ECIR2019 Cologne

Concept to Code™
Neural Networks for Sequence Learning

Omprakash Sonie
Flipkart
omprakash.s@flipkart.com

Muthusamy Chelliah
Flipkart
muthusamy.c@flipkart.com

Surender Kumar
Flipkart
surender.k@flipkart.com

Bidyut Kr. Patra
National Institute of Technology
patrabk@nitrkl.ac.in

Session Based Recommendation

Why Session based Recommender System?

- For common e-commerce company more than half (57.06%) of all sessions are non-logged users.
- No past interactions of the users are known.
- Most sessions are window-shopping ones, where users are only looking for the product availability, price, and information.
- Only 2.53% of all sessions converts to transaction.

Problem Statement of SRS

Session Based Recommender System (SRS)

The task of SRS is to predict what a user would like to click next when her current (click) session is given.

- Let $x = [x_1, x_2, \ldots, x_{n-1}, x_n]$ be a (click) session of a user, where $x_i \in \mathcal{I}$ is an item clicked by the user at i^{th} action. (Given)
- Build a model $\mathcal{M}$ which produces a ranking list $y = [y_1, y_2, \ldots, y_m]$ over all the next items that can occur in the next click. (Prediction)

Session-Based Recommendation with Recurrent Neural Networks [ICLR, 2016], [1]

GRU-based RNN:

$$\boxed{\begin{array}{l} h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t,\ z_t = \sigma(W_z[h_{t-1,x_t}] + b_z) \\ \tilde{h}_t = \tanh(W[r_t * h_{t-1}, x_t] + b),\ r_t = \sigma(W_r.[h_{t-1}, x_t] + b_r) \end{array}}$$

- GRU-based RNN is used.
- Input: Session of each user in form of parallel Mini-batch. (Represented by One-Hot encoding).
- Output: Preferences of each item to be appeared next.
- Network Description: Multiple GRU Layers and one Feedforward Layer before the output.

[1] B. Hidasi, A.Karatzoglou, L. Baltrunas and D. Tikk

Significant Contributions

1. RNN is exploited in SRS.
2. A new loss function TOP1 is introduced.
3. Make path for other researchers to explore RNN in SRS.

Session-Based Recommendation with Recurrent Neural Networks [ICLR, 2016]

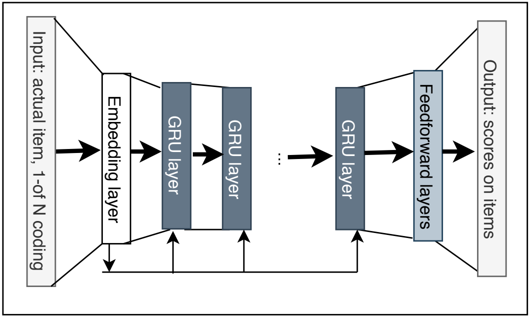


Figure: General Architecture of the GRU-based SRS [Hidasi]

Session-Based Recommendation with RNN (GRU4Rec, 2016)

- How to feed sessions of users to the network ?

Session-Based Recommendation with RNN (GRU4Rec, 2016)

- How to feed sessions of users to the network ?
- Session Parallel Mini-Batches:

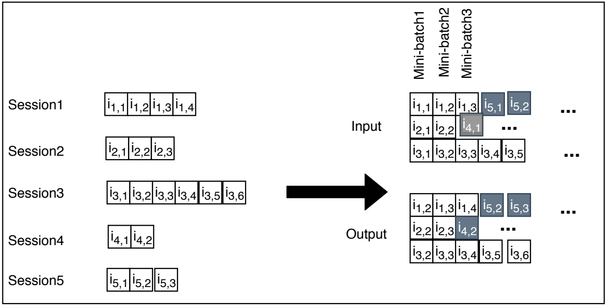


Figure: Parallel Mini-Batch in GRU4Rec [Hidasi et al.]

Loss Function Used in GRU4Rec:

1. TOP1: Loss at a given point in a session

$$L_s = \frac{1}{N_s} \sum_{j=1}^{N_s} \sigma(\hat{r}_{s,j} - \hat{r}_{s,i}) + \sigma(\hat{r}_{s,j}^2)$$

N_s = sample size,
$\hat{r}_{s,k}$ = score of on item k at a given point in the session s
i = desired item (next item in the session)

Few Observations

- GRU-based RNN has better performance than LSTM.

Few Observations

- GRU-based RNN has better performance than LSTM.
- Additional Feed-forward is not worthy.

Few Observations

- GRU-based RNN has better performance than LSTM.
- Additional Feed-forward is not worthy.
- Item embedding is found to be not working. (?)

Few Observations

- GRU-based RNN has better performance than LSTM.
- Additional Feed-forward is not worthy.
- Item embedding is found to be not working. (?)
- Single Layer GRU is actually found to be superior.

