Memory (Neuron) Networks

Petr Baudiš ⟨pasky@ucw.cz⟩

FEE CTU Prague; Ailao

Jan 2016
Motivation

“Normal” NNs: Map input FV to output label
MemNN task: Map input FV to a past memory
Perfect setting: NLP and Question Answering
bAbI dataset (simple stories)

Mary went to the bathroom. John is in the playground.
John moved to the hallway. John picked up the football.
Mary travelled to the office. Bob went to the kitchen.
**Where is Mary?** A: office
**Where is the football?** A: hallway
Motivation

“Normal” NNs: Map input FV to output label
MemNN task: Map input FV to a past memory
Perfect setting: NLP and Question Answering
bAbI dataset (simple stories)

Mary went to the bathroom. John is in the playground.
John moved to the hallway. John picked up the football.
Mary travelled to the office. Bob went to the kitchen.
Where is Mary? A: office
Where is the football? A: hallway
Motivation

“Normal” NNs: Map input FV to output label
MemNN task: Map input FV to a past memory
Perfect setting: NLP and Question Answering
bAbI dataset (simple stories)

Mary went to the bathroom. John is in the playground. John moved to the hallway. John picked up the football. Mary travelled to the office. Bob went to the kitchen.
Where is Mary? A: office
Where is the football? A: hallway
MemNN Architecture

- IGOR — Introduced in (Weston, 1410.3916)
- I: Map input to features (numeric vectors)
- G: Generalization — store input in memory
- O: Output — combine input and memory to output features
- R: Representation — convert numeric vectors to data
Basic MemNN (Weston, 1410.3916)

- I: Bag-of-words embeddings
- G: Memory is just a list we append to
- O: Pick memory by learning model $s(x, y) = x^T \cdot U^T \cdot U \cdot y^T$
- R: Select memory from the word by another $s$-like function

Many extensions — multi-hop inference, memory hashing, write time model, etc.

Mary went to the bathroom. John is in the playground. John moved to the hallway. John picked up the football. Mary travelled to the office. Bob went to the kitchen. Where is Mary? A: office
Deep Sentence Selection (Yu, 1412.1632)

Key problem: **Evaluating pairs of sentences**

- $U$ matrix is a common trick
- In (Yu, 1412.1632): $s(x, y) = \sigma(x^T M \cdot y^T)$
- In (Weston, 1410.3916): $s(x, y) = x^T U^T \cdot U y$
- **Generative model** — (i) the matrix projects between vector spaces of embeddings; (ii) dot-product is a measure of similarity
- Possible to train by simple regressions
- More complex projection models possible (e.g. GRU, LSTM)
Better MemNN (Weston, 1502.05698)

- MemNNs continue to develop besides the original paper
- (Originated at Facebook)
- Adaptive memory: Flexible number of hops
- Better relevancy scoring models (nonlinearity, MLP)
- Bag-of-ngrams
### Better MemNN performance (Weston, 1502.05698)

<table>
<thead>
<tr>
<th>TASK</th>
<th>Weakly Supervised</th>
<th>Uses External Resources</th>
<th>Strong Supervision (using supporting facts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Single Supporting Fact</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>2 - Two Supporting Facts</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>3 - Three Supporting Facts</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>4 - Two Arg. Relations</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>5 - Three Arg. Relations</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>6 - Yes/No Questions</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>7 - Counting</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>8 - Lists/Sets</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>9 - Simple Negation</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>10 - Indefinite Knowledge</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>11 - Basic Coreference</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>12 - Conjunction</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>13 - Compound Coref.</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>14 - Time Reasoning</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>15 - Basic Deduction</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>16 - Basic Induction</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>17 - Positional Reasoning</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>18 - Size Reasoning</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>19 - Path Finding</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>20 - Agent’s Motivations</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Mean Performance</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

**Note:** The table shows the performance metrics for different tasks, comparing weakly supervised, uses external resources, and strong supervision with supporting facts.
MemN2N (Sukhbaatar, 1503.08895)

End-to-end MemNN, weaker supervision
Conclusion

- Sorry for the brief presentation
- By itself still only toy tasks — simple questions, small number of words, small memory

Also check out:
- Keras MemNN examples
- Dynamic Memory Networks \((Kumar, 1506.07285)\)
- Reading Children’s Books \((Hill, 1511.02301)\)

pasky@ucw.cz
petr.baudis@gmail.com

Thank you for your attention!

*If you have good ideas, good data and fast computers, you can do almost anything.* – Geoffrey Hinton