GRU4REC

序列化推薦系統

6/21

柏崴

Source

- Session-based recommendations with recurrent neural networks. (ICLR 2016)

GRU4REC

Session-based
Neural network

Recommendation

Why Session based

Session 是伺服器端用來有時序關係的記錄、識別使用者的一種機制。
典型的場景比如購物車，服務端為特定的物件建立了特定的 Session，用於標識這個物件，並且跟蹤使用者的瀏覽點選行為。

傳統推薦演算法

- Content-based
- Collaborative Filtering

每個item互相獨立，沒有辦法考量時序性

如何考量時序性？

傳統解法（1）

- Item to item recommendation: 用item間的相似性預測下一個item

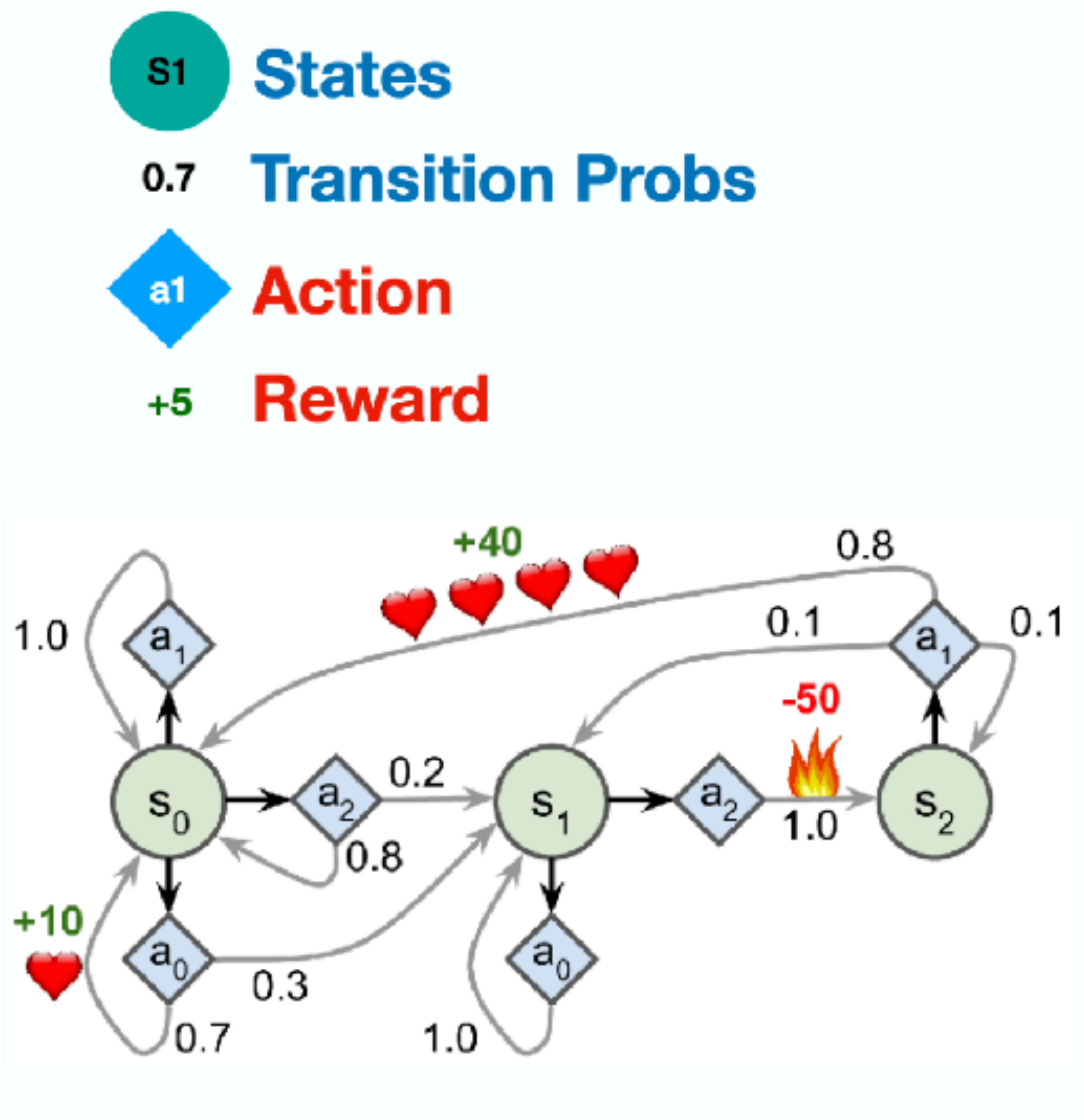

```
For each item in product catalog, I1
  For each customer C who purchased I1
    For each item I2 purchased by
       customer C
      Record that a customer purchased I1
       and I2
  For each item I2
    Compute the similarity between I1 and I2
```

缺點：只考慮最後一個click的item相似性

item-to-item recommendation approach (Sarwar et al.,2001; Linden et al., 2003)

傳統解法（2）

- Markov Decision Processes（MDPs）：馬可夫狀態轉移的四個狀態（S:狀態, A:動作, P:轉移機率, R:獎勵函數）用狀態轉移機率計算點擊下一個item的機率。



MDPs

- 轉移機率(*transition probability*)：
  在目前狀態S_t 為S，目前執行動作A_t為a的情況下，下一期狀態S_{t+1}為s'的機率。

