Recommendations Systems

Final Project Report - Session-based recommendations with recurrent neural networks

by
Gil Zeevi, 203909320
Gil Ayache, 200358612
Group 25

Links

- Paper: Article Github
- Our Work - Gil & Gil git repo
- References:
  - Phạm Thanh Hùng (hungthanhpham94) repo
  - Younghun Song (yhs-968) repo

Datasets

- RecSys Challenge 2015

1. Introduction

- What is the main objective of the paper, what are they trying to solve?

  The problem of having to base recommendations only on short session-based data instead of long user histories (as in the case of Netflix). In this situation the frequently popular matrix factorization approaches become pretty unscalable, as we could witness 'with our bare hands' in this project where applying matrix factorization technique with BPR loss scaled awfully and didnt yield as great results as other baselines as ITEM-KNN and Session popularity model.

  This problem is usually overcome in practice by resorting to item-to-item recommendations, i.e. recommending similar items. The paper argues that by modeling the whole session, more accurate recommendations can be provided.

- Evaluation - how are you going to evaluate performance?

  We will use the same evaluation metrics as the paper did, both for baselines and GRU's, with another small, but important, addition:

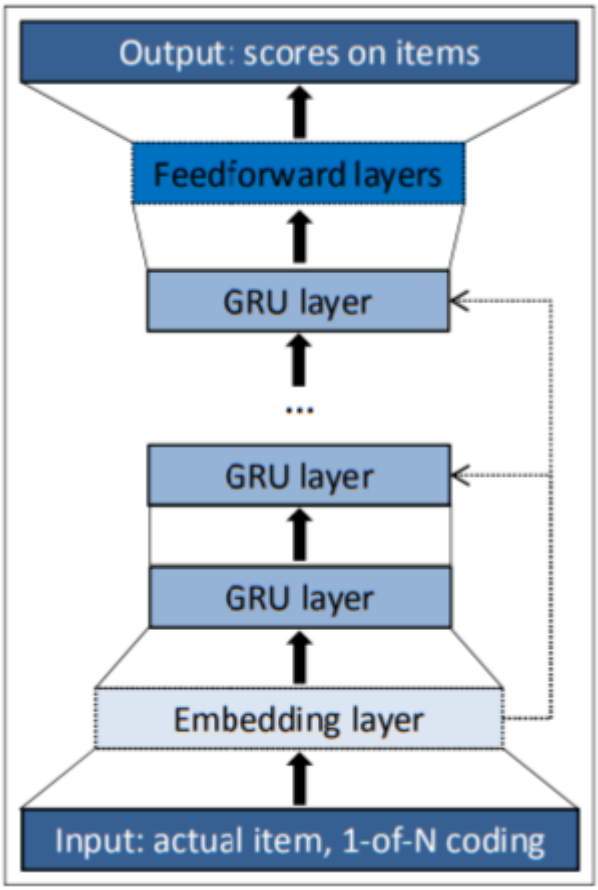
  1. MRR@20 - The inverse of harmonic mean, which indicates the quality of the recommender system where it gives a score to in which positing the first relevant item occured.
  2. Recall@20 - can be simplfied as a Measurement of success in recommending. the fraction of how many item were actually correctly predicted.
  3. Time -We will add a Training Time feature to each model in-order to compare running times.

2. Anchor paper

1. State the anchor paper: Session-based recommendations with recurrent neural networks
2. Provide a short summary of the approach presented in the paper:

