Deep Sequential Recommendation for Personalized Adaptive User Interfaces

Jongwoo Kim

0. CONTENTS

1) DRNN

Goal : Real-time contextual adaptation and assistance in the form of recommended actions that are personalized for user.

Proposed architecture

1. Using GRU, we are going to perform deep learning for user interaction patterns.

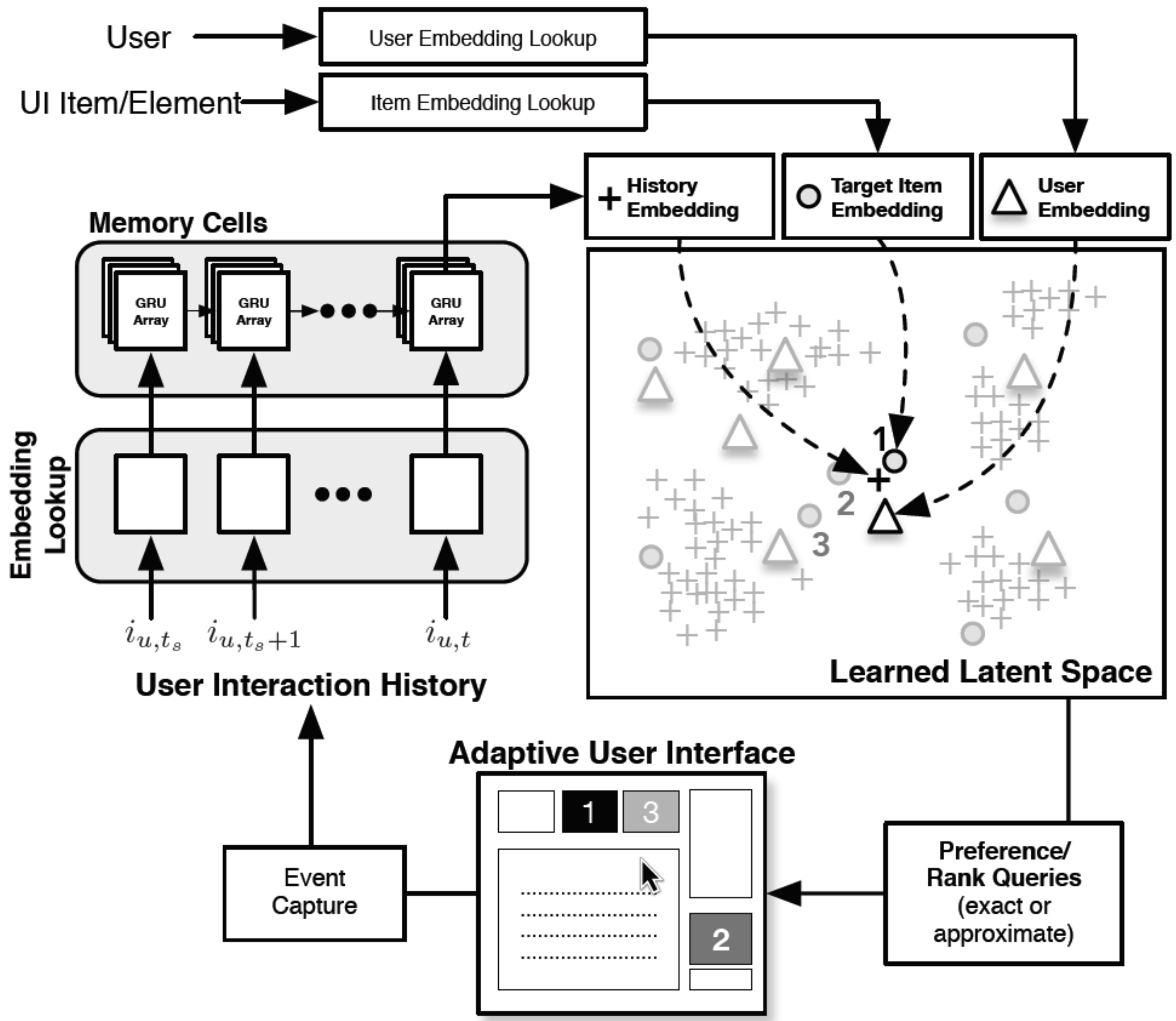
2. Collaborative filtering techniques that enable sharing of data among users

3. Fast approximate nearest-neighbor methods in Euclidean spaces for quick UI control and/or content recommendations.

-> outperforms state-of the-art tensor-factorization and metric embedding methods in contents recommendation