$$P_a(s, s') = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$$

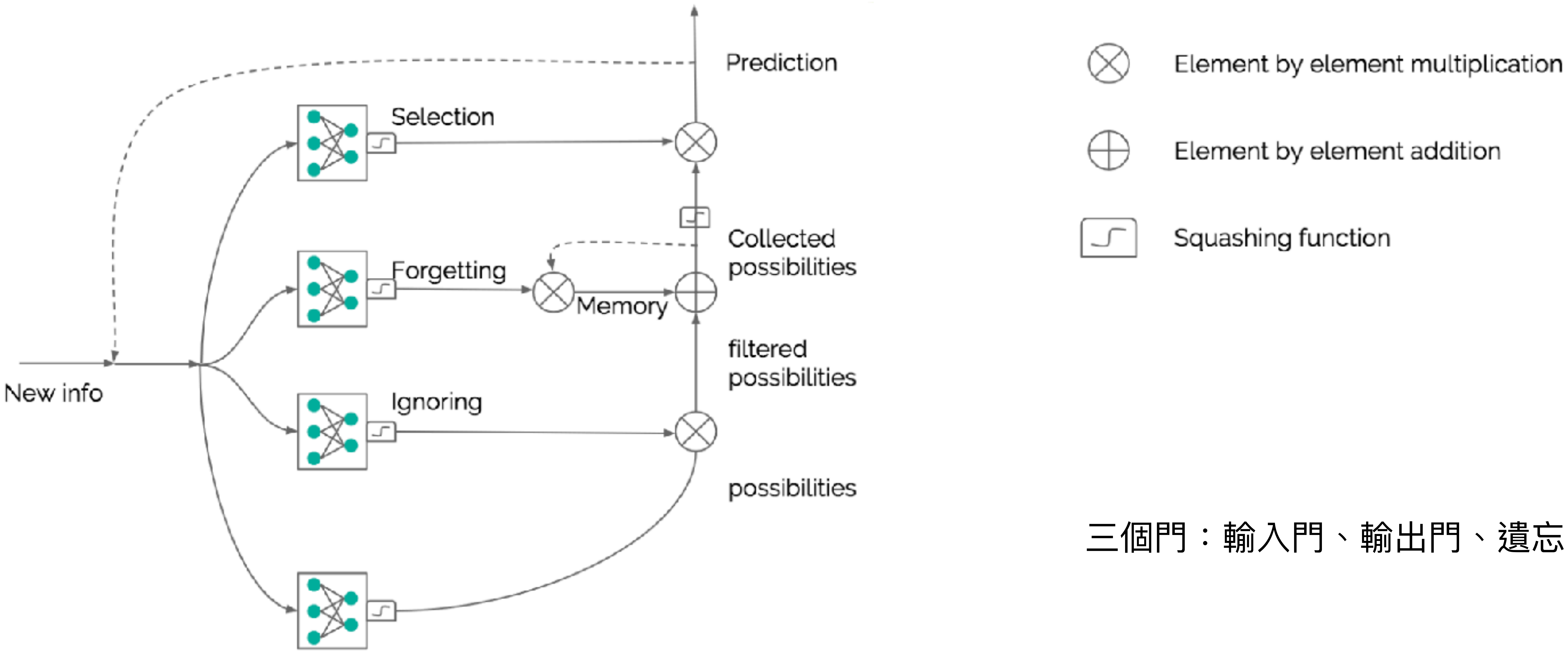
- 報酬函數 (*expected reward*)：
  ，在目前狀態S_t為s，目前執行動作A_t為a，下一期狀態S_{t+1}為s'的情況下，該動作可以得到的期望報酬。(S_{t+1}, $R_a(s)$) 存在聯合分配(*joint distribution*)。