Few Observations

- GRU-based RNN has better performance than LSTM.
- Additional Feed-forward is not worthy.
- Item embedding is found to be not working. (?)
- Single Layer GRU is actually found to be superior.
    - Unknown (Open)

Improved Recurrent Neural Networks for Session-based Recommendation [DLRS (RecSys), 2016][2]

- GRU is used in RNN Unit.
- Network Description: Multiple GRU Layers and one Feed-forward Layer before the output.

Significant Contributions

1. Each session of a user is feed separately. (No p-Mini-batch)
2. Data Augmentation via sequence pre-processing.
3. Temporal Adaptation during Training phase.
4. Trained via Back propagation Through-Time(BPTT) using cross-entropy loss.
5. Output is embedded to reduce the parameters of fully connected Layer.

[2] Yong Tan, X.Xu and Y.Liu

Generic Structure of the RNN Network [Tan, 2016]

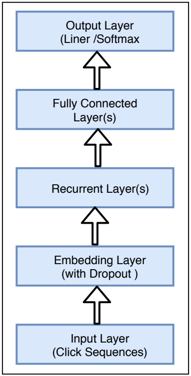


Figure: Generic Structure of the RNN network [Tan, 2016]

Input Pre-processing for Training

- Let $[x_1, x_2, \ldots, x_{n-1}, x_n]$ be a (click) session of a user. Labels of each sub sequence are generated as follows. $([x_1], V(x_2)), ([x_1, x_2], V(x_3)), \ldots, ([x_1, x_2, \ldots, x_{n-1}], V(x_n))$

Input Pre-processing for Training

- Embedding dropout to avoid over-fitting

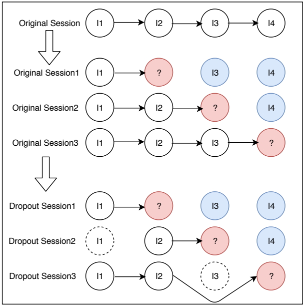


Figure: Pre-processing Step in RNN network [Tan, 2016]

Training Strategy

1. Temporal Adaptation :
   - Train on entire data-set.
   - Tune it using only recent subset of the data.
2. Training using Privileged Information.
   Let $x = [x_1, x_2, \ldots, x_r, x_{r+1}, \ldots, x_{n-1}, x_n]$ be a session.
   - Privileged Information: $[x_n, x_{n-1}, \ldots, x_{r+2}]$ with label x_{r+1}.
   - Build a model using this privileged sequence.
   - Tune the original model over entire sequence x.

Conclusion

- This approach adopted various techniques applied in other fields such as images, trajectory analysis.

Conclusion

- This approach adopted various techniques applied in other fields such as images, trajectory analysis.
- Single Layer GRU is actually found to be superior.
- Embedding in output layer can be explored further.

Modelling Contextual Information in Session-Aware Recommender Systems with NN [RecSys 2016][3]

- Network Description: RNN Layres, followed by Dropout Layers and FF Layers.
- GRU/LSTM of RNN.

Significant Contributions

- Each session $x = [x_1, x_2, \ldots, x_n]$ is represented as an ordered set of events. Each event x_i is described by a set of contextual information.
- Input: Event vector and Item vector.
- User session is modelled using Matrix Factorization.
- RNN is used to capture dependency between events in session.

[3] B.Twardowski

Architecture of Proposed NN Layer

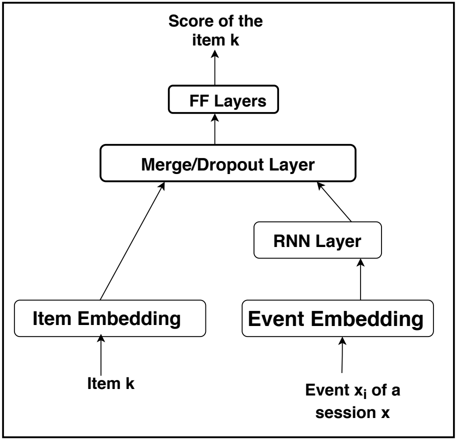


Figure: Architecture of NN Layer [Recsys 2016]

Conclusion

- Single Layer of GRU is used.
- Embedding is not working if dimension is low.
- RNN can be used for capturing long-term goal of a session.

Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks[4]

Drawback of GRU4REC

- Two sessions consist of same sequence of items will get same recommendations.

[4] M. Quadrana, A. Karatzoglou, B. Hidasi and P. Cremonesi.

Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks[4]

Drawback of GRU4REC

- Two sessions consist of same sequence of items will get same recommendations.
- Not Personalized.

[4] M. Quadrana, A. Karatzoglou, B. Hidasi and P. Cremonesi.

Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks[4]

Drawback of GRU4REC

- Two sessions consist of same sequence of items will get same recommendations.
- Not Personalized.

Significant Contributions

- User logged-in or user identifier present in the system.
- Two-level (Hierarchical) RNN is introduced.
  1. One is used to model a session of a user. (GRU4REC)
  2. Other is used to model sessions of a user.

[4] M. Quadrana, A. Karatzoglou, B. Hidasi and P. Cremonesi.