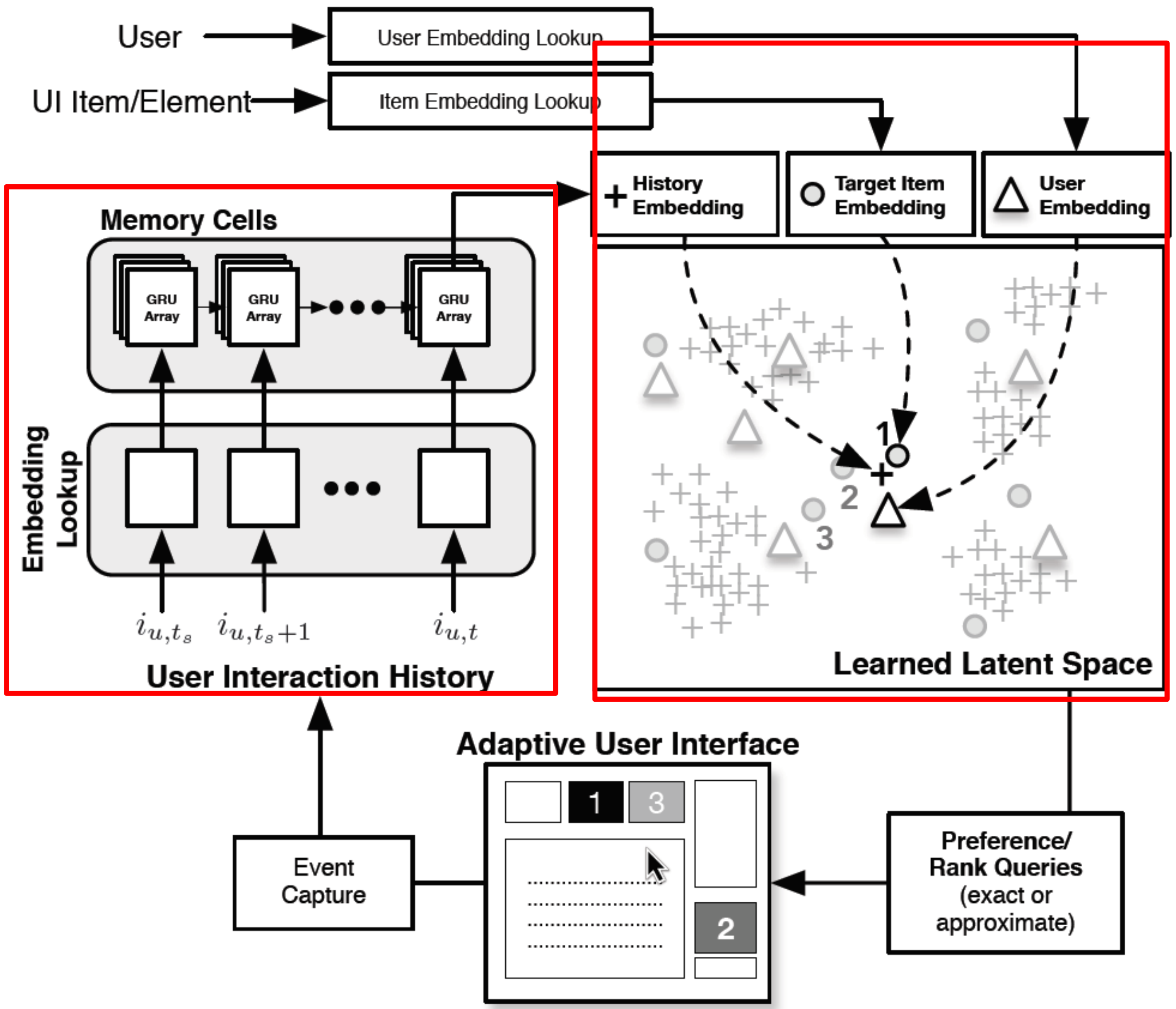
2. DRNN

2-1) Proposed personalized adaptive user interface



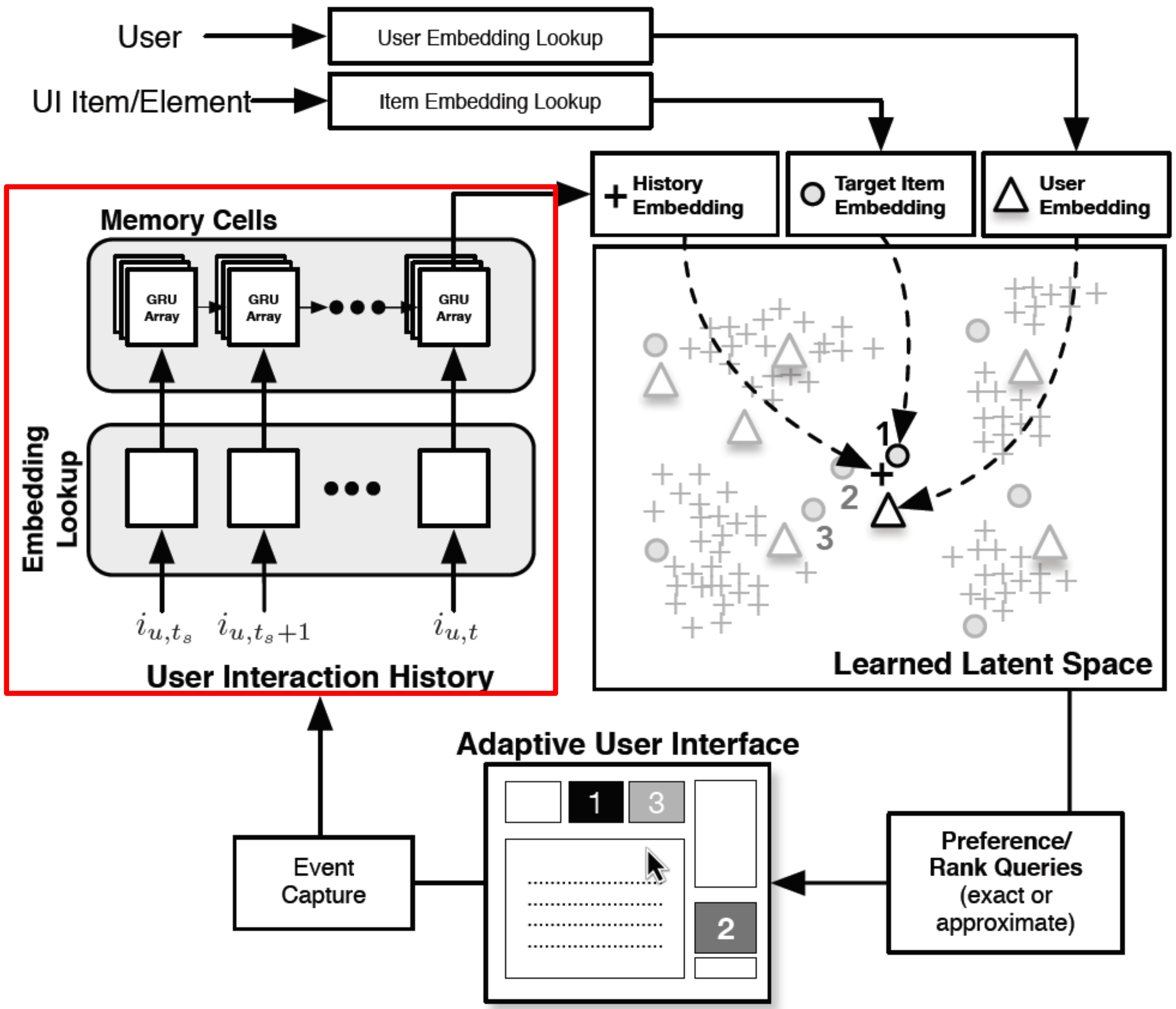
2. DRNN

2-1) Proposed personalized adaptive user interface

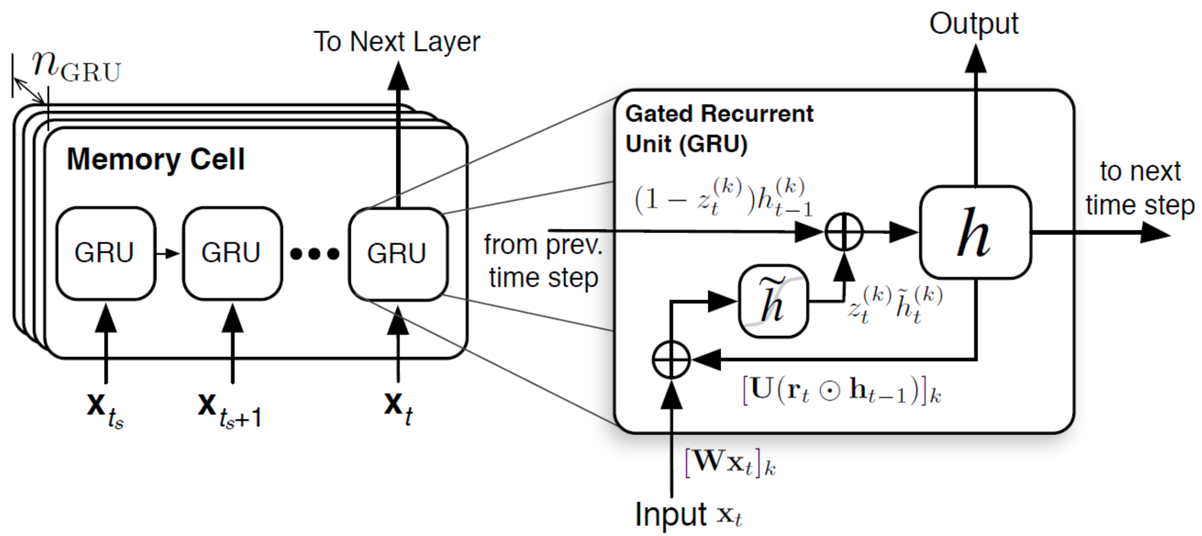


2. DRNN

2-1) Proposed personalized adaptive user interface

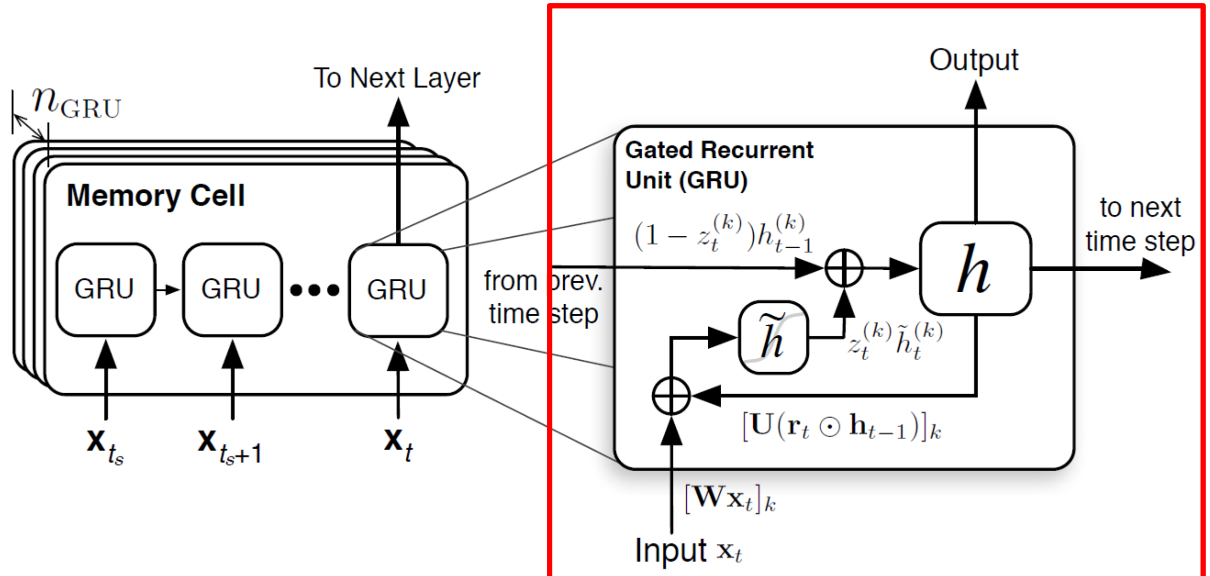


2-2) Memory: Gated Recurrent Units



Gate Recurrent units are used to recommend user interaction information in real time at ‘t’.

2-2) Memory: Gated Recurrent Units



Update gate : $z_t^{(k)} = sigm([\ W_z\, x_t\ +\ U_z\, h_{t-1}\ ]_k)$

