Neural Factorization Machines for Sparse Predictive Analytics

Xiangnan He, Tat-Seng Chua

Research Fellow

School of Computing
National University of Singapore

9 August 2017 @ SIGIR 2017, Tokyo, Japan

Sparse Predictive Analytics

- Many Web applications need to model categorical variables.
  - Search ranking: <query (words), document (words)>
  - Online Advertising: <user (ID+profiles), ads (ID+words)>

How to bridge the representation gap?

One-hot Encoding => Sparse Feature Vectors

- Standard supervised learning techniques deal with a numerical design matrix (feature vectors):
  - E.g., logistic regression, SVM, factorization machines, neural networks …

Linear/Logistic Regression (LR)

- Model Equation: $$\hat{y}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} = \sum_{i=1}^{n} w_i x_i$$
- Example:

Publisher	Advertiser
ESPN	Nike

$$s = w_{\text{ESPN}} + w_{\text{Nike}}$$

- Drawback: Cannot learn cross-feature effects like:

"Nike has super high CTR on ESPN"

Lab for Media Search

Example is adopted from:
Juan et al. WWW 2017. Field-aware Factorization Machines in a Real-world Online Advertising System

Degree-2 Polynomial Regression (Poly2)

- Model Equation: $\hat{y}(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{i,j} x_{i,j}$

- Example:

Publisher	Advertiser
ESPN	Nike

$$s = w_{\text{ESPN}} + w_{\text{Nike}} + w_{\text{ESPN,Nike}}$$

- Drawback: Weak generalization ability – cannot estimate parameter $w_{i,j}$ where *(i,j)* never co-occurs in feature vectors.

Lab for Media Search

Example is adopted from:
Juan et al. WWW 2017. Field-aware Factorization Machines in a Real-world Online Advertising System

Factorization Machine (FM)

- Model Equation: $\hat{y}(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathbf{v}_i^T \mathbf{v}_j \cdot x_i x_j,$
- Example:

Publisher	Advertiser
ESPN	Nike

$S = w_{ESPN} + w_{Nike} + \langle \boldsymbol{v}_{ESPN}, \boldsymbol{v}_{Nike} \rangle$

- Another Example:

Publisher (P)	Advertiser (A)	Gender (G)
ESPN	Nike	Male

$S = w_{ESPN} + w_{Nike} + w_{Gender} + \langle \boldsymbol{v}_{ESPN}, \boldsymbol{v}_{Nike} \rangle + \langle \boldsymbol{v}_{ESPN}, \boldsymbol{v}_{Male} \rangle + \langle \boldsymbol{v}_{Nike}, \boldsymbol{v}_{Male} \rangle$

Example is adopted from:
Juan et al. WWW 2017. Field-aware Factorization Machines in a Real-world Online Advertising System

Strong Generalization of FM

- FM has strong generalization in learning feature interactions, which is a key advantage brought by its interaction learning in latent space.

$+$	$-$	Publisher	Advertiser	s
950	50	ESPN	Nike	$\boldsymbol{v}_{\text{ESPN}} \cdot \boldsymbol{v}_{\text{Nike}} + \cdots$
2	0	ESPN	Gucci	$\boldsymbol{v}_{\text{ESPN}} \cdot \boldsymbol{v}_{\text{Gucci}} + \cdots$
0	0	Vogue	Nike	$\boldsymbol{v}_{\text{Vogue}} \cdot \boldsymbol{v}_{\text{Nike}} + \cdots$
950	50	Vogue	Gucci	$\boldsymbol{v}_{\text{Vogue}} \cdot \boldsymbol{v}_{\text{Gucci}} + \cdots$

  - $\boldsymbol{v}_{\text{Vogue}}$ is learned from 1000 data points
  - $\boldsymbol{v}_{\text{Nike}}$ is learned from 1000 data points
  - More accurate prediction than Poly2

Example is adopted from:
Juan et al. WWW 2017. Field-aware Factorization Machines in a Real-world Online Advertising System

Some Achievements by FMs

- After proposing FMs on 2010, Rendle used FM to win:
  - 1st award of ECML/PKDD 2009 Data Challenge on *personalized tag recommendation*
  - 1st award of KDD Cup 2010, Gr… Challenge on predicting student *performance on questions*

These DCs have a common property: most predictor variables are categorical and converted to one-hot sparse data.

