추 천 세 미 나

ToBig's 12기 이세윤

Deep Learning Recommendaion Model for Personalization and Recommendation Systems

Contents

범주형 변수를 다루기 위한 딥러닝 네트워크!

신경망 기반의 추천 모델을 이용해 개인화 추천 태스크를 해결해보자.

하지만 범주형 변수를 다루기 때문에 다른 딥러닝 네트워크와는 다르다.

-> sparse한 categorical data로 신경망이 효율적으로 작동하는 것이 어렵기 때문

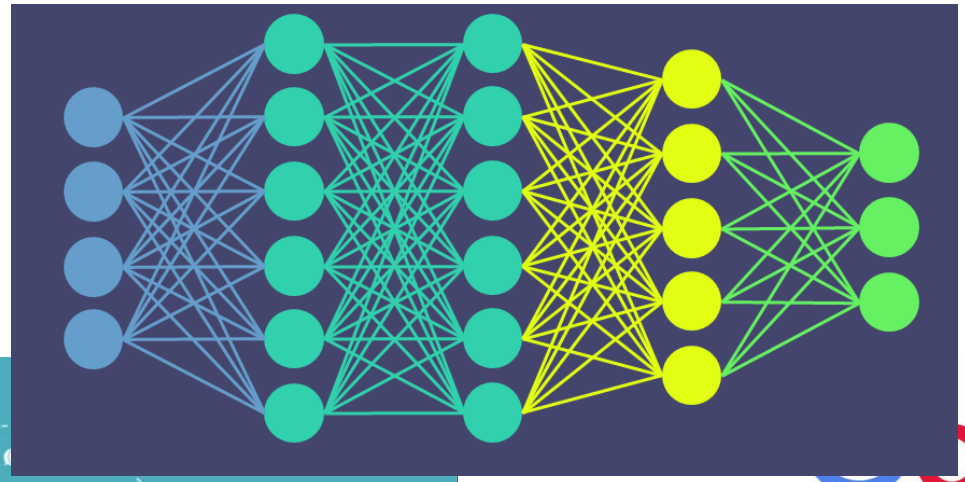
논문에서는 SOTA 딥러닝 추천 모델인 **DLRM**을 소개한다.

(Deep Learning Recommendation Model)

Pytorch 및 Caffe2 프레임워크로 구현 또한 제공한다.

Unit 01 | Introduction

DL



CTR

Ranking

딥러닝 모델 구조 설계의 두 가지 주요 관점

1. 추천 시스템의 관점

Content Filtering -> Collaborative Filtering

Neighborhood Method

Latent Factor Method

딥러닝 모델 구조 설계의 두 가지 주요 관점

2. 주어진 데이터에 기초하여 사건의 확률을 분류하거나 예측하기 위해 통계적 모델에 의존하는 예측 분석 관점

선형 및 로지스틱 모델

Embedding

깊은 네트워크 모델

one- / mutli-hot vector -> dense representation

추상적 공간 (잠재 요인 공간) -> 밀집된 표현

두 가지 관점의 결합에 의해 설계된 DLRM

- 범주형 데이터를 나타내는 sparse feature를 처리하기 위해 Embedding 사용
- dense feature를 처리하기 위해 MLP를 사용한 후 통계적 기법을 사용해 feature들을 명시적으로 상호작용
- 다른 MLP와의 교호작용을 post-processing 하여 사건 확률 찾음

왜 범주형과 연속형 변수 처리가 중요할까?

1. Embedding은 one-hot vector보다 훨씬 더 풍부한 데이터 표현으로, 훨씬 더 좋은 결과를 가져온다.

2. 연속형 및 범주형 feature를 단순히 많이 사용해서 fitting 시키는 것은 정확도를 높이지 못한다.
- feature 간 교호작용이 누락되었기 때문
ex) 코미디와 공포 영화를 좋아하는 사람이 공포 코미디 영화를 얼마나 좋아할지

따라서 단순 선형 회귀나 로지스틱 회귀 분석을 추천시스템에 사용할 수 없다.
(다항 회귀 분석에는 이 교호작용 성분이 있지만 보이지 않는 교호작용을 예측하지 못한다.)