Personalizing Session-based Recommendations with HRNN

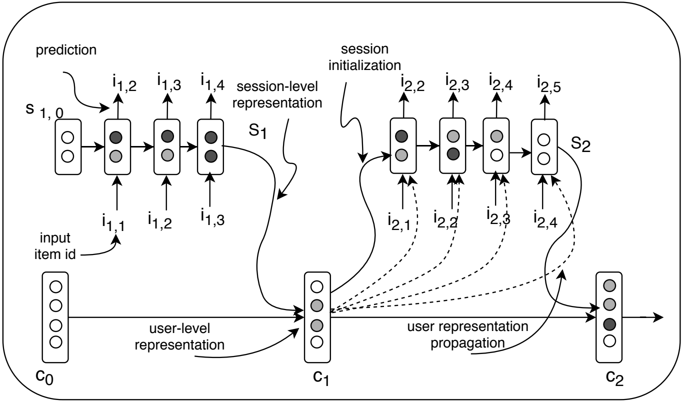


Figure: Architecture of HRNN Layer [Recsys 2017]

Personalizing Session-based Recommendations with HRNN

- Let $C^u = \{S_1^u, S_2^u, \dots, S_M^u\}$ be sessions of a user u.
- User level representation at session m be c_m.
  $c_m = GRU_{usr}(c_{m-1}, GRU_{ses}(\text{last hidden state of } m^{th}\text{session}))$
- c_m is used to initialized the first hidden sate of the session-level GRU of S_{m+1}.

Personalizing Session-based Recommendations with HRNN

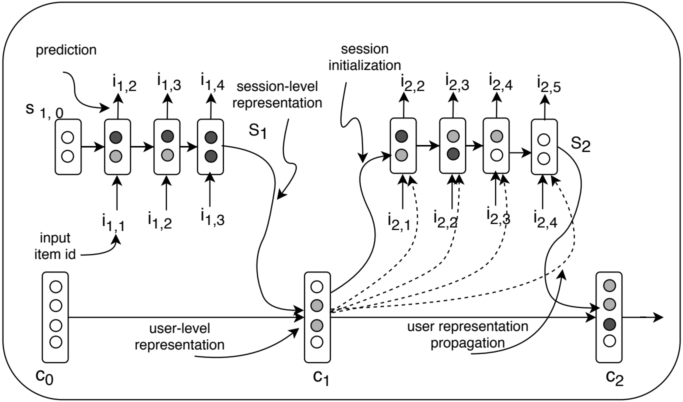


Figure: Architecture of HRNN Layer [Recsys 2017]

Conclusion:

- Personalized Recommendation to returning users.

When RNNs meet the *k*NN method for SRS[5]

Significant Contributions

- Exhaustive experiments are conducted on various datasets to verify the effectiveness of GRU4REC [Hidasi, ICLR 2016]
- A session based *k*NN is found be outperforming GRU4REC.
- A hybrid approach by combining GRU4REC with *k*NN is suggested.

[5] Dietmar Jannach. and M. Ludewig

Hybrid Approach GRU4REC and Session based KNN

- For each item x_i, a in-memory index data structure used to keep track of all sessions where the item x_i appears. Let S be the all sessions in a system.
- Collect all sessions where items of the session s appear. (Recommendation for session s.)
- Choose k nearest neighbor of the session s from this collection.
- Create a probable list of recommended items R that appear one of these k sessions.
- Find a score for each item $x \in R$ using the following equation. $score(x, s) = \sum_{t \in kNN(s)} sim(t, s) \times F_t(i)$

$$F_t(i) = \begin{cases} 1 & \text{if } i \in t, \\ 0 & \text{Otherwise.} \end{cases}$$

- Combine GRU4REC and Session based KNN in weighted manner.

Conclusion

- GRU4REC and Session based KNN are combined in weighted and cascade manner.
- Weighted hybrid approach led to the best results.
(RNN capable of capturing sequential patterns in the session.)

Session-Aware Information Embedding for E-commerce Product Recommendation [CIKM, 2017][6]

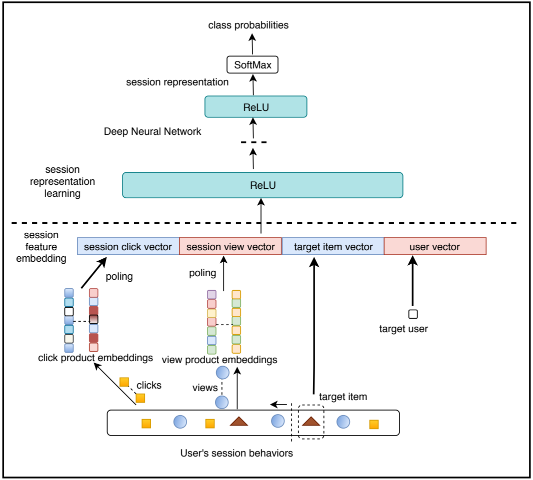
Significant Contributions

- User session includes click, view, purchase.
- Session embedding is introduced.
- List-wise deep neural network is used.

