Deep And Shallow Learning In Recommendation Systems

Gil Chamiel, Director of Data Science and Algorithm Engineering

October 2018

Outline

Problem setup

Challenges

Modeling

Case study

Problem Setup

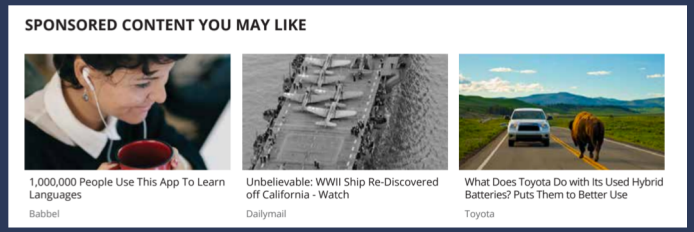
Millions of Possible Recommendations

Rank N recommendations by *CTR * CPC * f(Post-Click Engagement)*

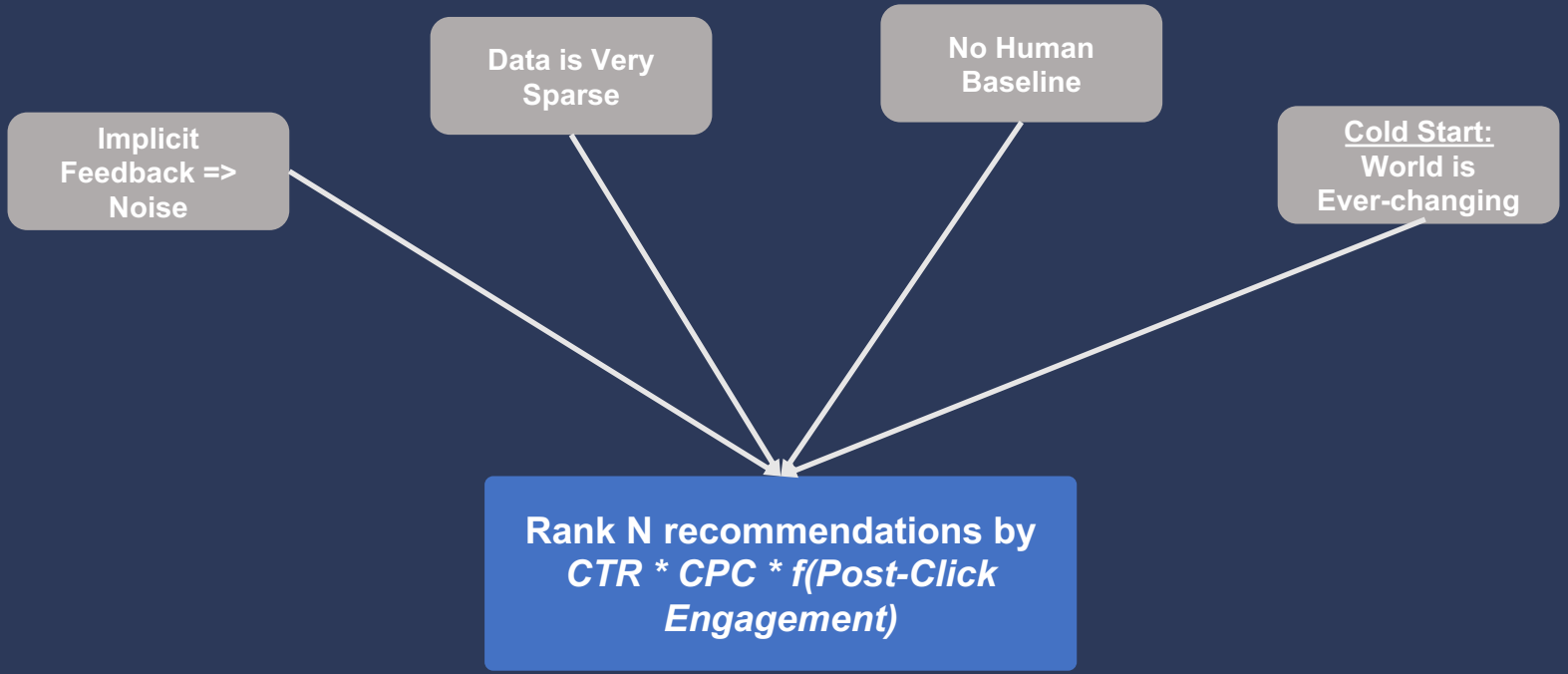
Context Metadata

Location Region-based Information

User Historical Data



Machine Learning + Recommendations = Hard



The Cold Start Problem

- Hard cold start: introduction of a new entity
- Soft cold start: existing entity in a new situation
- In Taboola, both issues are common
  - Borrow strength from similar informative situations

Matrix Factorization

$$loss = \frac{1}{n}\sum_{i,j\ \epsilon\ D}\left(r_{ij} - \overrightarrow{u_i}\overrightarrow{m_j}\right)^2 + reg_{trem}$$

	User 1	User 2	User 3	User 4	User 5
Movie 1	7.5	2.2	?	?	8.2
Movie 2	?	?	8.3	?	7
Movie 3	?	7.9	0.3	?	?
Movie 4	9.9	?	?	10	?
Movie 5	?	8.2	3.4	?	6.5

