Matching Networks for One-Shot Learning

By DeepMind:

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra

Samujjwal Ghosh

cs16resch01001@iith.ac.in

@Samujjwal_Sam

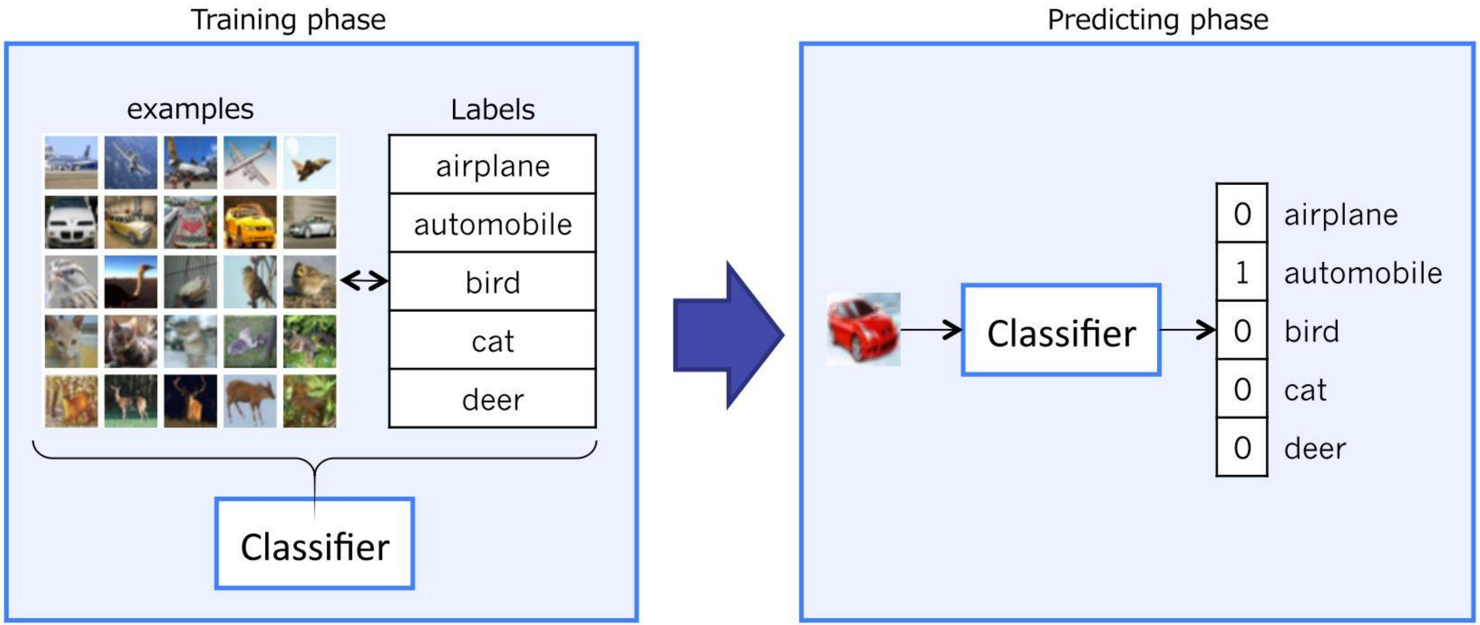
TWiML & AI Meetup EMEA, April 2, 2019

Abstract

- Techniques:
  - One-shot learning with attention and memory
  - Uniform training and testing strategy
- Advantage:
  - Utilize the advantage of both parametric and nonparametric learning
- Architecture Summary:
  - Differentiable nearest neighbor: incorporating the best characteristics from both parametric and nonparametric models
- Results:
  - Improved one-shot accuracy on ImageNet from 87.6% to 93.2% and on Omniglot from 88.0% to 93.8%