[6] C.Wu, M.Yan and L.Si

Session Representation

- Session Information Embedding Technique using FF Network.



List-wise Ranking with Session Embedding

- Match the embedded session representation with candidate items.
- Pass the candidate items to embedding layer to get a fixed length vector representation.
- Pre-trained the item embedding layer with Session Embedding.
- Fully connected Neural Network is used to match session representation and an item. score $(s, i) = s.W_{DNN}^{T}(i_e)$

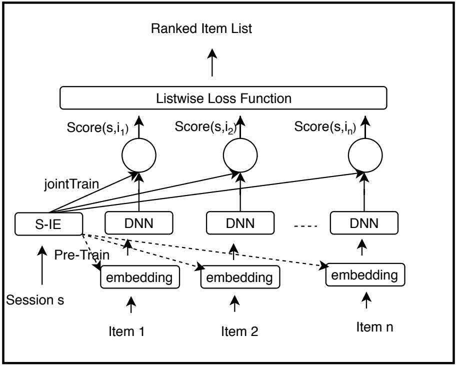


Figure: DNN with Session Embedding [CIKM, 2017]

Conclusion

- FFN is used for a compact representation of a session which includes click on items, view the items.
- Another Neural network (DNN) is used to match this session profile with candidate item profiles.
- RNN is not explored here in this method.

Neural Attentive Session-based Recommendation [CIKM 2017][7]

Main Drawback of Previous Work

- Main purpose of the session is not captured. Only sequential behavior in the session is considered.

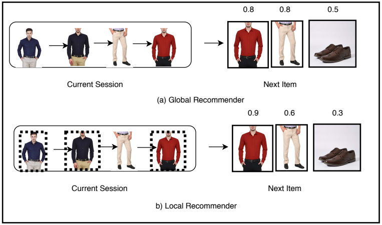


Figure: Two different recommenders [Li et. al., CIKM, 2017]

[7] J.Li, P Ren, Z.Cheb, Z, Ren, T. Lian and J.Ma

.

Significant Contributions

- Encoder-Decoder Framework in Session Based Recommender System is introduced.
- User sequential behavior and main purpose of the session are captured using two different RNN with Attention mechanism.
    1. **Global Encoder**: User sequential behavior.
    2. **Local Local Encoder**: Main purpose of the session.

General Framework of Encoder-Decoder based NARM

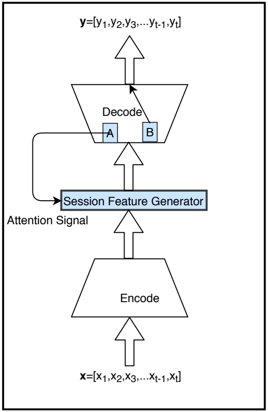


Figure: General Framework of Encoder-Decoder based NARM [Li et. al., CIKM, 2017]

Global Encoder

$$\boxed{\begin{array}{l} h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t,\ z_t = \sigma(W_z[h_{t-1,x_t}] + b_z) \\ \tilde{h}_t = \tanh(W[r_t * h_{t-1}, x_t] + b),\ r_t = \sigma(W_r.[h_{t-1}, x_t] + b_r) \end{array}}$$

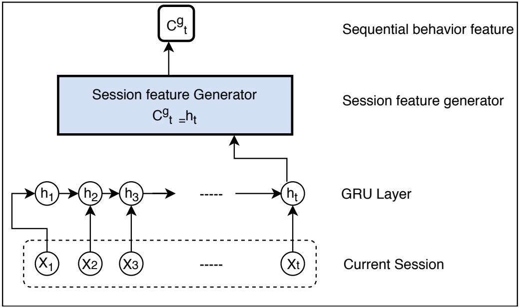


Figure: Graphical Model of Global Encoder in NARM [Li et. al., CIKM, 2017]

Local Encoder for Capturing Main Purpose

- $C_t^l = \sum_{j=1}^{t} \alpha_{tj} h_j; \;\; \alpha_{tj} = q(h_t, h_j);$
- q = similarity between h_t and h_j.

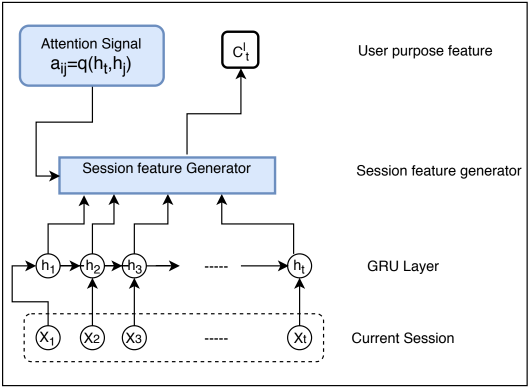


Figure: Graphical Model of Local Encoder in NARM [Li et. al., CIKM, 2017]

Neural Attentive Recommendation Machine (NARM)