Matrix Factorization

Item

User

	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

Rating Matrix

=

A	*1.2*	*0.8*
B	*1.4*	*0.9*
C	*1.5*	*1.0*
D	*1.2*	*0.8*

User Matrix

X

W	X	Y	Z
1.5	*1.2*	*1.0*	*0.8*
1.7	*0.6*	*1.1*	*0.4*

Item Matrix

Limitation of Matrix Factorization

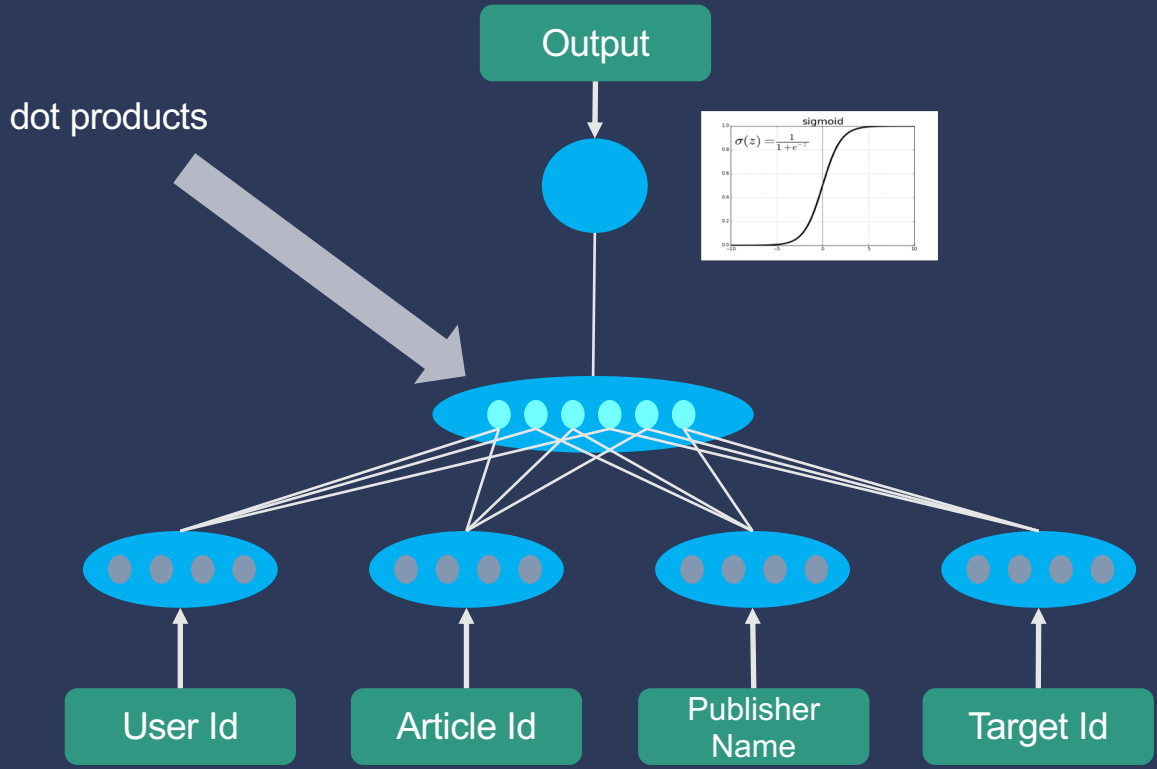
Or why the Netflix solution doesn't work for Taboola

- It doesn't handle cold start very well
  - For instance, input is sparse
- Input is complex and diverse
  - Could be sequential, textual, media, etc.
- In the Netflix challenge, feedback is explicit
- Need for additional generalizing features