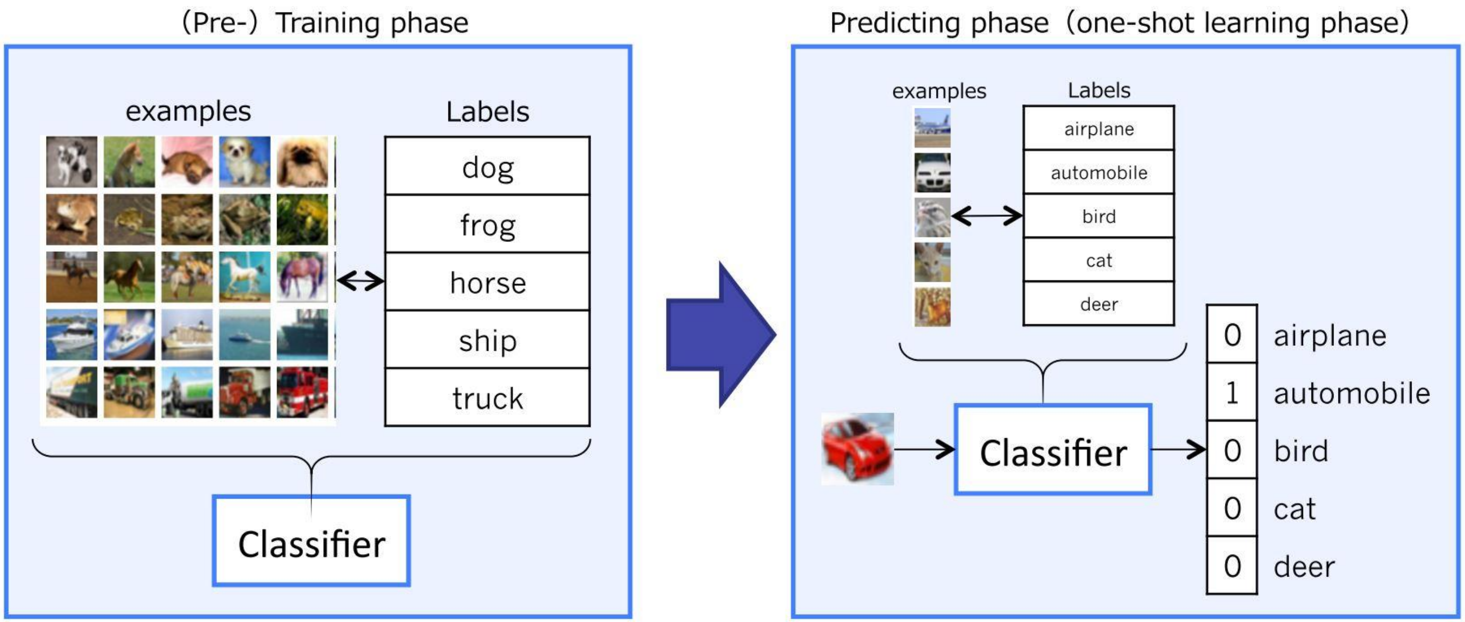
Supervised Learning

- Test labels are used during training



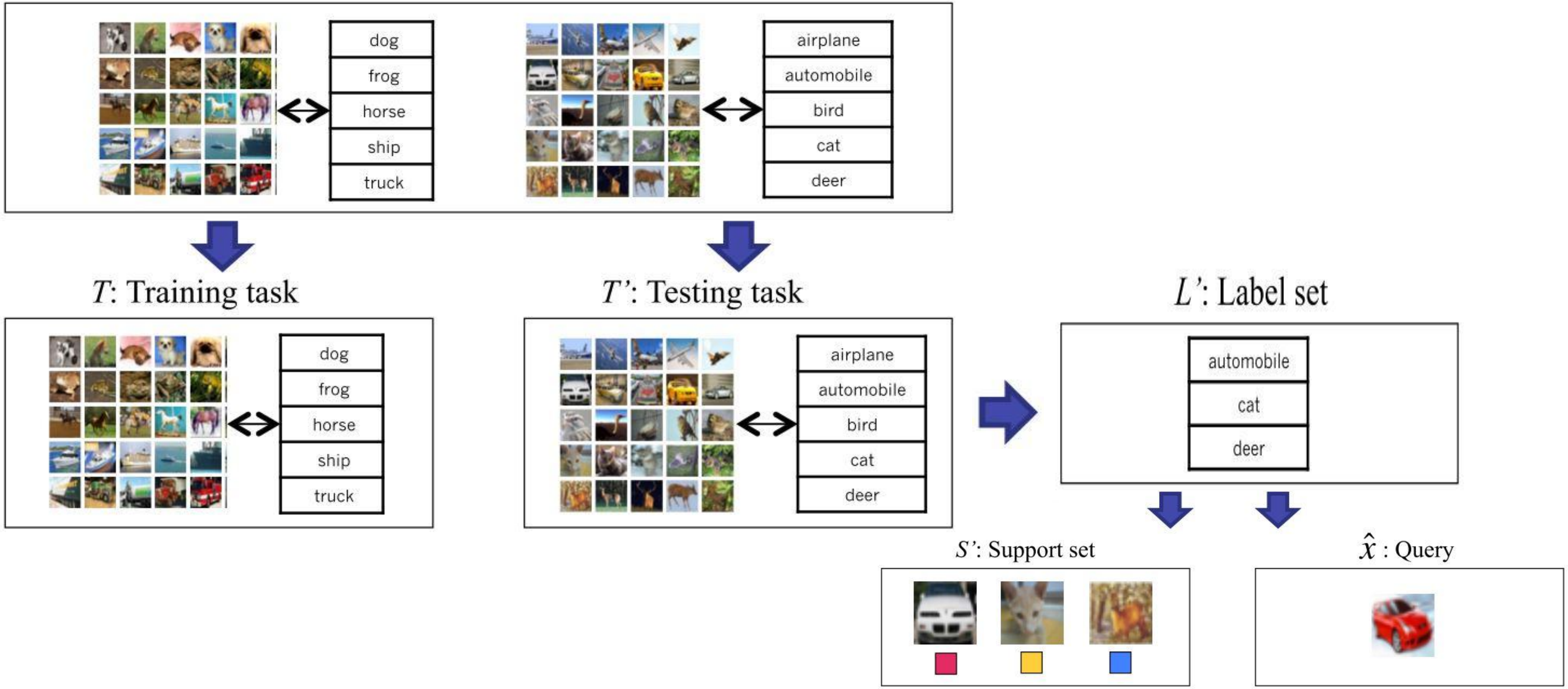
One-Shot Learning

- Test labels are not used during training. Disjoint label space
- **Idea**: A single image of Zebra is enough to show to a child



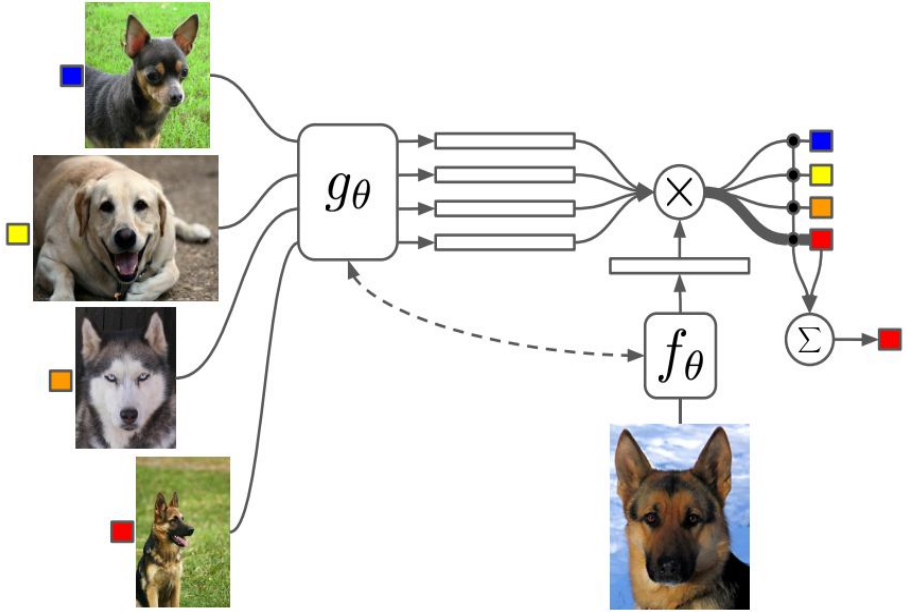
N-way k-shot Learning

- L’ has 3 labels and 1 sample per label, thus “**3-way 1-shot learning**”

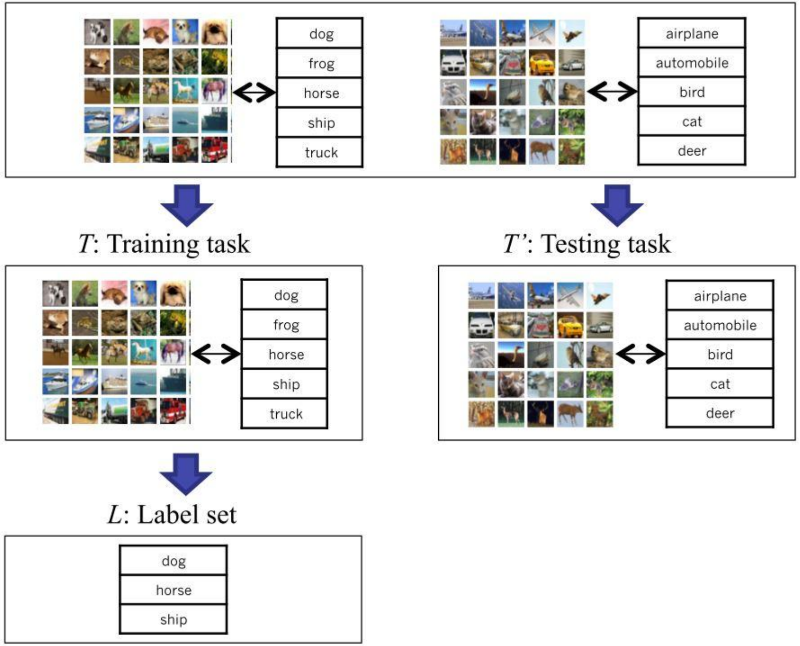


Contributions

Matching Networks



Training Strategy



Parametric & nonparametric learning

Parametric:

- Class properties are slowly learnt by models into its parameters
- Suffers from catastrophic forgetting

Nonparametric:

- Some models (e.g., k-NN) do not require any training
- Performance depends heavily on the “chosen” metric

Differentiable Nearest Neighbor

- Parametric Nearest Neighbor to embed inputs
- Define some parametric network to help us come up with a feature representation

