Matching Networks for One Shot Learning

Oriol Vinyals
Google DeepMind
vinyals@google.com

Charles Blundell
Google DeepMind
cblundell@google.com

Timothy Lillicrap
Google DeepMind
countzero@google.com

Koray Kavukcuoglu
Google DeepMind
korayk@google.com

Daan Wierstra
Google DeepMind
wierstra@google.com

Published at NIPS'16

Motivation

- One-shot learning
  - Learning with only one labelled sample for each class
- Few-shot learning

- Have a small labelled support set
- Map an unlabeled example to its label

Human can easily see two giraffes in this picture.

Would this be possible for algorithms?

Problems and Existing Techniques

- Data augmentation
- Regularization techniques

- Root cause: Parametric aspect of the model
  - Large number of parameters
  - Gradient descent

- Hope in non-parametric models
  - Nearest neighbors with chosen metric

Novelty of the proposed Matching Nets (MN)

- Modeling
  - Uses attention and memory → Rapid learning
- Training procedure
  - Simple ML principal: **"Test and train conditions must match."**
- **"The main novelty of our model lies in reinterpreting a well studied framework (neural networks with external memories) to do one-shot learning." – Sec. 2.1.2**

The Proposed Matching Net

Simplest form of the model

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

- Linear combination of the labels in the support set
  - KDE
  - kNN
  - Attention mechanism

The Attention Kernel

$$a(\hat{x}, x_i) = e^{c(f(\hat{x}), g(x_i))} / \sum_{j=1}^{k} e^{c(f(\hat{x}), g(x_j))}$$

- Embedding functions f and g being appropriate neural networks
  - VGG
  - Inception

- An opportunity to use self-supervised learning?
  - Typical: SSL + supervised fine tuning
  - Can we do?: supervise pre-train + SSL → one-shot?

Training Strategy

- "Typically we consider *T* to uniformly weight all data sets of up to a few unique classes (e.g., 5), with a few examples per class (e.g., up to 5)."

Omniglot – 1623 characters from 50 alphabets

Model	Matching Fn	Fine Tune	5-way Acc		20-way Acc	
			1-shot	5-shot	1-shot	5-shot
PIXELS	Cosine	N	41.7%	63.2%	26.7%	42.6%
BASELINE CLASSIFIER	Cosine	N	80.0%	95.0%	69.5%	89.1%
BASELINE CLASSIFIER	Cosine	Y	82.3%	98.4%	70.6%	92.0%
BASELINE CLASSIFIER	Softmax	Y	86.0%	97.6%	72.9%	92.3%
MANN (NO CONV) [21]	Cosine	N	82.8%	94.9%	–	–
CONVOLUTIONAL SIAMESE NET [11]	Cosine	N	96.7%	98.4%	88.0%	96.5%
CONVOLUTIONAL SIAMESE NET [11]	Cosine	Y	97.3%	98.4%	88.1%	97.0%
MATCHING NETS (OURS)	Cosine	N	**98.1%**	**98.9%**	**93.8%**	98.5%
MATCHING NETS (OURS)	Cosine	Y	97.9%	98.7%	93.5%	**98.7%**

Table 1: Results on the Omniglot dataset.

ImageNet Examples

Table 2: 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%**

Table 3: Results on full ImageNet on *rand* and *dogs* one-shot tasks. Note that $\neq L_{rand}$ and $\neq L_{dogs}$ are sets of classes which are seen during training, but are provided for completeness.

Model	Matching Fn	Fine Tune	ImageNet 5-way 1-shot Acc L_{rand}	$\neq L_{rand}$	L_{dogs}	$\neq L_{dogs}$
PIXELS	Cosine	N	42.0%	42.8%	41.4%	43.0%
INCEPTION CLASSIFIER	Cosine	N	87.6%	92.6%	**59.8%**	90.0%
MATCHING NETS (OURS)	Cosine (FCE)	N	**93.2%**	**97.0%**	58.8%	**96.4%**
INCEPTION ORACLE	Softmax (Full)	Y (Full)	$\approx 99\%$	$\approx 99\%$	$\approx 99\%$	$\approx 99\%$