
Kirill V. Demochkin
kvdymochkin@edu.hse.ru

Supervisor: Andrey V. Savchenko
Motivation
Outline

1. Proposed approach
2. Experimental results
3. Conclusion and future work
Task Formulation

Let there be given N users: n-th user is associated with $M_n$ images $\{X_n(m)\}$, $m = 1, 2, \ldots, M_n$ of products (single product on each image) that this user has purchased or interacted with. Each product belongs to one or more of D categories. The task is to predict the relevant classes of products to a user based on a collection of images of products that user has interacted with, i.e. generated a D-dimensional vector of scores (estimates of posterior probabilities) that the corresponding category is relevant to the user.
Transfer learning
Algorithms

1. Average pooling
2. Neural Aggregation
3. Neural Aggregation + Context Gating
Average Pooling

\[ x_n = \frac{1}{M_n} \sum_{m=1}^{M_n} (x_n(m)) x_n(m) \]
Learnable Pooling

\[ x_n = \sum_{m=1}^{M_n} w(x_n(m))x_n(m) \]
Neural Aggregation Module

Input: Face image set

Output: 128-d feature

Source: https://arxiv.org/abs/1603.05474
Context Gating Module

Input: Snow, Tree, Ski
Gates x Input: 0.9x, 0.1x, 0.9x
Output:

Source: https://arxiv.org/abs/1706.06905
Proposed Approach

User Image → MobileNet → Neural Aggregation

User Image → MobileNet → Context Gating

... → MobileNet

User Image → MobileNet

FC with ReLU Activation → D-dimensional vector of class probabilities

FC with Sigmoid Activation
Pretrained MobileNet v1 ➔ Transfer learning ➔ Neural aggregation training
Dataset

547700 entries
66519 unique users
28237 unique items
1000 categories

“Amazon Products”
Dependence of F1-measure on the number of recommendations $k$
Recall@k and Precision@k for different aggregation strategies

<table>
<thead>
<tr>
<th>k</th>
<th>Aggregation</th>
<th>Recall @k</th>
<th>Precision @k</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Average</td>
<td>0.704867</td>
<td>0.749925</td>
</tr>
<tr>
<td></td>
<td>Neural Aggregation</td>
<td>0.772574</td>
<td>0.839458</td>
</tr>
<tr>
<td></td>
<td>Neural Aggregation + Context Gating</td>
<td>0.792203</td>
<td>0.922438</td>
</tr>
<tr>
<td>10</td>
<td>Average</td>
<td>0.797340</td>
<td>0.595867</td>
</tr>
<tr>
<td></td>
<td>Neural Aggregation</td>
<td>0.901716</td>
<td>0.710123</td>
</tr>
<tr>
<td></td>
<td>Neural Aggregation + Context Gating</td>
<td>0.91846</td>
<td>0.881151</td>
</tr>
<tr>
<td>15</td>
<td>Average</td>
<td>0.815469</td>
<td>0.561431</td>
</tr>
<tr>
<td></td>
<td>Neural Aggregation</td>
<td>0.932418</td>
<td>0.710123</td>
</tr>
<tr>
<td></td>
<td>Neural Aggregation + Context Gating</td>
<td>0.942565</td>
<td>0.868210</td>
</tr>
<tr>
<td>20</td>
<td>Average</td>
<td>0.820141</td>
<td>0.553453</td>
</tr>
<tr>
<td></td>
<td>Neural Aggregation</td>
<td>0.943513</td>
<td>0.636783</td>
</tr>
<tr>
<td></td>
<td>Neural Aggregation + Context Gating</td>
<td>0.947498</td>
<td>0.864384</td>
</tr>
</tbody>
</table>
Conclusions:

1. The neural aggregation with context gating outperforms the naive averaging method by up to 34%.

Future work:

1. Mobile recommender system
2. Comparing the performance of the proposed approach with traditional recommender system
Thank you!