DLRM

DeepSync, South Korea

0. 기존의 personalization and recommendation

1. Recommendation system
   - content filtering (expert-user)
   - past user behavior
   - neighborhood method
   - latent factor method

2. Predictive analytics
   - statistical model (ex: linear regression, logistic regression)
   - deep networks (-> embedding)
   - latent factor method

1. DLRM (Deep Learning Recommendation Model)

앞에 소개한 방법 중 몇 개 + mlp + interacting dense feature

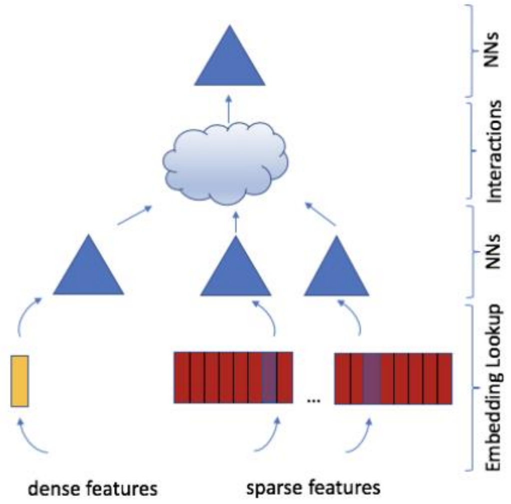


Figure 1: A deep learning recommendation model

Deep Sync

2. Model Architecture: embedding

	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8
man →	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
woman →	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
king →	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
queen →	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9

Word

Word embedding

Di

$m \times d$ 차원

만약 dog 차원을 알고 싶다면?

0 0 1 0 0 0 0 0

$1 \times m$ 차원

만약 dog, houses 차원을 알고 싶다면?

0 0 1 0 0 0 0 0

0 0 0 1 0 0 0 0

$2 \times m$ 차원

$$w_i^T = e_i^T W$$

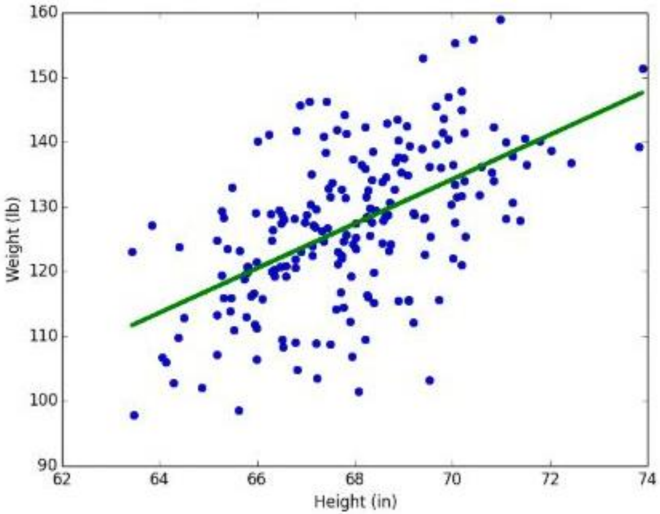
2. Model Architecture: latent factor method

Product vector: w ($\mathbb{R}^d$)
User vector: v ($\mathbb{R}^d$)

Dot product : $w_i^T \cdot v_j$ (i번째 상품과 j번째 사용자)
Ground truth: r_{ij}

Goal: $\min \sum_{(i,j)\in\mathcal{S}} r_{ij} - \boldsymbol{w}_i^T \boldsymbol{v}_j$

2. Model Architecture: Factorization Machine



Linear regression: $y = w_1 x_1 + w_0$

Multiple linear regression:

$$y = w_1 x_1 + w_2 x_2 + w_0$$

Multiple linear regression with interaction terms:

$$y = w_1 x_1 + w_2 x_2 + m_3 (x_1 \times x_2) + w_0$$
$$y = w_1 x_1 + w_2 x_2 + < v_1 \times v_2 > (x_1 \times x_2) + w_0$$

N term

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

$$\hat{y} = b + \boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{x}^T \mathtt{upper}(VV^T) \boldsymbol{x}$$