DLRM Architecture의 High-level 구성 요소

1. Embeddings

- 범주형 변수를 연속형 벡터에 매핑하는 것
- 이산형 변수를 저차원의 연속된 벡터 표현으로 만드는 것
- 범주형 변수의 차원을 줄일 수 있고 변환된 공간 내에서 범주를 의미있게 나타낼 수 있어 유용

2. Matrix Factorization

- 추천 시스템에서 사용되는 협업 필터링 알고리즘의 한 종류
- user-item interaction 행렬을 두 개의 낮은 차원의 직사각 행렬을 곱으로 분해하여 작동

DLRM Architecture의 High-level 구성 요소

3. Factorization Machine

- 분류와 회귀 모두에 사용할 수 있는 범용 지도학습 알고리즘
- 고차원의 sparse 데이터 내에서 feature 간 상호작용을 포착하도록 설계된 선형 모델의 확장

ex) 클릭 예측 시스템에서 FM 모델은 특정 범주의 광고를 특정 범주의 페이지에 배치할 때 관찰되는 클릭 비율 패턴을 알아낼 수 있다.

- 고차원 sparse 데이터셋 처리에 적합

4. Multilayer Perceptron (MLP)

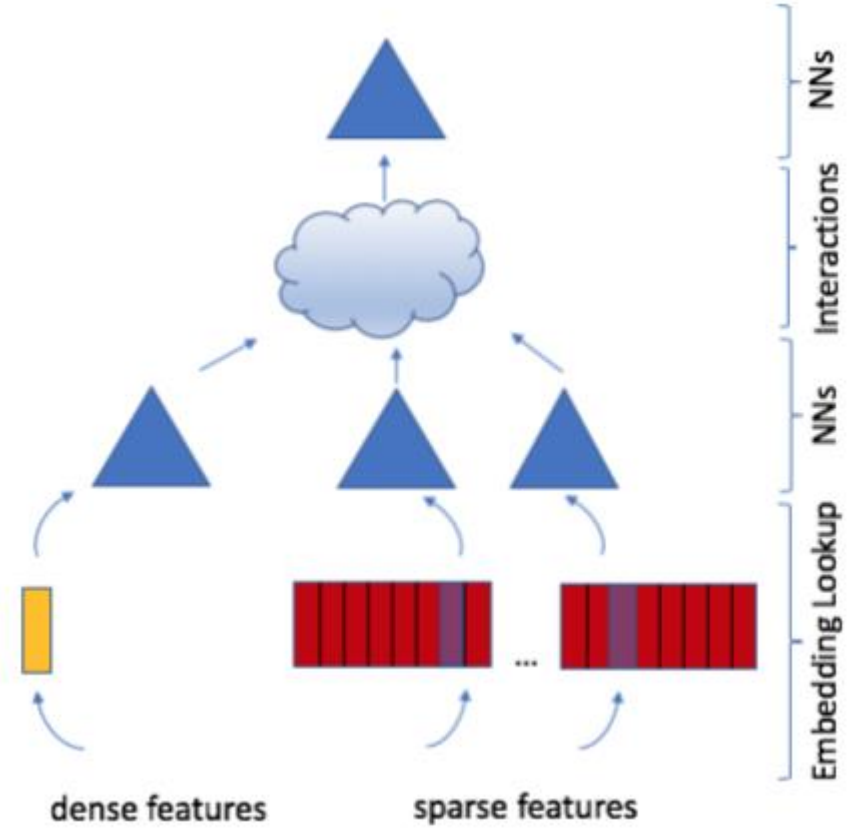
- feed forward 인공신경망
- backpropagation 지도학습 사용