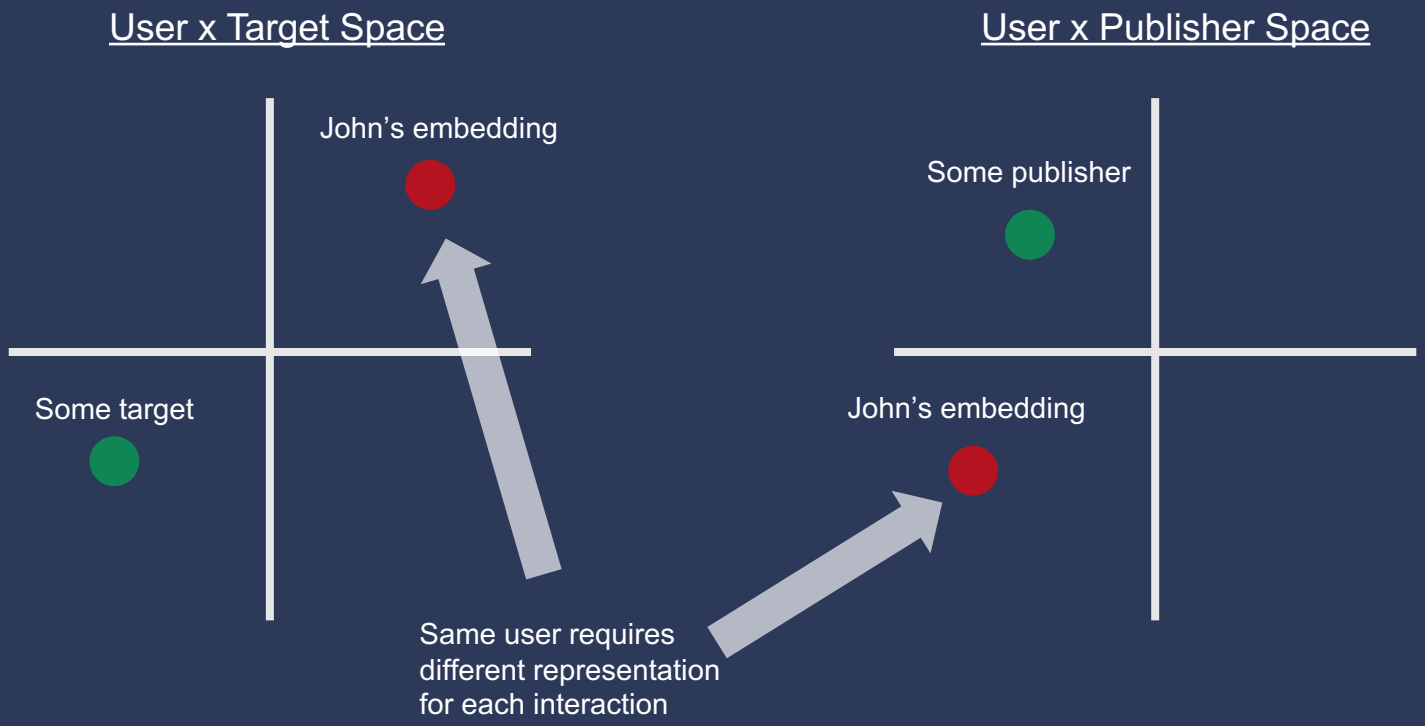
Factorization Machines

- Can factorize many pairs (i.e. features)
- Each feature gets embedding (its latent representation)
- $\emptyset_{FM}(w, x) = \sum_{i,j \in x} w_i \cdot w_j$
- n = number of features, k = latent feature vector length
- Works with any loss

Factorization Machines



Limitation of Factorization Machines



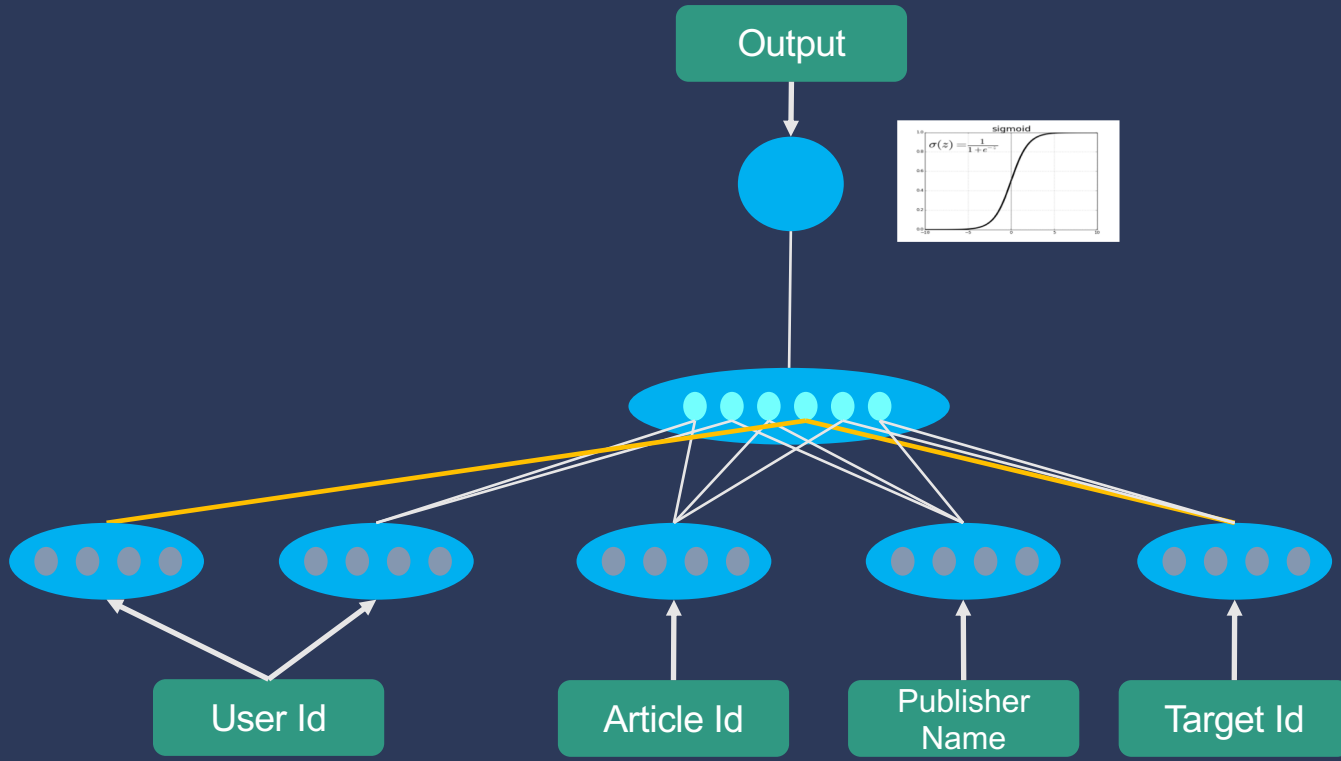
Field-aware Factorization Machines (FFM)