  - 1st (online track) and 2nd (offline track) award of ECML/PKDD 2013 on *recommending given names*
  - 3rd award of KDD Cup 2012 Track 1 of *click-through rate prediction*

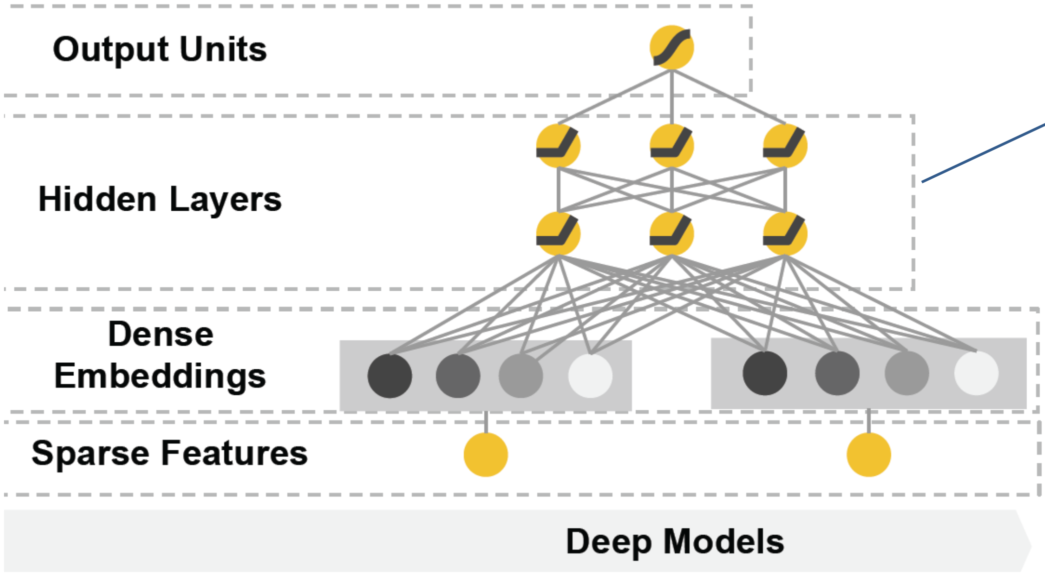
How about Deep Learning?

- The revolution brought by DL: CNNs for image data, and RNNs language data.
- What are DL solutions for such sparse data and how do they perform?

- In 2014, Field-aware FMs are proposed and win:
  - 1st award of 2014 Criteo *display ad CTR prediction*.
  - 1st award of 2015 Avazu *mobile ad CTR prediction*
  - 1st award of 2017 Outbrain *click prediction*.

Wide&Deep

- Proposed by Cheng et al. (Google) in RecSys 2016 for app recommendation:

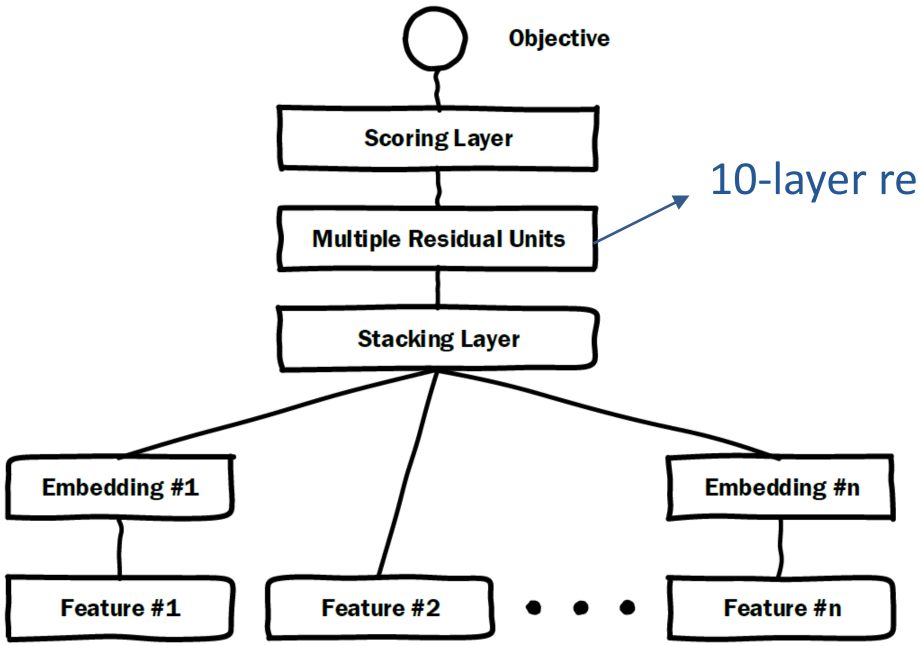


3-layer ReLU units:
1024 -> 512 -> 256

The deep part can learn high-order feature interactions in an **implicit** way.

DeepCross

- Proposed by Shan et al. (MSR) in KDD 2016 for sponsored search ranking.



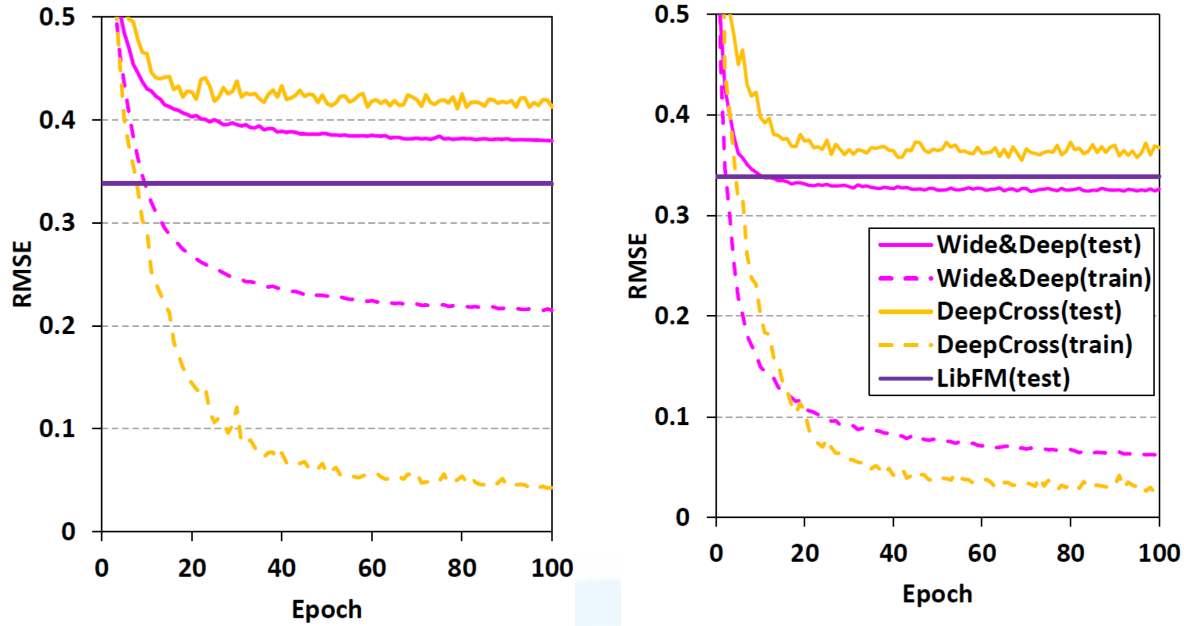
10-layer residual units

The deep part can learn high-order feature interactions in an **implicit** way.

Shan et al. KDD 2016. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features

How do Wide&Deep and DeepCross perform?

- Unfortunately, the original papers did not provide systematic evaluation on learning feature interactions.
- Contribution #1: We show that both state-of-the-art DL methods do not work well empirically for learning feature interactions.