Reset gate : $r_t^{(k)} = sigm([\ W_r\, x_t\ +\ U_r\, h_{t-1}\ ]_k)$

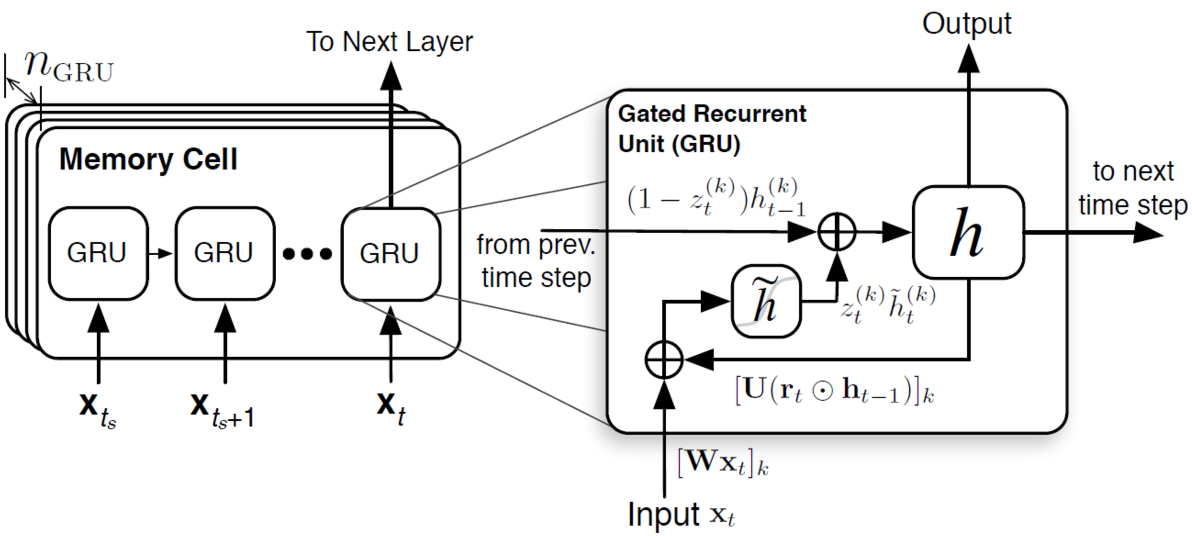
$sigm$: Sigmoid function (0~1)

W, U : Learning parameters

x_t : Input

h_{t-1} : Previous state value

2-2) Memory: Gated Recurrent Units



Update gate : $z_t^{(k)} = sigm([\,\boxed{W_z\, x_t} + \boxed{U_z\, h_{t-1}}\,]_k)$ -> ($\boxed{\text{Present input value}}$ + $\boxed{\text{past information}}$)

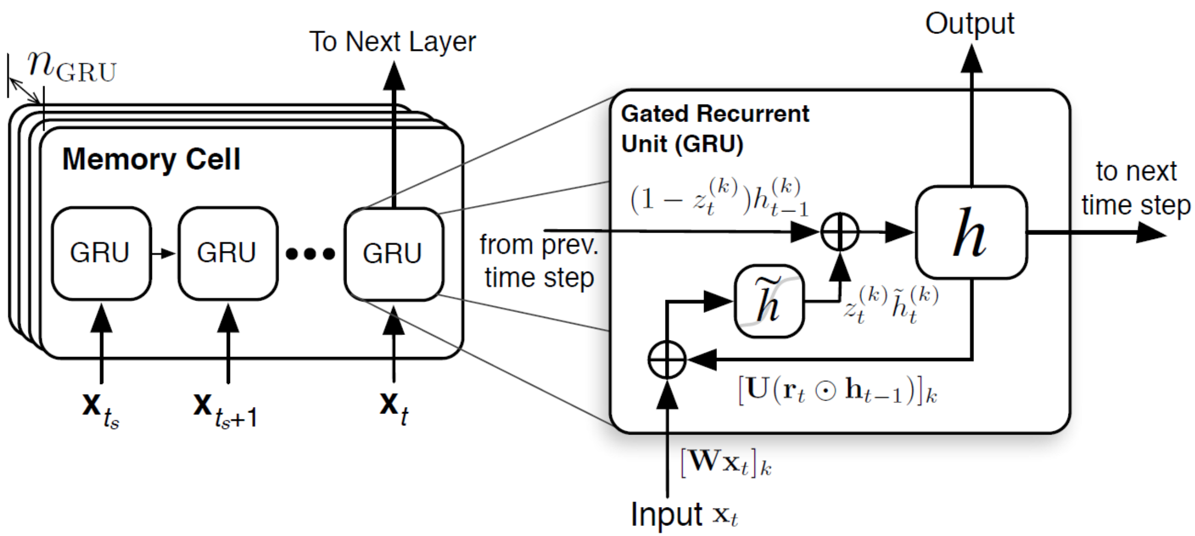
- The update gate controls how much of the previous hidden state to remember.

