, r_{k-1}], c_{k-1}) \\ h_k &= \hat{h}_k + f'(\hat{x}) \\ r_{k-1} &= \sum_{i=1}^{|S|} a(h_{k-1}, g(x_i)) g(x_i) \end{aligned}$$

$$a(h_{k-1}, g(x_i)) = \text{softmax}(h_{k-1}^T g(x_i))$$

$$\text{attLSTM}(f'(\hat{x}), g(S), K) = h_K$$

$$g(x_i, S) = \vec{h}_i + \overleftarrow{h}_i + g'(x_i)$$

$$\begin{aligned} \vec{h}_i, \vec{c}_i &= \text{LSTM}(g'(x_i), \vec{h}_{i-1}, \vec{c}_{i-1}) \\ \overleftarrow{h}_i, \overleftarrow{c}_i &= \text{LSTM}(g'(x_i), \overleftarrow{h}_{i+1}, \overleftarrow{c}_{i+1}) \end{aligned}$$



Example of two 5-way problem instance on ImageNet.

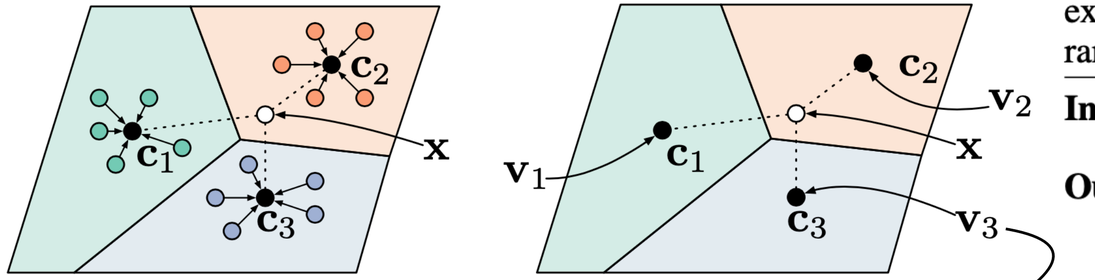
Results on *mini*ImageNet.

Model	Matching Fn	Fine Tune	5-way Acc 1-shot	5-way Acc 5-shot
PIXELS	Cosine	N	23.0%	26.6%
BASELINE CLASSIFIER	Cosine	N	36.6%	46.0%
BASELINE CLASSIFIER	Cosine	Y	36.2%	52.2%
BASELINE CLASSIFIER	Softmax	Y	38.4%	51.2%
MATCHING NETS (OURS)	Cosine	N	41.2%	56.2%
MATCHING NETS (OURS)	Cosine	Y	42.4%	58.0%
MATCHING NETS (OURS)	Cosine (FCE)	N	44.2%	57.0%
MATCHING NETS (OURS)	Cosine (FCE)	Y	**46.6%**	**60.0%**

Vinyals, Oriol, et al. "Matching networks for one shot learning." *arXiv preprint arXiv:1606.04080* (2016).

Prototypical Networks for Few-shot Learning

Few-shot classification is a task in which a classifier must be adapted to accommodate new classes not seen in training, given only a few examples of each of these classes.



(a) Few-shot

(b) Zero-shot

class meta-data vector

Inference

$S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$
↳ a small support set of N labeled examples

$x_i \in \mathbb{R}^D \rightarrow D$-dimensional feature vector of an example

$y_i \in \{1, 2, \ldots, K\} \rightarrow$ corresponding label

$S_k \rightarrow$ set of examples labeled with class k

$c_k \rightarrow$ prototype (M-dimensional representation of each class)

$f_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M \qquad \phi \rightarrow$ learnable parameters
↳ embedding function

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

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

$d : \mathbb{R}^M \times \mathbb{R}^M \rightarrow [0, +\infty)$
↳ distance function

$J(\phi) = -\log p_\phi(y = k|x) \rightarrow$ Training

Algorithm 1 Training episode loss computation for Prototypical Networks. N is the number of examples in the training set, K is the number of classes in the training set, $N_C \leq K$ is the number of classes per episode, N_S is the number of support examples per class, N_Q is the number of query examples per class. RANDOMSAMPLE(S, N) denotes a set of N elements chosen uniformly at random from set S, without replacement.

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where each $y_i \in \{1, \ldots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all elements $(\mathbf{x}_i, y_i)$ such that $y_i = k$.

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

$V \leftarrow$ RANDOMSAMPLE$(\{1, \ldots, K\}, N_C)$ ▷ Select class indices for episode

for k in $\{1, \ldots, N_C\}$ **do**

$S_k \leftarrow$ RANDOMSAMPLE$(\mathcal{D}_{V_k}, N_S)$ ▷ Select support examples

$Q_k \leftarrow$ RANDOMSAMPLE$(\mathcal{D}_{V_k} \setminus S_k, N_Q)$ ▷ Select query examples

$$\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_{\boldsymbol{\phi}}(\mathbf{x}_i)$$ ▷ Compute prototype from support examples

end for

$J \leftarrow 0$ ▷ Initialize loss

for k in $\{1, \ldots, N_C\}$ **do**

for $(\mathbf{x}, y)$ in Q_k **do**

$$J \leftarrow J + \frac{1}{N_C N_Q} \left[d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k)) + \log \sum_{k'} \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_{k'})) \right]$$ ▷ Update loss

end for

end for

The first term does not affect the softmax probabilities!

$$-\|f_\phi(\mathbf{x}) - \mathbf{c}_k\|^2 = -f_\phi(\mathbf{x})^\top f_\phi(\mathbf{x}) + 2\mathbf{c}_k^\top f_\phi(\mathbf{x}) - \mathbf{c}_k^\top \mathbf{c}_k$$
$$2\mathbf{c}_k^\top f_\phi(\mathbf{x}) - \mathbf{c}_k^\top \mathbf{c}_k = \mathbf{w}_k^\top f_\phi(\mathbf{x}) + b_k$$
$$\mathbf{w}_k = 2\mathbf{c}_k \text{ and } b_k = -\mathbf{c}_k^\top \mathbf{c}_k$$

The miniImageNet dataset consists of 60,000 color images of size 84 × 84 divided into 100 classes with 600 examples each. The splits use a set of 100 classes, divided into 64 training, 16 validation, and 20 test classes.

miniImageNet Few-shot Classification

Model	Dist.	Fine Tune	5-way Acc. 1-shot	5-way Acc. 5-shot
BASELINE NEAREST NEIGHBORS*	Cosine	N	28.86 ± 0.54%	49.79 ± 0.79%
MATCHING NETWORKS [32]*	Cosine	N	43.40 ± 0.78%	51.09 ± 0.71%
MATCHING NETWORKS FCE [32]*	Cosine	N	43.56 ± 0.84%	55.31 ± 0.73%
META-LEARNER LSTM [24]*	-	N	43.44 ± 0.77%	60.60 ± 0.71%
MAML [9]	-	N	**48.70 ± 1.84%**	63.15 ± 0.91%
PROTOTYPICAL NETWORKS (OURS)	Euclid.	N	**49.42 ± 0.78%**	**68.20 ± 0.66%**

Snell, Jake, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning." *Advances in neural information processing systems*. 2017.

Boulder

Questions?