- Separate embedding for different interactions
- $Nike * usatoday.com \; vs \; Nike' * User_j$
- $K_{ffm} \ll K_{fm}$

$$\phi_{\text{FFM}}(\boldsymbol{w}, \boldsymbol{x}) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\boldsymbol{w}_{j_1,f_2} \cdot \boldsymbol{w}_{j_2,f_1}) x_{j_1} x_{j_2}$$

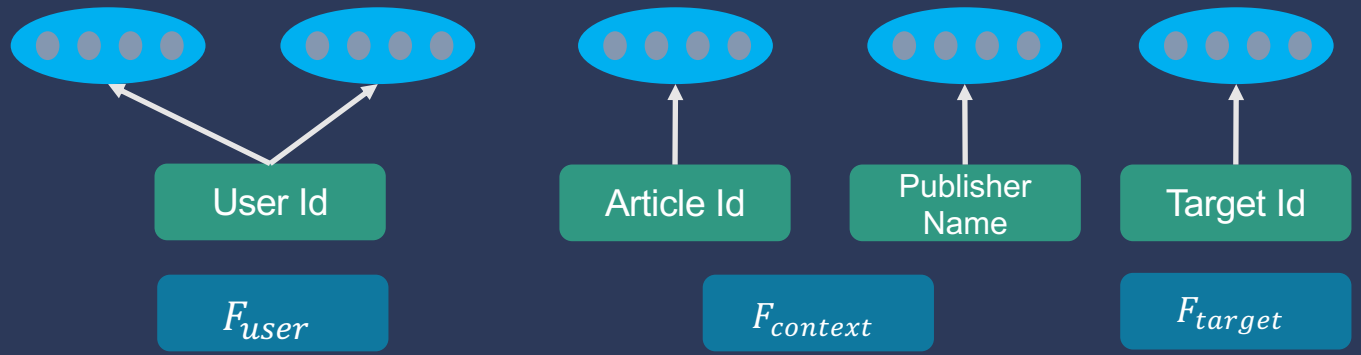
Field-aware Factorization Machines for CTR Prediction, RecSys 2016

Factorization Machines



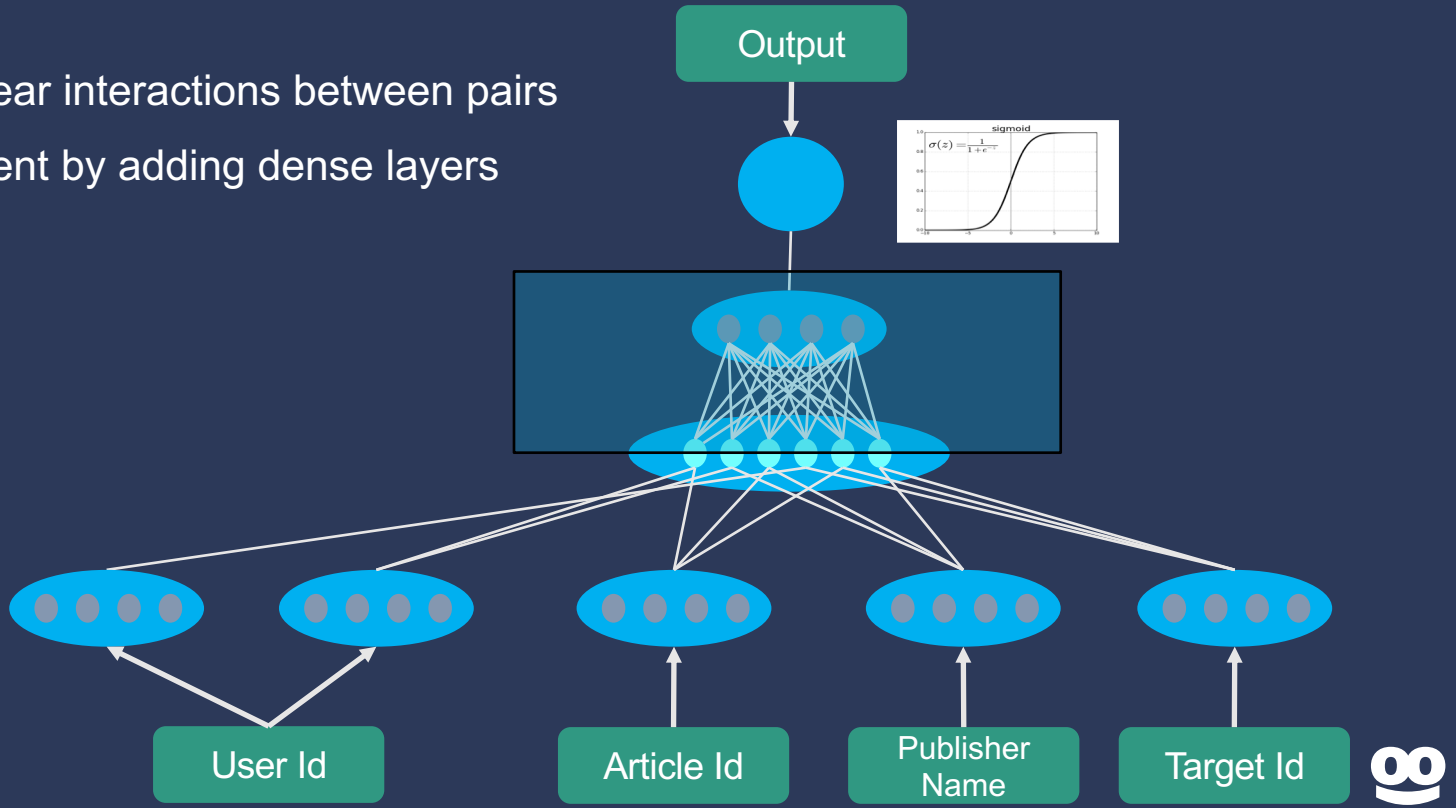
What's a Field in FFM?