Reset gate : $r_t^{(k)} = sigm([\; W_r\, x_t \; + \; U_r\, h_{t-1} \;]_k)$

- The update gate controls how much of the previous hidden state to forget.

$sigm$: Sigmoid function (0~1)
W, U : Learning parameters
x_t : Input(user interaction history)
h_{t-1} : Previous state value

2-2) Memory: Gated Recurrent Units



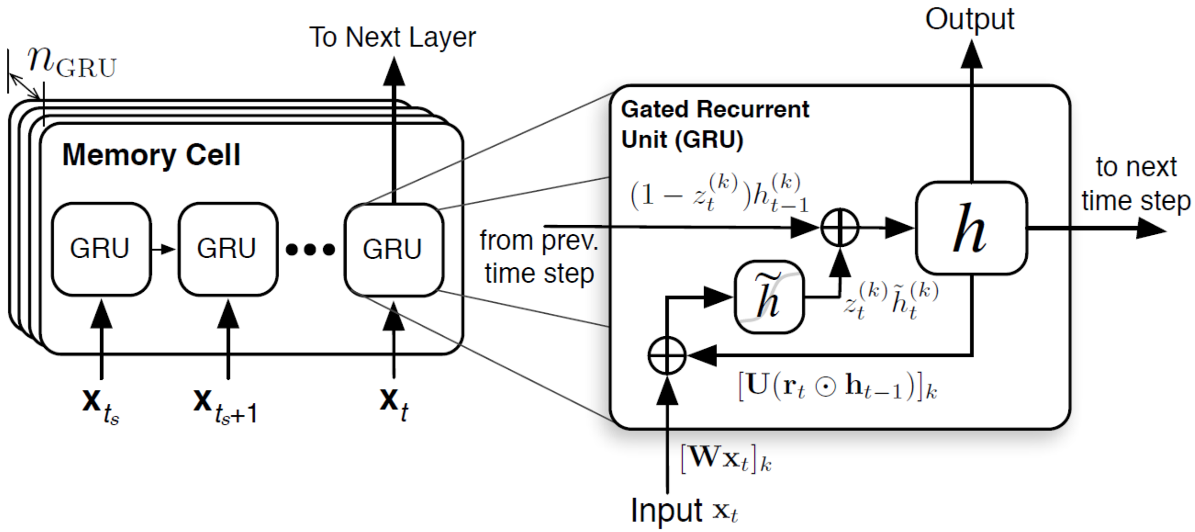
$$\tilde{h}_t^{(k)} = \tanh\left([\, W_r\, x_t + r_t \odot U_r\, h_{t-1} \,]_k \right)$$

$\tanh$: Hyperbolic tangent

$\tilde{h}_t^{(k)}$: Candidate activation (memorable information)

$\odot$: Element-wise multiplication

2-2) Memory: Gated Recurrent Units



$$\tilde{h}_t^{(k)} = \tanh\left(\left[Wx_t + r_t \odot U\,h_{t-1} \right]_k\right)$$

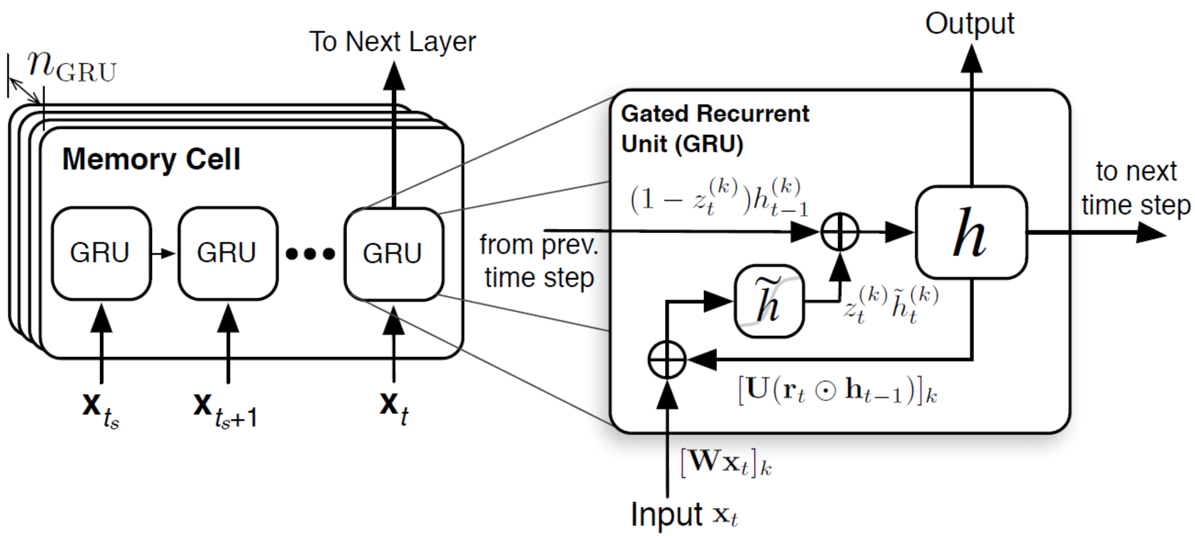
-> Reflects the present information (Wx_t) and the past information (Uh_{t-1}), but how much it stores depends on reset gate(r_t) value.

r_t = 0 -> forget all past information
r_t = 1 -> remember all past information

Current information (Wx_t) is computed regardless of reset gate(r_t) value.