DLRM Architecture

- user와 item은 많은 연속형 변수, 범주형 변수로 설명

- 범주형 변수를 처리하기 위해 MF에 사용되는 잠재적 요인의 개념을 일반화하면서 동일한 차원의 임베딩 벡터로 표현

- 연속형 변수를 다루기 위해 임베딩 벡터와 같은 길이의 밀도를 산출하는 MLP에 의해 변형



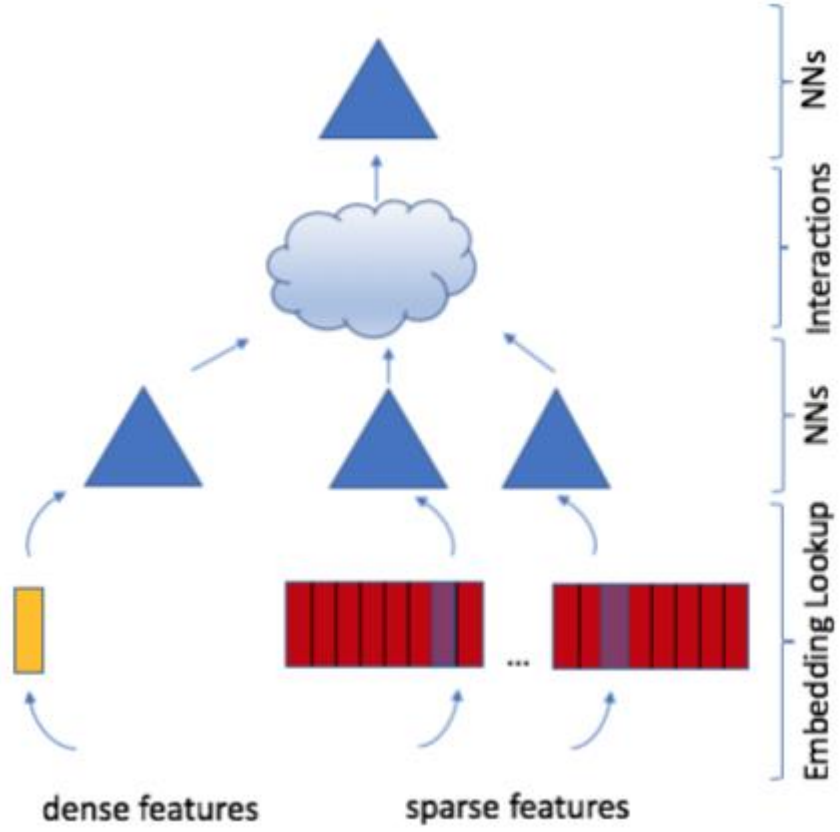
DLRM Architecture

- FM에 제공된 sparse 데이터를 처리하기 위해 직관에 따라 다양한 feature의 2차 상호작용 계산이 명시적으로 수행됨

- 선택적으로 2차 상호작용을 MLP를 통해 전달

(임베딩 벡터와 dense feature의 모든 쌍들 사이에서 dot product를 취함으로써 이루어짐)