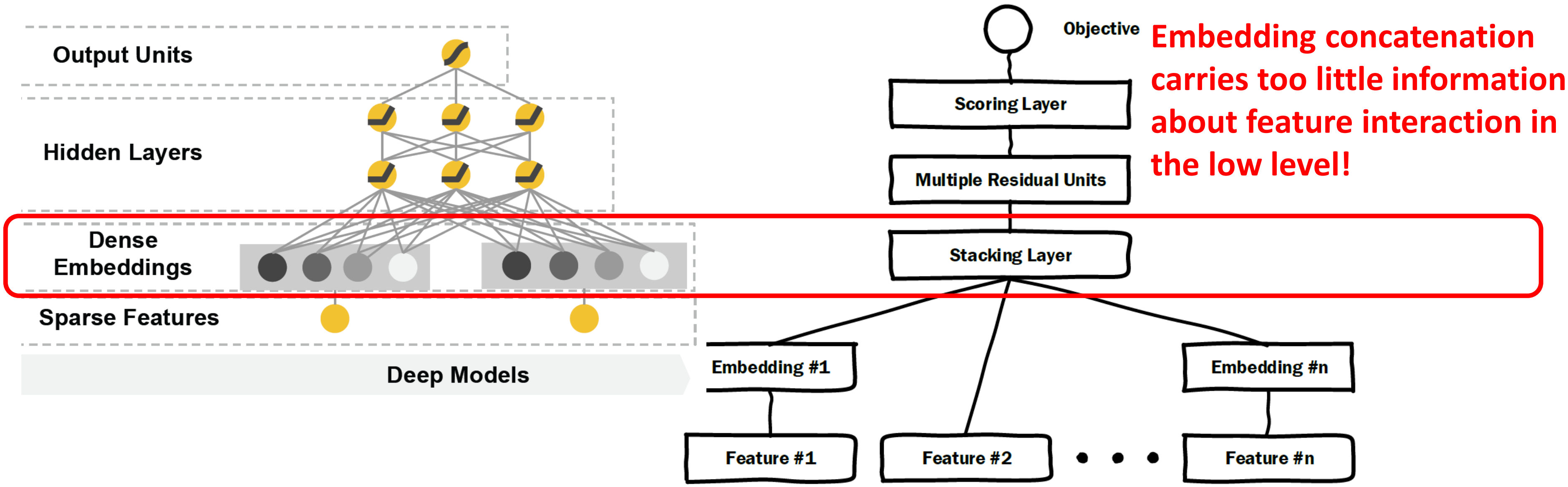


(a) Random initialization

(b) FM as pre-training

Limitation of Existing DL Methods

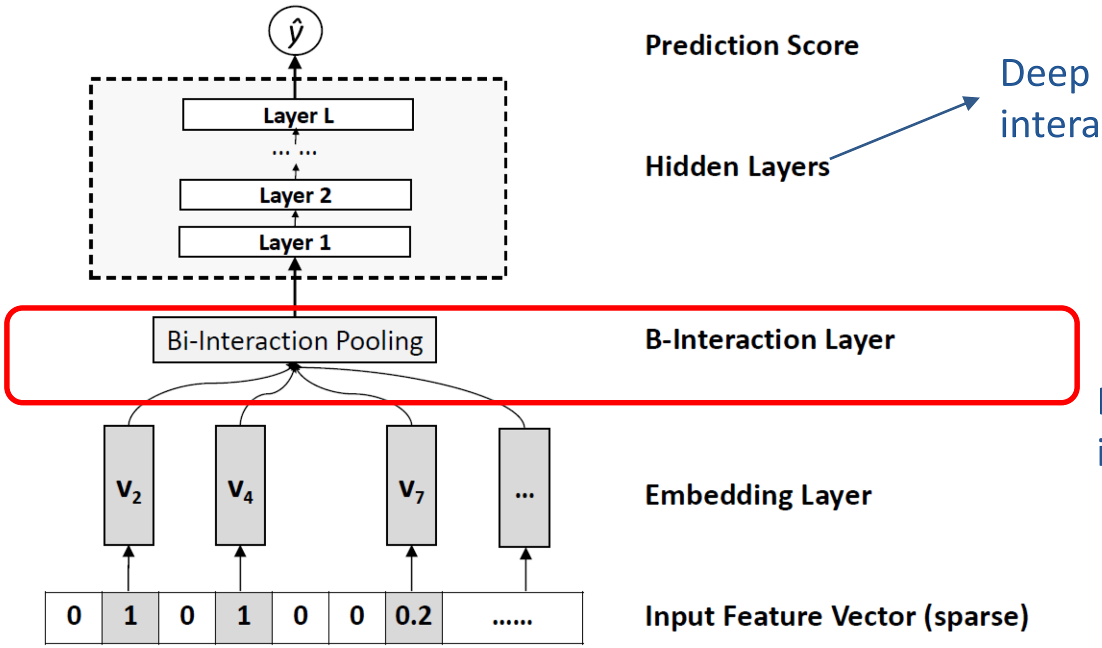
However, we find that both DL methods can hardly outperform the shallow FM.