- Number of variables $n * f * k$
- $f = number\ of\ fields$
- Determine feature families, e.g. Target, User, Context
- f Embedding for each feature
- $Nike_{user}, Nike_{context}, Nike_{target}$



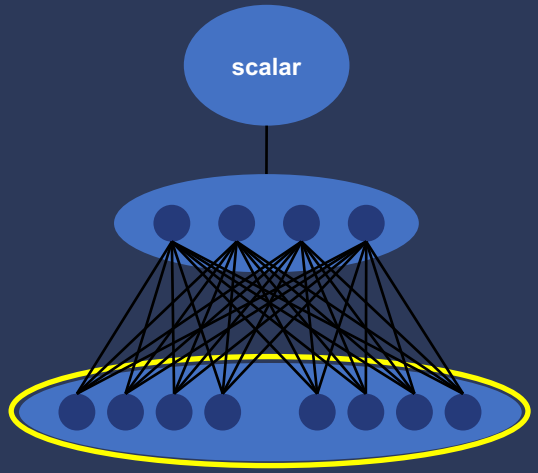
Representing additional relations in FFM

- Capture non-linear interactions between pairs
- Easy to implement by adding dense layers

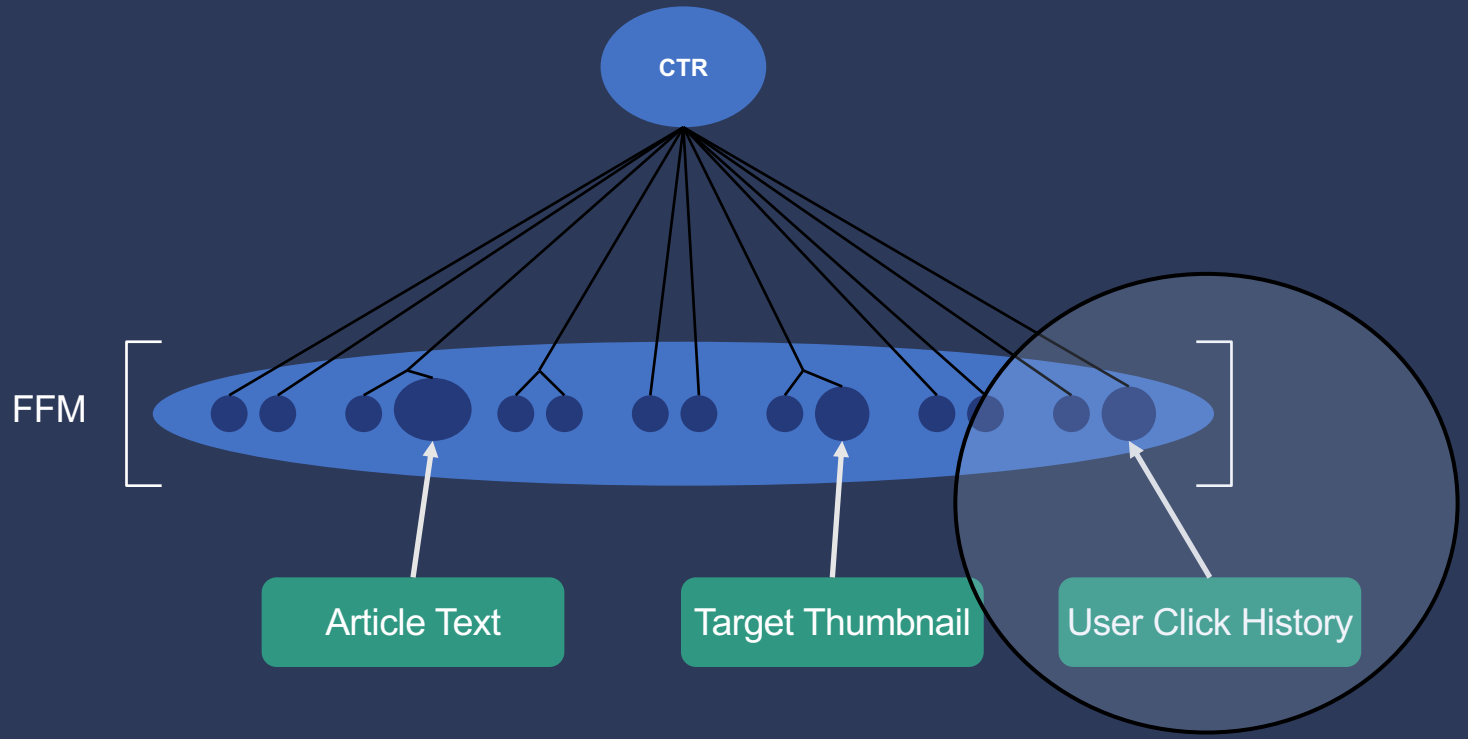


