user2vec: user modeling using LSTM networks

Konrad Żołna & Bartłomiej Romański, June 24th 2016

Jagiellonian University & RTB House

User modeling

User modeling describes the process of building up and modifying a **state (internal representation)** of the user.

The main goal of user modeling is customization and adaptation of systems to the **user's specific needs**.

Real-time bidding

Real-time bidding (RTB) is an online advertising auction-based model where the advertiser **valuates every single impression** opportunity.

A **bid value** is usually based on a predicted impression value evaluated using **low level features** such as the **history of the user's activity** on the advertiser's webpage or the **size of the ad slot**.

User history as an input?

Typically the history of the user is projected into a **fixed number of manually-crafted features** which are believed to help in prediction.

These features are usually extracted using a baseline feature extraction methods like **counting** or **binning**.

Manually-crafted features

Manual crafting requires a **human expert** whose work is **laborious** and **expensive.**

Usefulness of features may depend on the advertiser, so a human has to **revise them frequently** and **reexplore** for every new advertiser.

Since features are snapshot at the time of the impression, models don’t learn from events which follow the last impression of the user and ignores the data for users who have never seen any impressions. **Data is lost.**

Sequential input

Our LSTM **model is fed sequentially** with **every event** originating **from the user's activity** on the advertiser's website.

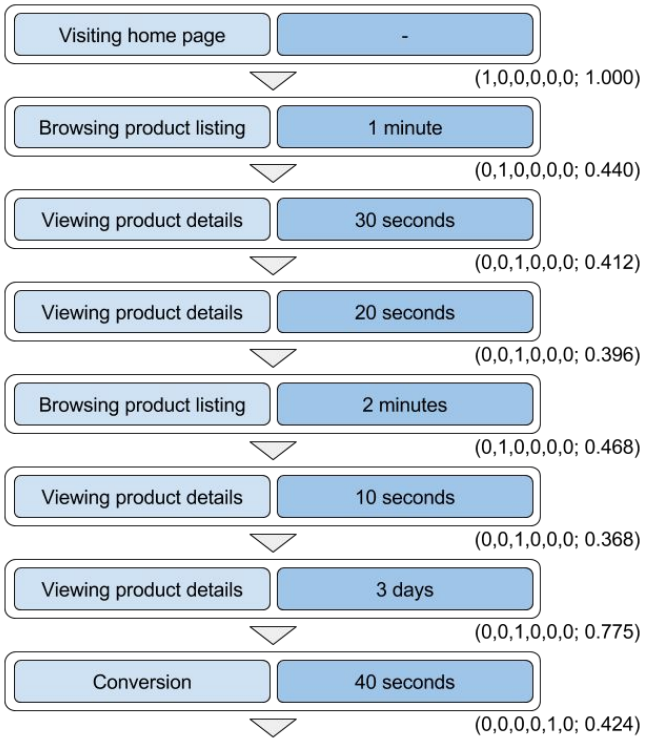
Input to a single step is represented as a vector of seven real numbers: **one-hot encoded type of the event** and normalized **time to the previous event**.

Sequential input (example)

In the first session a user **visited home page**, **viewed details of three products** with **browsing two listings** between.

The second session (3 days after the first one) is started by **browsing product details** and finalizes with **a conversion**.

The figure also shows how these actions are encoded to be interpretable by the LSTM model: **one-hot encoded event's type** first and **normalized time to the previous event** last.