The model has to fully rely on "deep layers" to learn meaningful feature interactions, which is difficult to achieve, especially when no guidance info is provided.

Neural Factorization Machines

- We propose a new operator – *Bilinear Interaction pooling* – to model the second-order feature interactions in the low level.



Deep layers learn **high-order** feature interactions only, being much easier to train.

$$f_{BI}(\mathcal{V}_x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i \mathbf{v}_i \odot x_j \mathbf{v}_j,$$

BI layer learns **second-order** feature interactions, e.g., *female* likes *pink*

Figure 2: Neural Factorization Machines model (the first-order linear regression part is not shown for clarity).

Appealing properties of Bi-Interaction Pooling

$$f_{BI}(\mathcal{V}_x) = \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_i\mathbf{v}_i \odot x_j\mathbf{v}_j,$$

1. It is a standard pooling operation that converts a set of vectors (of variable length) to a single vector (of fixed length).
2. It is more informative than mean/max pooling and concatenation, but has the same time complexity *O(kN$_x$)* :

$$f_{BI}(\mathcal{V}_x) = \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_i\mathbf{v}_i \odot x_j\mathbf{v}_j = \frac{1}{2}\left[(\sum_{i=1}^{n} x_i\mathbf{v}_i)^2 - \sum_{i=1}^{n}(x_i\mathbf{v}_i)^2\right]$$

3. It is differentiable and can support end-to-end training:

$$\frac{d f_{BI}(\mathcal{V}_x)}{d\mathbf{v}_i} = (\sum_{j=1}^{n} x_j\mathbf{v}_j)x_i - x_i^2\mathbf{v}_i = \sum_{j=1, j\neq i}^{n} x_i x_j\mathbf{v}_j.$$

FM as a Shallow Neural Network

- By introducing the Bi-Interaction pooling, we provide a novel neural network view for FM.

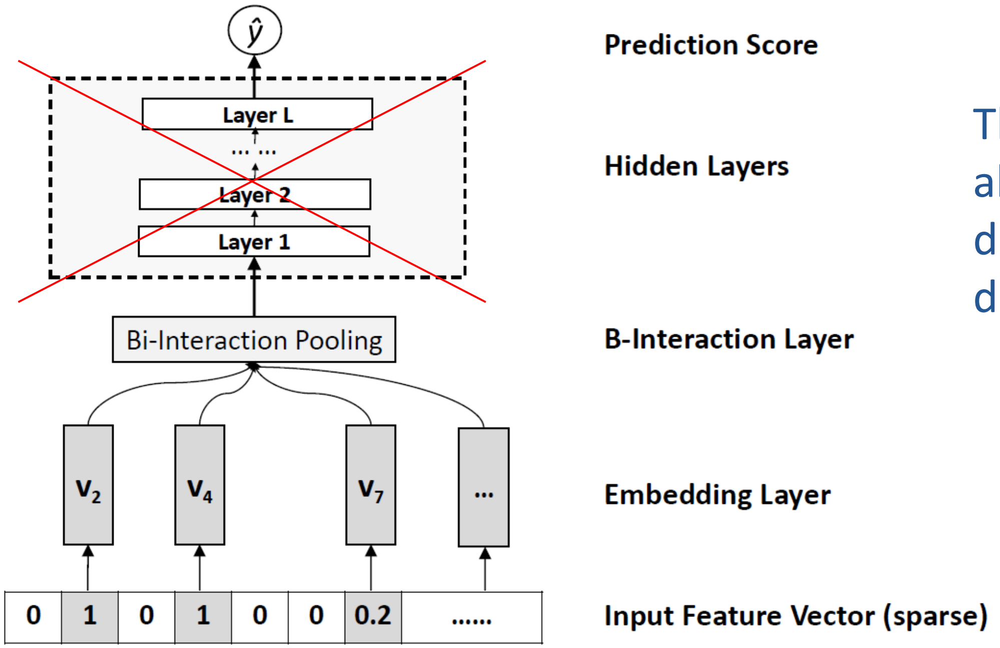


Figure 2: Neural Factorization Machines model (the first-order linear regression part is not shown for clarity).