Handling numerical feature in FFM

- Option 1: Bucketing (not a trivial task)
- Option 2: Vectorization via small deep net

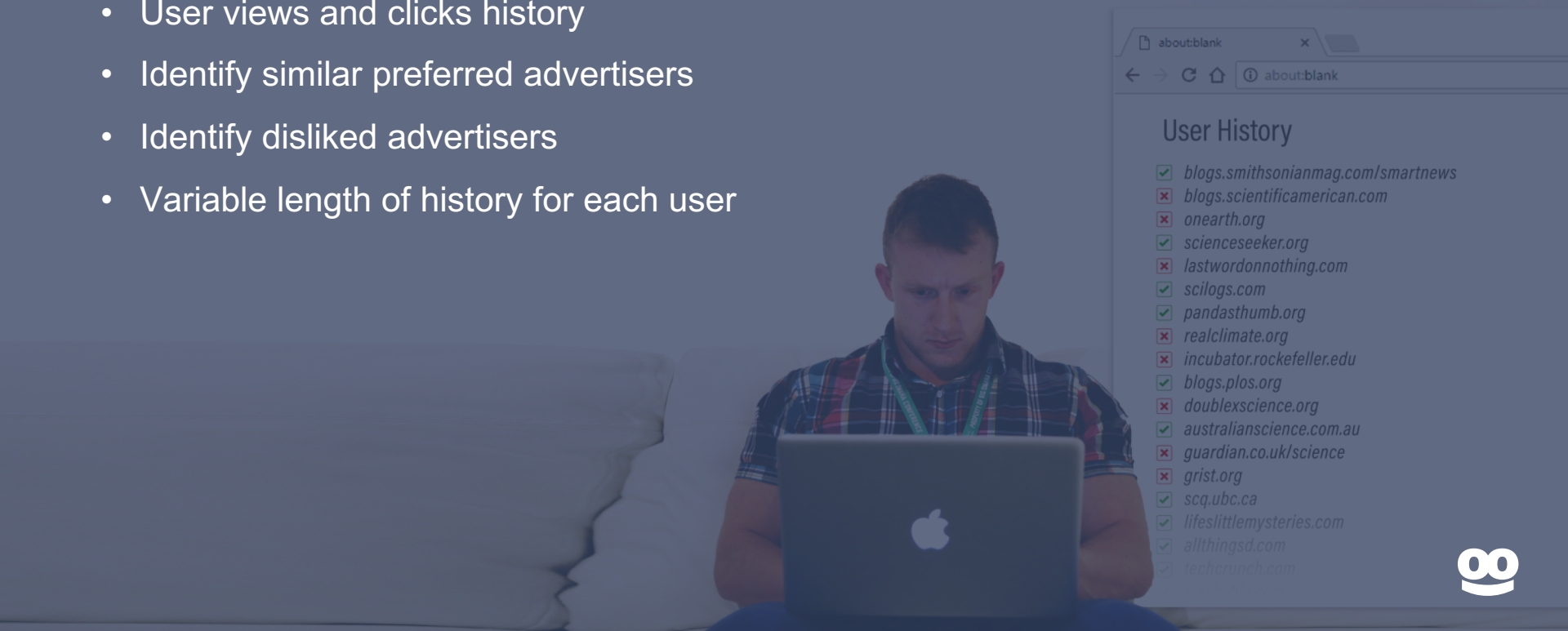


Deep & Shallow Learning



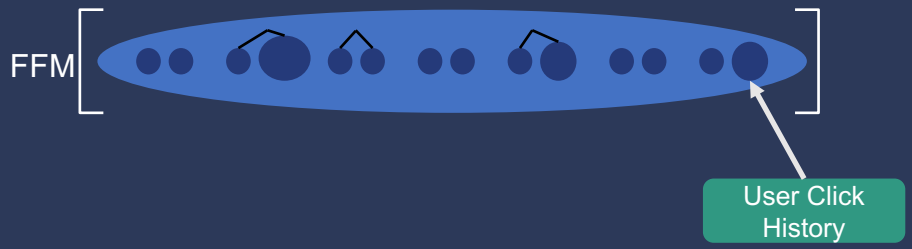
Case study – user advertiser history

- User views and clicks history
- Identify similar preferred advertisers
- Identify disliked advertisers
- Variable length of history for each user