- $c_t = [c_t^g; c_t^l]; \; S_i = emb_i{}^T \mathbf{B} c_t;$

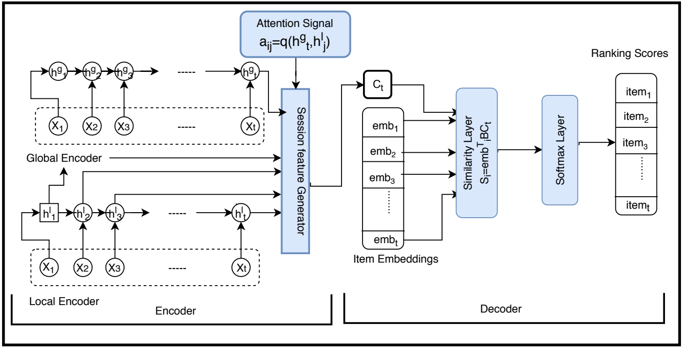


Figure: The NARM [Li et. al., CIKM, 2017]

Conclusion and Research Direction

- Attention mechanism is incorporated into RNN to capture main purpose of a session.

Conclusion and Research Direction

- Attention mechanism is incorporated into RNN to capture main purpose of a session.
- Nearest Neighbor sessions can be explored further.
- More item attributes can be incorporated.

Modeling User Session and Intent with Attention-based Encoder-Decoder Architecture [RecSys 2017][8]

Significant Contributions

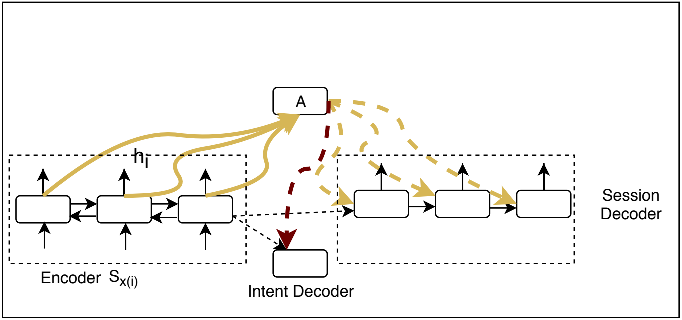
- Transition regularity from one item to next item in a session is captured using Encoder-Decoder Framework.
- Attention mechanism is incorporated to learn more expressive portion (intent estimation) of a session using bi-directional RNN.
- Two decoders are used.
  1. Session Decoder: User sequential behavior.
  2. Intent Decoder: Main purpose of the session. (purchasing intention/browsing only)

[8] P.Loyola, C. Liu and Y. Hirate

Encoder Architecture

1. Let $x = [x_1, x_2, \ldots, x_n]$ be a user session. Identify source-target transition pair within a session.
   I) Source set: $S_x = \{x_1, x_2, \ldots, x_{n-1}\}$
   Target set: $T_x = \{x_2, \ldots, x_n\}$
   From a set of item transition pair:
   $P_s = \{(x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n)\}$
2. The source S_x is feed as input to a bi-directional RNN (Encoder).
3. Encoder generates a fixed length representation of the whole source sequence.

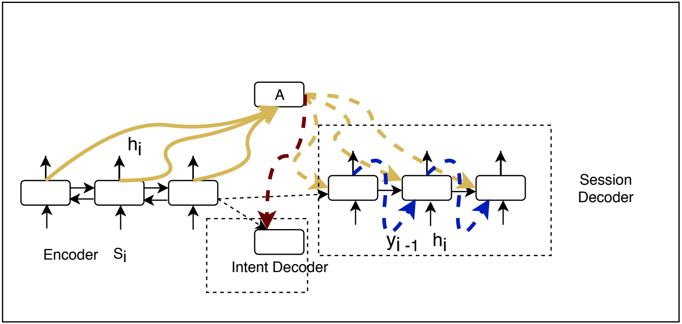
Attention-based Encoder-Decoder Architecture [RecSys 2017]



- Generates a fixed length representation $c = g(h_1, h_2, \ldots, h_t)$
- Internal State of j^{th} unit in Encoder: $h_j = [\overrightarrow{h_j}, \overleftarrow{h_j}]$.
- Internal State of i^{th} unit in Decoder: $d_i = f(d_{i-1}, c_i)$
- A computes attention factor (c_i) of the session.

  $c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j$; $\alpha_{i,j}$ is computed using FFN.

Decoder Architecture [RecSys 2017]



- Session Decoder: i^{th} unit decoder state: $d_i = f(d_{i-1}, y_{i-1}, c_i, h_i)$
- Intent Decoder: $I = f(h_{n-1}, c_i)$

Conclusion [2017]

- Bi-directional RNN acts as Encoder of a session.
- Two decoder Networks are used for finding Intention and Session characteristics.
- Parameters of Attention unit is computed using FF network.

STAMP: Short-Term Attention/Memory Priority Model for SR [KDD 2018][9]

Significant Contribution

- A model termed as STAMP is introduced to capture two aspects simultaneously.
  1. Long-term (general interest) interests.
  2. Short-term interests (current interests).

