Recsys Challenge – G26

Algorithms: NRMS, GRU, LSTUR

Neural Recommendation with Multi-Head Self-Attention (NRMS)

Contextualized representations of user interests, generated with CNNs.

Context = Relatedness between different news articles that a user has read.

Components:

1. **News Encoder**: Capture relationships between words in news content.
2. **User Encoder**: Capture user interests based on session history.

Additive attention mechanisms in both encoders to emphasize most important sections in text.

Both effective and efficient, used in production *(Neural News Recommendation with Multi-Head Self-Attention: Wu et al., EMNLP-IJCNLP 2019)*

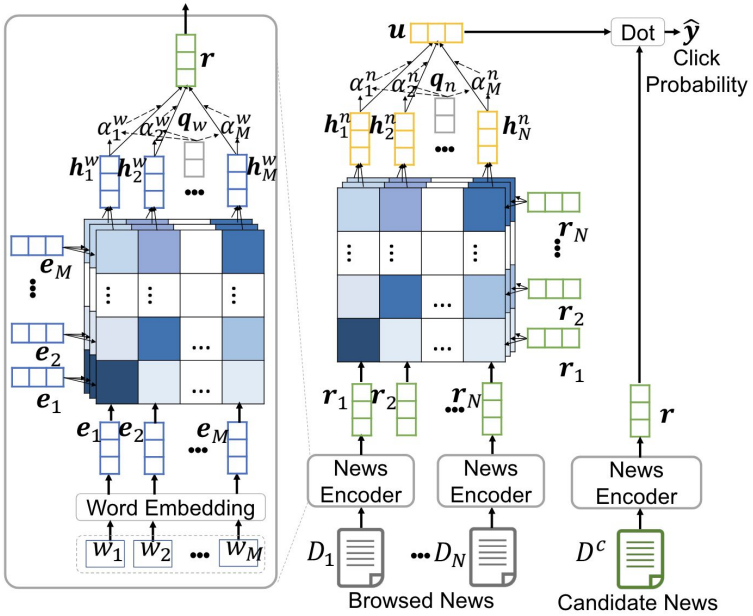
Summary: https://typeset.io/questions/what-is-neural-news-recommendation-with-multi-head-self-36nr72w0v
One file implementation: https://github.com/wuch15/EMNLP2019-NRMS

Neural Recommendation with Multi-Head Self-Attention (NRMS)

Components:

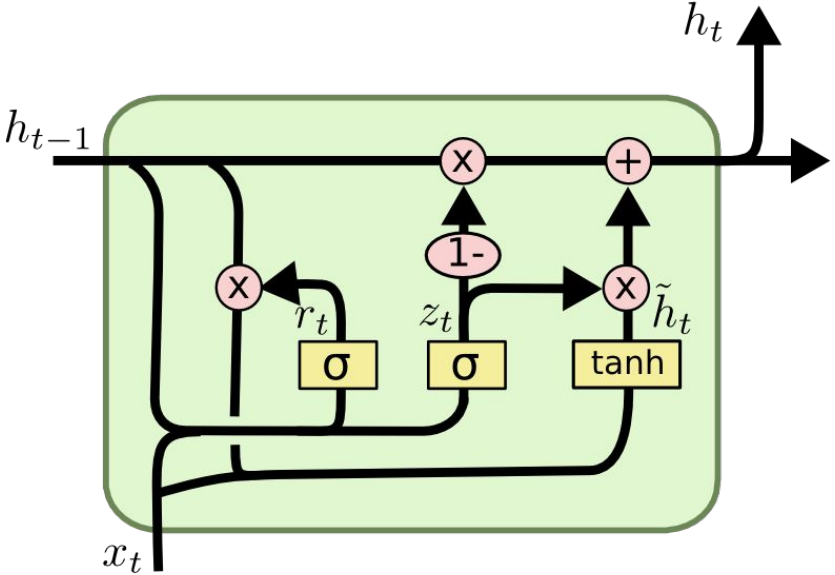
1. **News Encoder**: Capture relationships between words in news content.
2. **User Encoder**: Capture user interests based on session history.