- dot product는 원래 처리된 dense feature와 결합되어 다른 MLP와 후처리되며 확률을 주기 위해 sigmoid 함수에 공급



DLRM Architecture

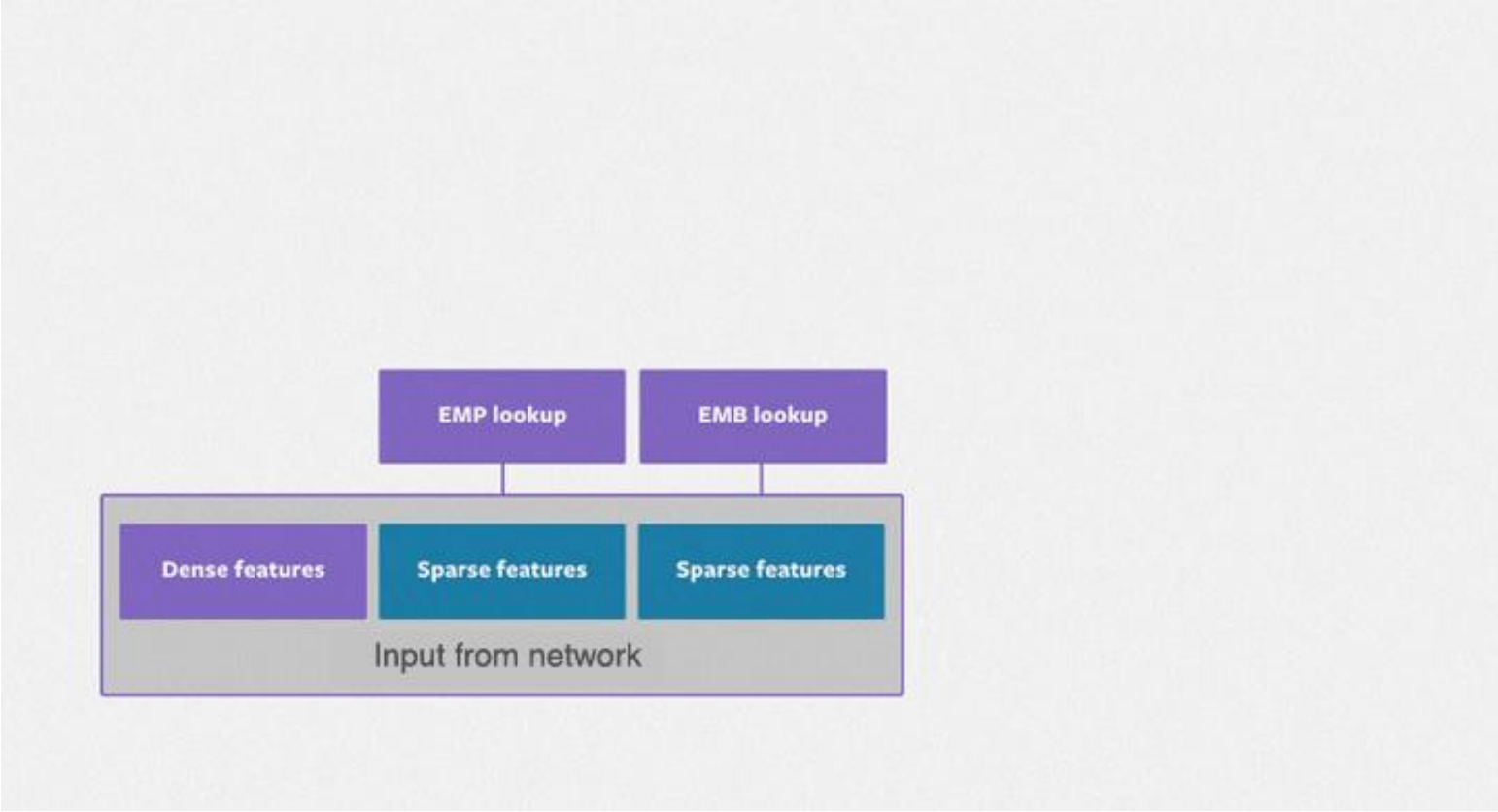
```
DLRM_Net(
  (emb_l): ModuleList(
    (0): EmbeddingBag(4, 16, mode=sum)
    (1): EmbeddingBag(3, 16, mode=sum)
    (2): EmbeddingBag(2, 16, mode=sum)
  )
  (bot_l): Sequential(
    (0): Linear(in_features=13, out_features=512, bias=True)
    (1): ReLU()
    (2): Linear(in_features=512, out_features=256, bias=True)
    (3): ReLU()
    (4): Linear(in_features=256, out_features=64, bias=True)
    (5): ReLU()
    (6): Linear(in_features=64, out_features=16, bias=True)
    (7): ReLU()
  )
  (top_l): Sequential(
    (0): Linear(in_features=22, out_features=512, bias=True)
    (1): ReLU()
    (2): Linear(in_features=512, out_features=256, bias=True)
    (3): ReLU()
    (4): Linear(in_features=256, out_features=1, bias=True)
    (5): Sigmoid()
  )
)
```

