Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks

(Zhang, et al. 2017)

Presented by Jiin Seo

June 28, 2018

Outline

Outline

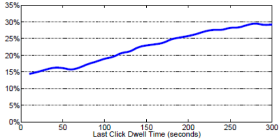
1. Introduction

Sequential Click prediction (Zhang, et al. 2017)

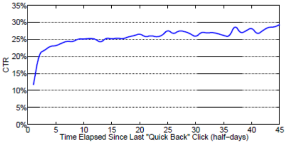
- Goal : Estimating the click-through rate (CTR) of ads with sequential information
- Modeling sequential dependency between user's behaviors
  $\rightarrow$ RNN

Outline

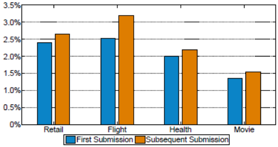
2. Data Analysis on Sequential Dependency



(a) Correlation between last click dwell time and current CTR.



(b) CTR right after a quick back click on the same ad.



(c) CTR on users' first and subsequent submissions of a certain type of query.

Figure: Sequential information

- The model have to learn sequential dependency between user behaviors by itself.

Outline

3. The Proposed Framework

Feature Construction

- Ad features
- User features
- Sequential feature
- We re-organize the input features along with the user dimension.

3. The Proposed Framework

Architecture

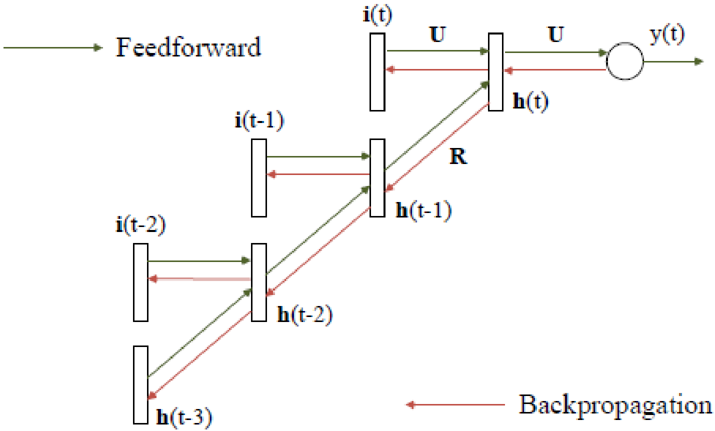


Figure: RNN training process with BPTT algorithm. Unfolding step is set to 3 in this figure.

3. The Proposed Framework

Model

$$
\begin{aligned}
\mathbf{h}(t) &= f(i(t)\mathbf{U}^T + \mathbf{h}(t-1)\mathbf{R}^T), \\
y(t) &= \sigma(\mathbf{h}(t)\mathbf{V}^T)
\end{aligned}
$$

- $\mathbf{R}$: the recurrent connections between $\mathbf{h}(t-1)$ and $\mathbf{h}(t)$,
- $f()$: tanh function , $\sigma()$: sigmoid
- $i(t)$: Features correlated to user's current behavior.
- $\mathbf{h}(t)$: Sequential information of user's previous behaviors.

3. The Proposed Framework

Learning

- Loss (averaged cross entropy)

$$L = \frac{1}{M}\sum_{i=1}^{M}(-y_i log(p_i) - (1 - y_i)log(1 - p_i))$$

- $y_i \in \{0, 1\}$: labeled sample , p_i : the predicted click probability
- Learning Algorithm (Back Propagation Through Time : BPTT)

3. The Proposed Framework

Learning Algorithm(BPTT)

- The gradient of the output layer :

$$e_o(t) = y(t) - l(t)$$

  y(t) : the predicted click prob., l(t) : the binary true label

- The weights($\mathbf{V}$) between the hidden layer($\mathbf{h}(t)$) and output ($y(t)$) :

$$\mathbf{V}(t+1) = \mathbf{V}(t) - \alpha \times e_o(t) \times \mathbf{h}(t)$$

- Gradients of errors :

$$\begin{aligned} e_{\mathbf{h}}(t) &= e_o(t)\mathbf{V} * (\overrightarrow{1} - \mathbf{h}(t) * \mathbf{h}(t)), \\ e_{\mathbf{h}}(t-\tau-1) &= e_{\mathbf{h}}(t-\tau)\mathbf{R} * (\overrightarrow{1} - \mathbf{h}(t-\tau-1) * \mathbf{h}(t-\tau-1)), \\ & \tau \in [0, T) \text{ and } T : \# \text{ of unfolding steps.} \end{aligned}$$

3. The Proposed Framework

Learning Algorithm(BPTT)

- The weight matrix **U** and the recurrent weights **R** :

$$\begin{aligned}
\mathbf{U}(t+1) &= \mathbf{U}(t) - \alpha\left[\sum_{z=0}^{T-1} e_{\mathbf{h}}(t-z)^T i(t-z)\right] \\
\mathbf{R}(t+1) &= \mathbf{R}(t) - \alpha\left[\sum_{z=0}^{T-1} e_{\mathbf{h}}(t-z)^T \mathbf{h}(t-z)\right]
\end{aligned}$$

3. The Proposed Framework

Testing process

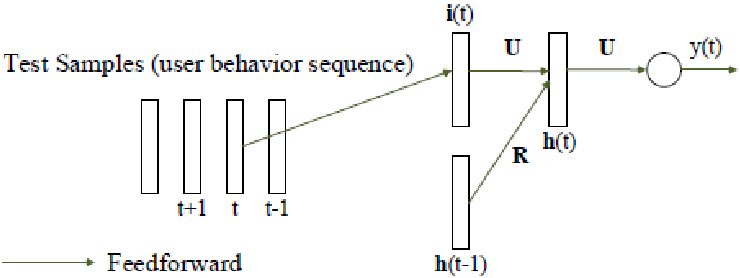


Figure: RNN testing process with sequential input samples.

- We only record the hidden state of the last test sample.

Outline

4. Experiments

Overall performance

- Data : Logs of search engine for 2 weeks (randomly sample a set of users)
- Metric
  - Area Under ROC Curve (AUC)
  - Relative Information Gain (RIG)

Model	AUC	RIG
LR	87.48%	22.30%
NN	88.51%	23.76%
RNN	**88.94%**	**26.16%**

Figure: Overall performance of models

4. Experiments

Performance with History

- RNN model performs the best in all settings
- Long sequences help further improve the accuracy of CTR.

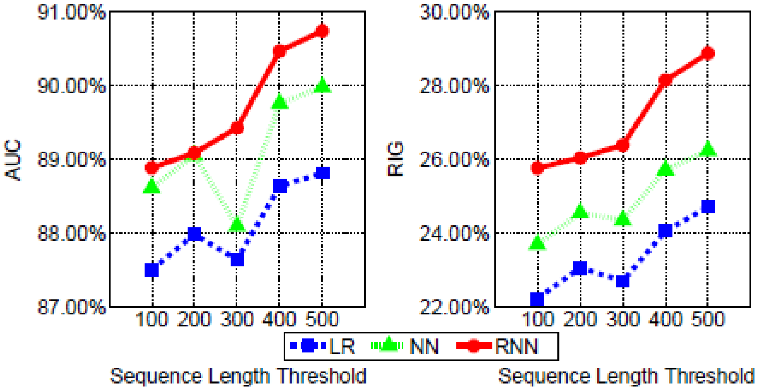


Figure: Performance with Long vs. Short History

4. Experiments

Effect of RNN Unfolding Step

- Unfolding 3 steps is the best.
- The backpropagated error vanishes after 3 steps of unfolding.