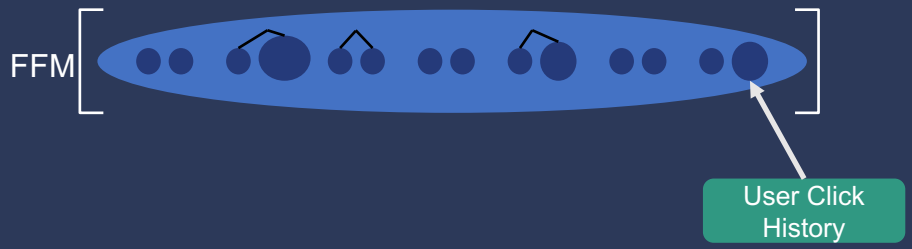
Case study - input

- $[a_1, a_2, \ldots . . a_n]$ - Nike, coca cola
- $[c_1, c_1, \ldots . . c_n]$ – Number of clicks
- $[v_1, v_2, \ldots . . v_n]$ – Number of views



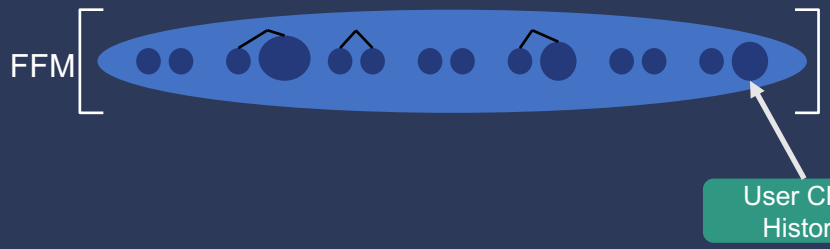
We could one hot encode…

- $[1,0,1,0,\ldots\ldots,1]$ - ids
- $[2,0,0,0,\ldots\ldots,1]$ – clicks
- $[23,0,3,0,\ldots\ldots,5]$ - views



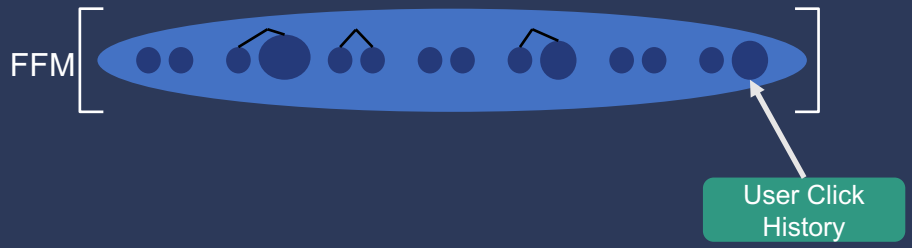
Embedding to the rescue (BoW style)

- $[a_1, \ldots a_n]$ => $[\overrightarrow{a_1}, \ldots . \overrightarrow{a_n}]$
- $res = \frac{1}{n}\sum_1^n A_i$
- Filter non-clicks, filter OOV
- $sign_i = c_i > 0 \,?\, 1 : -1$
- $res = \frac{1}{n}\sum_1^n sign_i * A_i$



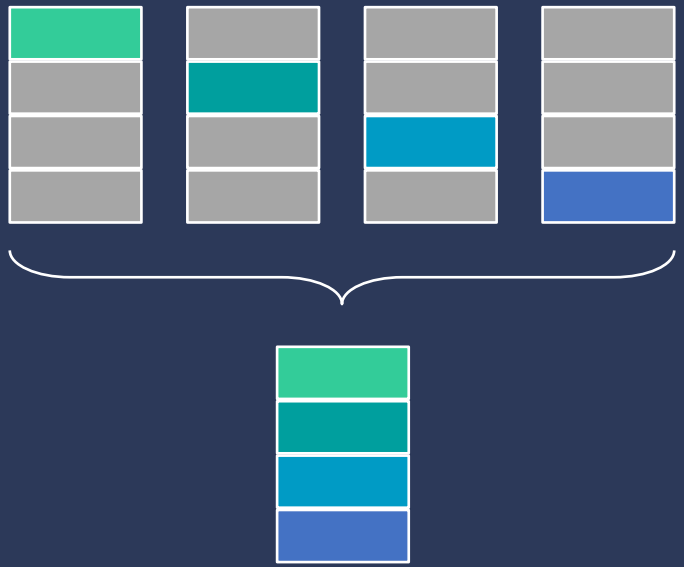
Integrating clicks & views

- $[a_1, \dots a_n]$ => $[\overrightarrow{a_1}, \dots . \overrightarrow{a_n}]$
- $[c_1, \dots c_n], [v_1, \dots v_n]$
- $res = \sum_1^n \frac{Ci}{V_i} * A_i \ * norm_{factor}$
- Integrating priors $p_c, p_v \ - network\ weights$
- $res = \sum_1^n \frac{Ci+p_c}{V_{i+p_v}} * A_i \ * norm_{factor}$



Max pooling

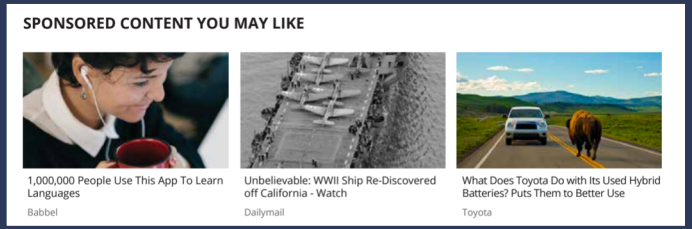
- Extract the max from each dimension of the latent vector



And Many Other Ways..

