Automated Embedding Size Search in Deep Recommender Systems

Haochen Liu¹, Xiangyu Zhao¹, Chong Wang², Xiaobing Liu² and Jiliang Tang¹

1: Data Science and Engineering Lab, Michigan State University
2: Applied Machine Learning, Bytedance

Presenter: Haochen Liu

July, 2020
Recommender Systems

• Goal
  • Predict a user’s preference for an item

• Applications
  - E-commerce websites
  - Music/video platforms
  - Social media
Deep Recommender Systems

- Advantages
  - ✓ Learning the feature representations of users and items
  - ✓ Modeling the non-linear relationships between users and items

- Architecture
  - Representation layer
  - Inference layer
Motivations

• Dynamically search the embedding sizes for different users and items

• Why
  • Users and items have highly varied frequencies
  • The frequency of a user/item changes dynamically
  • More efficient in memory
Introduction

• Challenges
  • Automatic search
  • Non-differentiability
  • Soft selection

• Our Design
  • Embedding Size Adjustment Policy Network (ESAPN)
  • Reinforcement learning (RL)
  • Hard selection
Overview

Policy Network

Embedding Table

Reward Function

Recommendation

User $u_k$ → $emb\ size^{(u_k)}$ → $emb\ size^{(i_k)}$ → Prediction → Recommender

Item $i_k$
Deep Recommendation Model

- Candidate embedding sizes

\[ D = \{d_1, d_2, \ldots, d_n\} \quad \text{with} \quad d_1 < d_2 < \cdots < d_n \]

- Linear transformations

\[
\begin{align*}
e_2 &= W_{1\rightarrow 2} e_1 + b_{1\rightarrow 2} \\
e_3 &= W_{2\rightarrow 3} e_2 + b_{2\rightarrow 3} \\
&\quad \vdots \\
e_n &= W_{n-1\rightarrow n} e_{n-1} + b_{n-1\rightarrow n}
\end{align*}
\]
Policy Network

- **Environment**
  The deep recommendation model

- **State**
  \[ s = (f, e) \]
  
  \[ f : \text{frequency} \quad e : \text{current embedding size} \]
Policy Network

- Action
  - Enlarge
  - Unchange

- Reward
  \[ L^{(u)} = (L_1^{(u)}, \ldots, L_T^{(u)}) \]
  \[ R^{(u)} = \frac{1}{T} \sum_{t=1}^{T} L_t^{(u)} - L \]
  \[ L^{(i)} = (L_1^{(i)}, \ldots, L_T^{(i)}) \]
  \[ R^{(i)} = \frac{1}{T} \sum_{t=1}^{T} L_t^{(i)} - L \]
Optimization

Policy network

1. Sample a validation batch
2. Sample actions
3. Temporarily adjust embedding sizes
4. Calculate rewards
5. Update policy network

Recommendation Model

1. Obtain a training batch
2. Select actions
3. Permanently adjust embedding sizes
4. Calculate losses
5. Update recommendation model
Experimental Settings

• Datasets
  • MovieLens 20M Dataset (ml-20m)
  • MovieLens Latest Dataset (ml-latest)

• Candidate Embedding Sizes

\[ D = \{2, 4, 8, 16, 64, 128\} \]
### Experimental Results

- **Baselines**
  - FIXED
  - DARTS
  - AutoEmb

- **Tasks**
  - Binary Classification
  - Multiclass Classification

<table>
<thead>
<tr>
<th>Models</th>
<th>ml-20m Binary</th>
<th>ml-latest Binary</th>
<th>ml-20m Multiclass</th>
<th>ml-latest Multiclass</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>MSE Loss</td>
<td>Accuracy (%)</td>
<td>CE Loss</td>
</tr>
<tr>
<td>FIXED</td>
<td>72.13</td>
<td>0.1845</td>
<td>49.45</td>
<td>1.1517</td>
</tr>
<tr>
<td>DARTS</td>
<td>72.18</td>
<td>0.1836</td>
<td>49.85</td>
<td>1.1423</td>
</tr>
<tr>
<td>AutoEmb</td>
<td>72.27</td>
<td>0.1828</td>
<td>49.94</td>
<td>1.1399</td>
</tr>
<tr>
<td>ESAPN</td>
<td>72.98</td>
<td>0.1785</td>
<td>51.10</td>
<td>1.1126</td>
</tr>
<tr>
<td>FIXED</td>
<td>72.13</td>
<td>0.1845</td>
<td>50.01</td>
<td>1.1414</td>
</tr>
<tr>
<td>DARTS</td>
<td>72.22</td>
<td>0.1834</td>
<td>50.46</td>
<td>1.1304</td>
</tr>
<tr>
<td>AutoEmb</td>
<td>72.36</td>
<td>0.1823</td>
<td>50.47</td>
<td>1.1311</td>
</tr>
<tr>
<td>ESAPN</td>
<td>72.88</td>
<td>0.1790</td>
<td>51.11</td>
<td>1.1147</td>
</tr>
</tbody>
</table>
Experimental Results

• Performance Comparison with Frequency
Experimental Results

• Performance Comparison with Frequency

• Memory Consumption Comparison

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>64</th>
<th>128</th>
<th>Total Dim</th>
<th>FIXED</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>user (ml-20m)</td>
<td>1,041</td>
<td>10,854</td>
<td>18,146</td>
<td>30,836</td>
<td>22,830</td>
<td>54,786</td>
<td>9,157,770</td>
<td>17,727,104</td>
<td>51.66%</td>
</tr>
<tr>
<td>movie (ml-20m)</td>
<td>17,032</td>
<td>216</td>
<td>380</td>
<td>623</td>
<td>2,237</td>
<td>6,790</td>
<td>1,060,224</td>
<td>3,491,584</td>
<td>30.37%</td>
</tr>
<tr>
<td>user (ml-latest)</td>
<td>62,808</td>
<td>83,190</td>
<td>31,450</td>
<td>22,778</td>
<td>31,613</td>
<td>51,389</td>
<td>9,675,448</td>
<td>36,253,184</td>
<td>26.69%</td>
</tr>
<tr>
<td>movie (ml-latest)</td>
<td>45,430</td>
<td>2,089</td>
<td>2,006</td>
<td>1,678</td>
<td>1,545</td>
<td>5,350</td>
<td>925,792</td>
<td>7,436,544</td>
<td>12.45%</td>
</tr>
</tbody>
</table>
Future Works

• Incorporate other information to determine an appropriate embedding size

• Automatically design the network structure in the inference layer
Thanks!

Haochen Liu: [http://www.cse.msu.edu/~liuhaoc1/](http://www.cse.msu.edu/~liuhaoc1/)
DSE Lab: [http://dse.cse.msu.edu/](http://dse.cse.msu.edu/)