2. Model Architecture: Factorization Machine

Feature vector **x**

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

2. Model Architecture: Factorization Machine

Deep Sync

장점1 : linear complexity

$O(kn^2)$

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle x_i x_j$$

$$=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle x_i x_j - \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_i\rangle x_i x_i$$

$$=\frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i x_j - \sum_{i=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i x_i\right)$$

$$=\frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)\left(\sum_{j=1}^{n} v_{j,f}\, x_j\right) - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)$$

$$\boxed{=\frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)}$$

$O(kn)$

2. Model Architecture: Factorization Machine

Deep Sync

장점2: spare한 상황에서 interaction을 예상 가능

Task: 사람 Alice의 Star Trek의 rating을 예측 하고 싶음

- no interaction ($w_{A,ST} = 0$)

- Bob과 Charlie는 star wars를 봤고 비슷한 평을 남김
($< v_B, v_{SW} >$ $and < v_c, v_{SW} >$ are similar)

- Alice는 Charlie와 different factor vector를 보임
($< v_A, v_{SW} >$ $and < v_c, v_{SW} >$ are not similar)

- Bob은 star wars와 star trek의 평을 비슷하게 함
($< v_B, v_{SW} >$ and $< v_B, v_{ST} > are\ similar)$

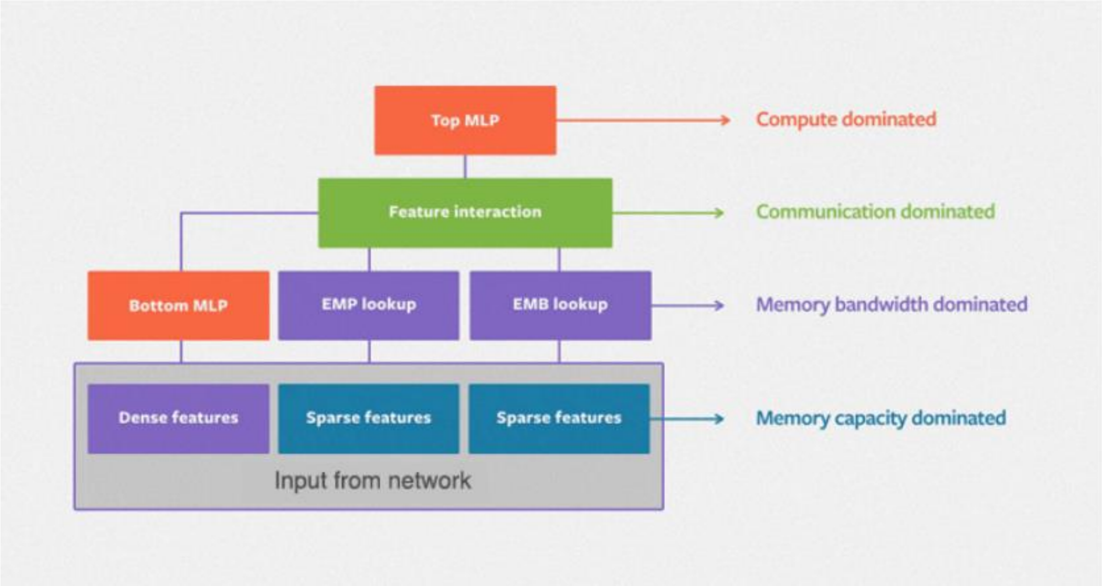
=> ($< v_A, v_{ST} >$ and $< v_A, v_{SW} > are\ similar)$

2. Model Architecture: MLP

Deep Sync

$$\hat{y} = W_k\sigma(W_{k-1}\sigma(...\sigma(W_1\boldsymbol{x} + \boldsymbol{b}_1)...) + \boldsymbol{b}_{k-1}) + \boldsymbol{b}_k$$

2. Model Architecture and Parallelism



1. Categorical features -> emb
2. Continuous features -> MLP
3. Second-order interaction
4. 3의 결과 ->Top MLP, Dense features로
5. Sigmoid function -> T= {+1,-1}