DLRM Summary

	Embedding	MLP	Interactions	Loss
PyTorch	nn.EmbeddingBag	nn.Linear/addmm	matmul/bmm	nn.CrossEntropyLoss
Caffe2	SparseLengthSum	FC	BatchMatMul	CrossEntropy

Table 1: DLRM operators by framework

DLRM Architecture



다른 네트워크와의 차이

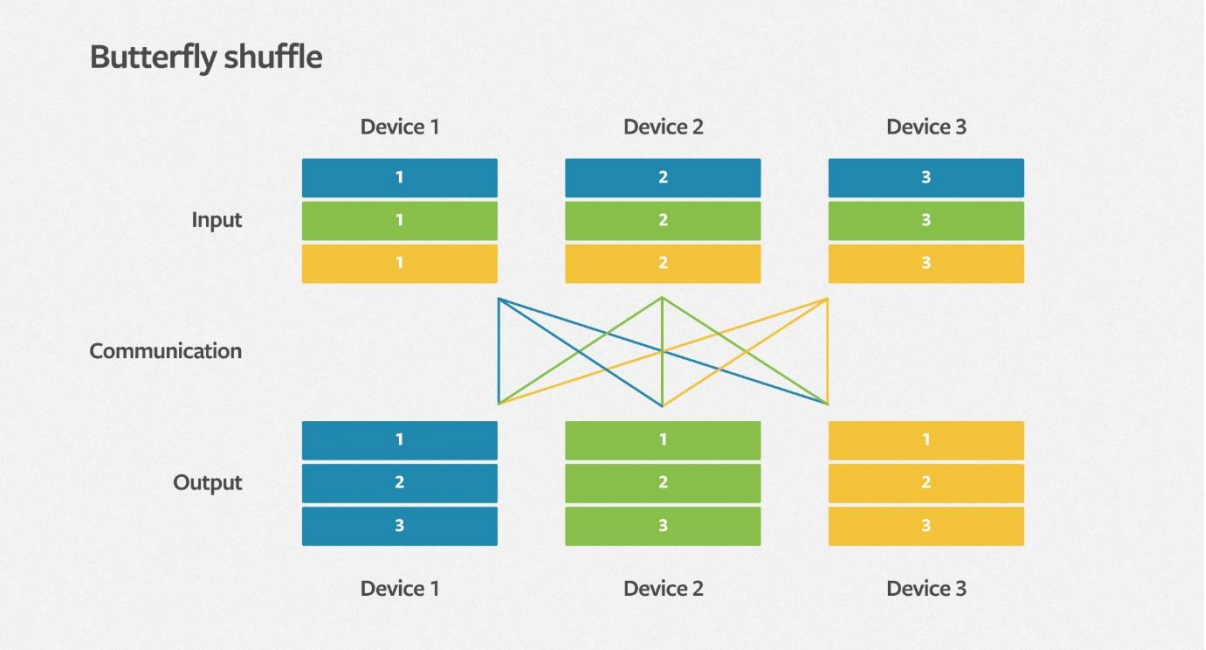
- Wide and Deep, Deep and Cross, DeepFM, xDeepFM과 차이는
임베딩 feature vector와 그 교차항을 처리하는 방법에 있음