This new view of FM is very instructive, allowing us to adopt techniques developed for DNN to improve FM, *e.g.* dropout, batch normalization etc.

Experiments

- Task #1: Context-aware App Usage Prediction
  - Frappe data: userID, appID, and 8 context variables (*sparsity: 99.81%*)
- Task #2: Personalized Tag Recommendation
  - MovieLens data: userID, movieID and tag *(sparsity: 99.99%)*

Table 1: Statistics of the evaluation datasets.

Dataset	Instance#	Feature#	User#	Item#
Frappe	288,609	5, 382	957	4,082
MovieLens	2,006,859	90, 445	17,045	23,743

- Randomly split: 70% (training), 20% (validation), 10% (testing)
- Evaluated prediction error by RMSE (lower score, better performance).

http://baltrunas.info/research-menu/frappe
http://grouplens.org/datasets/movielens/latest

Baselines

- 1. LibFM:
  - The official implementation of second-order FM
- 2. HOFM:
  - A 3rd party implementation of high-order FM.
  - We experimented with order size 3.
- 3. Wide&Deep:
  - Same architecture as the paper: 3 layer MLP: 1024->512->256
- 4. DeepCross:
  - Same structure as the paper: 10 layer (5 ResUnits): 512->512->256->128->64)
- Our Neural FM (NFM):
  - Only 1-layer MLP (same size as the embedding size) above Bi-Interaction

I. NFM is a new state-of-the-art

Table: Parameter # and testing RMSE at embedding size 128

	Frappe		MovieLens	
Method	**Param#**	**RMSE**	**Param#**	**RMSE**
Logistic Regression	5.38K	0.5835	0.09M	0.5991
FM	0.69M	0.3437	11.67M	0.4793
HOFM	1.38M	0.3405	23.24M	0.4752
Wide&Deep (3 layers)	2.66M	0.3621	12.72M	0.5323
Wide&Deep^{+} (3 layers)	2.66M	0.3311	12.72M	0.4595
DeepCross (10 layers)	4.47M	0.4025	12.71M	0.5885
DeepCross^{+} (10 layers)	4.47M	0.3388	12.71M	0.5084
NFM (1 layer)	**0.71M**	**0.3127**	**11.68M**	**0.4557**

$^{+}$ means using FM embeddings are pre-training.
K means *thousand*, M means *million*

1. Modelling feature interactions with embeddings is very useful.
2. Linear way of high-order modelling has minor benefits.
3. For end-to-end training, both DL methods underperform FM.
4. Pre-training is crucial for two DL methods: Wide&Deep slightly betters FM while DeepCross suffers from overfitting.
5. **NFM significantly betters FM by end-to-end training with fewest additional parameters.**

Lab for Media Search

II. Impact of Hidden Layers

- 1. One non-linear hidden layer improves FM by a large margin.