$$R_a(s) = R_{t+1} | S_t = s, a_t = a$$

缺點：狀態數量大，隨問題維度指數增加

Deep Neural Network

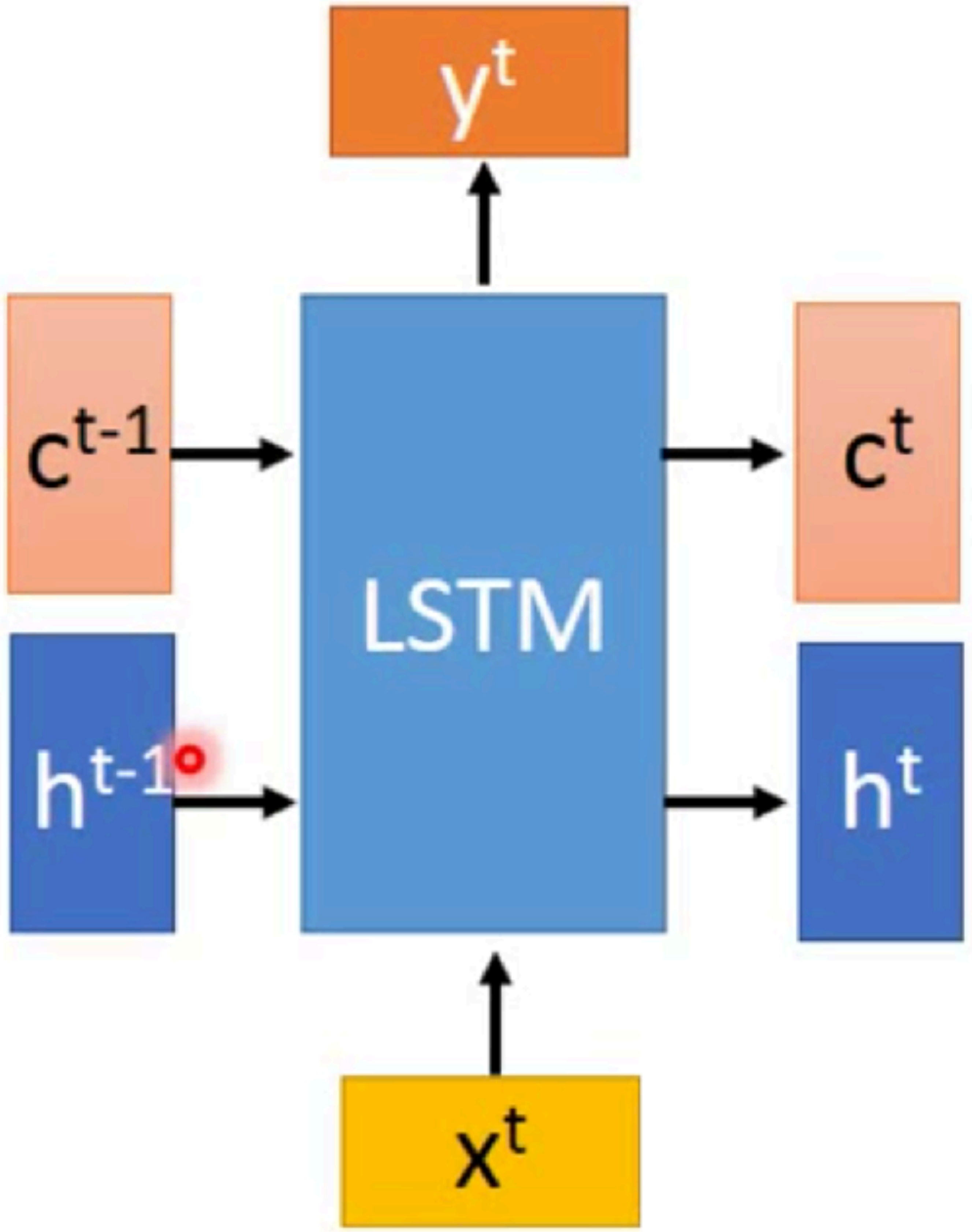
先複習一下LSTM



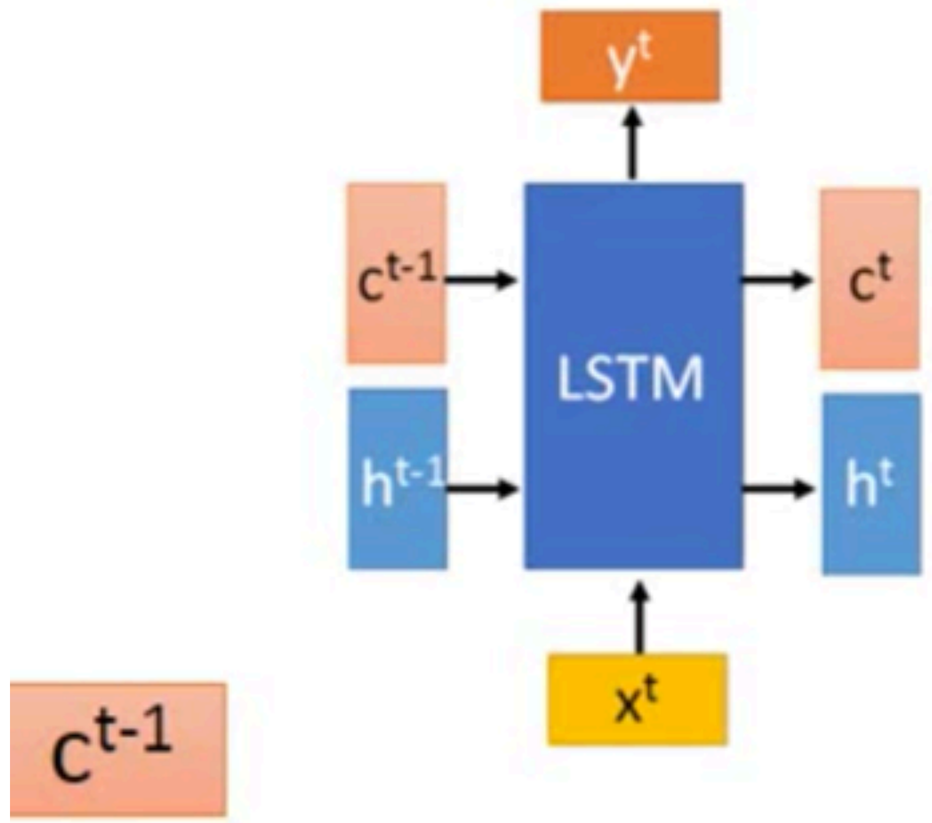
三個門：輸入門、輸出門、遺忘門

LSTM

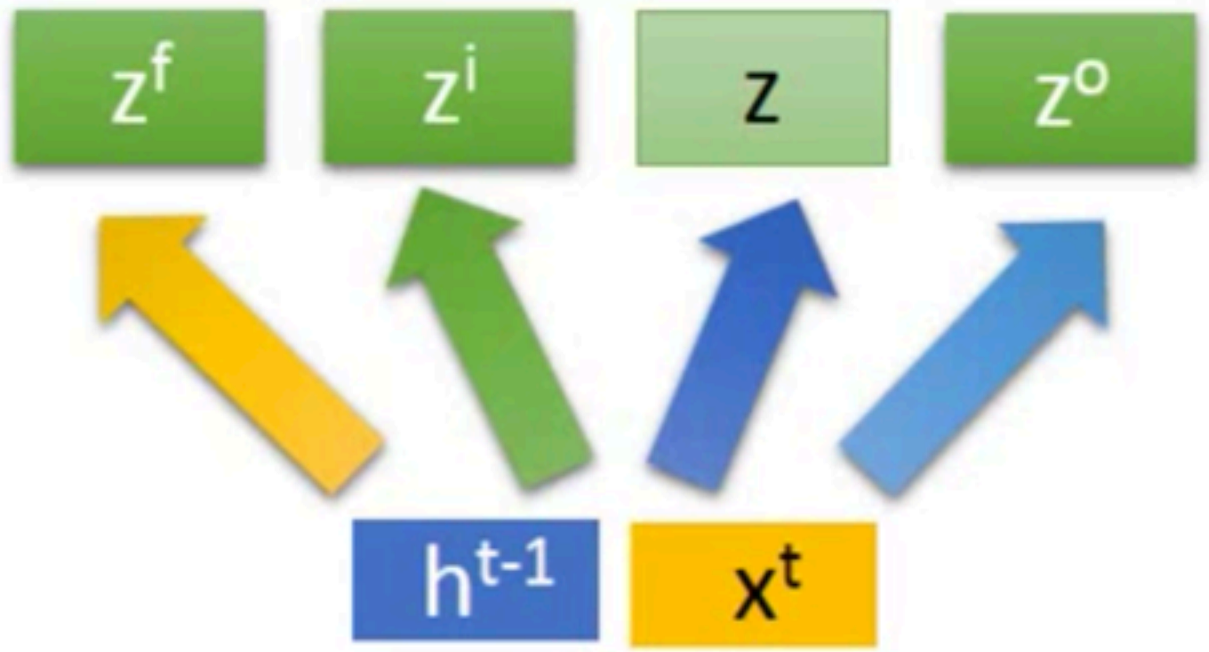
- C 變化緩慢（記憶功能的訊息傳遞）
- h變化較快