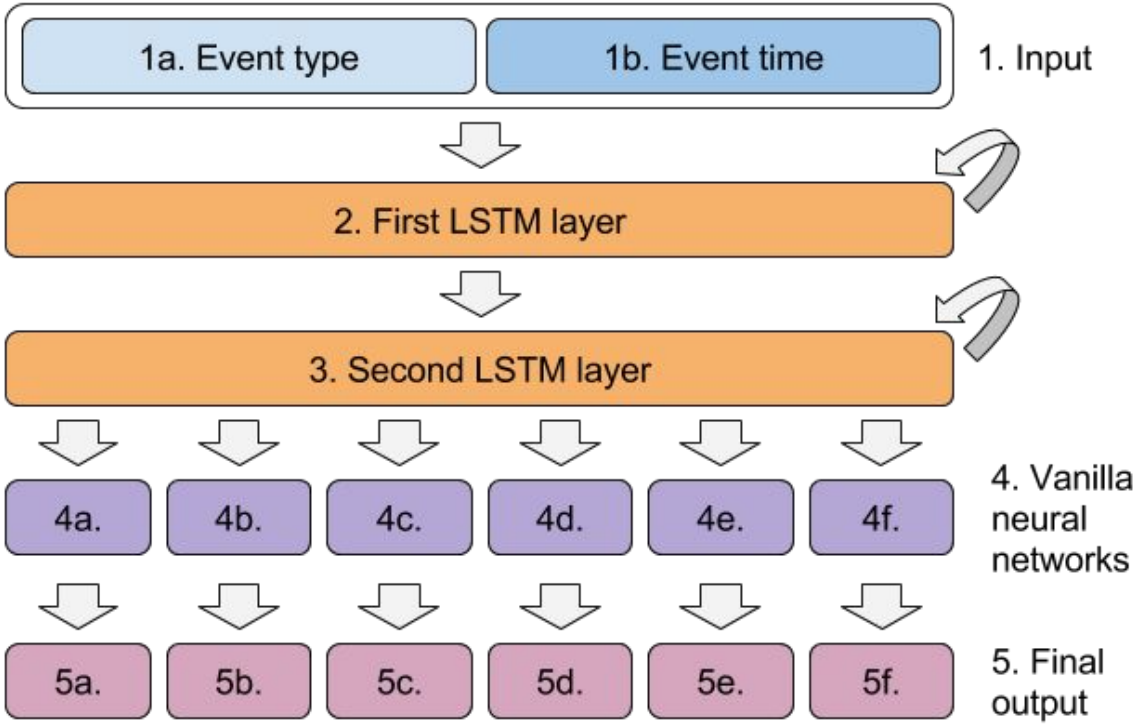


Targets of our LSTM model

A single input for the user is the sequence of all the events and **targets are answers to** a fixed list of a few **questions asked at the time of every event.**

a. Will the **user come back** in less than 30 days after this session ends?
b. What is the **type of the next event**?
c. Will **this session end** in 20 secs / 2 mins / 20 mins / more than 20 mins?
d. Will the **next session start** in 16 hrs / more than 16 hrs / never?
e. Will the **next conversion** be in this session / after this session / never?
f. Will the **user convert** in the next 30 days?

Our LSTM model



Memory cells of LSTM

State of every LSTM model is stored in two fixed size vectors of real numbers called the **memory cells** and the **last output**.

Since our LSTM model is trained to predict user's behavior, elements of these vectors are the **natural candidates** for the **user-dependent features** (they **depict a user's state**).

They can be **extended by** the **resulting predictions** (answers to the questions).

user2vec

Learned on historic data **LSTM is set up** and **constantly monitors** all events performed on the advertiser's website.

At any time one can **ask the LSTM about** it's state for the particular user which can be understood as **the user's state**.

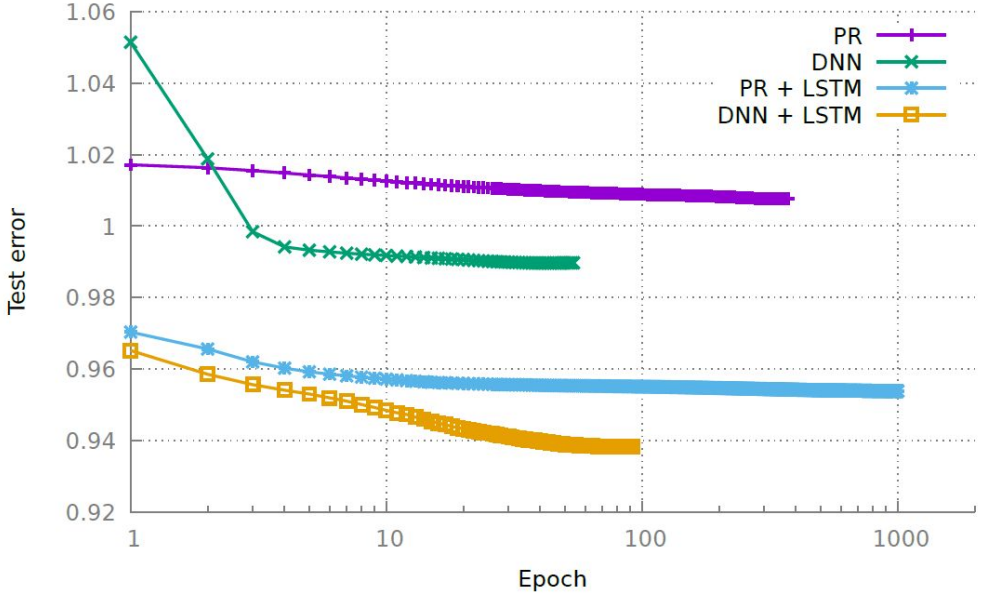
This procedure is called *user2vec* and obtained features can be used further by more specialist models like CR model.