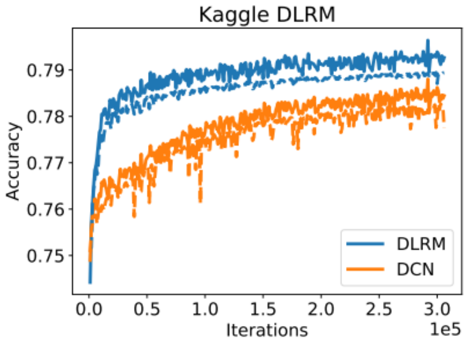
1. Model parallel (size of emb-> large parameter)
2. Data parallel (smaller parameter, large computation)
3.
4. Data parallel

3. experiments

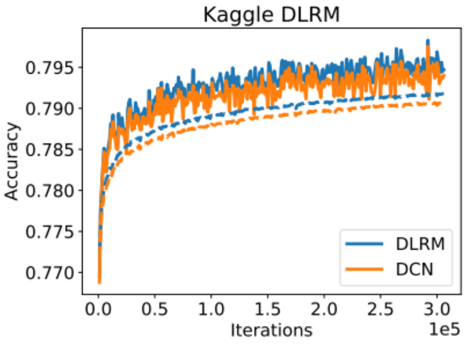
- Kaggle data set
- DCN과 비교

- DLRM
  1. bottom MLP: 512, 256 ,64
  2. top MLP: 512, 256
  3. emb dimension: 16

- DCN
  1. 6 cross layer
  2. deep network: 512, 256
  3. emb dimension: 16

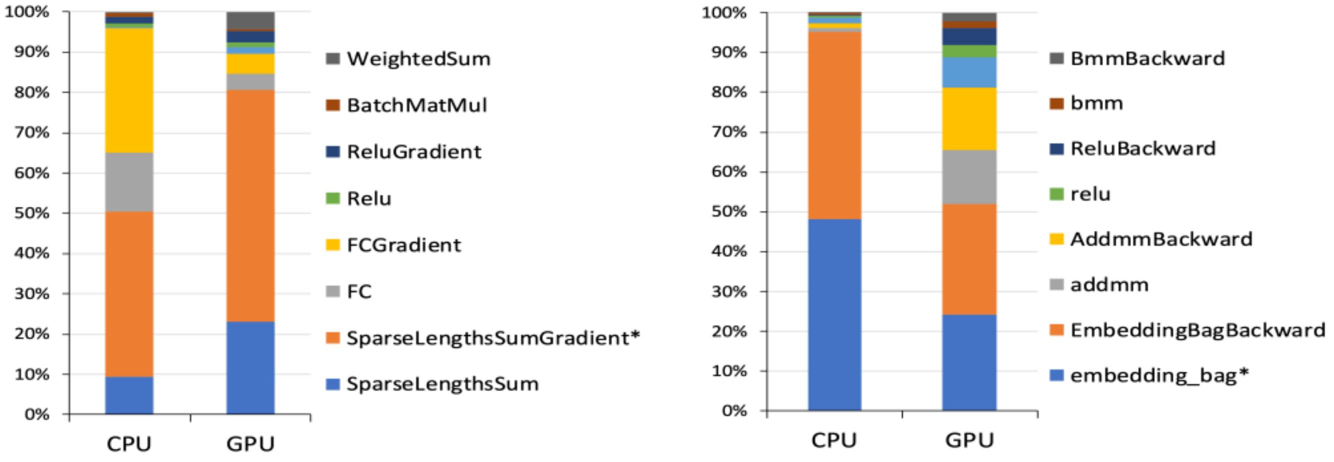


(a) SGD



(b) Adagrad

3. experiments



(a) Caffe2

(b) PyTorch

Figure 6: Profiling of a sample DLRM on a single socket/device

reference

[추천시스템][paper review][구현] Factorization Machines (tistory.com)

DLRM: An advanced, open source deep learning recommendation model (meta.com)

paper.dvi (ntu.edu.tw)

Thank you for your time