1. Tầm quan trọng của việc khai thác thông tin tuần tự trong dữ liệu người dùng

Hệ gợi ý tuần tự là một trong những hướng nghiên cứu quan trọng trong lĩnh vực hệ gợi ý, tập trung vào việc khai thác thông tin từ chuỗi hành vi của người dùng để dự đoán hành động tiếp theo. Khác với các hệ gợi ý truyền thống chỉ dựa trên thông tin tĩnh, như lịch sử tương tác tổng quát hoặc các thuộc tính người dùng, hệ gợi ý tuần tự tận dụng các thay đổi động trong sở thích và hành vi người dùng theo thời gian.

Nhờ sự phát triển của học sâu, các phương pháp hiện đại như GRU4Rec, SASRec và BERT4Rec đã cải thiện đáng kể khả năng khai thác thông tin tuần tự:

- **GRU4Rec**: Giúp mã hóa chuỗi sự kiện tuần tự, nhưng còn hạn chế trong việc xử lý chuỗi dài.
- **SASRec**: Loại bỏ hạn chế của RNN bằng cách sử dụng self-attention để nắm bắt các mối quan hệ giữa các sự kiện mà không bị giới hạn bởi khoảng cách.
- **BERT4Rec**: Mở rộng SASRec với khả năng khai thác ngữ cảnh hai chiều, tối ưu hóa thông tin từ cả phía trước và phía sau trong chuỗi.

Việc áp dụng các phương pháp này đã mở ra khả năng gợi ý chính xác và hiệu quả hơn, đặc biệt trong các môi trường thực tế như thương mại điện tử, nơi hành vi người dùng thay đổi nhanh chóng và có tính cá nhân hóa cao.

2. Cấu trúc GRU

Để xử lý vấn đề gradient biến mất hoặc bùng nổ khi chuỗi trở nên quá dài, các biến thể như GRU (Gated Recurrent Unit) và LSTM (Long Short-Term Memory) đã được giới thiệu. Chúng sử dụng các cổng kiểm soát (gates) để điều chỉnh dòng thông tin trong quá trình lan truyền ngược.

GRU sử dụng hai cổng chính, gồm **cổng cập nhật** (z_t) và **cổng xoá bỏ** (r_t), để kiểm soát dòng thông tin trong quá trình cập nhật trạng thái ẩn. Công thức cập nhật trạng thái trong GRU được định nghĩa như sau:

- Cổng cập nhật:

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

Cổng này xác định tỷ lệ thông tin từ trạng thái cũ h_{t-1} cần giữ lại để sử dụng trong trạng thái hiện tại.

- Cổng xóa bỏ:

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

Cổng xoá bỏ kiểm soát mức độ ảnh hưởng của trạng thái trước đó h_{t-1} khi tạo trạng thái mới.

- Trạng thái ứng viên:

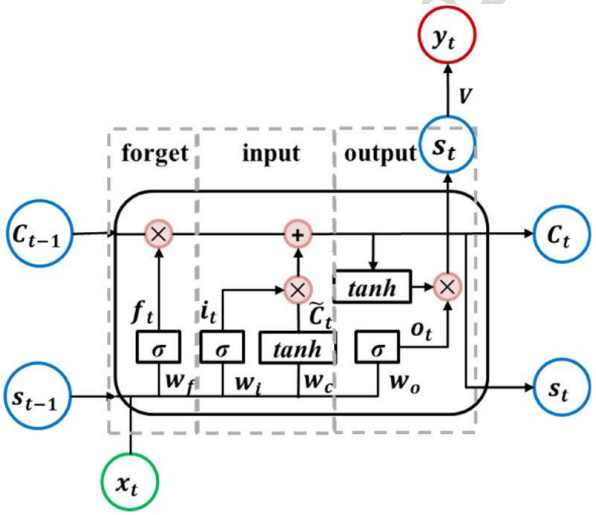
$$\tilde{h}_t = tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h)$$

Trạng thái ứng viên $\tilde{h}_t$ là biểu diễn trung gian, chịu tác động bởi cổng xoá bỏ r_t và thông tin đầu vào x_t.