CR model comparison 1/2

Two CR models were considered each one **in two versions**:
a **core version** (only core features),
an **extended version** (with additional *user2vec* features).

Considered models are:
Poisson regression (*PR, PR + LSTM*),
Deep neural net (*DNN, DNN + LSTM*).

CR model comparison 2/2



Model name	Test error	Improvement
PR	1.0076	0.00%
DNN	0.9897	1.78%
PR + LSTM	0.9538	5.34%
DNN + LSTM	0.9383	6.88%
DNN + LSTM - Core	0.9626	4.47%

Current directions

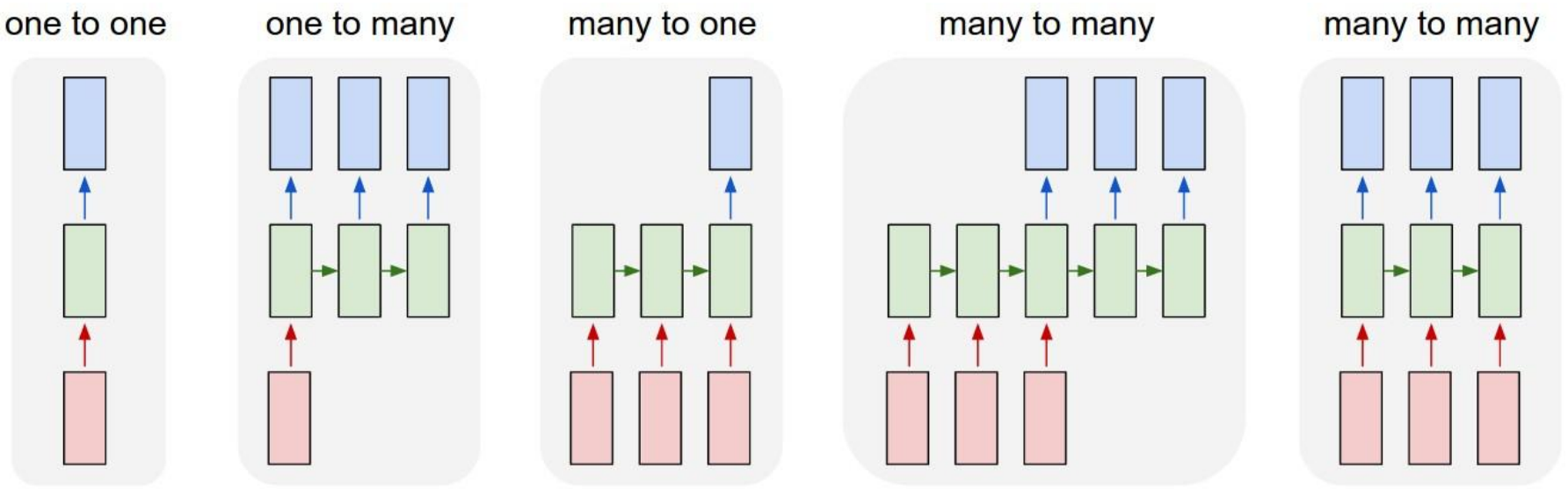
The LSTM can be fed with more **detailed descriptions of the event**. For example, for a viewed product, the LSTM can also get the **identifier of the product**. It may result in two benefits:

a. the projection is more **sophisticated** and **accurate**,
b. possibility of performing useful **hallucination**.