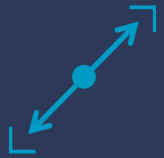
- Using an attention layer (learn a weighted average)
- Using an RNN (if time matters)
- Take user's favorite embedding

- Could model text/image/video input
  - Using your deep subnet of your choice



Summary

Recommendation system in a dynamic environment

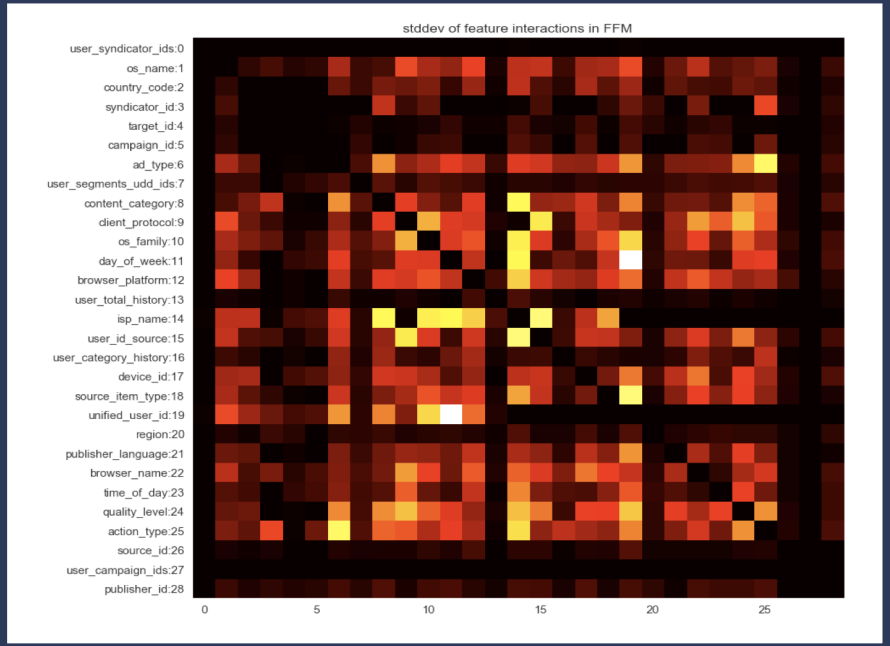


Deep learning – not one size fit all

Modeling matters – modularity is king

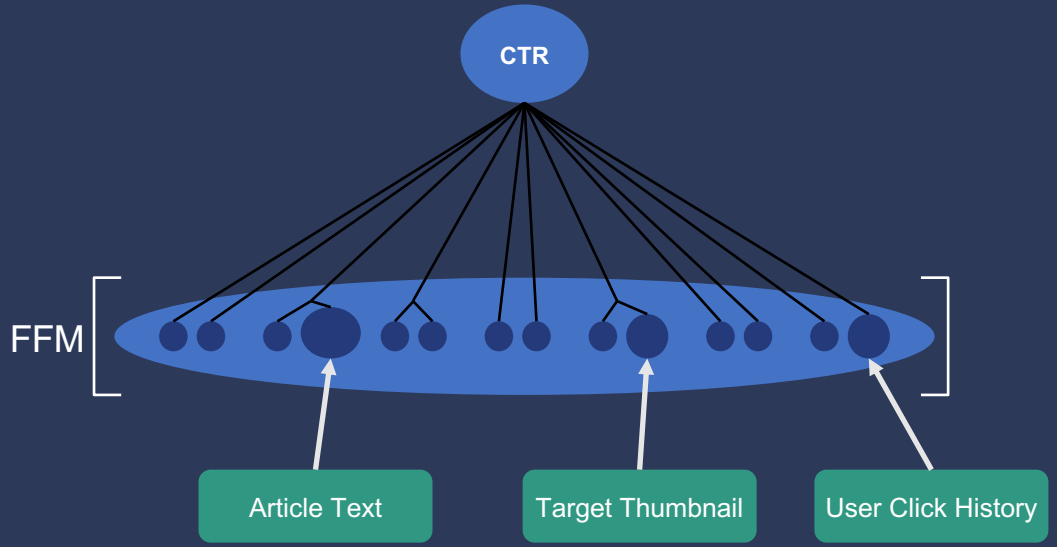
Interpretable FFM

- Using a simplified model for analysis
  - Without intermediate dense layers
- TensorFlow makes it easy…



Additional Architecture

- Plain fully connected deep network
- Deep FM
- Wide and Deep



Thank You!

Questions?

Where is this used in Taboola

- Dataset of ~40M rows
- Goals: CTR and CVR prediction
- Tensorflow over GPU for training
- Tensorflow Serving for serving
- 1T estimations per day