[9] Liu,Zeng,Mokhosi and Zhang

STAMP: Short-Term Attention/Memory Priority Model for SR [KDD 2018][9]

Significant Contribution

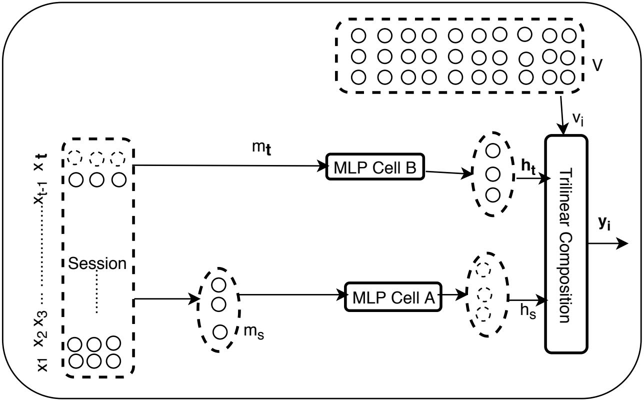
- A model termed as STAMP is introduced to capture two aspects simultaneously.
  1. Long-term (general interest) interests.
  2. Short-term interests (current interests).

STAMP Model: Let $x = \{x_1, x_2, \ldots, x_t\}$ be a session.

1. Long-term (general interest) interests are captured from all historical clicks (x) in a session including last click.
2. Short-term interests (current interests) is derived from last click (x_t).

[9] Liu,Zeng,Mokhosi and Zhang

Short-Term Memory Priority Model (STMP)

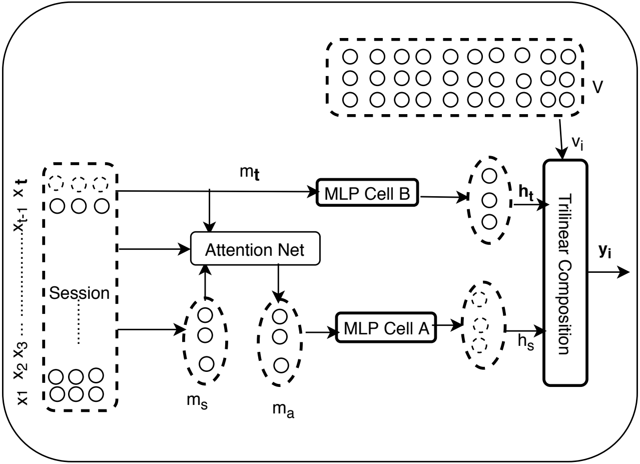


- $m_s = \frac{1}{t}\sum_{i=1}^{t} x_i$; $h_s = g(W_s m_s + b_s)$
- Score for a candidate item v_i, $\hat{z}_i = \sigma(< h_s^T.(h_t * v_i) >)$

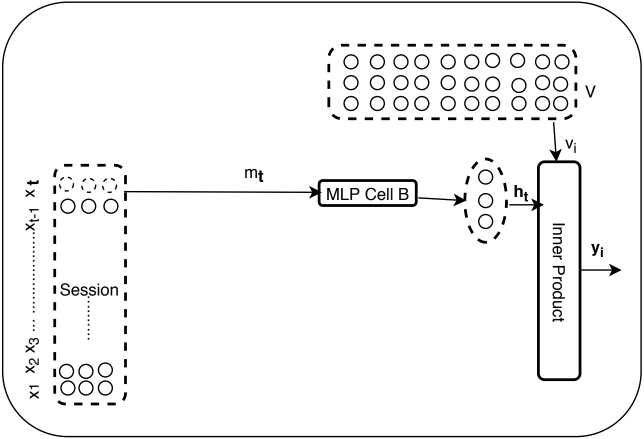
Short-Term Attention/Memory Priority (STAMP) Model

- Attention Net (a FFN) is incorporated to generate attention weights for each of the items present in the session.
- $m_a = \sum_{i=1}^{t} \alpha_i x_i$;
  $\alpha_i = F(x_i, x_t, m_s) = W_0 \sigma(W_1 x_i + W_2 x_t + W_3 m_s + b_a)$
  User's general interest [Long-term interest]

STAMP


V
v_i
m_t
MLP Cell B
h_t
Attention Net
Trilinear Composition
y_i
Session
x_1 x_2 x_3 ...... x_{t-1} x_t
MLP Cell A
h_s
m_s
m_a

Short-Term Memory Only Model (STMO)



Conclusion (Difference between NARM and STAMP)

- STAMP explicitly emphasizes current interest reflected by last click.

Micro Behaviors: A **New** Perspective in E-Commerce Recommender Systems [WSDM][10]

A session is a sequence of items clicked by a user (Macro Interaction).