1. Full Context Embedding (FCE)
    a. Embedding supports: $g(x_i)$
    b. Embedding targets: f(\hat{x})
2. Attention Kernel

Full Context Embedding (g)

- **Idea**: Encode each support in context of its neighbors within support set (S)
- **Using**: Use Bidirectional LSTM

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

$$\vec{h}_i, \vec{c}_i = \text{LSTM}(g'(x_i), \vec{h}_{i-1}, \vec{c}_{i-1})$$

$$\overleftarrow{h}_i, \overleftarrow{c}_i = \text{LSTM}(g'(x_i), \overleftarrow{h}_{i+1}, \overleftarrow{c}_{i+1})$$

g’: neural network (e.g., VGG or Inception)

Full Context Embedding (g)

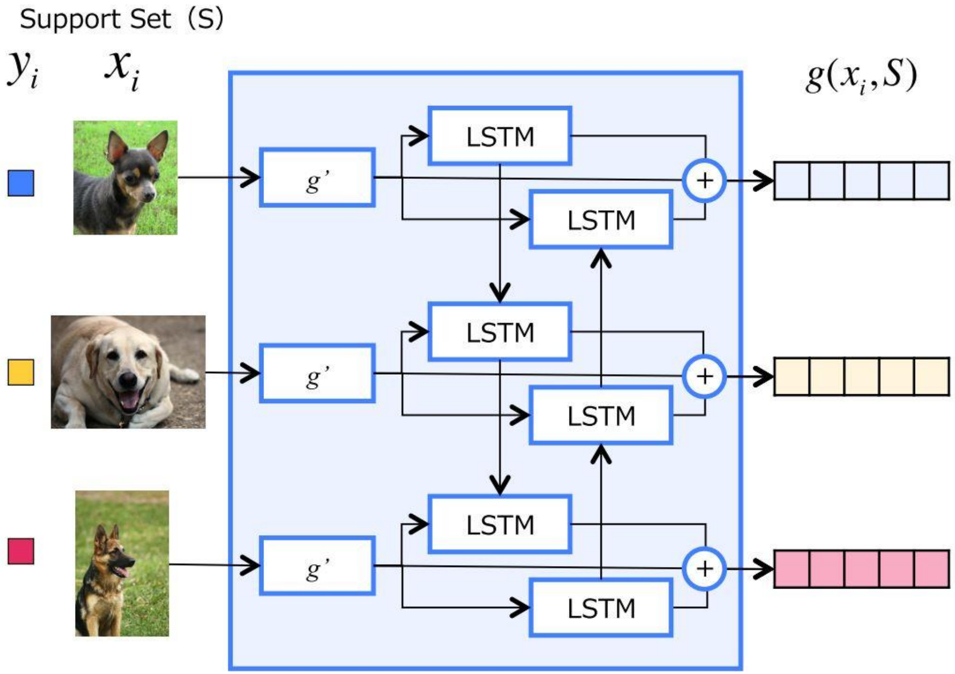
- **Idea**: Encode each support in context of its neighbors within support set (S)
- **Using**: Use Bidirectional LSTM

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

$$\vec{h}_i, \vec{c}_i = \mathrm{LSTM}(g'(x_i), \vec{h}_{i-1}, \vec{c}_{i-1})$$

$$\overleftarrow{h}_i, \overleftarrow{c}_i = \mathrm{LSTM}(g'(x_i), \overleftarrow{h}_{i+1}, \overleftarrow{c}_{i+1})$$

g': neural network (e.g., VGG or Inception)



Full Context Embedding (f)