2. DRNN

2-2) Memory: Gated Recurrent Units



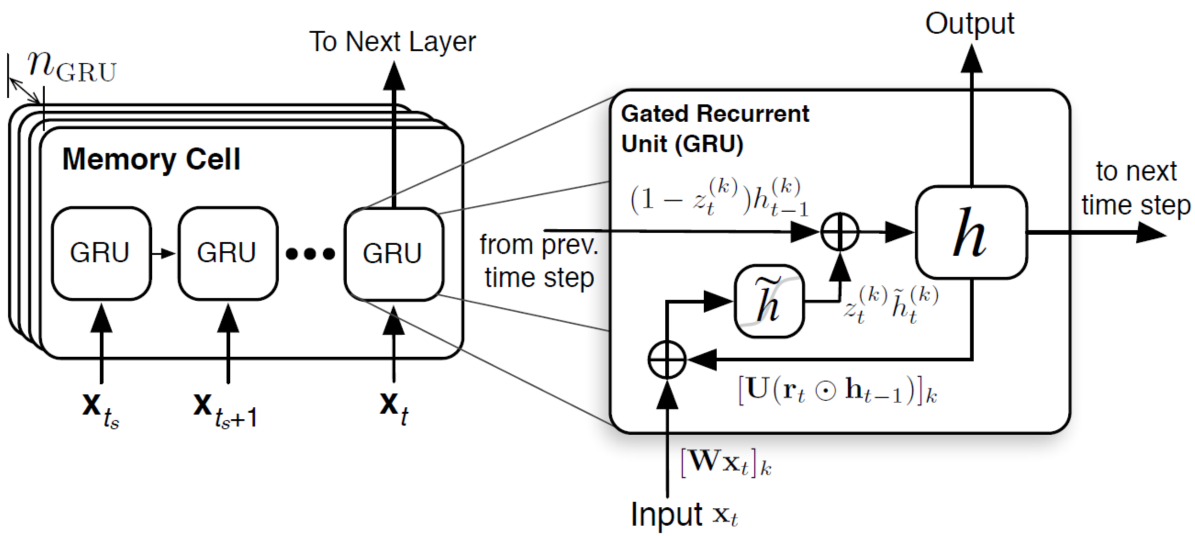
$$z_t^{(k)} = sigm([\ W_z\, x_t\ +\ U_z\, h_{t-1}\ ]_k\ )$$

$$r_t^{(k)} = sigm([\ W_r\, x_t\ +\ U_r\, h_{t-1}\ ]_k\ )$$

$$\tilde{h}_t^{(k)} = \tanh\ (\ [\ W_r\, x_t\ +\ r_t\ \odot\ U_r\, h_{t-1}\ \ ]_k\ )$$

-> state update : $h_t^{(k)} = \left(1\ -\ z_t^{(k)}\right) h_{t-1}^{(k)}\ +\ z_t^{(k)}\ \tilde{h}_t^{(k)}$

2-2) Memory: Gated Recurrent Units



Output : $h_t^{(k)} = \left(1 - z_t^{(k)}\right) h_{t-1}^{(k)} + z_t^{(k)} \tilde{h}_t^{(k)}$

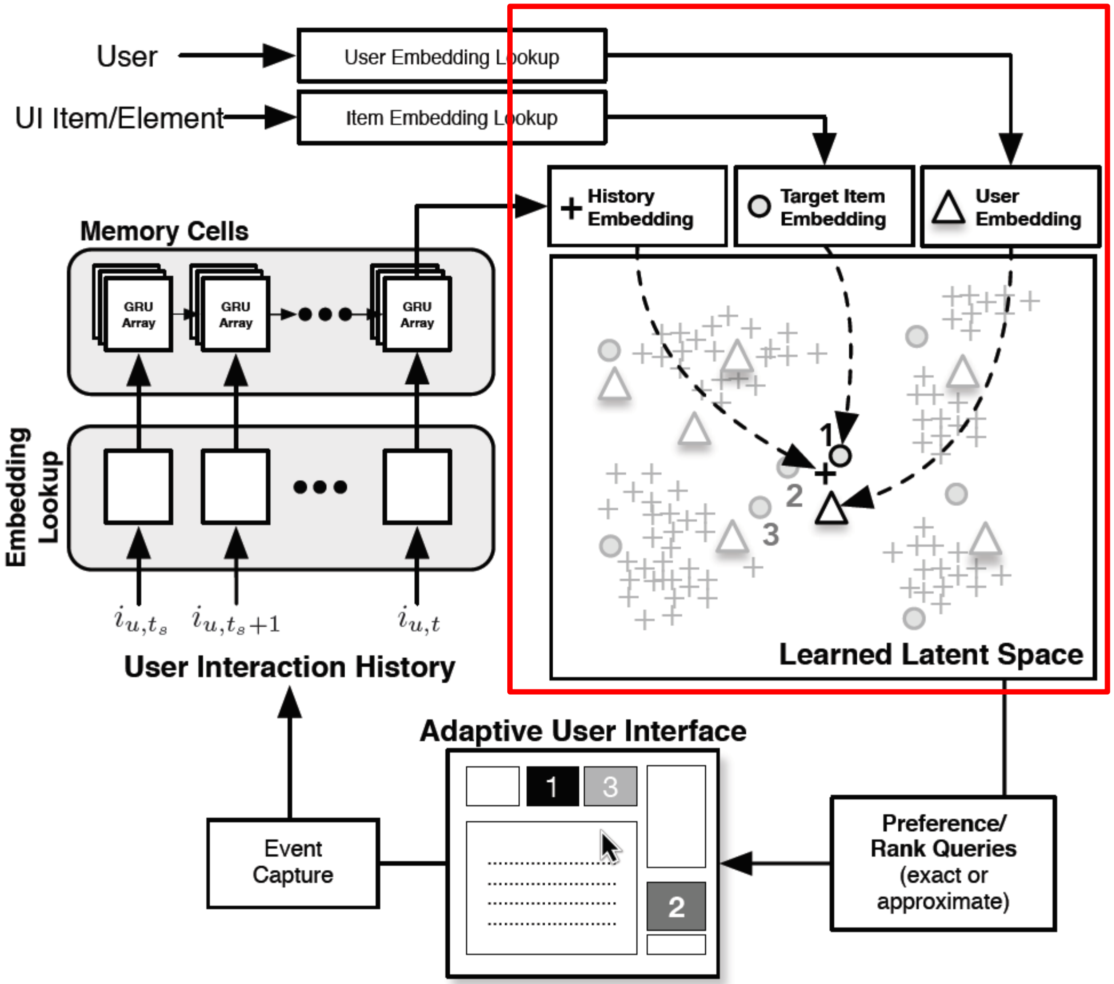
So output is a combination of past state value and candidate value.

- The update gate($z_t^{(k)}$) determines the combination ratio.