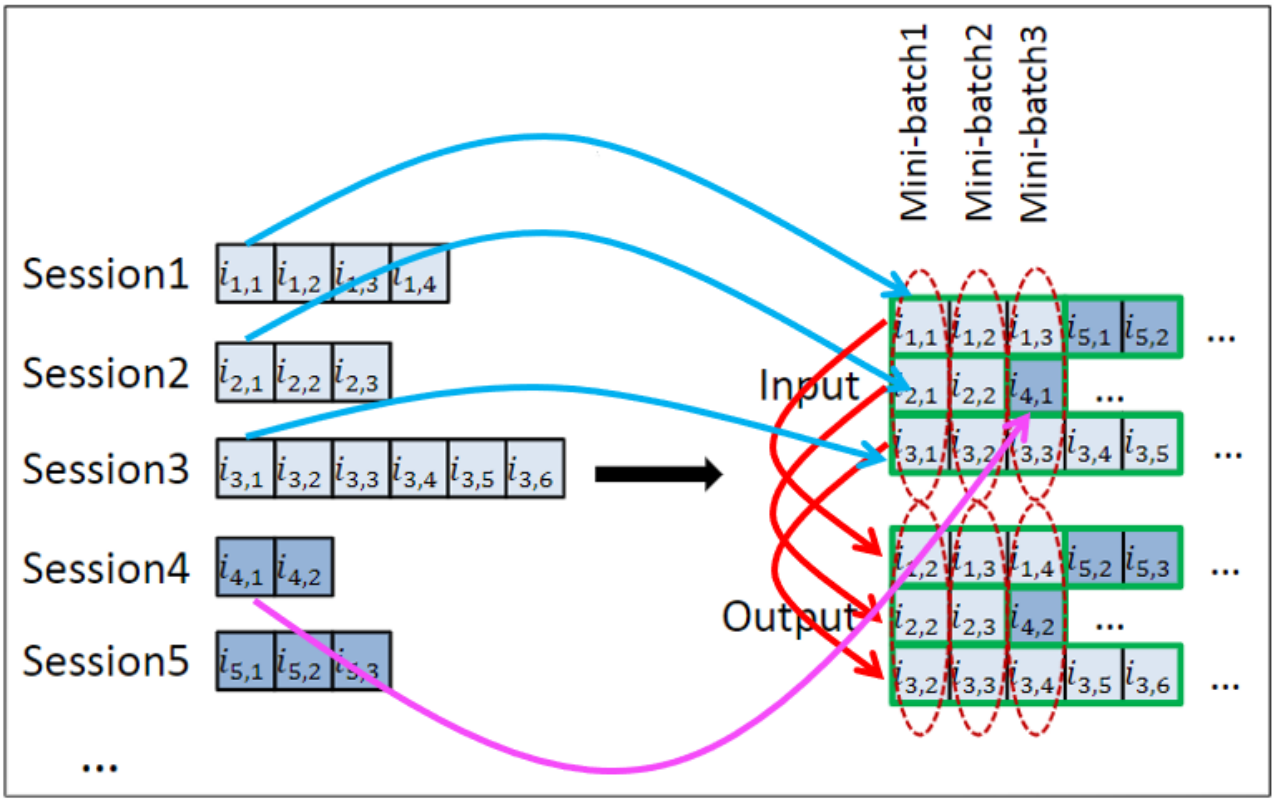
   The ancor paper using the following improvement in order to overcome the problem of using only short session and the lack of user information in the seesion:

- The model: GRU - it is a more elaborate model of an RNN unit that aims at dealing with the vanishing gradient problem. In general RNN makes predictions with data that comes in a form of a sequence.



taken from the session based with RNN paper

- Session - parallel mini batches: The dataset which is feeded inside the GRU is first being reordered by sessions. Then the first event of the first X sessions, used to form an input of the first mini-batch which will feed the GRU as input. then, the second mini batch is formed from the second event of the active sessions and so on. if the session ends, the next available session is put in its place.watch the following image which demonstrates the process:



[taken from the session based with RNN paper]

: GRU - it is a more elaborate model of an RNN unit that aims at dealing with the vanishing gradient problem. In general RNN makes predictions with data that comes in a form of a sequence.

- The loss functions: The paper use **pairwise ranking** loss instead of pointwise ranking loss, they did test the pointwise loss on cross-entropy that were unstable even with regularization.

1. BPR - it optimizes a pairwise ranking loss, using Stochastic Gradient Descent.
   To apply this method on sessions-based problems, the current state of the session is modeled as the average of the feature vectors of the items that have occurred in it so far. the similarites of the feature vectors between a recommendable item and the items of the session so far are being averaged. The loss which is being optimized is denoted as the following:

$$L_s = -\frac{1}{N_s} \cdot \sum_{j=1}^{N_s} log\left(\sigma\left(r_{s,i} - r_{s,j}\right)\right)$$

N_s $-$ $Sample\ Size$
$r_{s,i}$ $-$ $Score\ on\ item\ i\ (or\ negative\ sampling\ j\ )\ at\ the\ given\ point\ of\ session$
σ $-$ $Sigmoid\ Function\ \frac{1}{1+e^{-x}}$

2. TOP1 - The first part aims to push the target score above the score of the samples, while the second part lowers the score of negative samples towards zero. The latter acts as a regularizer, but instead of constraining the model weights directly, it penalizes high scores on the negative examples. Since all items act as a negative score in one training example or another, it generally pushes the scores down.