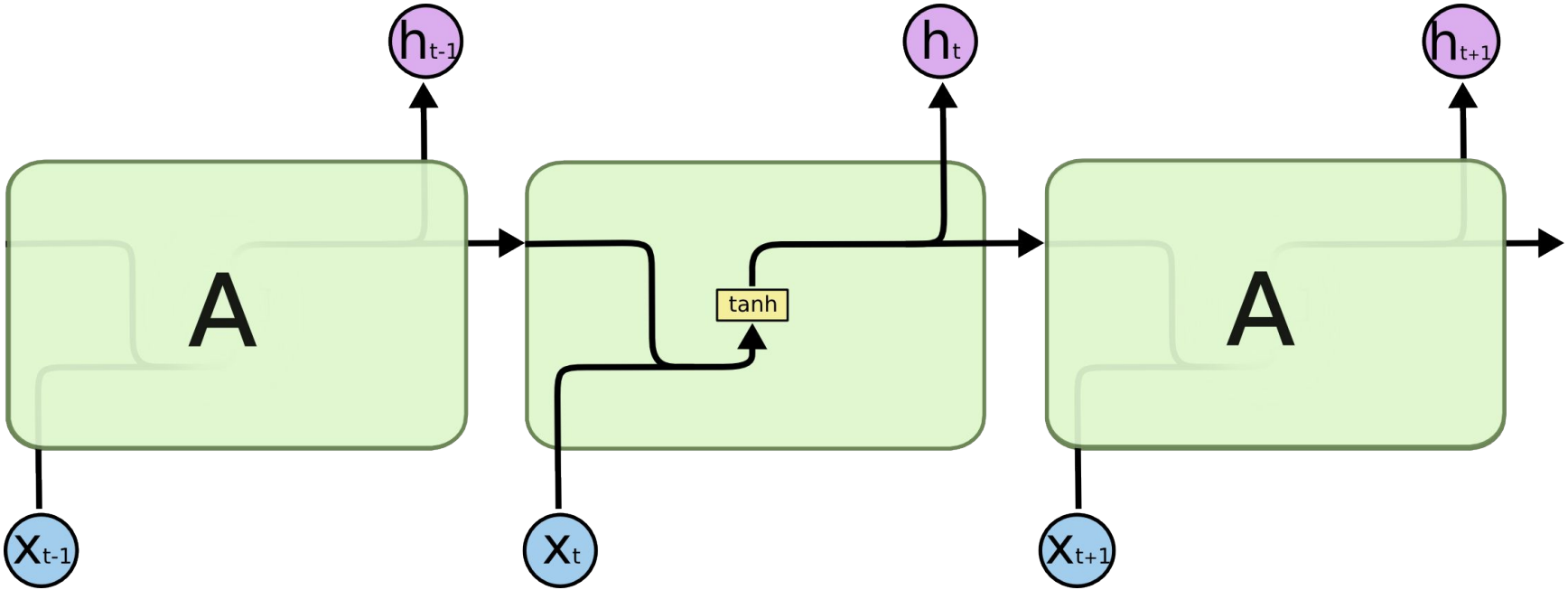
End of presentation

Thank you for your attention.