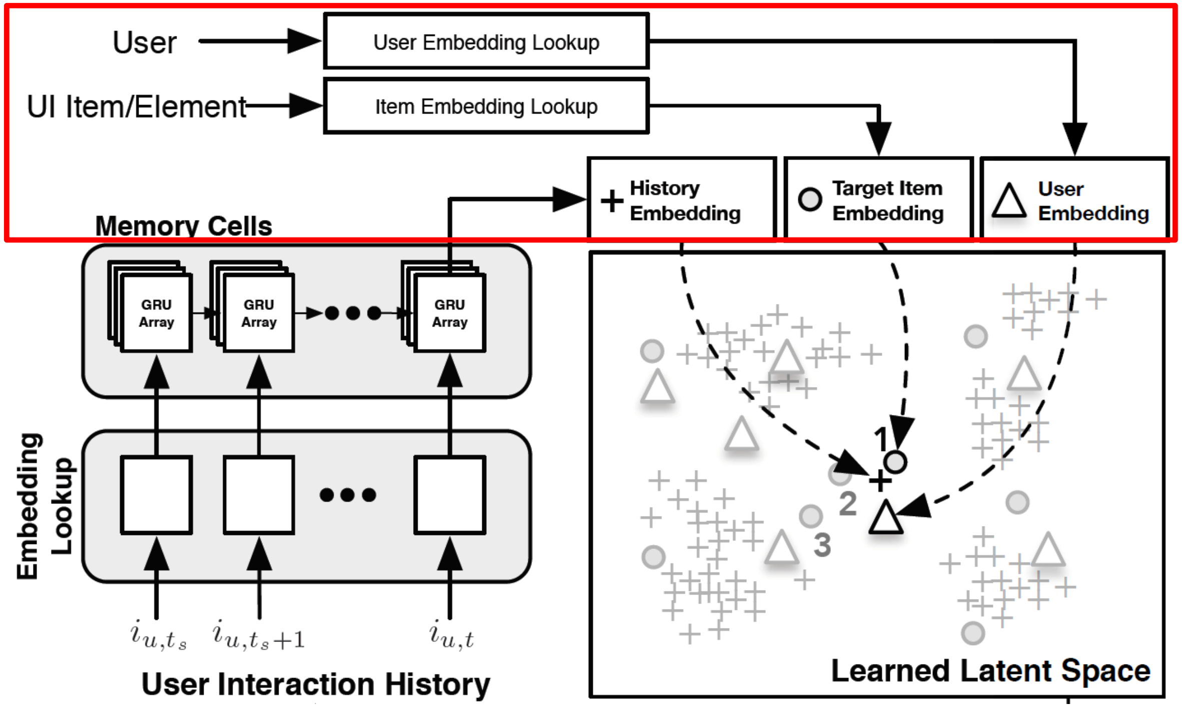
user interaction history information is returned through the GRUs.

2. DRNN

2-3) Embedding Symbols as Vector



2-3) Embedding Symbols as Vector



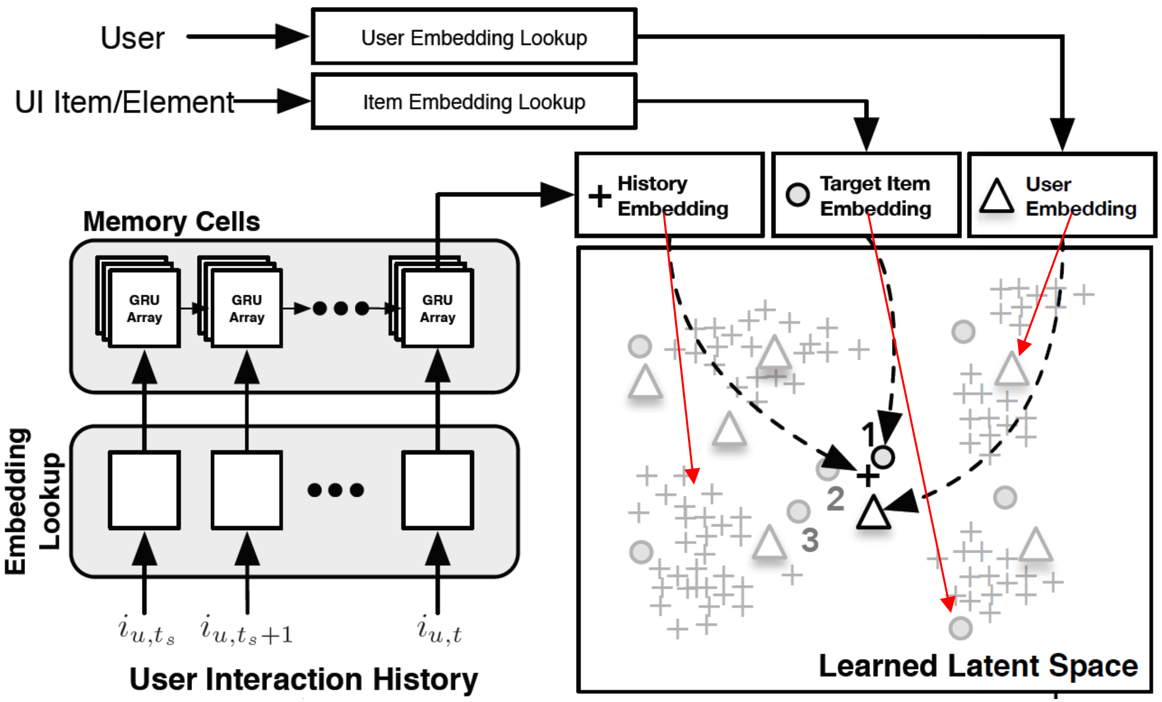
1. Embedding symbols as vectors

   - set the embedding sizes to be equal $n_u = n_i = n_{EMB}$ since the learnt space is shared.

2. To project interaction histories into this latent space, along with users and target items.

3. Find KNN

2-3) Embedding Symbols as Vector



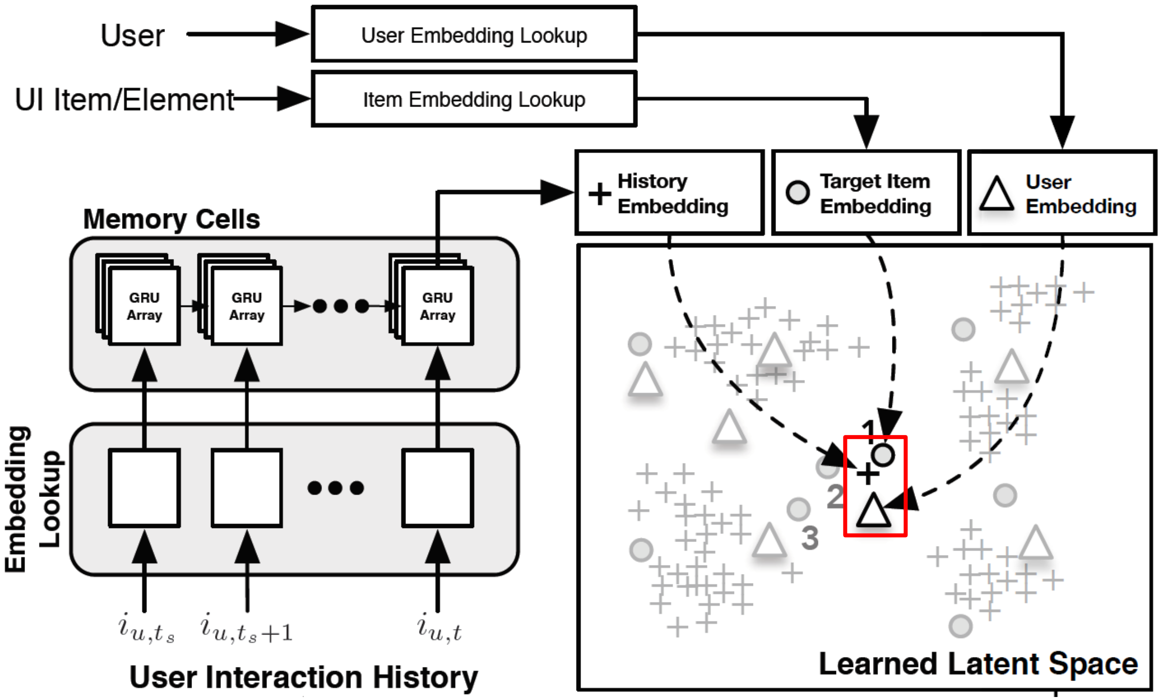
1. Embedding symbols as vectors

   - set the embedding sizes to be equal $n_u = n_i = n_{EMB}$ since the learnt space is shared.