$$L_{TOP1} = \frac{1}{N_s} \cdot \sum_{j=1}^{N_s} \sigma\left(r_{s,j} - r_{s,i}\right) + \sigma(r_{s,j}^2)$$

- Baselines results:

Table 1: Recall@20 and MRR@20 using the baseline methods

Baseline	RSC15		VIDEO	
	Recall@20	MRR@20	Recall@20	MRR@20
POP	0.0050	0.0012	0.0499	0.0117
S-POP	0.2672	0.1775	0.1301	0.0863
Item-KNN	0.5065	0.2048	0.5508	0.3381
BPR-MF	0.2574	0.0618	0.0692	0.0374

- Best evaluation metrics:

Table 3: Recall@20 and MRR@20 for different types of a single layer of GRU, compared to the best baseline (item-KNN). Best results per dataset are highlighted.

Loss / #Units	RSC15		VIDEO	
	Recall@20	MRR@20	Recall@20	MRR@20
TOP1 100	0.5853 (+15.55%)	0.2305 (+12.58%)	0.6141 (+11.50%)	0.3511 (+3.84%)
BPR 100	0.6069 (+19.82%)	0.2407 (+17.54%)	0.5999 (+8.92%)	0.3260 (-3.56%)
Cross-entropy 100	0.6074 (+19.91%)	0.2430 (+18.65%)	0.6372 (+15.69%)	0.3720 (+10.04%)
TOP1 1000	0.6206 (+22.53%)	**0.2693 (+31.49%)**	**0.6624 (+20.27%)**	**0.3891 (+15.08%)**
BPR 1000	**0.6322 (+24.82%)**	0.2467 (+20.47%)	0.6311 (+14.58%)	0.3136 (-7.23%)
Cross-entropy 1000	0.5777 (+14.06%)	0.2153 (+5.16%)	–	–

3. Innovative part

- Choosing a smaller dataset consisting of 4.5 days but still outperforming the baselines with pretty low training time and siginficant higher score in RECALL@20

- Showing training time comparisons between all models.
- Presenting the Validation loss grpahs which were not presented at the original paper. its proves how all the GRU candidates doesnt overfit.
- We questioned the paper's statement which claimed that GRU model with final activation of Tanh and Adagrad optimizer performs the best. we presented all sort of different model which performed almost as the paper's model, but with much less training time. our chosen model even outperformed the paper's model in terms of RECALL@20 and was 3 times faster in terms of training time.
- Interactive notebook to see the affect of change in hyperparameters where all related work in github consists only python/Shell files as far as we've encourted.

4. Summary of work and conclusion

- We worked at a different,smaller, scale as opposed to the GRU4REC paper, we did that in order to 'scale' the RNN network to our available limited resources (colab,local gpu & kaggle). this fact helped us examine a whole bunch of different GRU's with different optimizers and different losses, even though we couldnt afford applying the complete dataset.
- We see that the basic Popularity model fails big time at big scales due to multiple items and too many relevant options.
- We see how a slight change in POP model into per-session popularity model can still be really strong baseline model. Nevertheless we know from the original paper that as the dataset expands - popularity models performance will naturally go down
- Most of our best picked GRU models (performed more than 500 parameters inspections) yielded around the same RECALL@20. We thus conclude that as opposed to what's presented in the paper, each GRU model can outperform all the baseline models presented.after careful hyperparameters tuning of course.
- The chosen GRU model in paper had worse training time performance than our pick, even though it had slightly better MRR@20. The RECALL@20 was also in our favor and has beaten the paper model, at our chosen scale of course. Actually having a MRR@20 score of 0.22 or 0.24 doesnt really make significant difference in our opinion.
- We managed to significantly outperform, in term of RECALL@20, the baselines with GRU. futhermore, we know that as the dataset grows bigger, expanding the GRU units even will increase its performance.
- We were surprised by the poor performance of classical MF. it performed poorly on the dataset, compared to the cpu time it took to train. BPR-MF demand many iterations in order to perform well, and at growing and expanding datasets, it becomes not scalable.

6. Future work and inspections:

If we had more time, we would definitely want to inspect multi features session based recommendations. it can be in adding context or other features to see how the GRU performs. it is actually also a work of Balázs Hidasi (the author of the original inspected paper).