  => Non-linear function is useful to learn high-order interactions

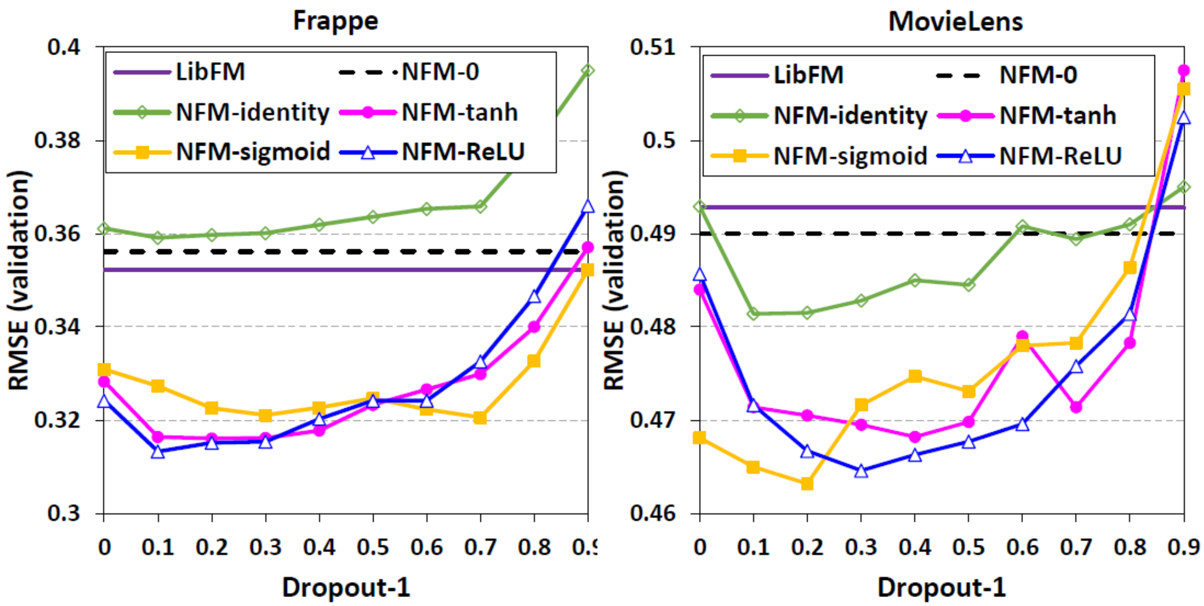


Figure 6: Validation error of LibFM, NFM-0 and NFM with different activation functions on the first hidden layer.

II. Impact of Hidden Layers

- 2. More layers do not further improve the performance.

  => The informative Bi-Interaction pooling layer in the low level eliminates the needs of deep models for learning high-order feature interactions.

Table 2: NFM *w.r.t.* different number of hidden layers.

Methods	Frappe	MovieLens
NFM-0	0.3562	0.4901
NFM-1	**0.3133**	**0.4646**
NFM-2	0.3193	0.4681
NFM-3	0.3219	0.4752
NFM-4	0.3202	0.4703

Lab for Media Search

III. Study of Bi-Interaction Pooling

- We explore how *dropout* and *batch norm* impact NFM-0 (i.e., our neural implementation of FM)
- 1. Dropout prevents overfitting and improves generalization:

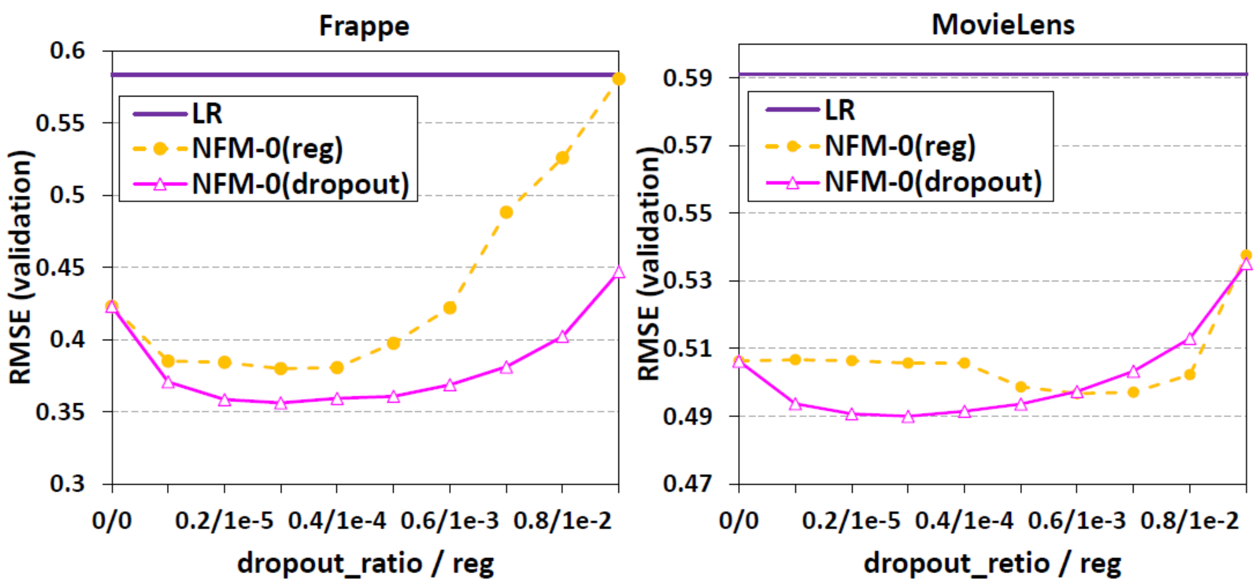


Figure 3: Validation error of NFM-0 *w.r.t.* dropout on the Bi-Interaction layer and L_2 regularization on embeddings.

