# Summary of Matching Networks for One Shot Learning (NIPS 2016)

Muthu Chidambaram

Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

# Matching Networks for One Shot Learning (NIPS 2016)

---

- Authors: Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra
- Aims to approach learning representations from little data by drawing from non-parametric approaches (KNN)
- Learns a network that maps a small labelled support set and an unlabelled example to its label

# Matching Networks for One Shot Learning (NIPS 2016)

- Model architecture based on memory networks/pointer networks/attention models
- Casts one-shot learning as a set-to-set problem
  - Map from a small support set of k examples of image-label pairs S to a classifier C
  - Classifier defines probability distribution over output labels given a test example

# Matching Networks for One Shot Learning (NIPS 2016)

---

- Computes labels for an unseen example x hat as $\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$
  - X_i, y_i from support set, a is an attention mechanism
  - Attention mechanism subsumes both KDE and KNN, non-parametric in nature
- F, g are embeddings: $a(\hat{x}, x_i) = e^{c(f(\hat{x}), g(x_i))} / \sum_{j=1}^{k} e^{c(f(\hat{x}), g(x_j))}$
  - I.e. word embedding model for NLP or CNN for images
- Embeddings are functions of entire support set as well as specific example

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

# Matching Networks for One Shot Learning (NIPS 2016)

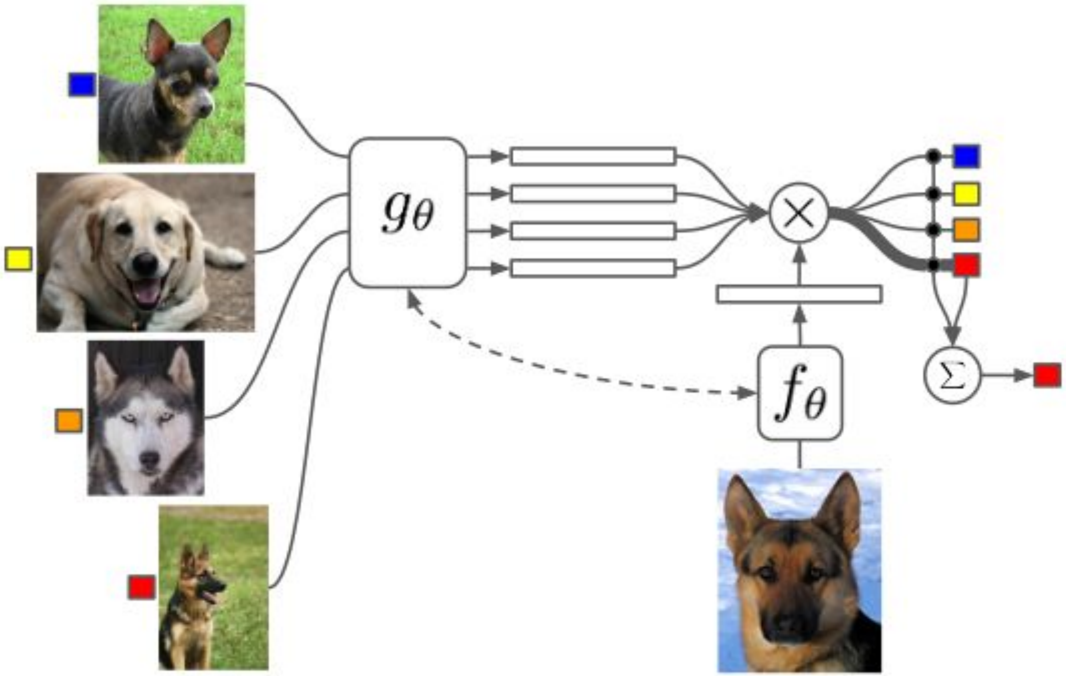

Figure 1: Matching Networks architecture

# Matching Networks for One Shot Learning (NIPS 2016)

- Define a task T as a distribution over possible label sets L
- Use L to sample the support set S and batch B, matching net is trained to minimize error in batch B conditioned on S

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

# Matching Networks for One Shot Learning (NIPS 2016)

- Experimenting done with k labelled examples from N classes not previously trained upon
  - Task is to classify a disjoint batch of unlabelled examples into one of these N classes
- Tested on Omniglot and ImageNet
- Baselines: raw pixel matching, matching on discriminative features, convolutional siamese net

# Matching Networks for One Shot Learning (NIPS 2016)

| Model | Matching Fn | Fine Tune | 5-way Acc 1-shot | 5-way Acc 5-shot | 20-way Acc 1-shot | 20-way Acc 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.

# Matching Networks for One Shot Learning (NIPS 2016)

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) | $\approx 99\%$ | $\approx 99\%$ | $\approx 99\%$ | $\approx 99\%$ |

1. O Vinyals, S Bengio, and M Kudlur. Order matters: Sequence to sequence for sets. arXiv preprint arXiv:1511.06391, 2015.

our process block based on an attention mechanism uses the following:

$$
\begin{aligned}
q_t &= LSTM(q^*_{t-1}) && (3)\\
e_{i,t} &= f(m_i, q_t) && (4)\\
a_{i,t} &= \frac{\exp(e_{i,t})}{\sum_j \exp(e_{j,t})} && (5)\\
r_t &= \sum_i a_{i,t} m_i && (6)\\
q^*_t &= [q_t \; r_t] && (7)
\end{aligned}
$$

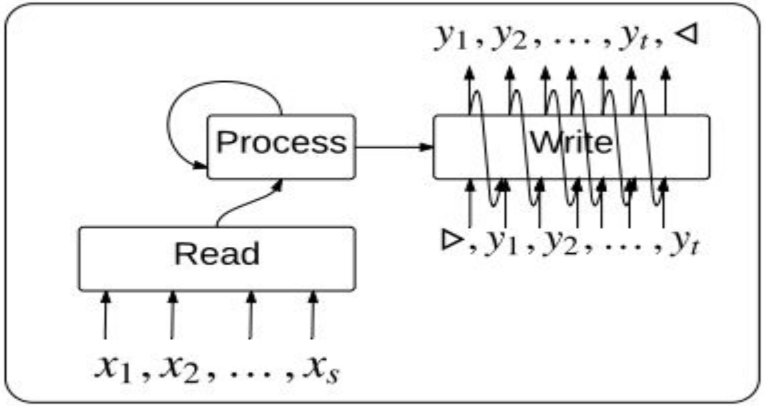


Figure 1: The Read-Process-and-Write model.

where $i$ indexes through each memory vector $m_i$ (typically equal to the cardinality of $X$), $q_t$ is a query vector which allows us to read $r_t$ from the memories, $f$ is a function that computes a single scalar from $m_i$ and $q_t$ (e.g., a dot product), and $LSTM$ is an LSTM which computes a recurrent state but which takes no inputs. $q^*_t$ is the state which this LSTM evolves, and is formed by concatenating the query $q_t$ with the resulting attention readout $r_t$. $t$ is the index which indicates how many "processing steps" are being carried to compute the state to be fed to the decoder. Note that permuting $m_i$ and $m_{i'}$ has no effect on the read vector $r_t$.

0. End-To-End Memory Networks

https://arxiv.org/pdf/1503.08895v5.pdf

1. Sequence to sequence learning with neural networks

https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

2. Meta-Learning with Memory-Augmented Neural Networks

http://jmlr.org/proceedings/papers/v48/santoro16.pdf

# Matching Networks for One Shot Learning (NIPS 2016)

- Conclusion: Introducing specific one-shot loss and non-parametric structure in neural network models leads to significant gains in one-shot classification tasks