Ask Me Anything: Dynamic Memory Networks for Natural Language Processing

Ankit Kumar, Peter Ondruska, Mohit Ivyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher

MetaMind

ICML, 2017
Presenter: Tianlu Wang
Introduction

- Tasks in natural language processing can be cast as a question answering problem:
  - Machine Translation ⇒ What is the translation into French?
  - Name entity recognition ⇒ What are the name entity tags in this sentence?

\[\text{Input sequences} \downarrow \text{Questions} \downarrow \text{Dynamic Memory Network (DMN)} \downarrow \text{Answers}\]

- State-of-the-art on multiple dataset:
  - Question answering (Facebook bAbI dataset)
  - Text classification for sentiment analysis (Stanford Sentiment Treebank)
  - Sequence modeling for part-of-speech tagging (WSJ-PTB)
The episodic memory in humans stores specific experiences in their spatial and temporal context.

Provide a vector representation to capture all relevant information from input sequences and questions.
Outline

1 Introduction

2 Dynamic Memory Network
   - Model Overview
   - Encoding and Mutations
   - More Details

3 Results
   - Progress of experiments
   - Comparisons
   - Meta-parameters

4 Summary
Model Overview

- Input: a population of models, each model is a trained single-layer nonconvolutional model with learning rate = 0.1
- Measurement: accuracy on validation dataset
Model Overview

- Input: a population of models, each model is a trained single-layer nonconvolutional model with learning rate = 0.1
- Measurement: accuracy on validation dataset
Model Overview

- Input: a population of models, each model is a **trained single-layer nonconvolutional model** with learning rate = 0.1
- Measurement: accuracy on validation dataset

When to stop?

---

Ankit Kumar, Peter Ondruska, Mohit Ivyer, JAsk Me Anything: Dynamic Memory Network
Outline

1. Introduction

2. Dynamic Memory Network
   - Model Overview
   - Encoding and Mutations
   - More Details

3. Results
   - Progress of experiments
   - Comparisons
   - Meta-parameters

4. Summary
Model Encoding

Individual model is encoded as a graph:

- **Vertices**
  - rank-3 tensor(image width * image height * channels)
  - activations(batch normalization with ReLU or plain linear layer)

- **Edges**
  - Identity connections
  - Convolutions
Model Encoding

Individual model is encoded as a graph:

- **Vertices**
  - rank-3 tensor($image\_width \times image\_height \times channels$)
  - activations(batch normalization with ReLU or plain linear layer)

- **Edges**
  - Identity connections
  - Convolutions

Inconsistent input:

- pick and keep primary one
- reshape(interpolation/truncation/padding) non-primary ones
Mutations

The worker picks a mutation at random from a set:

- ALTER-LEARNING-RATE
- IDENTITY (effectively means keep training)
- RESET-WEIGHTS
- INSERT/REMOVE CONVOLUTION
- ALTER-STRIDE
- ALTER-NUMBER-OF-CHANNELS
- FILTER-SIZE
- INSERT-ONE-TO-ONE
- INSERT/REMOVE SKIP
Outline

1 Introduction

2 Dynamic Memory Network
   - Model Overview
   - Encoding and Mutations
   - More Details

3 Results
   - Progress of experiments
   - Comparisons
   - Meta-parameters

4 Summary
- Poor initial conditions (12th slide)
- 45,000 training; 5,000 validation; 10,000 test
- SGD with momentum of 0.9, batch size 50, weight decay 0.0001
- Computation cost: floating-point operations
- Inherit parameters’ weights whenever possible
Outline