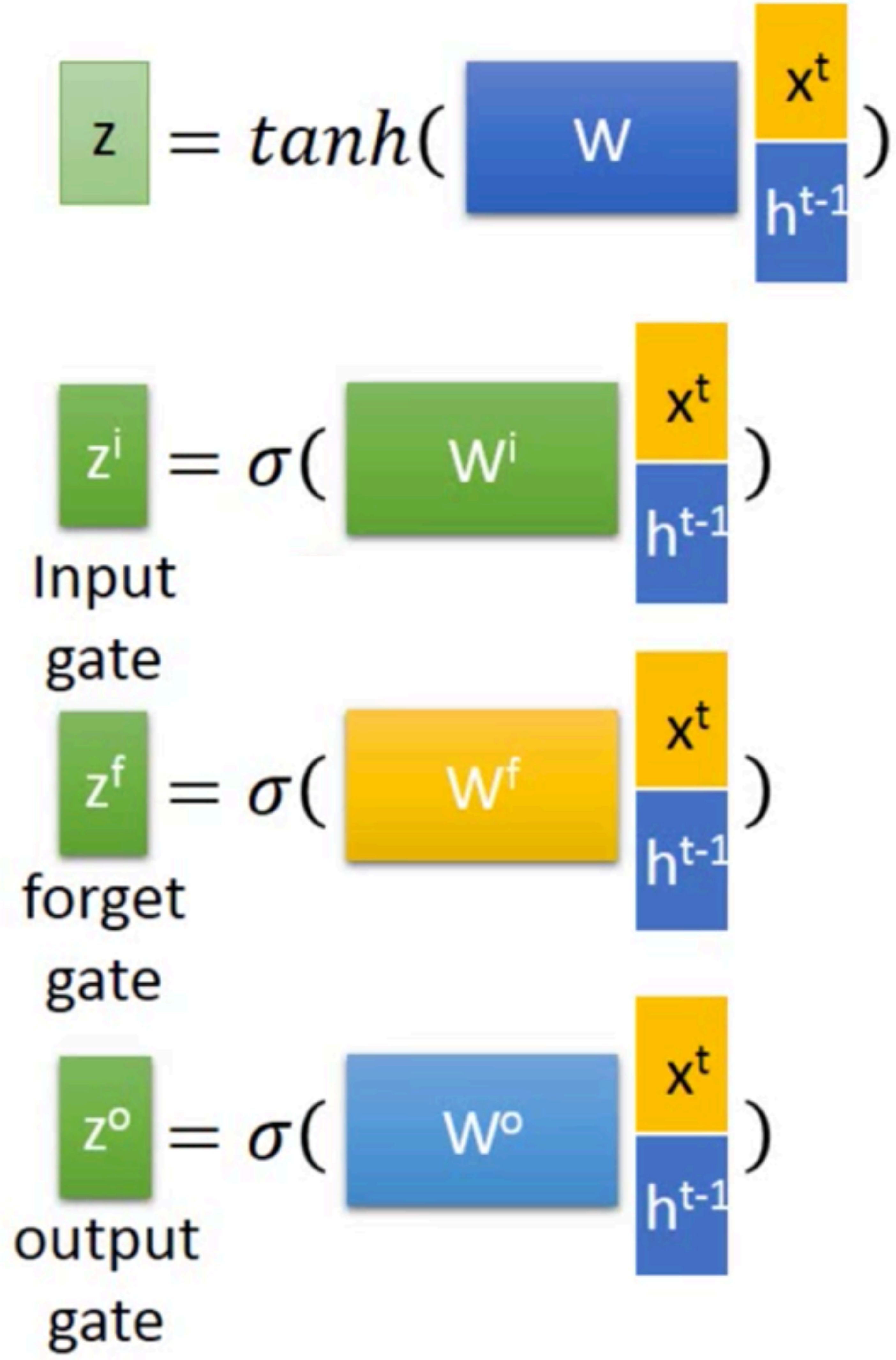
LSTM



y^t
c^{t-1}
LSTM
c^t
h^{t-1}
h^t
x^t
c^{t-1}

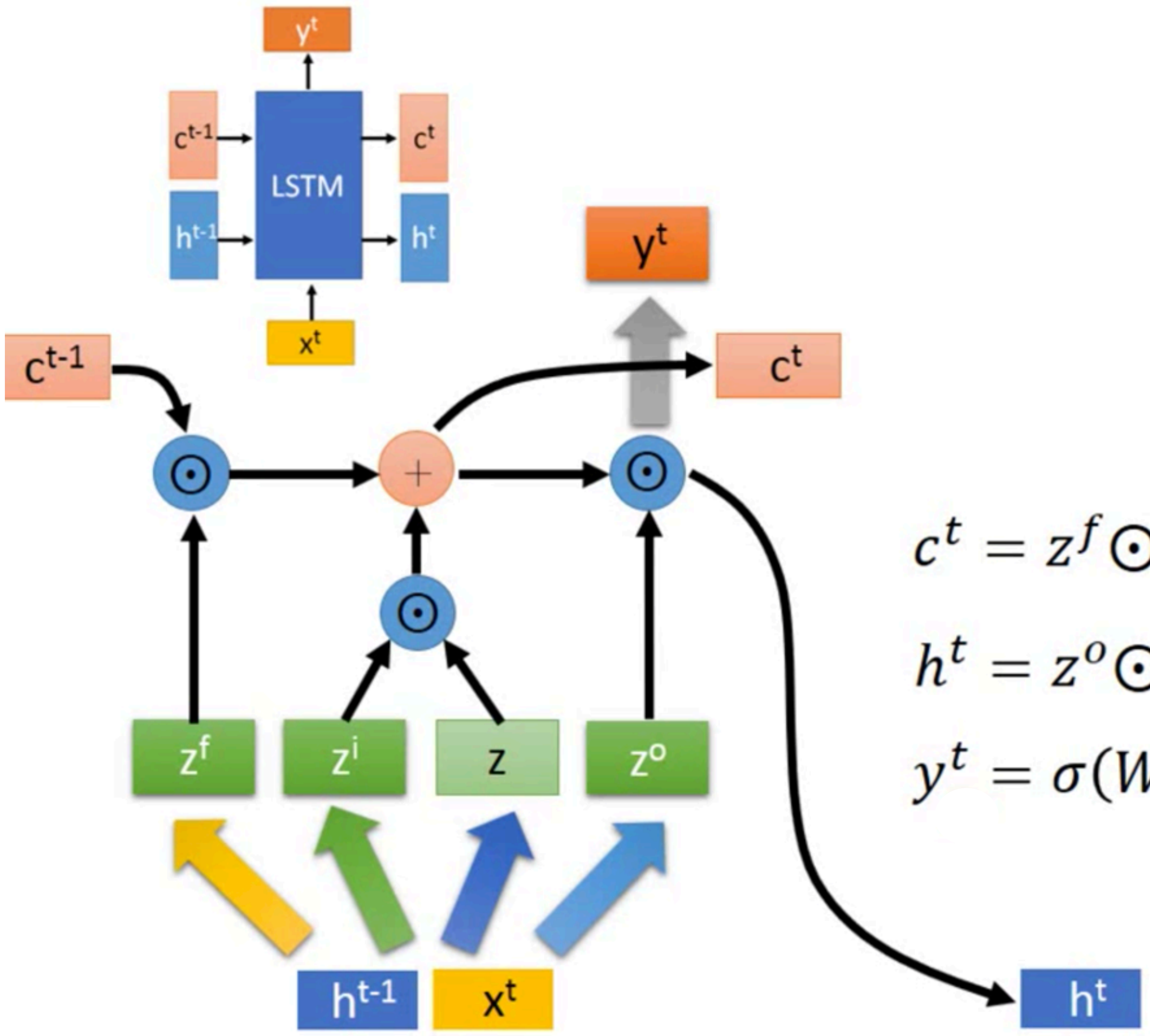


z^f
z^i
z
z^o
h^{t-1}
x^t



z = tanh(W x^t h^{t-1})
z^i = σ(W^i x^t h^{t-1})
Input gate
z^f = σ(W^f x^t h^{t-1})
forget gate
z^o = σ(W^o x^t h^{t-1})
output gate

