Siamese Network &
Matching Network
for one-shot learning

Reference Papers
Siamese Neural Networks for One-Shot Image Recognition (Gregory Koch, Ruslan Salakhutdinov)
Matching Network for One-shot Learning (Oriol Vinyals et al.)
Order matters: Sequence to Sequence for Sets (Oriol Vinyals, Samy Bengio)
Pointer Networks (Oriol Vinyals et al.)
Face verification

- Verify whether a given test image is in the same class
- Large number of classes of data
- Number of training samples for a target class is very small

[Solution] Learning a similarity metric from data and then used it for target class
Verification to One-shot task

Verification tasks (training)

One-shot tasks (test)

Siamese Neural Networks for One-Shot Image Recognition (Gregory Koch, Ruslan Salakhutdinov, 2016)
Siamese Network

Energy function

\[ p = \sigma(\sum_j \alpha_j |h_{1,L-1}^{(j)} - h_{2,L-1}^{(j)}|) \]

Optimization

\[ L(x_1^{(i)}, x_2^{(i)}) = y(x_1^{(i)}, x_2^{(i)}) \log p(x_1^{(i)}, x_2^{(i)}) + (1 - y(x_1^{(i)}, x_2^{(i)})) \log (1 - p(x_1^{(i)}, x_2^{(i)})) + \lambda^T |w|^2 \]

One-shot classification

\[ C^* = \text{argmax}_c p^{(c)} \]

Figure 3. A simple 2 hidden layer siamese network for binary classification with logistic prediction \( p \). The structure of the network is replicated across the top and bottom sections to form twin networks, with shared weight matrices at each layer.

Siamese Neural Networks for One-Shot Image Recognition (Gregory Koch, Ruslan Salakhutdinov, 2016)
Experiments

Table 2. Comparing best one-shot accuracy from each type of network against baselines.

<table>
<thead>
<tr>
<th>Method</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Humans</td>
<td>95.5</td>
</tr>
<tr>
<td>Hierarchical Bayesian Program Learning</td>
<td>95.2</td>
</tr>
<tr>
<td>Affine model</td>
<td>81.8</td>
</tr>
<tr>
<td>Hierarchical Deep</td>
<td>65.2</td>
</tr>
<tr>
<td>Deep Boltzmann Machine</td>
<td>62.0</td>
</tr>
<tr>
<td>Simple Stroke</td>
<td>35.2</td>
</tr>
<tr>
<td>1-Nearest Neighbor</td>
<td>21.7</td>
</tr>
<tr>
<td>Siamese Neural Net</td>
<td>58.3</td>
</tr>
<tr>
<td>Convolutional Siamese Net</td>
<td>92.0</td>
</tr>
</tbody>
</table>

Table 3. Results from MNIST 10-versus-1 one-shot classification task.

<table>
<thead>
<tr>
<th>Method</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Nearest Neighbor</td>
<td>26.5</td>
</tr>
<tr>
<td>Convolutional Siamese Net</td>
<td>70.3</td>
</tr>
</tbody>
</table>
Matching Network

One(few)-shot prediction

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

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

\( \hat{x} \): test data
\( x_i \): support set
\( f \): input data embedding function
\( g \): support set embedding function
\( c \): cosine similarity

Key idea: context embedding for one(few)-shot sets

Matching Network for One-shot Learning (Oriol Vinyals et al., NIPS 2016)
Training objective

Objective: maximize the conditional probability given data and support set

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

- \(T\): full task set
- \(L\): label set
- \(S\): support set (one or few-shot set)
- \(B\): training batch

Matching Network for One-shot Learning (Oriol Vinyals et al., NIPS 2016)
Context Embedding

Embedding for $f$
(input data)
: Attention LSTM

\[
f(\tilde{x}, S) = \text{attLSTM}(f'(\tilde{x}), g(S), K)
\]

\[
\hat{h}_k, c_k = \text{LSTM}(f'(\tilde{x}), [h_{k-1}, r_{k-1}], c_{k-1})
\]

\[
h_k = \hat{h}_k + f'(\tilde{x})
\]

\[
r_{k-1} = \sum_{i=1}^{\left|S\right|} a(h_{k-1}, g(x_i))g(x_i)
\]