1. Introduction

2. Dynamic Memory Network
   - Model Overview
   - Encoding and Mutations
   - More Details

3. Results
   - Progress of experiments
   - Comparisons
   - Meta-parameters

4. Summary
Progress of an evolution experiment
Repeatability of results and controls

![Graph showing test accuracy over wall-clock time](image-url)
Outline

1 Introduction

2 Dynamic Memory Network
   • Model Overview
   • Encoding and Mutations
   • More Details

3 Results
   • Progress of experiments
   • Comparisons
   • Meta-parameters

4 Summary
Compared to hand-designed networks

<table>
<thead>
<tr>
<th>Study</th>
<th>Params.</th>
<th>C10+</th>
<th>C100+</th>
<th>Reachable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXOUT (Goodfellow et al., 2013)</td>
<td>–</td>
<td>90.7%</td>
<td>61.4%</td>
<td>No</td>
</tr>
<tr>
<td>Network in Network (Lin et al., 2013)</td>
<td>–</td>
<td>91.2%</td>
<td>–</td>
<td>No</td>
</tr>
<tr>
<td>All-CNN (Springenberg et al., 2014)</td>
<td>1.3 M</td>
<td>92.8%</td>
<td>66.3%</td>
<td>Yes</td>
</tr>
<tr>
<td>Deeply Supervised (Lee et al., 2015)</td>
<td>–</td>
<td>92.0%</td>
<td>65.4%</td>
<td>No</td>
</tr>
<tr>
<td>Highway (Srivastava et al., 2015)</td>
<td>2.3 M</td>
<td>92.3%</td>
<td>67.6%</td>
<td>No</td>
</tr>
<tr>
<td>ResNet (He et al., 2016)</td>
<td>1.7 M</td>
<td>93.4%</td>
<td>72.8%</td>
<td>Yes</td>
</tr>
<tr>
<td>Evolution (Ours)</td>
<td>5.4 M</td>
<td>94.6%</td>
<td>77.0%</td>
<td>N/A</td>
</tr>
<tr>
<td>Wide ResNet 28-10 (Zagoruyko &amp; Komodakis, 2016)</td>
<td>36.5 M</td>
<td>96.0%</td>
<td>80.0%</td>
<td>Yes</td>
</tr>
<tr>
<td>Wide ResNet 40-10+d/o (Zagoruyko &amp; Komodakis, 2016)</td>
<td>50.7 M</td>
<td>96.2%</td>
<td>81.7%</td>
<td>No</td>
</tr>
<tr>
<td>DenseNet (Huang et al., 2016a)</td>
<td>25.6 M</td>
<td>96.7%</td>
<td>82.8%</td>
<td>No</td>
</tr>
</tbody>
</table>
Compared to auto-discovered networks

<table>
<thead>
<tr>
<th>Study</th>
<th>Starting Point</th>
<th>Constraints</th>
<th>Post-Processing</th>
<th>Params.</th>
<th>C10+</th>
<th>C100+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian</td>
<td>3 layers</td>
<td>Fixed architecture, no skips</td>
<td>None</td>
<td>–</td>
<td>90.5%</td>
<td>–</td>
</tr>
<tr>
<td>(Snoek et al., 2012)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q-Learning</td>
<td>–</td>
<td>Discrete params., max. num. layers, no skips</td>
<td>TUNE, RETRAIN</td>
<td>11.2 M</td>
<td>93.1%</td>
<td>72.9%</td>
</tr>
<tr>
<td>(Baker et al., 2016)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL (Zoph &amp; Le, 2016)</td>
<td>20 layers, 50%</td>
<td>Discrete params., exactly 20 layers</td>
<td>SMALL GRID SEARCH, RETRAIN</td>
<td>2.5 M</td>
<td>94.0%</td>
<td>–</td>
</tr>
<tr>
<td>RL (Zoph &amp; Le, 2016)</td>
<td>39 layers, 2 pool</td>
<td>Discrete params., exactly 39 layers, 2 pool layers at 13 and 26</td>
<td>ADD MORE FILTERS, SMALL GRID SEARCH, RETRAIN</td>
<td>37.0 M</td>
<td>96.4%</td>
<td>–</td>
</tr>
<tr>
<td>Evolution (Ours)</td>
<td>Single layer, zero convs.</td>
<td>Power-of-2 strides</td>
<td>None</td>
<td>5.4 M</td>
<td>94.6%</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40.4 M</td>
<td></td>
<td>77.0%</td>
</tr>
</tbody>
</table>
Outline

1. Introduction

2. Dynamic Memory Network
   - Model Overview
   - Encoding and Mutations
   - More Details

3. Results
   - Progress of experiments
   - Comparisons
     - Meta-parameters

4. Summary
Improve the method

- Large population size
- More training steps
- Increase mutation rate
- Reset all weights
Summary

- Neuro-evolution starts from trivial initial conditions and yields fully trained models
- Construct large, accurate networks for two challenging and popular image classification benchmarks
- Large search space and high computation cost