Drawbacks of Previous Work

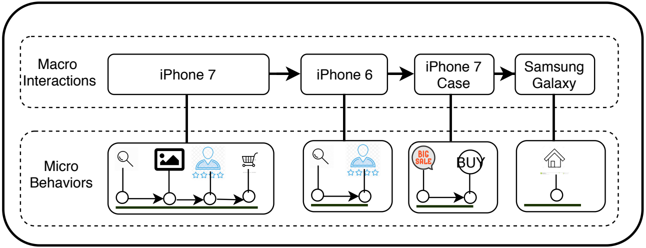
- This framework ignores sequence of tasks (reading, carting, ordering, dwell timing) performed by a user between two consecutive clicks. (Micro behaviors)

Significant Contribution

- A recommendation framework is proposed which uncovers the effects of micro behaviors which are inherently sequence in nature.
- Techniques are introduced to capture these sequence of micro behaviors in systematic way.

[10] M. Zhou, Zhuoye Ding, Jiliang Tang and Dawei Yin

Macro Interactions and Micro Behaviors



These micro behaviors can provide fine-grained understandings about users.

RIB (Recommendation framework from the mIcro Behavior perspective)

$S_u = \{x_1, x_2, \ldots, x_n\}$	where $x_i = (p_v, a_m, d_k)$
$p_v \in \mathbb{R}^V$	item vector
$a_m \in \mathbb{R}^M$	micro behavior vector
$d_k \in \mathbb{R}^K$	dwell vector

1. Input data (Session representation) is very sparse and high dimensional.
2. Micro behaviors in the sequence are correlated, then how to uncover this sequential information of micro behaviors ?
3. Different micro behaviors have distinct importance; how to capture this?

RIB (Recommendation framework from the mIcro Behavior perspective)

The RIB consists of five layers.

1. Input data (session representation) is very sparse and high dimensional. (An Embedding layer)

RIB (Recommendation framework from the mIcro Behavior perspective)

The RIB consists of five layers.

1. Input data (session representation) is very sparse and high dimensional. (An Embedding layer)
2. Micro behaviors in the sequence are correlated, then how to uncover this sequential information of micro behaviors ? (A RNN layer)

RIB (Recommendation framework from the mIcro Behavior perspective)

The RIB consists of five layers.

1. Input data (session representation) is very sparse and high dimensional. (An Embedding layer)
2. Micro behaviors in the sequence are correlated, then how to uncover this sequential information of micro behaviors ? (A RNN layer)
3. Different micro behaviors have distinct importance; how to capture this? (An Attention layer)

RIB (Recommendation framework from the mIcro Behavior perspective)

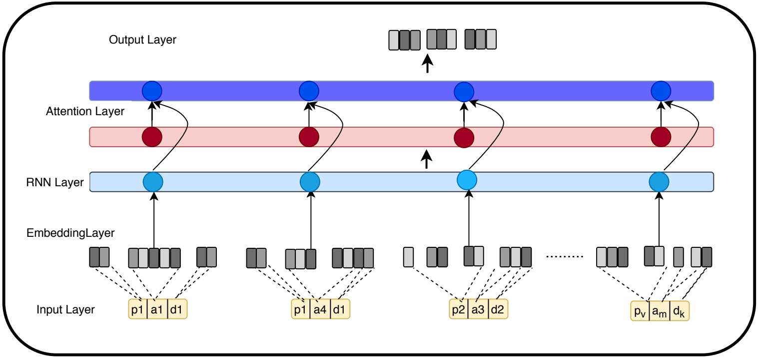


Figure: Architecture of RIB

RIB Description: Input and Embedding Layers

$S_u = \{x_1, x_2, \ldots, x_n\}$	where $x_i = (p_v, a_m, d_k)$
$p_v \in \mathbb{R}^V$	item vector
$a_m \in \mathbb{R}^M$	micro behavior vector
$d_k \in \mathbb{R}^K$	dwell vector

- Transform $x_t \in \mathbb{R}^{V+M+K}$ into low-dimensional dense vector e_t
- $e_t = \text{Concat}(W_p p_v, W_A a_m, W_d d_k)$
  $W_p \in \mathbb{R}^{d_A \times V}, d_p << V$
  $W_A \in \mathbb{R}^{d_A \times M}, d_A << M$
  $W_d \in \mathbb{R}^{d_D \times K}, d_D << K$
- $e_t \in \mathbb{R}^{d_p + d_A + d_D}$

RIB Description: RNN Layer

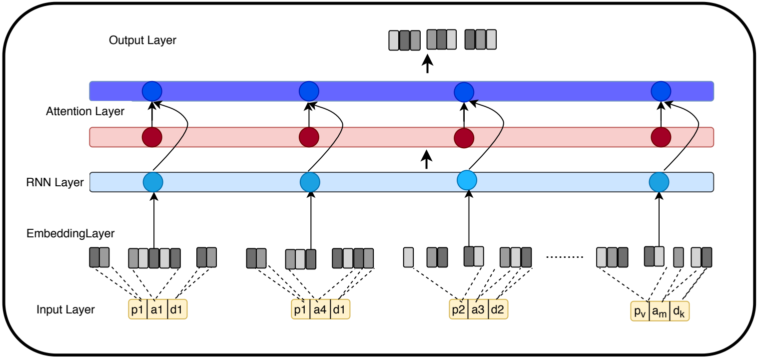


Figure: Architecture of RIB

- RNN Layer: Consists of GRU units.