2. To project interaction histories into this latent space, along with users and target items.

3. Find KNN

2-3) Embedding Symbols as Vector



1. Embedding symbols as vectors

   - set the embedding sizes to be equal $n_u = n_i = n_{EMB}$ since the learnt space is shared.

2. To project interaction histories into this latent space, along with users and target items.

3. Find Nearest Neighbor (FLANN) - > recommend

3. EXPERIMENTS

3-1) Training Data

Training Data

- Equation Grapher UI dataset(EqGraph)
  51 users , 16 items , 1277 interactions

- Microsoft Web Data
  1205 users , 246 items, 6623 interactions

- 2015 ASSISTment dataset
  4130 users , 100 items, 51,318 interactions

- Loss optimize - stochastic gradient descent (Adam)

$$L = \sum_{u,t} l_{\mathrm{SM}}(u,t)$$

$$l_{\mathrm{SM}} = -a_{u,t+1,y} + \log \frac{\sum_{j \in S} \exp(a_{u,t+1,j} - \log |I|)}{|I|}$$

$a_{u,t+1,y}$: Preference score for item j

3-2) Model Training and Parameters

Model training and Parameters

200 epochs
100-sample mini-batches
15 negative sample

Embedding size { 32, 64 ,128 }

Dropout regularization {0.05 , 0.10 , 0.15 , 0.20 }

10-fold cross-validation : Training data : 90% / Test data : 10%

Performance evaluation

MAP (Mean Average Precision)_

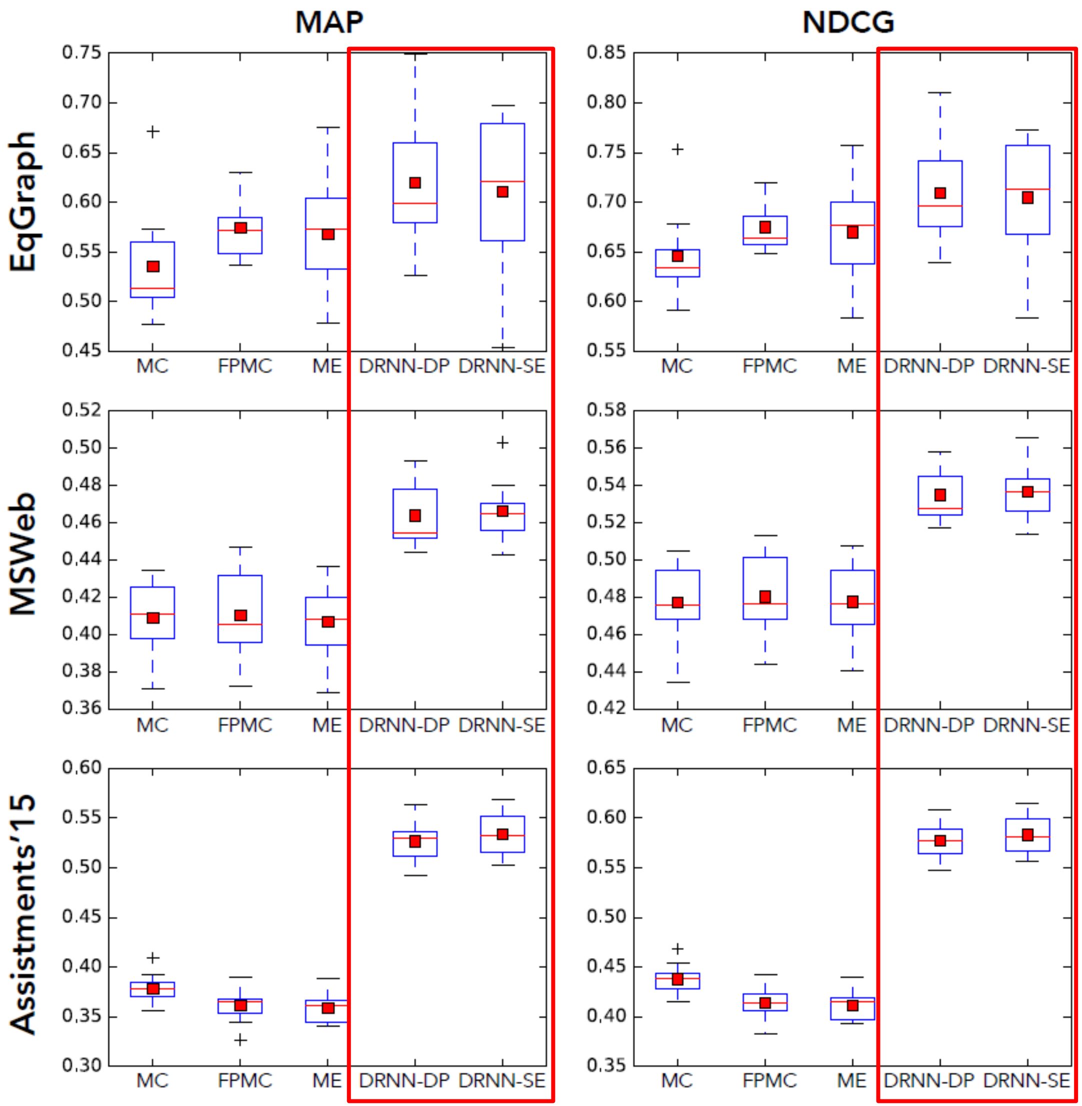
$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{Q}$$

NDCG (Normalized DCG)

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCGp} \qquad \mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