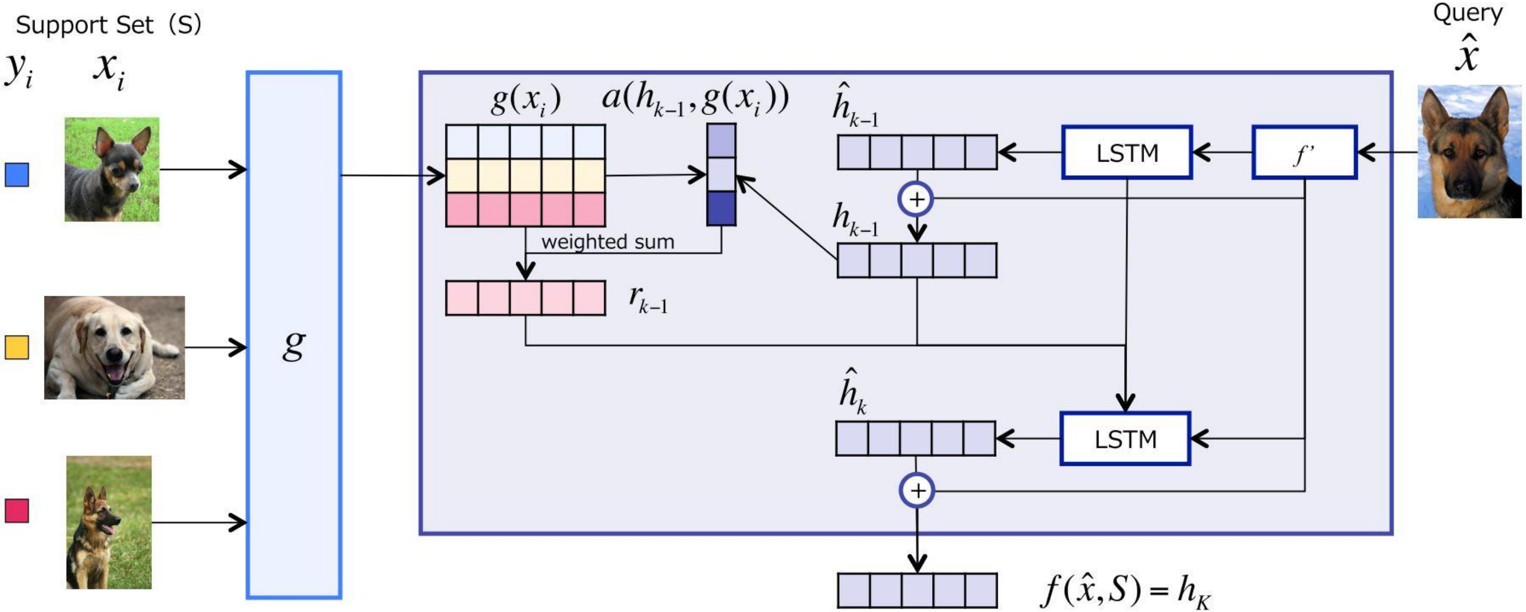
- **Idea**: Encode targets in context of its supports
- **Using**: Use Bidirectional LSTM with attention

$$
\begin{aligned}
\hat{h}_k, c_k &= \mathrm{LSTM}(f'(\hat{x}), [h_{k-1}, r_{k-1}], c_{k-1}) \\
h_k &= \hat{h}_k + f'(\hat{x}) \\
r_{k-1} &= \sum_{i=1}^{|S|} a(h_{k-1}, g(x_i)) g(x_i) \\
a(h_{k-1}, g(x_i)) &= \mathrm{softmax}(h_{k-1}^T g(x_i))
\end{aligned}
$$

Full Context Embedding (f)

- **Idea**: Encode targets in context of its supports
- **Using**: Use Bidirectional LSTM with attention

$$
\begin{aligned}
\hat{h}_k, c_k &= \mathrm{LSTM}(f'(\hat{x}), [h_{k-1}, r_{k-1}], c_{k-1}) \\
h_k &= \hat{h}_k + f'(\hat{x}) \\
r_{k-1} &= \sum_{i=1}^{|S|} a(h_{k-1}, g(x_i))g(x_i) \\
a(h_{k-1}, g(x_i)) &= \mathrm{softmax}(h_{k-1}^T g(x_i))
\end{aligned}
$$



