

# wide & deep

Wide & deep은 2016년 구글에서 발표한 논문으로, **Memorization**을 담당하는 **wide** set of cross-product feature transformations와 **Generalization**을 담당하는 **Deep** neural network를 결합한 모델을 제안하여 Google Play의 추천시스템을 개선하였다고 말한다.

## Memorization & Generalization

### Memorization

논문에 따르면 Memorization는 learning the frequent co-occurrence of items features and exploiting the correlation available in the historical data라고 한다.

간단하게 말하자면, 말 그대로 학습 데이터를 얼마나 잘 “기억하는가”를 의미한다.

Cross-product transformation이라는 방법을 이용하는데, 이는 One-hot vector 사이의 AND 연산을 통해 쉽게 표현이 될 수 있다. Decision Tree의 개념을 생각하면 이해가 쉬울 것이다.

이러한 Memorization은 학습에 존재하지 않았던 데이터가 주어지면 이렇게 query-item feature pair를 만들 수 없기 때문에 generalization ability가 떨어진다.

### Generalization

Embedding-based models (FM & DNN) 은 학습에 존재하지 않았던 데이터를 low-dimensional embedding vector를 통해 잘 표현할 수 있지만 Sparse and high-rank한 특성을 가진 데이터에서는 over-generalize하거나 less relevant recommendation을 생성할 수 있다는 문제점이 존재한다.

### Novelty

이러한 Memorization과 Generalization의 문제점을 해결하기 위해 Memorization에 해당하는 Wide 부분과 Generalization에 해당하는 deep 부분을 결합하여 jointly training 하는 방식을 제안하였다.

## Wide & Deep Learning

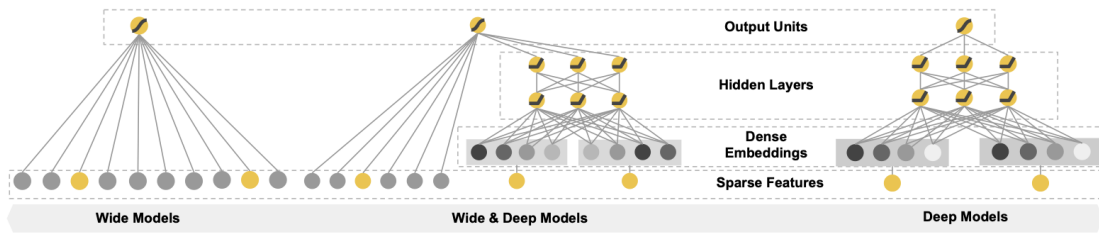


Figure 1: The spectrum of Wide & Deep models.

## Wide Component

### Cross-product Transformation

$$\phi_k(\mathbf{x}) = \prod_{i=1}^d x_i^{c_{ki}} \quad c_{ki} \in \{0, 1\}$$

$c_{ki}$  : a boolean variable that is 1 if the  $i$ -th feature is part of the  $k$ -th transformation  $\phi_k$ , and 0 otherwise.

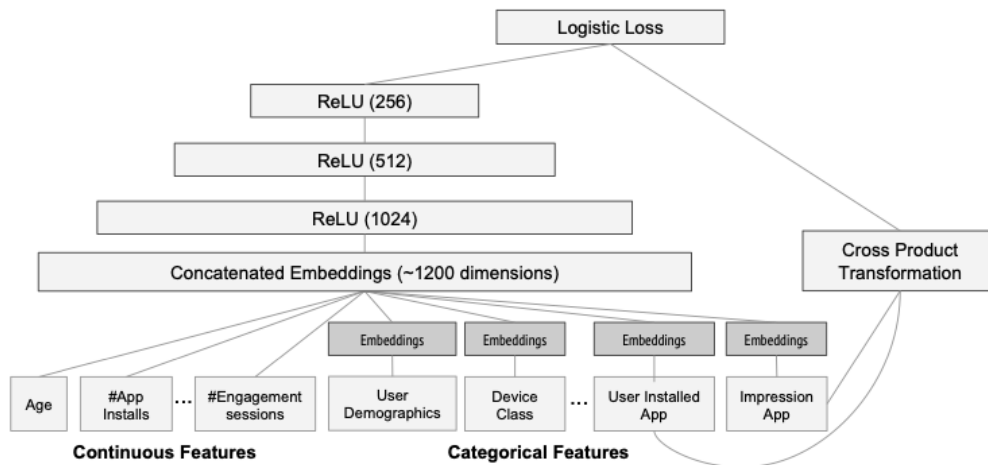
- 쉽게 말해서 AND 연산을 의미.
- Decision Tree에서 여러 조건 Node들을 지나야 최종 Class Prediction 이 1이 될 수 있는 방식과 유사

## Deep Component

$$a^{(l+1)} = f(W^{(l)}a^{(l)} + b^{(l)})$$

- 일반적인 Multi-layer Perceptron Architecture

## Joint Training of Wide & Deep Model



**Figure 4: Wide & Deep model structure for apps recommendation.**

## Ensemble vs Joint Training

- 과정 설명에 앞서 저자는 Ensemble과 Joint Training이 다른 방식이라고 명시하여 언급하고 있음
  - Ensemble은 독립적인 모델의 Output을 조합하여 Inference하는 것
  - 반면 Joint Training은 하나의 loss를 통해 각 모델에 서로가 영향을 끼치면서 학습됨

## Joint Training

- Wide component와 deep component는 log odds를 weighted sum방식을 이용해서 결합됨
- 이후 하나의 logistic loss로 표현되어 전체 학습에 영향을 끼침

$$P(Y = 1|\mathbf{x}) = \sigma(\mathbf{w}_{wide}^T[\mathbf{x}, \phi(\mathbf{x})] + \mathbf{w}_{deep}^T a^{(l_f)} + b)$$

- Wide part: FTRL (Follow-the-regularized-leader) algorithm
- deep part: mini-batch gradient descent

## Experiment Results

**Table 1: Offline & online metrics of different models.  
Online Acquisition Gain is relative to the control.**

Model	Offline AUC	Online Acquisition Gain
Wide (control)	0.726	0%
Deep	0.722	+2.9%
Wide & Deep	0.728	+3.9%

- A/B 테스트 진행
- 실험군과 대조군을 전체 중 1%를 Random Sampling을 통해 도출
- 대조군은 wide-only logistic regression model을 이용
- 결과를 살펴봤을 때, offline AUC에 대해서도 slightly하게 높지만 Online traffic을 감당하는 능력이 더 커졌음

## Conclusion

- 이렇게 Memorization과 Generalization을 결합한 wide & deep model을 살펴봤다.
- wide component는 training-set으로부터의 명백한 feature들을 표현하고, deep component는 unseen data에 대한 representation 능력을 키우면서 기존 모델을 향상시킬 수 있었다.