3. EXPERIMENTS

3-3) MAP and NDCG score



MC : Markov Chain

FPMC : Factorized Personalized Markov Chain

ME : Euclidean Metric

DRNN-DP : DRNN with dot product

DRNN-SE : squared Euclidean

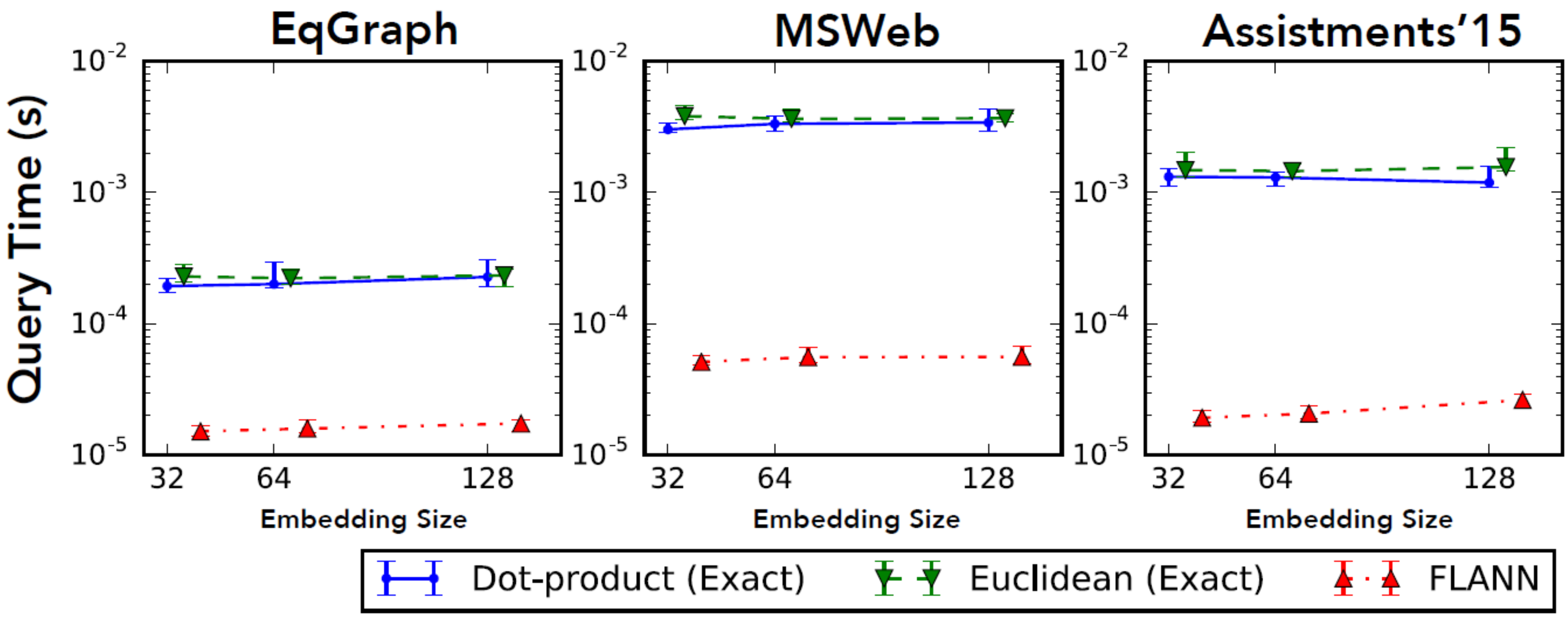
DRNN achieves the highest scores

3-4) MAP scores

Dataset	MC	FPMC	ME	DRNN-DP	DRNN-SE
EqGraph	0.535 (0.054)	0.574 (0.031)	0.568 (0.057)	0.620 (0.062)	0.611 (0.078)
MSWeb	0.409 (0.020)	0.410 (0.023)	0.407 (0.021)	0.475 (0.022)	0.466 (0.016)
Assist'15	0.378 (0.015)	0.361 (0.016)	0.359 (0.014)	0.526 (0.020)	0.534 (0.022)

DRNN achieves 15~50 % higher MAP score

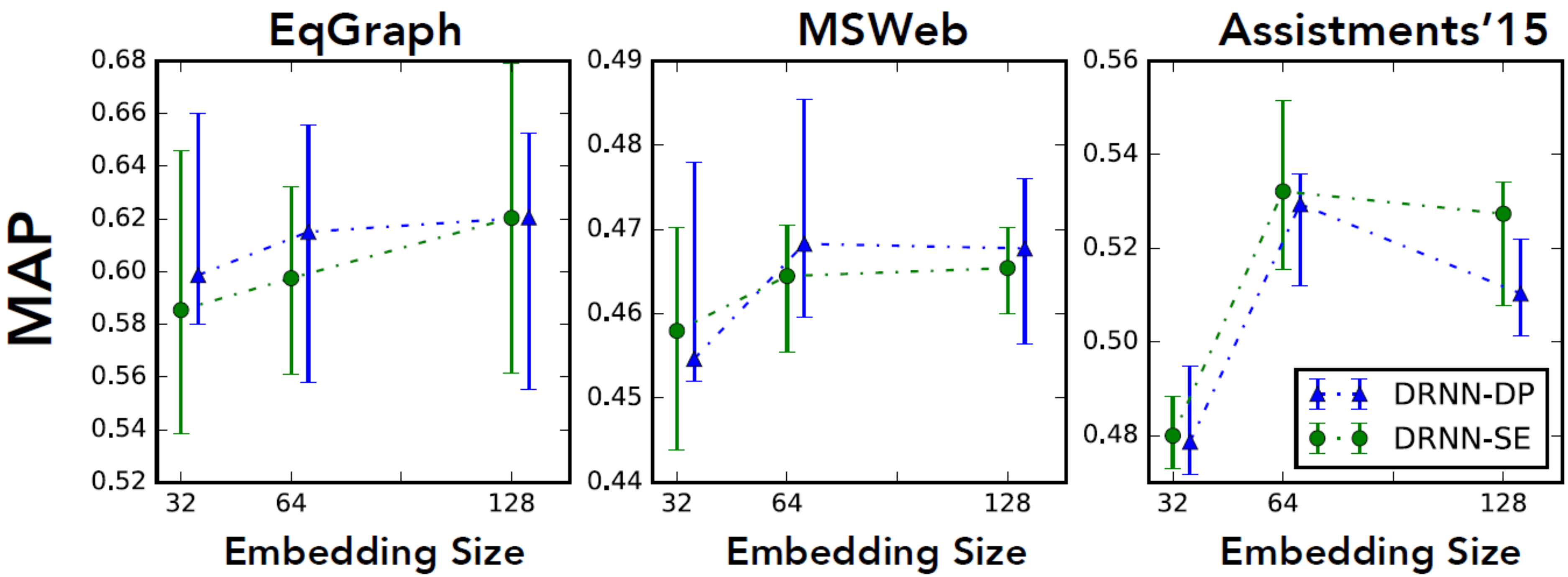
3-5) Query times



FLANN faster than dot-product or Euclidean distances

3. EXPERIMENTS

3-6) MAP scores vs Embedding size



DRNN's performance generally improved with embedding size, with a few exceptions where nEMB = 128

4. CONCLUSION

- **This paper presented an architecture for an adaptive user interface with a RNN** that performs sequential recommendation of content and control elements.

- Relative to the compared methods, **the DRNN provided higher (or similar) accuracies, with fast prediction**.

- **The use of FLANN to perform item recommendations improved query times** by an order of magnitude over standard exact techniques

- **DRNN can be easily extended to project side-information**, such as user profiles and interaction element features into the latent space

Q & A

Thank you!