Attention Kernel

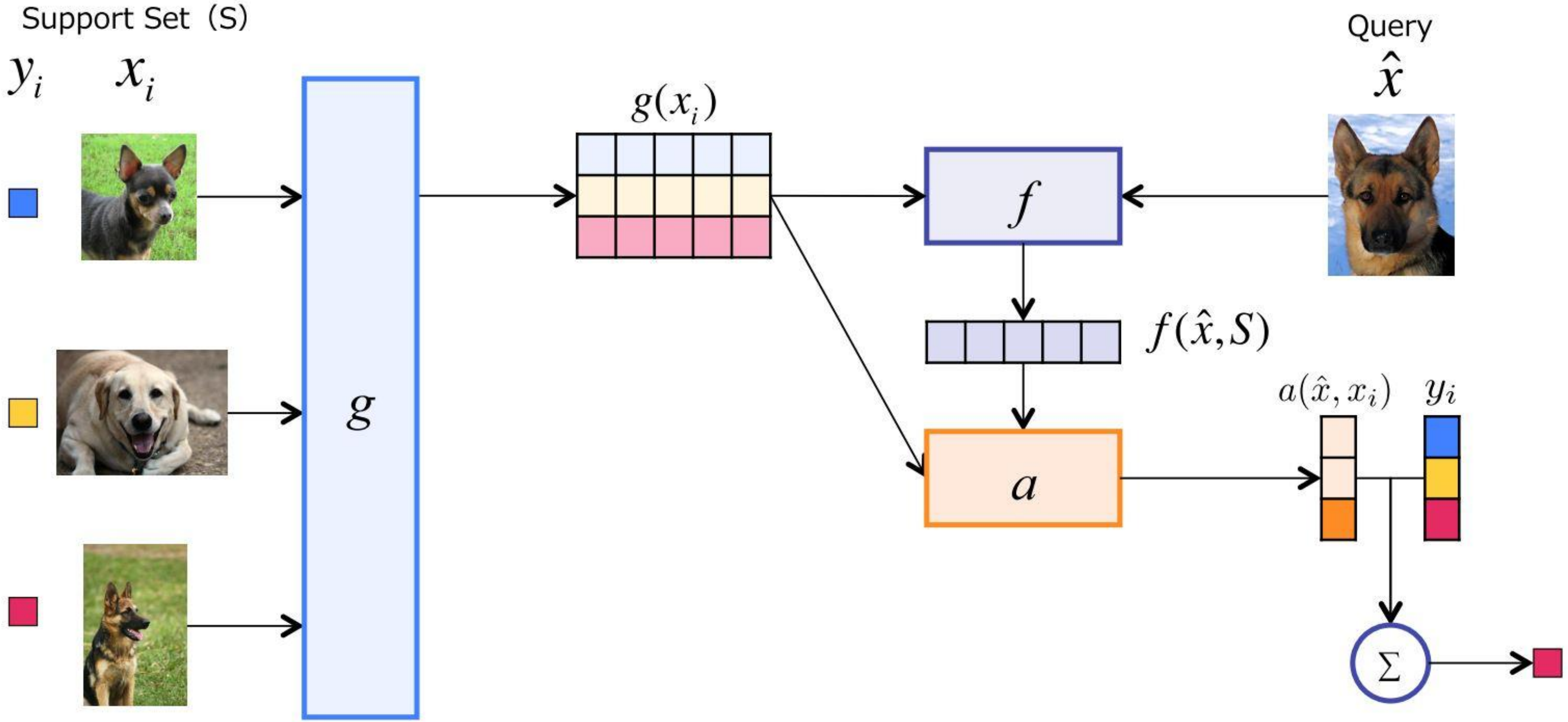
- Attention: Softmax over cosine distance between f(x,S) and g(x_i)

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

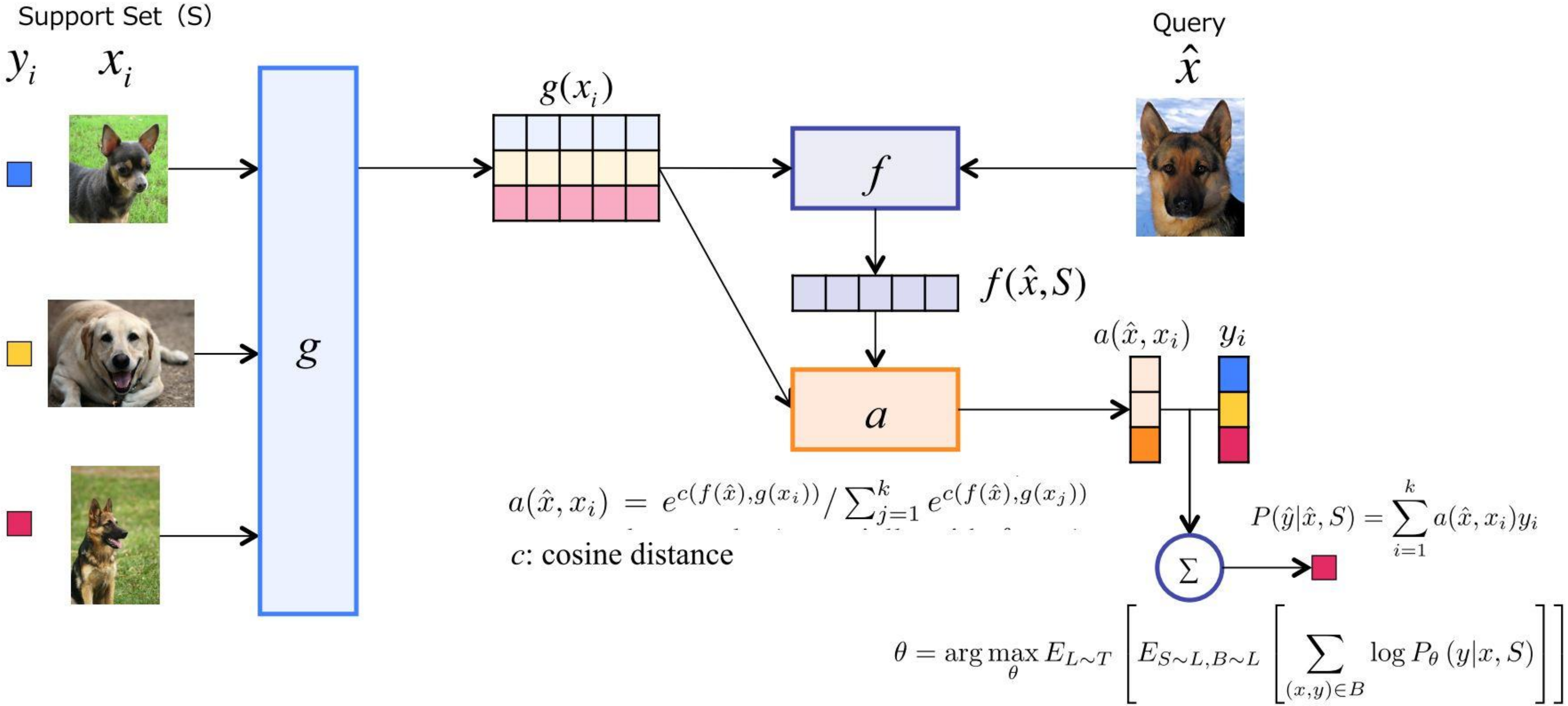
$$a(\hat{x}, x_i) = e^{c(f(\hat{x}), g(x_i))} / \sum_{j=1}^{k} e^{c(f(\hat{x}), g(x_j))}$$