LSTM

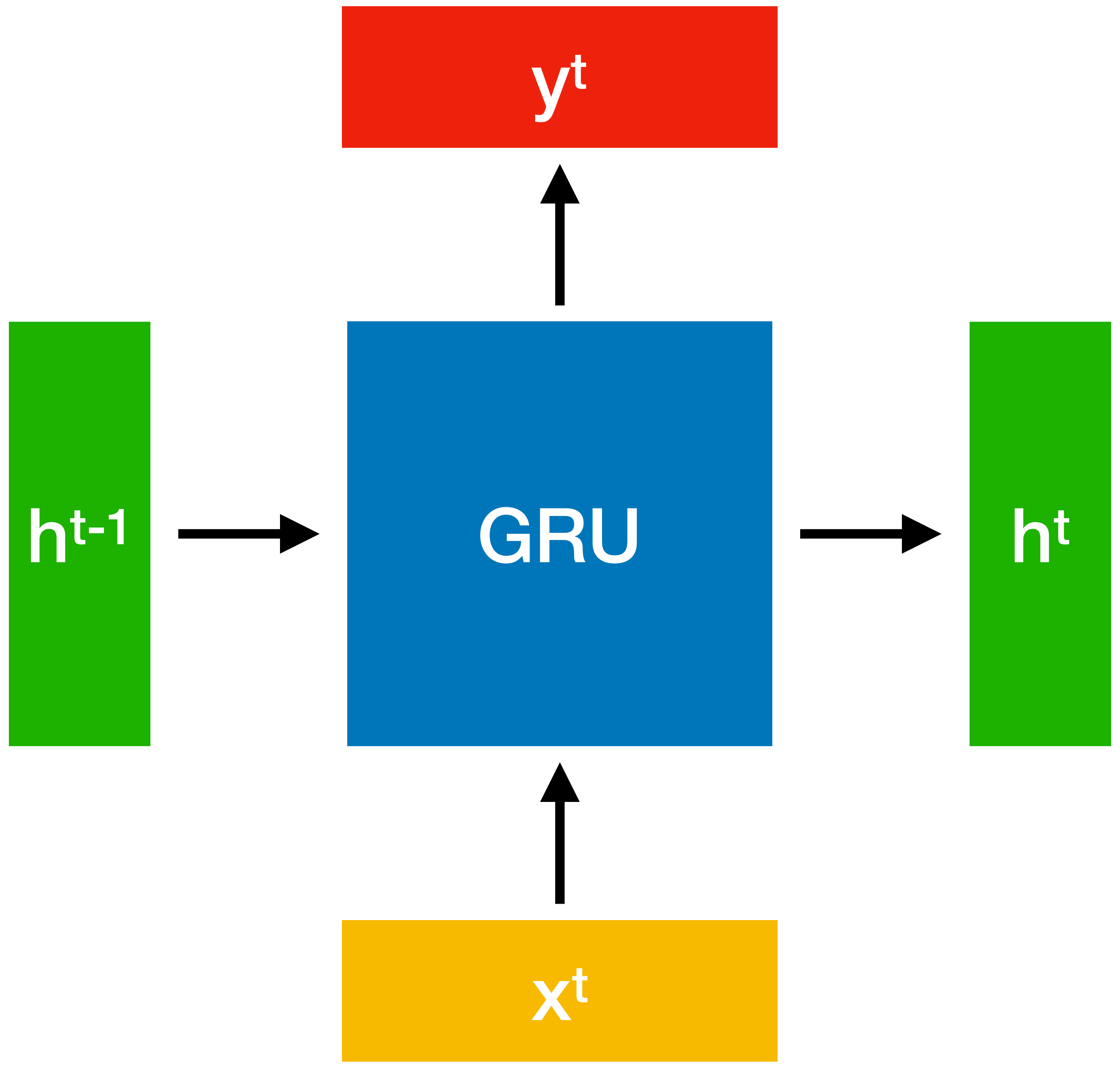


$$c^t = z^f \odot c^{t-1} + z^i \odot z$$

$$h^t = z^o \odot tanh(c^t)$$

$$y^t = \sigma(W' h^t)$$

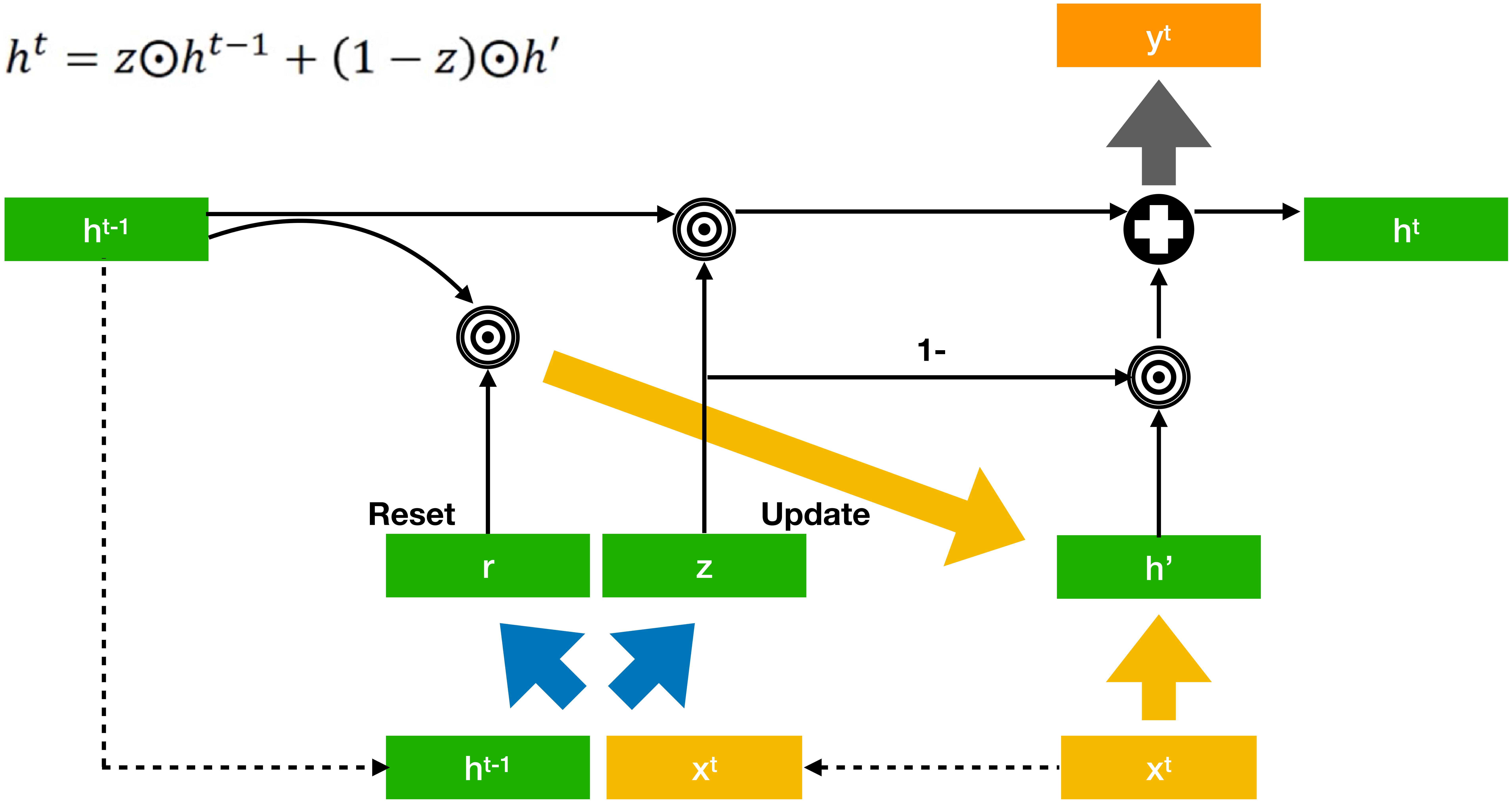
GRU(Gated Recurrent Unit)



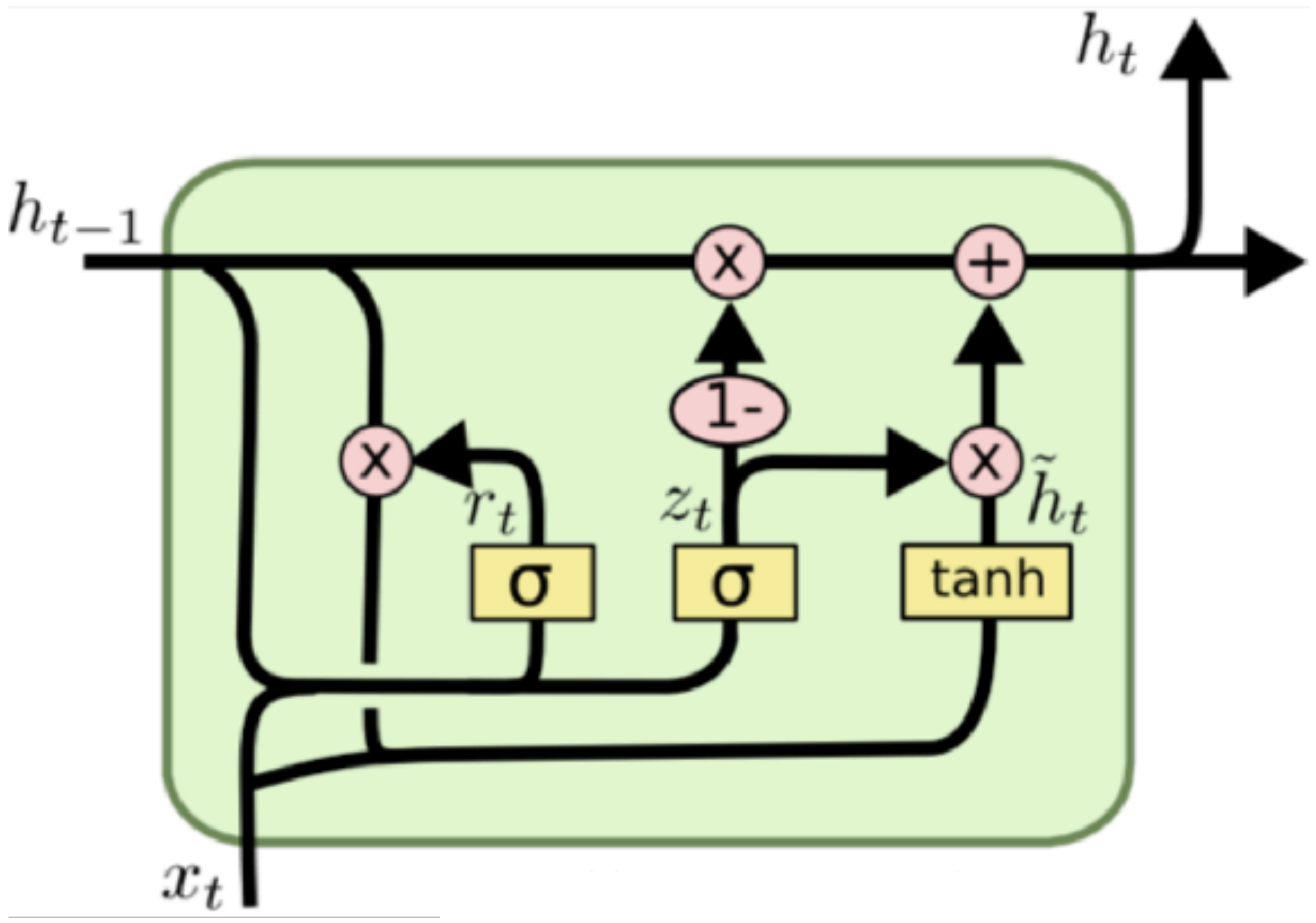
參數較少
輸入輸出較少
但performance不一定比較好

GRU

$$h^t = z \odot h^{t-1} + (1-z) \odot h'$$



GRU



$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$	Update Gated
$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$	Reset Gated
$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$	Candidate hidden layer
$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$	Output Gated