- Trạng thái ẩn cuối cùng:

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

Trạng thái cuối cùng h_t là sự kết hợp giữa trạng thái trước đó h_{t-1} (được điều chỉnh bởi z_t) và trạng thái ứng viên $\tilde{h}_t$.



- Ở đây:
  - x_t là đầu vào tại thời điểm t (ví dụ: embedding của sản phẩm).
  - h_{t-1} là trạng thái ẩn tại thời điểm trước đó.
  - σ là hàm sigmoid, còn tanh làm hàm kích hoạt phi tuyến.
  - W_z, W_r, W_h là các trọng số cần học.
  - b_z, b_r, b_h là bias.
- Dự đoán đầu ra: Dựa trên trạng thái ẩn h_t, GRU dự đoán phần tử tiếp theo trong chuỗi thông qua một lớp softmax:

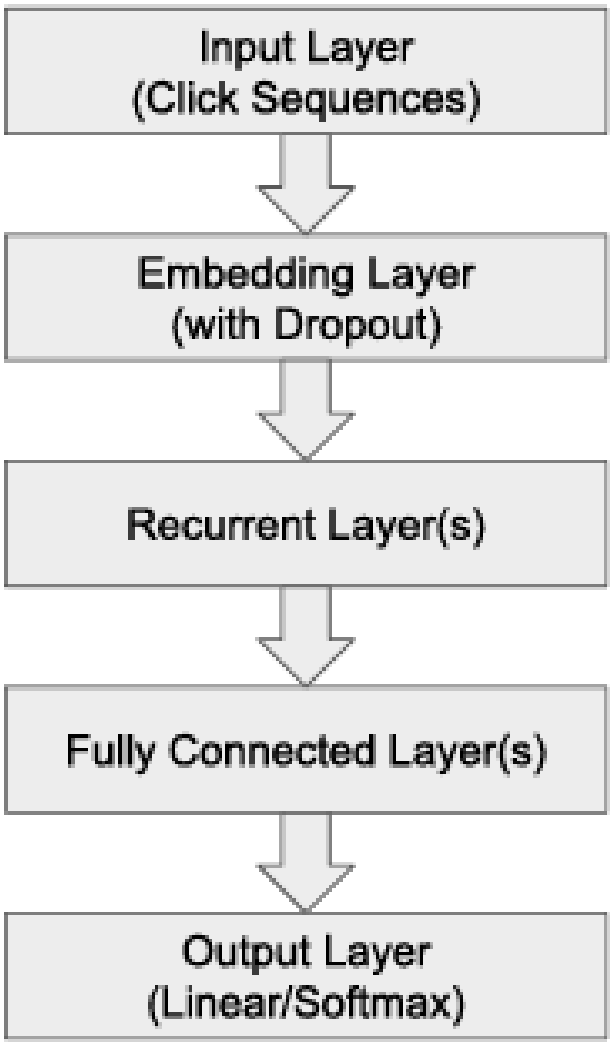
$$y_t = softmax(W_y \cdot h_t + b_y)$$

Hàm mất mát thường được sử dụng là cross-entropy giữa phân phối dự đoán y_t và nhãn thực y_t^*.

3. GRU4Rec

Cấu trúc mạng sử dụng trong GRU4Rec được tổ chức theo các tầng sau:

- Tầng đầu vào (Input Layer): Nhận chuỗi nhấp chuột của người dùng.
- Tầng nhúng (Embedding Layer): Biểu diễn sản phẩm dưới dạng vector nhúng và có thể áp dụng dropout để giảm overfitting.
- Tầng hồi tiếp (Recurrent Layer - GRU): Mô hình hóa thông tin tuần tự dựa trên GRU.
- Tầng fully connected: Hợp nhất thông tin từ trạng thái ẩn của GRU.
- Tầng đầu ra (Output Layer): Có thể sử dụng hàm softmax hoặc linear để dự đoán sản phẩm tiếp theo.



Kiến trúc tổng quát của mạng sử dụng trong GRU4Rec, bao gồm các tầng xử lý từ đầu vào đến đầu ra