- c(f(),g()) is cosine distance between target and support embedding
- Train using Cross Entropy loss
- Prediction is linear combination of labels in the support set:
  - 0.2 [1, 0, 0] + 0.5 [0, 1, 0] + 0.3 [0, 0, 1] = [0.2, 0.5, 0.3]

Matching Networks



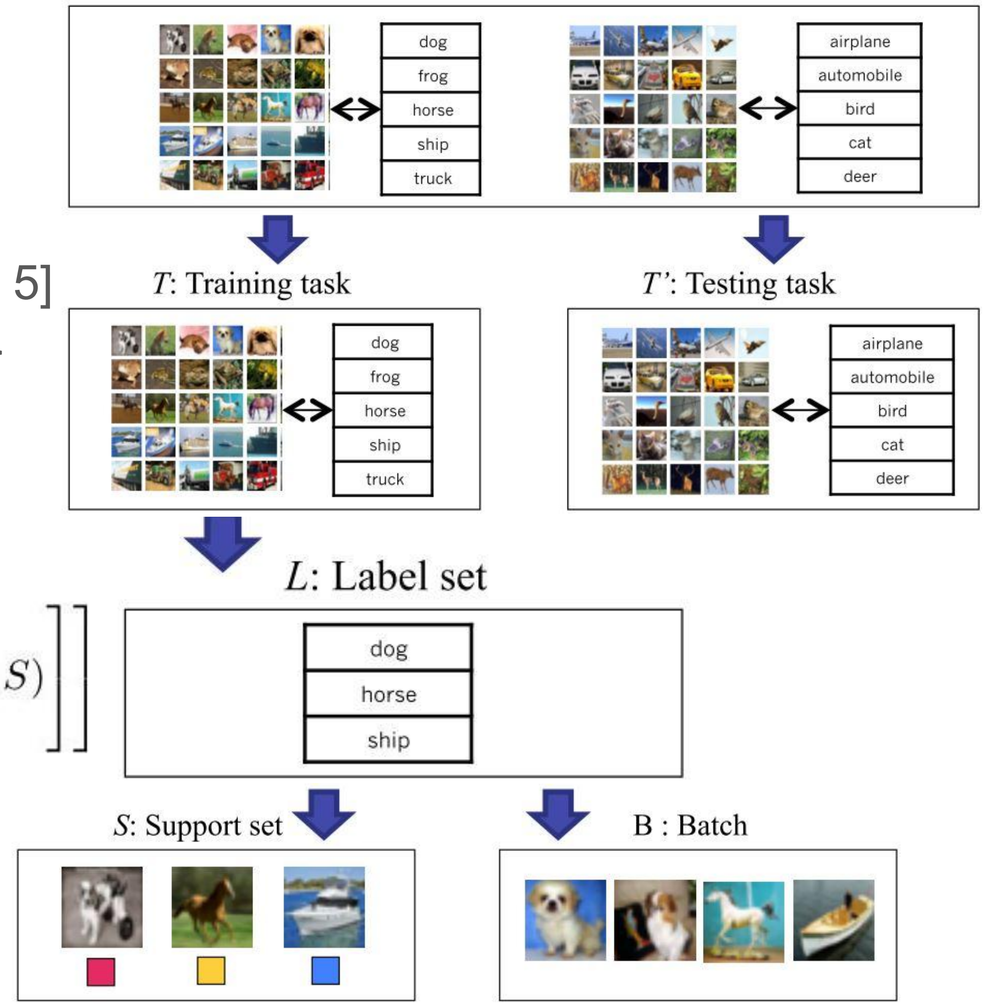
Matching Networks



Training strategy

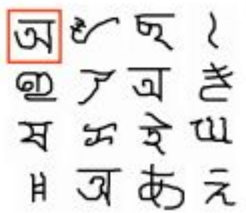
- Training task T
- L is sampled from T [Typically, |L| = 5]
- L could be the label set {cats, dogs}
- Sample Support Set S from L
- Sample Target Set B from L