III. Study of Bi-Interaction Pooling

- We explore how *dropout* and *batch norm* impact NFM-0 (i.e. our neural implementation of FM)
- 2. Batch norm speeds up training and leads to slightly better performance:

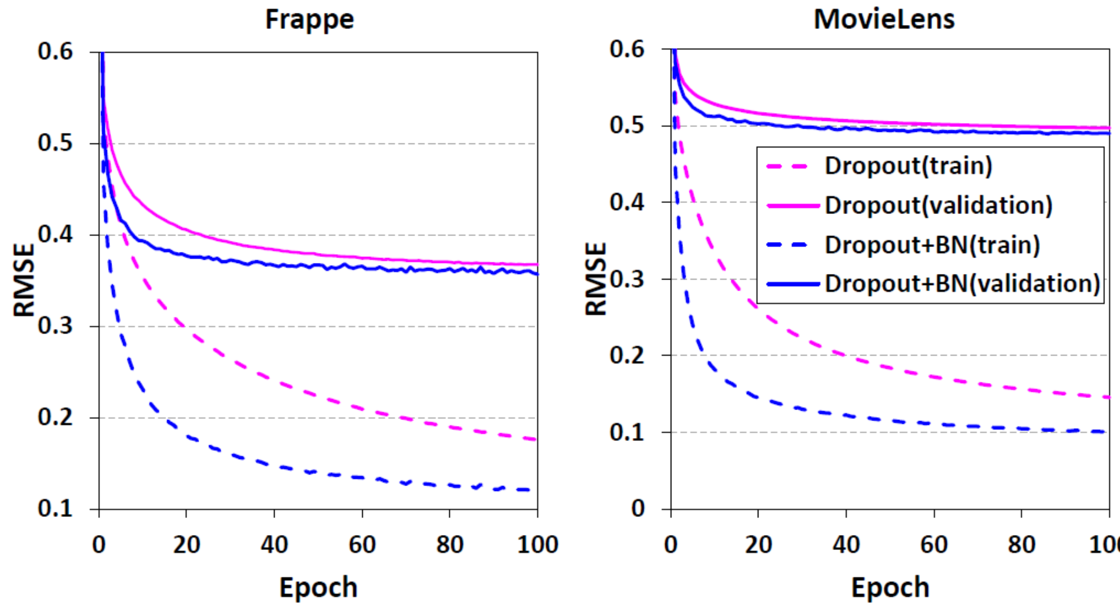


Figure 5: Training and validation error of each epoch of NFM-0 with and without BN on the Bi-Interaction layer.

Conclusion

- In sparse predictive tasks, existing DL methods can hardly outperform shallow FM:
  - Deep models are difficult to train and tune;
  - Low-level operation is not informative for capturing feature interactions

- We propose a novel *Neural FM* model.
  - Smartly connects FM and DNN with an informative Bi-Interaction pooling.
  - FM/DNN accounts for second-/high- order feature interactions, respectively.
  - Being easier to train and outperform existing DL solutions.

Personal Thoughts

- In many IR/DM tasks, shallow models are still dominant.
  - E.g. logistic regression, factorization, and tree-based models.
- Directly apply existing DL methods may not work.
  - Strong representation => Over-generalization (overfitting).
- Our key finding is that early crossing features is useful for DL.
  - Applicable to other tasks that need to account for feature interactions.
- Future research should focus on designing **better** and **explainable** neural components that can meet the specific properties of a task.
  - We can well explain second-order feature interactions by using **attention** on Bi-Interaction pooling [IJCAI 2017]
  - How to interpret high-order interactions learned by DL?

Lab for Media Search

Thanks!

Codes: https://github.com/hexiangnan/neural_factorization_machine