- 논문에서는 위의 네트워크에서 발생하는 2차 이상의 고차 상호작용까지 필요하지 않다고 주장

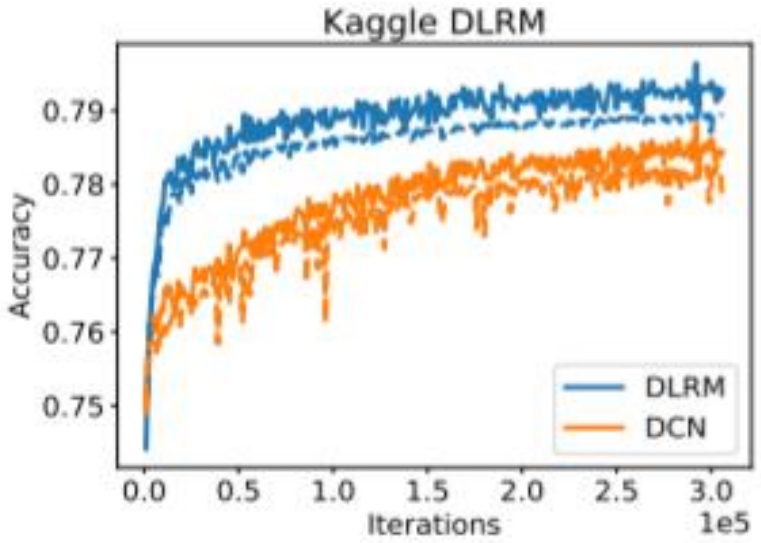
- DLRM에서는 dot product에 의해 생산된 교차항만을 고려하여 차원을 현저히 감소시키는 FM을 모방한 구조로 임베딩을 상호작용함

Parallelism

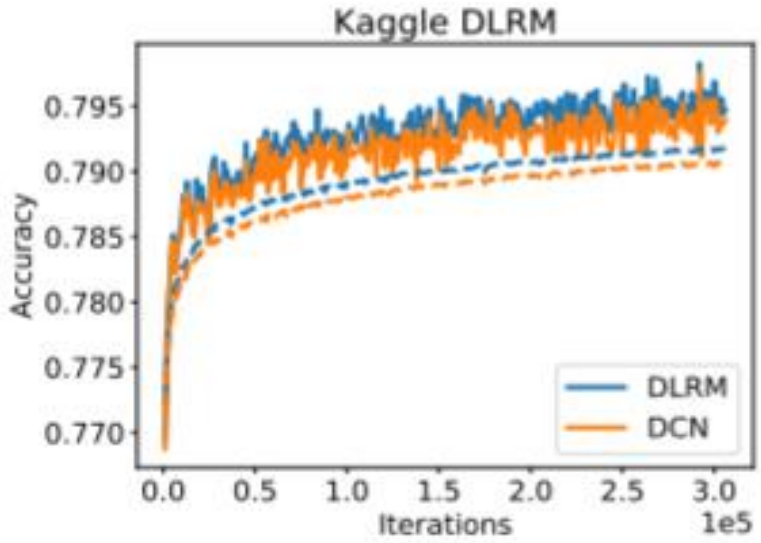
- DLRM에는 매우 많은 파라미터 존재
- 따라서 모델의 효율적인 병렬화 필요
- 앞서 말했듯이 DLRM은 범주형 feature(임베딩)와 연속형 feature(MLP)를 결합하여 처리함.
- 임베딩은 파라미터의 대부분을 차지하지만 임베딩의 경우 모든 기기에 임베딩의 복제가 필요하므로 데이터 병렬화 안하며 모델 병렬화만 진행
- MLP의 매개변수는 적지만 상당한 양의 컴퓨팅으로 변환되며 업데이트를 할때에만 통신을 필요로 하므로 데이터 병렬화 가능
- DLRM에서 병렬화는 임베딩에 대한 모델 병렬화와 MLP에 대한 데이터 병렬화를 결합



Experiments



(a) SGD



(b) Adagrad

- Criteo Ad Kaggle data set 사용하여 DLRM과 DCN(Deep and Cross Network)의 성능 비교 (extensive한 하이퍼파라미터 튜닝 없이)

Unit 05 | User-Centered Evaluation

Reference

- https://arxiv.org/pdf/1906.00091v1.pdf
- https://ai.facebook.com/blog/dlrm-an-advanced-open-source-deep-learning-recommendation-model/
- https://medium.com/@rehan_ahmad/deep-learning-recommendation-machines-dlrm-92e37733d07f

Code

- https://github.com/facebookresearch/dlrm

들어주셔서 감사합니다.