ADER: Adaptively Distilled Exemplar Replay Towards Continual Learning for Session-based Recommendation

Presentation by Alec Pauli
Session-based Recommendation Systems
Recommender Systems

YouTube:
• 168,055,344,000 hours of video
• Enough to watch for more than 200’000 lives

Amazon:
• 350 million products
• Enough to buy every day around 12000 products

* Based on a life expectancy of 80 years
We have to cool them down to ~273°C to be able to capture anything at all.
Amazon

514 Billions * 1% * 10% = 0.5 Billion
Recommender Systems

Content-based
Future interactions are predicted based on the characteristics of a specific item.

Collaborative filtering
Finds similar actions taken by other users.
Session based recommender Systems

**No login required**
As a session is normally a short continuous interaction with the service already are able to predict without a large history

**High quality datasets**
With the rising popularity of new social networks and content platforms new datasets were open-sourced/collected

**Privacy regulations**
For example, GDPR makes storing large datasets more complicated. Even more important is the new E-Privacy regulation

**Deep Learning advances**
Naturally fits into paradigms of Deep learning. Thus advances in Deep Learning such as RNNs can be directly applied
Recommender Systems are dynamic systems.
Catastrophic forgetting

Learning task 1

Learning task 2

Learning task f
Catastrophic cartoon solution

Learning task 2
Ader

How to split available space

Which datapoints
Sample Dataset

Stock items

Dataset
Split on the available space
Selection of Exemplars of the 🎈 class
Selection of Exemplars of the 🎍 class
Open questions

- How do we compute the loss for training?
- Which type of network is used?
KD - Loss
CE - Loss
Combination
Model – SASRec – Previous models

Markov Chain

• Good at short term realtionships

Recurrent Neuronal Network

Perform best with long term semantics
Model – SASRec – High level idea

SASRec
Tries to combine the strengths of MC and RNN’s via an attention mechanism
Adaptively Distilled Exemplar Replay

\[ S = D_{t-1} \]

Recommender Network \( f(w_0) \)

Predict action

\[ S = D_t \cup E_{t-1} \]

Recommender Network \( f(w_1) \)

Predict action

\[ S = D_{t+1} \cup E_t \]

Recommender Network \( f(w_2) \)

Predict action
Experiments

Recall@k:

\[
\text{Element 1} \quad \text{Element 2} \quad \text{Element 3} \\
\text{Element 4} \quad \text{Element 5} \\
\ldots \\
\ldots \\
\ldots \\
\frac{\text{#in first k}}{\text{#total}}
\]

MRR@k:

\[
\text{Element 1} \\
\text{Element 2} \\
\text{Element 3} \\
\text{Element 4} \\
\text{Element 5} \\
\frac{1}{\text{#total}} \sum_{i=1}^{\text{#total}} \frac{1}{\text{corresp. m}}
\]
Comparison against Ader

- Finetune
- Dropout
- Joint
- EWC
Dropout

Figure 1. Networks trained with dropout tend to forget at a slower rate. The lines represent the evolution of the validation accuracy of the first task, as networks learn new tasks.
EWC model
Diginetica

Click stream data of e-commerce site

5 Months
YouChoose

Click stream data of different e-commerce site

6 Months

Less dynamic
YouChoose Diginetica

For YouChoose the update interval is daily and for Diginetica weekly

Still Youchoose has around 4 times more Actions in each interval
Results of the Diginetica dataset

Performance on DIGINETICA with 30k exemplars

Recall@20

Recall@10

- Finetune
- Dropout
- EWC
- Joint
- Ader
Results of the Diginetica dataset

Performance on DIGINETICA with 30k exemplars
Results of the YOOCHOOSE dataset

Performance on Youchoose with 30k exemplars

Recall@20

Recall@10
Results of the YOOCHOOSE dataset

Performance on YouChoose with 30k exemplars

- MRR@20
- MRR@10
Performance over the weeks

DIGINETICA

YOOCHOOSE

Recall@20(%) vs week

MRR@20(%) vs day

- Finetune
- Dropout
- EWC
- Joint
- ADER
Effect of exemplar size
Ablation study

random

loss

herding

equal

fix

Ader
Ablation study
Personal opinion

How to use space
Generally good written
Better than a upper baseline
Herding technique
Exemplar sizes

**Diginetica**
Around 50’000 samples per iteration

**YouChoose**
Around 200’000 samples per iteration
Main sources:

- https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421
- https://www.google.com/search?client=safari&rls=en&q=session+based+recommender+springer&ie=UTF-8&oe=UTF-8
- https://github.com/kang205/SASRec
Backup Slides / Support for discussion
CE Loss

Cross entropy according to current data

\[ L_{CE}(\theta_t) = -\frac{1}{|D_t|} \sum_{(x, y) \in D_t} \sum_{i=1}^{|I_t|} \delta_{i=y} \cdot \log(p_i) \]
KD Loss

\[ L_{KD}(\theta_t) = -\frac{1}{|E_{t-1}|} \sum_{(x,y)\in E_{t-1}} \sum_{i=1}^{I_{t-1}} \hat{p}_i \cdot \log(p_i), \]
Total Loss

\[ L_{ADER} = L_{CE} + \lambda_t \cdot L_{KD}, \quad \lambda_t = \lambda_{base} \sqrt{\frac{|I_{t-1}|}{|I_t|} \cdot \frac{|E_{t-1}|}{|D_t|}} \]

Small if either a lot of new actions are available or a lot of new data
Algorithm for choosing exemplars

Pseudoalgorithm for selection in loop t:

For all items y:

\[ P_y = \text{elements with the same } y \]

\[ \mu = \text{Average of the } y \text{ according to the output of the model} \]

for k from 1 to number of elements to store for this action

\[ \argmin_{x \in P_y} \left| \mu - \frac{1}{k} (\phi(x) + \sum_{j=1}^{k-1} \phi(x_j)) \right| \]

Use the union of all elements chosen
Important training parameters

- SASRec used 150 hidden units and 2 stacked self-attention blocks
- Batch size is 256 for Diginetica and 512 for YOOCHOOSE
- The Adam Optimizer was used with a learning rate of 5e-4
- Train default was 100 epochs that were lowered if Recall@20 didn’t improve for 5 epochs