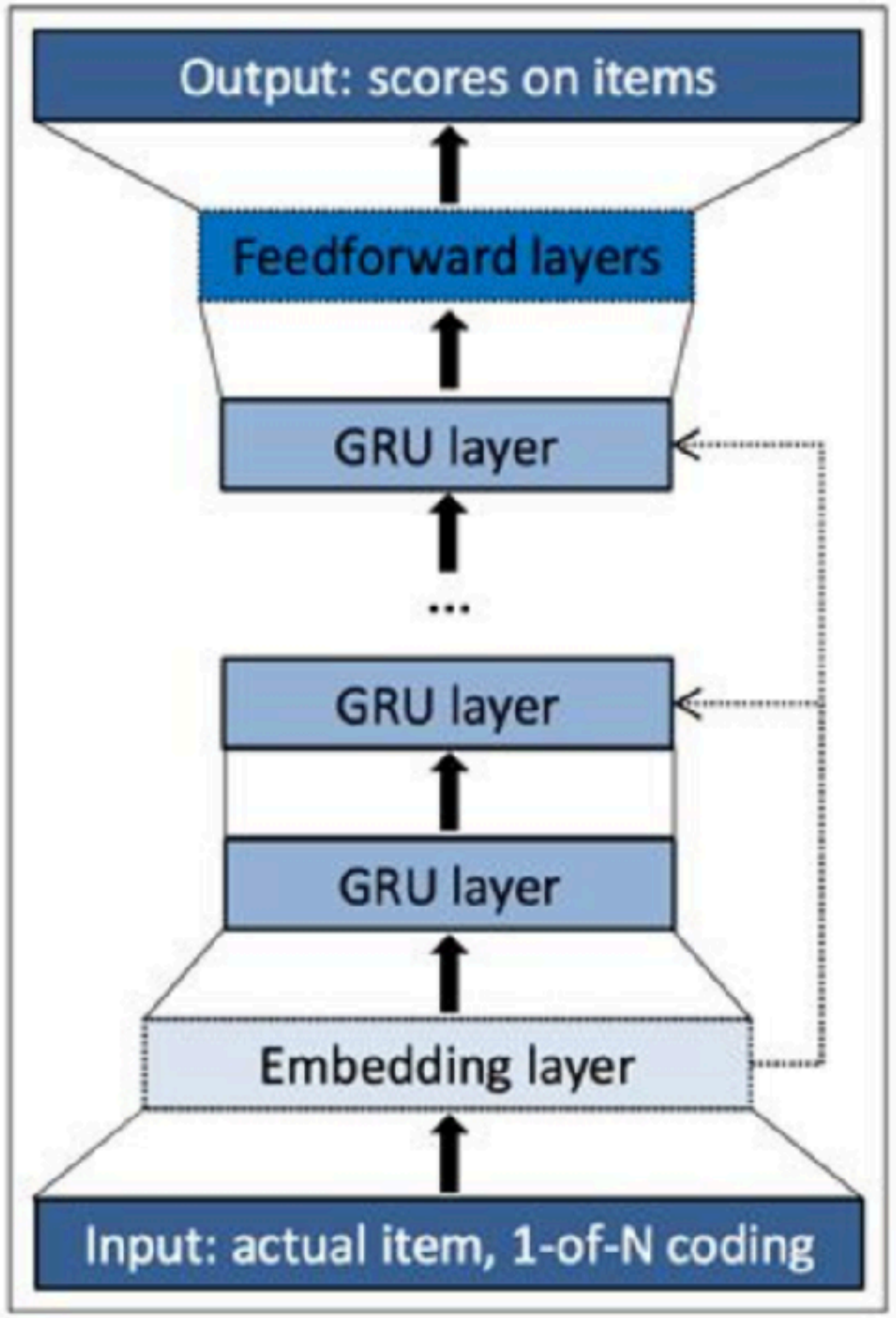
兩個門：更新門、重置門（將LSTM的輸入、遺忘門用更新門代替）

GRU4REC

- 模型架構

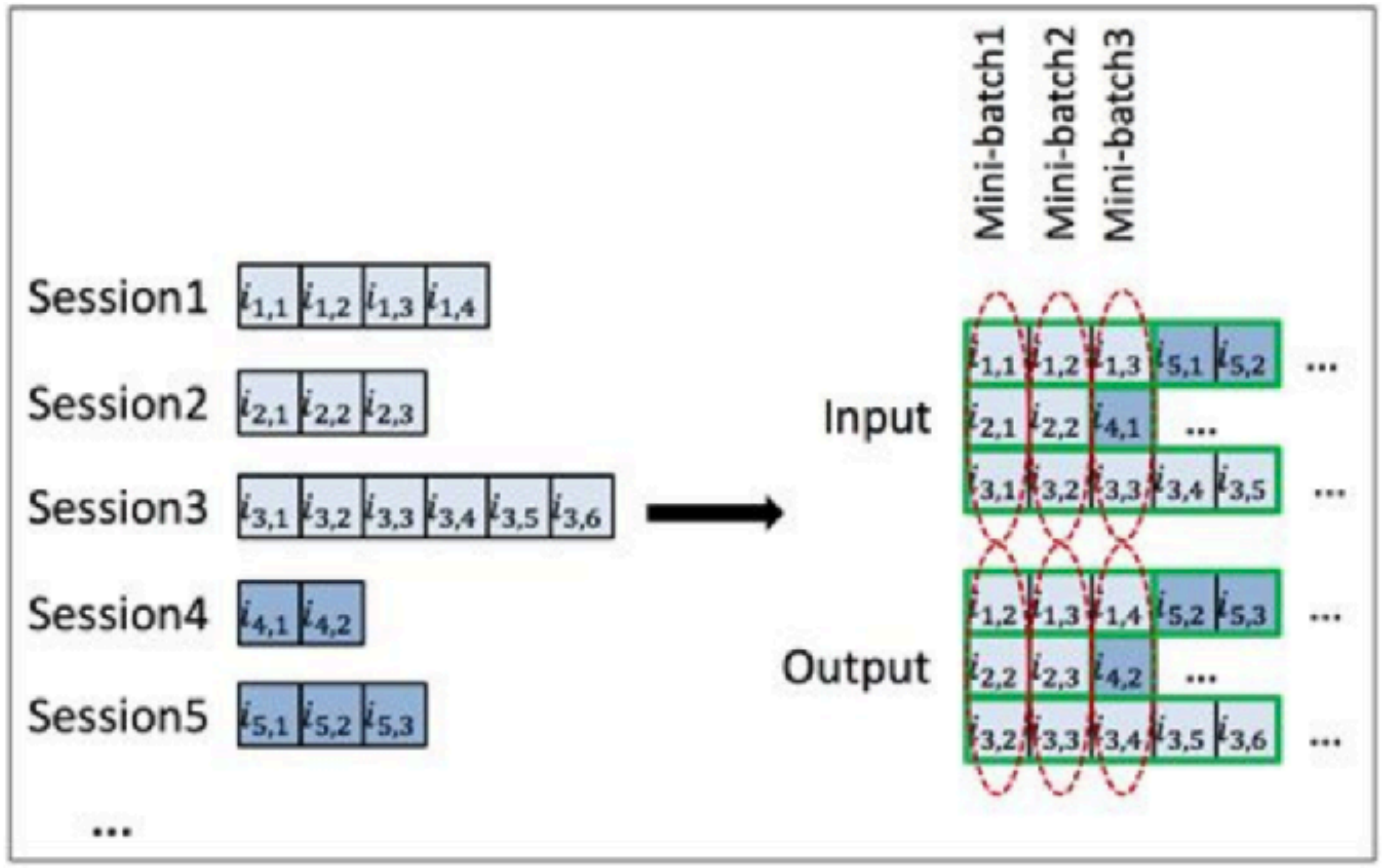
  - Input: Session中的點擊序列$X=[X_1, X_2, ..., X_{r-1}, X_r]$通過One hot encoding，再通過embedding層壓縮為連續低微向量作為GRU輸入

  - Output: 每個Item被點擊的預測機率，$y = M(X)$, where $y=[y_1, y_2, …, y_m]$其中M是模型函數、y_i是item i的預測點擊機率



GRU4REC

- 訓練策略（提高訓練效率、簡化訓練代價）
  - Training Strategy: Mini-batch
  - Training data sample: 選取當前正樣本（下一個點擊的item）加上隨機抽取的負樣本（選取同一個mini-batch中其他sequence下一個點擊的item作為負樣本）



GRU4REC

- **Loss Function**
  - Point-wise ranking loss
  - Pair-wise ranking loss:
    - BPR: 基於貝葉斯理論的矩陣分解
    - TOP1: 一種正則估計

BPR

$$L_s = -\frac{1}{N} * \sum_{j=1}^{N_s} log(\sigma(\tilde{y}_{s,i} - \tilde{y}_{s,j}))$$

Ns: Sample size

$y_{s,k}$: *item k* 在當前*session*的分數

*i: session*中的下一個*item*

j: 負樣本

Bayesian Personalized Ranking(BPR)概念

- 背景：如果商品的數量非常多且用戶非常少，我們關心的是哪些極少數的商品在用戶心中的排序=> 排序問題

u

i

j

<u, i, j> 表示user對i的優先級高於j，假設有m個用戶、n個用品
訓練成果是兩個分解後的矩陣W和H：
W的維度：m x k
H的維度：n x k
其中k是一個自定義的維度。
對於用戶u，商品i的排序評分為 $\bar{x}_{ui} = w_u \bullet h_i = \sum_{f=1}^{k} w_{uf} h_{if}$

目標：找到合適的矩陣W, H讓$\bar{X}$和X相似

Top1

Paper設計的排序損失計算方式，加入類似L2正則化的誤差項