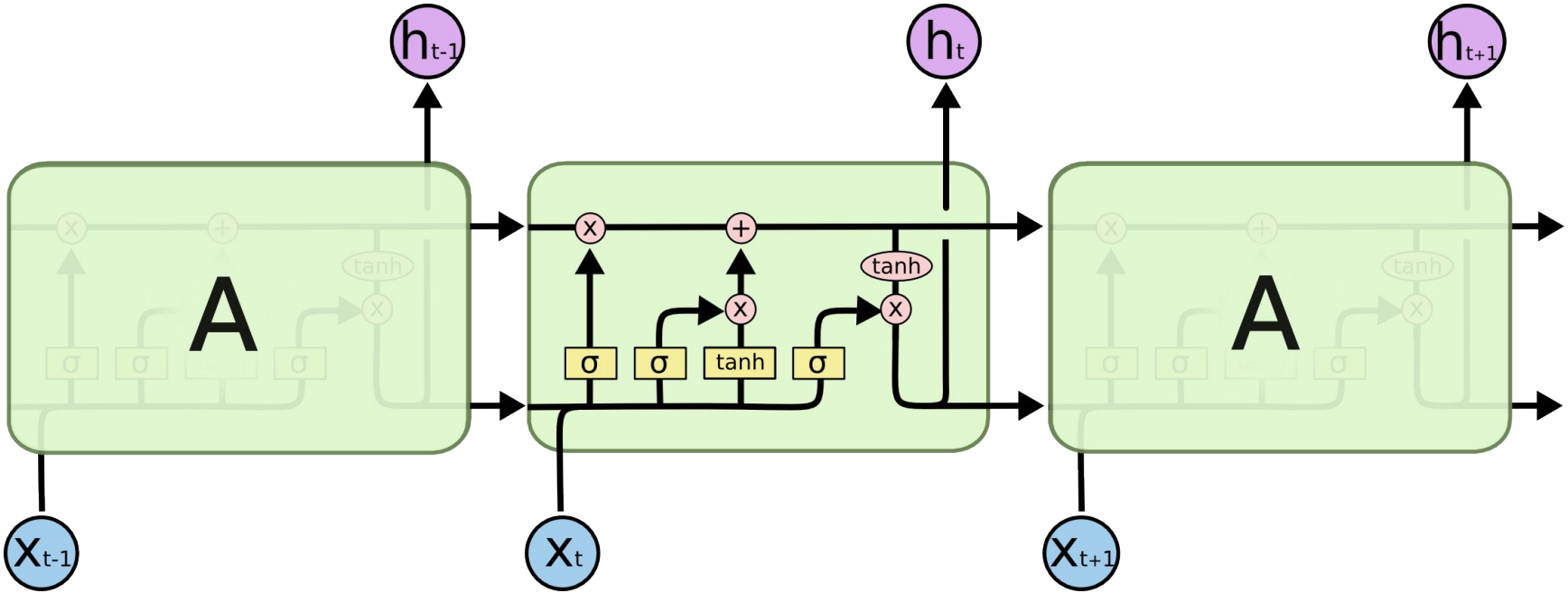
Sequential data



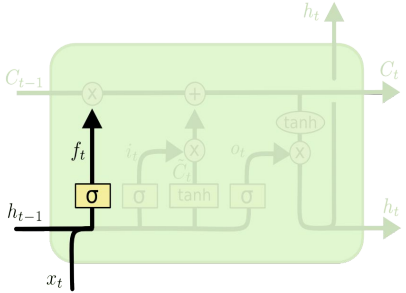
Recurrent Neural Networks



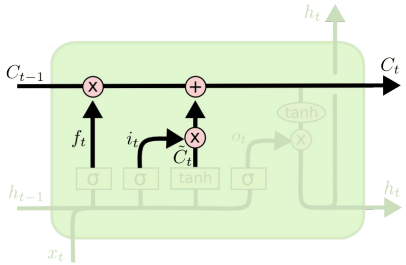
Long short-term memory



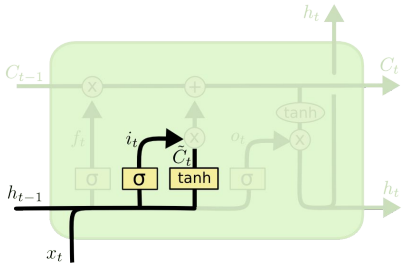
LSTM, step by step



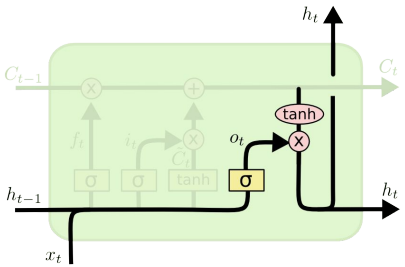
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

End of presentation

Part of LSTM is taken from the blog of **Andrej Karpathy** (The Unreasonable Effectiveness of Recurrent Neural Networks) and the blog of **Christopher Olah** (Understanding LSTM Networks).

Thank you for your attention.