RIB Description: Attention Layer

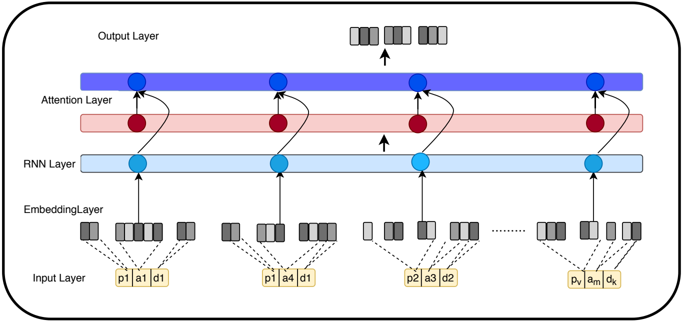


Figure: Architecture of RIB

- Attention Layer: Consists of FFT with *tanh* and σ activation functions. $\alpha_t = \sigma(\tanh(h_t))$.

RIB Description: Output Layer

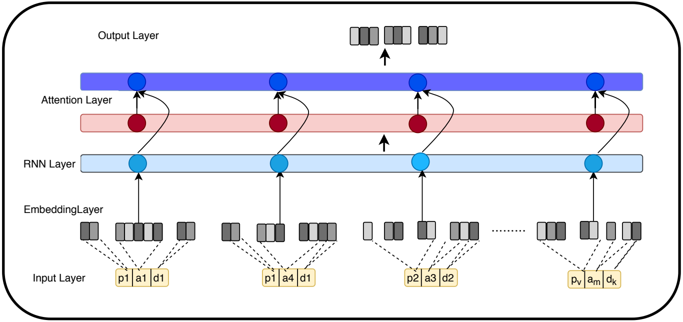


Figure: Architecture of RIB

- Final output is attention weighted pooling of RNN Layers.
- $output = \sum_{t=1}^{T} \alpha_t h_t, output \mathbb{R}^k$

Conclusion

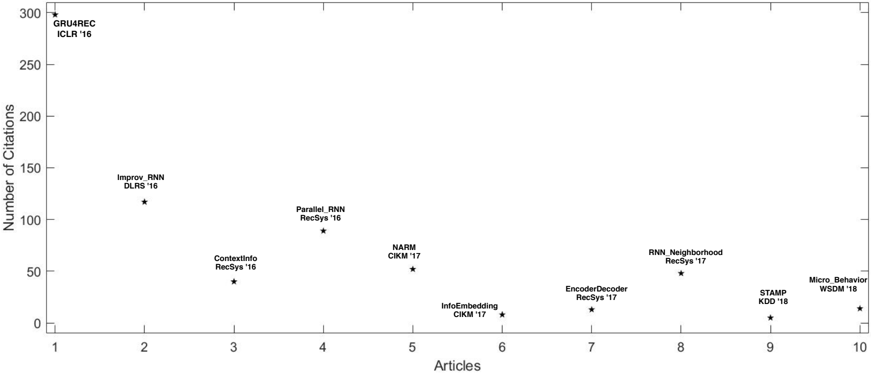


Figure: Citation of SRS articles

Future Research Directions

- Attention based RNN will dominate to uncover sequential relationship among various micro activities within a session.
- Neighbor session can play important role in improving furhter accuracy.
- RIB (Recommendation framework from the mIcro Behavior perspective)) framework can be further studied.

1. B. Hidasi, A.Karatzoglou, L. Baltrunas and D. Tikk. Session-Based Recommendation with Recurrent Neural Networks, ICLR 2016.
2. Yong Tan, X.Xu and Y.Liu. Improved Recurrent Neural Networks for Session-based Recommendations, DLRS, 2016.
3. B.Twardowski. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks, RecSys 2016.
4. B. Hidasi, M. Quadrana, A. Karatzoglou and D. Tikk. Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations, RecSys 2016.
5. M. Quadrana, A. Karatzoglou, B. Hidasi and P. Cremonesi. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks, RecSys 2017.
6. Dietmar Jannach. and M. Ludewig. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation, RecSys 2017.
7. C.Wu, M.Yan and L.Si. Session-Aware Information Embedding for E-commerce Product Recommendation, CIKM 2017.
8. J.Li, P Ren, Z.Cheb, Z, Ren, T. Lian and J.Ma. Neural Attentive Session-based Recommendation, CIKM 2017.
9. P.Loyola, C. Liu and Y. Hirate. Modeling User Session and Intent with Attention-based Encoder-Decoder Architecture, RecSys 2017.

10. Liu,Zeng,Mokhosi and Zhang. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation, KDD 2018.

11. M. Zhou, Zhuoye Ding, Jiliang Tang and Dawei Yin. Micro Behaviors: ANewPerspective in E-Commerce Recommender System, WSDM 2018.

THANK YOU