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



Datasets

	OmniGlot	miniImageNet	Penn Treebank
Training:	1200 chars	80 classes	9000 words
Testing:	423 chars	20 classes	1000 words



Results: OmniGlot

Training: 1200 chars; **Testing**: 423 chars

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%**

- Fully Conditional Embedding (FCE) did not seem to help much
- Baseline and Siamese Net were improved with fine-tuning

Results: ImageNet

miniImageNet: Training: 80 classes; Testing: 20 classes

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%**

- Matching Networks overtook baseline
- Fully Conditional Embedding (FCE) was shown effective to improve the performance in this task

Results: ImageNet (Contd.)

randImageNet **dogsImageNet**

Training: random classes (882 classes) All non-dog classes (882 classes)

Testing: remaining classes (118 classes) Dog classes (118 classes)

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)	≈ 99%	≈ 99%	≈ 99%	≈ 99%

- Matching Net outperformed Inception in L_{rand} but degraded in L_{dogs}
- Decrease in performance in L_{dogs} might be cause training and testing data comes from different distribution.

Results

Penn Treebank

1. an experimental vaccine can alter the immune response of people infected with the aids virus a **<_>** u.s. scientist said.	prominent
2. the show one of five new nbc **<_>** is the second casualty of the three networks so far this fall.	series
3. however since eastern first filed for chapter N protection march N it has consistently promised to pay creditors N cents on the **<_>**.	dollar
4. we had a lot of people who threw in the **<_>** today said <unk> ellis a partner in benjamin jacobson & sons a specialist in trading ual stock on the big board.	towel
5. it's not easy to roll out something that **<_>** and make it pay mr. jacob says.	comprehensive
Q: in late new york trading yesterday the **<_>** was quoted at N marks down from N marks late friday and at N yen down from N yen late friday.	dollar

Oracle LSTM-LM: Trained on all the words (not one-shot), upper bound.

Model	**5 way accuracy**		
	1-shot	2-shot	3-shot
Matching Nets	32.4%	36.1%	38.2%
Oracle LSTM-LM	(72.8%)	-	-

Conclusion

- Nonparametric structure gives Matching Networks the ability to assimilate unseen classes very effectively
- Trainable end-to-end fully differentiable nearest neighbour with metric learning capability
- Matching Network is effective in handling unknown labels as seen on 3 different datasets
- Training a model “one-shot” way makes learning easier

Remarks

- Introduced Matching Networks
- Parametric perspective: Metric Learning
- Nonparametric perspective: Linear combination of labels of nearest neighbors
- Training and Support set label distributions should be close
- Ordering of inputs during FCE is not specified, however, it will matter.
    - $f(x', S)$: $(x', x_1, \ldots, x_k, \ldots, x_n) \neq (x', x_k, \ldots, x_1, \ldots, x_n)$
- Sample size during training and testing are fixed - not very suitable if training set grows online
- Becomes computationally expensive if support set is large

Resources

- Discussions:
  - https://vitalab.github.io/deep-learning/2018/01/24/MatchingNet.html
  - https://github.com/karpathy/paper-notes/blob/master/matching_networks.md
  - https://github.com/GokuMohandas/casual-digressions/blob/master/notes/oneshot.md#detailed-notes
  - https://blog.acolyer.org/2017/01/03/matching-networks-for-one-shot-learning/
  - https://www.slideshare.net/KazukiFujikawa/matching-networks-for-one-shot-learning-71257100
- Code:
  - TensorFlow: https://github.com/AntreasAntoniou/MatchingNetworks
  - Others: https://www.paperswithcode.com/paper/matching-networks-for-one-shot-learning

References

- **Attention**
  - Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014)
  - Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly. "Pointer networks." Advances in Neural Information Processing Systems 2015
  - Vinyals, Oriol, Samy Bengio, and Manjunath Kudlur. "Order matters: Sequence to sequence for sets." In ICLR 2016
- **Datasets**
  - Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009)
  - Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on. IEEE
  - Lake, Brenden M., et al. "One shot learning of simple visual concepts." Proceedings of the 33rd Annual Conference of the Cognitive Science Society. Vol. 172. 2011
  - Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini. "Building a large annotated corpus of English: The Penn Treebank." Computational linguistics 19.2 (1993): 313-330
- **Matching Networks**
  - Vinyals, Oriol, et al. "Matching networks for one shot learning." Advances in Neural Information Processing Systems 2016
- **One-shot Learning**
  - Koch, Gregory. Siamese neural networks for one-shot image recognition. Diss. University of Toronto, 2015.
  - Santoro, Adam, et al. "Meta-learning with memory-augmented neural networks." Proceedings of The 33rd International Conference on Machine Learning 2016.
  - Bertinetto, Luca, et al. "Learning feed-forward one-shot learners." Advances in Neural Information Processing Systems 2016.