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

Embedding for $g$
(support set)
: Bidirectional LSTM

\[
g(x_i, S) = \tilde{h}_i + \bar{h}_i + g'(x_i)
\]

\[
\tilde{h}_i, \tilde{c}_i = \text{LSTM}(g'(x_i), \tilde{h}_{i-1}, \tilde{c}_{i-1})
\]

\[
\bar{h}_i, \bar{c}_i = \text{LSTM}(g'(x_i), \bar{h}_{i+1}, \bar{c}_{i+1})
\]

Matching Network for One-shot Learning (Oriol Vinyals et al., NIPS 2016)
Sequence-to-sequence model

\[(X^i, Y^i)\] : a pair of an input and its corresponding target

Sequence-to-sequence paradigm both X and Y are represented by sequences, of possibly different lengths: \[X^i = \{x^i_1, x^i_2, \ldots, x^i_{n_i}\} \quad Y^i = \{y^i_1, y^i_2, \ldots, y^i_{m_i}\}\]

\[P(Y|X) = \prod_{t=1}^{T} P(y_t|y_1, y_2, \ldots, y_{t-1}, X)\]

[ref] Sequence to Sequence Learning with Neural Networks (Ilya Sutskever, Oriol Vinyals, NIPS 2014)
Sequence-to-sequence model

What if input does not naturally correspond to a sequence?

Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

\[
\begin{align*}
    h_s &= f_{enc}(h_{s-1}, x_s) \\
    g_1 &= h_s \\
    g_t &= f_{dec}(g_{t-1}, y_{t-1})
\end{align*}
\]

\[
P(y_t|y_1, y_2, \ldots, y_{t-1}, X) = \text{softmax}(\text{affine}(g_t))
\]

[ref] Sequence to Sequence Learning with Neural Networks (Ilya Sutskever, Oriol Vinyals, NIPS 2014)
Order matters

- Altering the order of sequence in the context of machine translation: performance changes
  - English to French; reversing the order of input sentence Sutskever et al. (2014) got 5.0 BLEU score improvement
  - Constituency parsing; reversing the order of input sentence 0.5% increase in F1 score (Vinyals et al., 2016)
  - Convex hull computation presented in Vinyals et al. (2015) by sorting the points by angle, the task becomes simpler and faster

Empirical findings point to the same story: input order matters

[ref] Order matters: Sequence to Sequence for Sets (Oriol Vinyals, Samy Bengio, ICLR 2016)
Attention LSTM

\[ q_t = LSTM(q_{t-1}^*) \] (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)

\( q_t \) : query vector
\( m_i \) : memory vector
\( f \) : dot product

Sequential content based addressing => input order invariant

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

[ref] Order matters: Sequence to Sequence for Sets (Oriol Vinyals, Samy Bengio, ICLR 2016)
Attention LSTM

- A reading block which simply embeds each element $x_i$ onto a memory vector $m_i$
- A process block which is an LSTM without inputs or outputs performing $T$ steps of computation over the memories $m_i$. This LSTM keeps updating its state by reading $m_i$ repeatedly using attention mechanism.
- A write block, which is an LSTM pointer network that takes in $q_t$ and points at elements of $m_i$, one step at a time.

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

[ref] Order matters: Sequence to Sequence for Sets (Oriol Vinyals, Samy Bengio, ICLR 2016)
When dealing with combinatorial problem, (e.g. convex hull, Traveling Salesman Problem) output dictionary relies on the length of input sequence. To solve this, decoder focuses on the previous encoder state by attention mechanism.