$$L_s = \frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \sigma\left(\hat{r}_{s,j} - \hat{r}_{s,i}\right) + \sigma\left(\hat{r}_{s,j}^2\right)$$

實驗結果

POP: 推薦最受歡迎的item

S-POP: 推薦當前session最受歡迎的item

Item-KNN: 推薦與實際item相似的item，session向量間的cos similarity

Table 1: Recall@20 and MRR@20 using the baseline methods

Baseline	**RSC15**		**VIDEO**	
	Recall@20	MRR@20	Recall@20	MRR@20
POP	0.0050	0.0012	0.0499	0.0117
S-POP	0.2672	0.1775	0.1301	0.0863
Item-KNN	0.5065	0.2048	0.5508	0.3381
BPR-MF	0.2574	0.0618	0.0692	0.0374

Loss / #Units	**RSC15**		**VIDEO**	
	Recall@20	MRR@20	Recall@20	MRR@20
TOP1 100	0.5853 (+15.55%)	0.2305 (+12.58%)	0.6141 (+11.50%)	0.3511 (+3.84%)
BPR 100	0.6069 (+19.82%)	0.2407 (+17.54%)	0.5999 (+8.92%)	0.3260 (-3.56%)
Cross-entropy 100	0.6074 (+19.91%)	0.2430 (+18.65%)	0.6372 (+15.69%)	0.3720 (+10.04%)
TOP1 1000	0.6206 (+22.53%)	**0.2693 (+31.49%)**	**0.6624 (+20.27%)**	**0.3891 (+15.08%)**
BPR 1000	**0.6322 (+24.82%)**	0.2467 (+20.47%)	0.6311 (+14.58%)	0.3136 (-7.23%)
Cross-entropy 1000	0.5777 (+14.06%)	0.2153 (+5.16%)	–	–

Reference

- Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling
- Session-based recommendations with recurrent neural networks. (ICLR 2016)
- Gated RNN and Sequence Generation (Recorded at Fall, 2017)