(Neural News Recommendation with Multi-Head Self-Attention: Wu et al., EMNLP-IJCNLP 2019)



Summary: https://typeset.io/questions/what-is-neural-news-recommendation-with-multi-head-self-36nr72w0v
One file implementation: https://github.com/wuch15/EMNLP2019-NRMS

Neural Recommendation with Gated recurrent units (GRUs)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

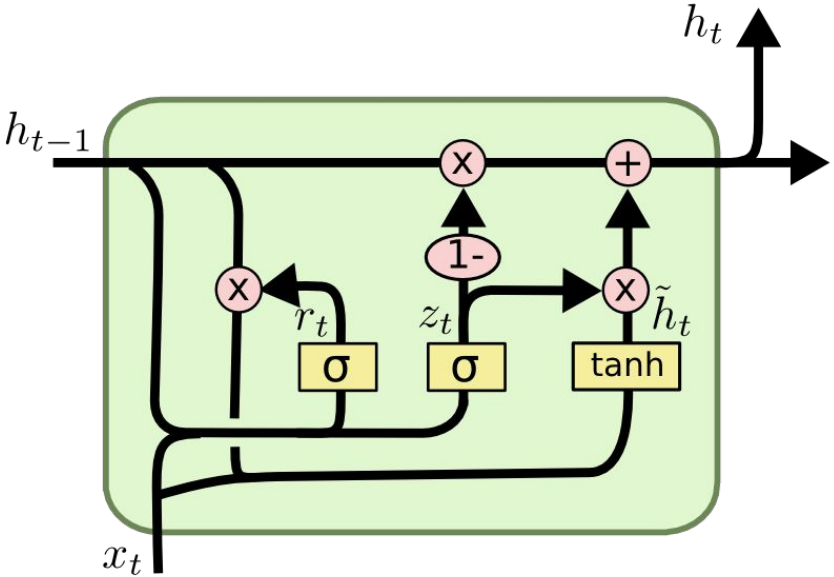
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Neural Recommendation with Gated recurrent units (GRUs)



Improvement of LSTM.

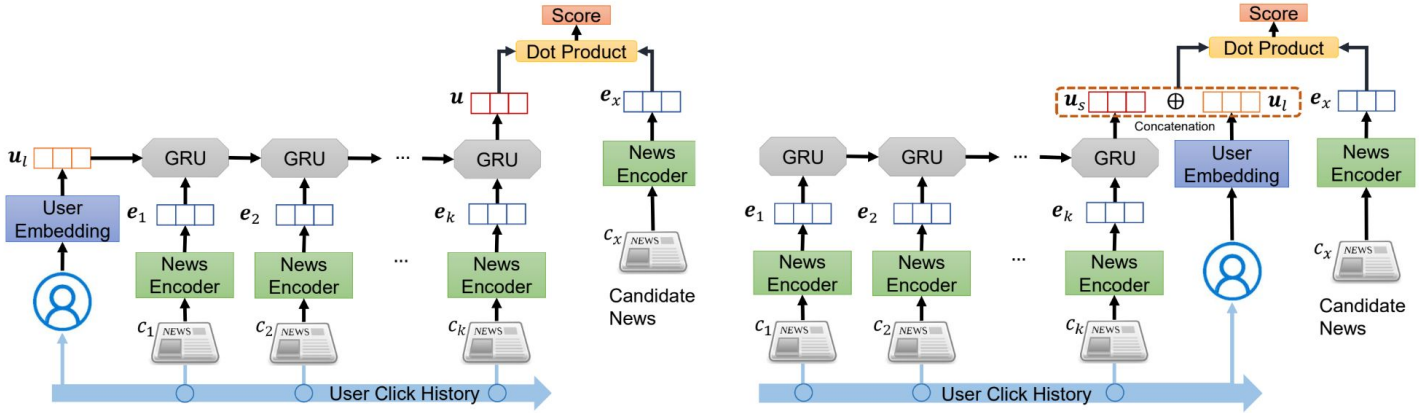
Simpler and more performant.

- Combines the forget and input gates into a single “update gate”.
- Merges the cell state and hidden state.

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Neural Recommendation with Long- and Short-term User Representations (LSTUR)

Two possible approaches:



(a) LSTUR-ini. (b) LSTUR-con.

(Neural News Recommendation with Long- and Short-term User Representations: An et al., ACL 2019)

Neural Recommendation with Long- and Short-term User Representations (LSTUR)

Built upon GRU.

Captures both users' long- and short-term interests:

- **Long-term user-representations**: From user ID embeddings to capture stable preferences.
- **Short-term user-representations**: From recently browsed news using a GRU network to reflect current interests.
- **News encoding**: Contextualized representations of news content.

Two ways to combine long-term and short-term user representations:

a. Long-term representations initialize the GRU's hidden state for short-term representations
b. Concatenating both representations into a single user vector

(Neural News Recommendation with Long- and Short-term User Representations: An et al., ACL 2019)

One file implementation: https://github.com/recommenders-team/recommenders/blob/main/examples/00_quick_start/lstur_MIND.ipynb or https://github.com/yusanshi/news-recommendation
Benchmark: https://www.researchgate.net/figure/The-effectiveness-of-incorporating-long-tern-user-representations-LTUR-and-short-term_fig3_333173219