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_i) \quad j \in (1, \ldots, n)$$

$$p(C_1|C_1, \ldots, C_{i-1}, P) = \text{softmax}(u^i)$$

[ref] Pointer Networks (Oriol Vinyals et al., 2015)
Conclusion

• Employed Attention LSTM for set problem (instead of sequence) – Memory network
• Context embedding for support set
• What if support set becomes larger?
• Classification on existing categories

Matching Network for One-shot Learning (Oriol Vinyals et al., NIPS 2016)
Experiments for matching network

### Table 1: Results on the Omniglot dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Matching Fn</th>
<th>Fine Tune</th>
<th>5-way Acc 1-shot</th>
<th>5-way Acc 5-shot</th>
<th>20-way Acc 1-shot</th>
<th>20-way Acc 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIXELS</td>
<td>Cosine</td>
<td>N</td>
<td>41.7%</td>
<td>63.2%</td>
<td>26.7%</td>
<td>42.6%</td>
</tr>
<tr>
<td>BASELINE CLASSIFIER</td>
<td>Cosine</td>
<td>N</td>
<td>80.0%</td>
<td>95.0%</td>
<td>69.5%</td>
<td>89.1%</td>
</tr>
<tr>
<td>BASELINE CLASSIFIER</td>
<td>Cosine</td>
<td>Y</td>
<td>82.3%</td>
<td>98.4%</td>
<td>70.6%</td>
<td>92.0%</td>
</tr>
<tr>
<td>BASELINE CLASSIFIER</td>
<td>Softmax</td>
<td>Y</td>
<td>86.6%</td>
<td>97.6%</td>
<td>72.9%</td>
<td>92.3%</td>
</tr>
<tr>
<td>MANN (NO CONV) [21]</td>
<td>Cosine</td>
<td>N</td>
<td>82.8%</td>
<td>94.9%</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>CONVOLUTIONAL SIAMESE NET [11]</td>
<td>Cosine</td>
<td>N</td>
<td>96.7%</td>
<td>98.4%</td>
<td>88.0%</td>
<td>96.5%</td>
</tr>
<tr>
<td>CONVOLUTIONAL SIAMESE NET [11]</td>
<td>Cosine</td>
<td>Y</td>
<td>97.3%</td>
<td>98.4%</td>
<td>88.1%</td>
<td>97.0%</td>
</tr>
<tr>
<td>MATCHING NETS (OURS)</td>
<td>Cosine</td>
<td>N</td>
<td>98.1%</td>
<td>98.9%</td>
<td>93.8%</td>
<td>98.5%</td>
</tr>
<tr>
<td>MATCHING NETS (OURS)</td>
<td>Cosine</td>
<td>Y</td>
<td>97.9%</td>
<td>98.7%</td>
<td>93.5%</td>
<td>98.7%</td>
</tr>
</tbody>
</table>

### Table 2: Results on miniImageNet.

<table>
<thead>
<tr>
<th>Model</th>
<th>Matching Fn</th>
<th>Fine Tune</th>
<th>5-way Acc 1-shot</th>
<th>5-way Acc 5-shot</th>
<th>20-way Acc 1-shot</th>
<th>20-way Acc 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIXELS</td>
<td>Cosine</td>
<td>N</td>
<td>23.0%</td>
<td>26.6%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BASELINE CLASSIFIER</td>
<td>Cosine</td>
<td>N</td>
<td>36.6%</td>
<td>46.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BASELINE CLASSIFIER</td>
<td>Cosine</td>
<td>Y</td>
<td>36.2%</td>
<td>52.2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BASELINE CLASSIFIER</td>
<td>Softmax</td>
<td>Y</td>
<td>38.4%</td>
<td>51.2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCHING NETS (OURS)</td>
<td>Cosine</td>
<td>N</td>
<td>41.2%</td>
<td>56.2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCHING NETS (OURS)</td>
<td>Cosine</td>
<td>Y</td>
<td>42.4%</td>
<td>58.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCHING NETS (OURS)</td>
<td>Cosine (FCE)</td>
<td>N</td>
<td>44.2%</td>
<td>57.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCHING NETS (OURS)</td>
<td>Cosine (FCE)</td>
<td>Y</td>
<td>46.6%</td>
<td>60.0%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Matching Network for One-shot Learning (Oriol Vinyals et al., NIPS 2016)