DeepViNE: Virtual Network Embedding with Deep Reinforcement Learning

M. Dolati, S. B. Hassanpour, M. Ghaderi, A. Khonsari

Presenter: Mahdi Dolati
mahdidolati@ut.ac.ir
University of Tehran and University of Calgary
Virtual Network Embedding

• Network Virtualization
  • Is one of the key technologies of future networks.
  • Allows multiple virtual networks (VN) to coexist on the same physical network (PN).
Virtual Network Embedding

- Node mapping
  - CPU demand
- Link mapping
  - Bandwidth demand

- **Request arrives**
  - Little information about future
- **Long-term objective**
Options

- Optimal
- Heuristic
- Online-algorithm-based
- Learning-based
Reinforcement Learning

- Learn from interaction -> Brings adaptability
- Curse of dimensionality

\[ Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left( r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right) \]
Reinforcement Learning

- Learn from interaction -> Brings adaptability
- Curse of dimensionality -> Use neural networks

Deep Learning

- Convolutional neural network
- With origins in image recognition area
  - Automates feature extraction
  - End-to-end learning
- Over-fitting

Architecture

- Deep Reinforcement Learning (DRL)
Two Challenging Problems

Action Space Affects learning efficiency

Image Representation Perceivable by a DNN
Algorithm Realization

- Action definition
- Image construction
- Reward Signal
- Neural network architecture
Action Definition

- Exploration and Solution
  - Move a pointer: UP, DOWN, LEFT, RIGHT (8 actions)
  - Embed (1 action)
Image

Image Construction

Encode the input as a 2D matrix
Image Construction

PN Encoding

Physical Pointer

Available CPU
Physical Pointer
Embedded Virtual Nodes
Available Bandwidth

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1</td>
<td>0000</td>
<td>0.12</td>
<td>0.2</td>
</tr>
<tr>
<td>0.14</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0</td>
<td>0000</td>
<td>0.45</td>
<td>0.5</td>
</tr>
<tr>
<td>0.47</td>
<td>0.58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>0</td>
<td>0000</td>
<td>0.78</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Image Construction

PN Encoding – CPU
Image Construction

PN Encoding – Pointer
Image Construction

PN Encoding – Embedded

Available CPU
Physical Pointer
Embedded Virtual Nodes
Available Bandwidth

<table>
<thead>
<tr>
<th></th>
<th>0.1</th>
<th>0.12</th>
<th>0.2</th>
<th>0</th>
<th>0001</th>
<th>0.23</th>
<th>0.3</th>
<th>0</th>
<th>0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.14</td>
<td>0.25</td>
<td>0.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0</td>
<td>0000</td>
<td>0.45</td>
<td>0.5</td>
<td>0</td>
<td>0100</td>
<td>0.56</td>
<td>0.6</td>
<td>0</td>
</tr>
<tr>
<td>0.47</td>
<td>0.58</td>
<td>0.69</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>0</td>
<td>0000</td>
<td>0.78</td>
<td>0.8</td>
<td>0</td>
<td>0000</td>
<td>0.89</td>
<td>0.9</td>
<td>0</td>
</tr>
</tbody>
</table>
Image Construction

VN Encoding

Embedding Status
Virtual Pointer
Virtual Node ID
CPU Demand

Bandwidth Demand

<table>
<thead>
<tr>
<th>Embedding Status</th>
<th>Virtual Pointer</th>
<th>Virtual Node ID</th>
<th>CPU Demand</th>
<th>Bandwidth Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.001</td>
<td>1</td>
<td>1</td>
<td>0.012</td>
</tr>
<tr>
<td>0.013</td>
<td></td>
<td></td>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td>0.034</td>
<td>0.100</td>
<td>0</td>
<td>1</td>
<td>0.034</td>
</tr>
<tr>
<td>0.034</td>
<td></td>
<td></td>
<td></td>
<td>0.04</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
One-hot encoding scheme for IDs

- Unique
- Easy to represent the union
  - $0001 + 0010 \rightarrow 0011$
Take advantage of channels

Use three channels: Each channel represents the image from a specific viewpoint, e.g., an RGB image has three channels to represent Red, Green, and Blue.

**Channel 2**: When a virtual node or link is embedded we replace its corresponding encoding in the VN representation with an array of appropriate number of 1’s.

**Channel 3**: When remaining capacity of a physical node or link is not enough to embed any virtual node and link we show it by an array of 0’s with appropriate length.
Reward Signal

The cumulative reward is maximized when the maximum number of VNs are embedded.

Complete embedding: 1

Embedding a single virtual node: 0.01

Violating the resource constraints: −0.01
- The episode is terminated

Moving the virtual and physical pointers: 0
- Allows the agent to explore
Neural Network Architecture

Dueling technique

- $a_1$. Move Physical Pointer UP
- $a_2$. Move Physical Pointer DOWN
- $a_3$. Move Physical Pointer LEFT
- $a_4$. Move Physical Pointer RIGHT
- $a_5$. Move Virtual Pointer UP
- $a_6$. Move Virtual Pointer DOWN
- $a_7$. Move Virtual Pointer LEFT
- $a_8$. Move Virtual Pointer RIGHT
- $a_9$. EMBED
### Evaluation

<table>
<thead>
<tr>
<th>PN has 25 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each VN has 9 nodes</td>
</tr>
<tr>
<td>Physical Resources from [50, 100]</td>
</tr>
<tr>
<td>Virtual Demand from [1, 10]</td>
</tr>
</tbody>
</table>

**Episodic Operation:**
- Each episode is a sequence of 800 iterations of action selection.
- VNs arrive sequentially
- On average 40 VNs
- End of episode: All VNs leave the system

**Tensorflow**

**Competing approaches**
- BestFit, FirstFit, Random, NeuroViNE, GRC
Evaluation

- Reward and objective are aligned
- Fast convergence
Evaluation
Evaluation

- **DeepViNE** considers the physical link and node resources simultaneously, which eventually leads to higher CPU and link utilization, but also lower blocking probability.

- **Underlying assumptions**
  - Load-balancing (GRC, Neuro-ViNE)
  - Aggressive packing
Conclusion

• We presented a DRL-based VN embedding algorithm and overcame two important challenges
  • Small action space for fast convergence
  • Image representation of problem inputs
  • We presented simulation results with several existing works

• An interesting extension of this work is to modify the reward signal to model other system objectives such as minimizing energy consumption of the physical network.