https://doi.org/10.5392/JKCA.2020.20.03.267

클릭률 예측 성능 향상을 위한 다중 배열 CNN 모형 설계

Design of a Multi-array CNN Model for Improving CTR Prediction

김태석
배재대학교 경영학과

Tae-Suk Kim(itmkim@pcu.ac.kr)

요약

클릭률(CTR) 예측은 사용자가 주어진 항목을 클릭할 확률을 추정하는 것으로 온라인 광고 수익 극대화를 위한 전략 결정에 중요한 역할을 한다. 최근 CTR 예측을 위해 CNN을 활용하는 시도가 이루어지고 있다. CTR 데이터는 특징 정보가 연관성 측면에서 의미 있는 순서를 갖지 않기 때문에, 임의의 순서로 배열될 수 있다. 하지만 CNN은 필터 사이즈에 의해 제한된 로컬 정보만을 학습하기 때문에 데이터 배열이 성능에 큰 영향을 줄 수 있다. 이 논문에서는 CNN이 수집할 수 있는 모든 로컬 특징 정보를 추출할 수 있는 데이터 배열 집합을 생성하고 생성된 배열들에 대하여 개별 CNN 모듈들이 특징들을 학습할 수 있는 다중 배열 CNN 모델을 제안한다. 대규모 데이터 세트에 대한 실험 결과에 따르면 제안된 모델은 기존 CNN 대비 AUC의 RI에서 22.6% 상승 효과를, 제안된 배열 생성 방법은 임의 생성 방법보다 3.87% 성능 향상을 달성하였다.

■ 중심어 : | 클릭률 | CNN | 특징 생성 | 딥러닝 |

Abstract

Click-through rate (CTR) prediction is an estimate of the probability that a user will click on a given item and plays an important role in determining strategies for maximizing online ad revenue. Recently, research has been performed to utilize CNN for CTR prediction. Since the CTR data does not have a meaningful order in terms of correlation, the CTR data may be arranged in any order. However, because CNN only learns local information limited by filter size, data arrays can have a significant impact on performance. In this paper, we propose a multi-array CNN model that generates a data array set that can extract all local feature information that CNN can collect, and learns features through individual CNN modules. Experimental results for large data sets show that the proposed model achieves a 22.6% synergy with RI in AUC compared to the existing CNN, and the proposed array generation method achieves 3.87% performance improvement over the random generation method.

■ keyword : | CTR | CNN | Feature Generation | Deep Learning |

I. 서 론

추천 광고를 클릭한 사용자와 광고가 포함된 페이지를 보는 총 사용자 수의 비율을 측정하는 click-through rate (CTR)은 온라인 광고를 평가하는 데 널리 사용되는 지표 중 하나이다[1]. 온라인 광고 시스템에서 광고에 대한 수익은 해당 광고의 CTR과 광고 입찰 가격의 곱으로 결정되기 때문에 정확한 CTR 예측에 기반한 효

* 본 논문은 2019학년도 배재대학교 교내학술연구비 지원에 의하여 수행된 것임
접수일자 : 2020년 02월 10일
수정일자 : 2020년 02월 26일
심사완료일 : 2020년 02월 26일
교신저자 : 김태석, e-mail : itmkim@pcu.ac.kr

과적인 광고 노출 전략은 광고 수익의 극대화에 매우 중요한 역할을 한다.

CTR 예측의 핵심 과제는 데이터에 내재하는 특징(feature) 간 상호 작용을 효과적으로 모델링하는 것이다. 특징 상호 작용의 모델링을 위한 전통적인 접근 방식은 Factorization machine(FM)을 기반으로 한 선형 모델링이다. FM은 원래 협업 추천(collaborative recommendation)을 위해 제안되었다[2][3]. 실수값을 가지는 특징 벡터가 주어지면 FM은 인수분해된(factorized) 상호 작용 매개 변수를 통해 각 특징 쌍(pairwise) 사이의 모든 상호 작용을 모델링하여 목표를 추정한다. FM은 감독 학습(supervised learning)에서 실수값을 가지는 특징 벡터의 어떠한 조합에도 적용이 가능하고 선형/로지스틱 회귀 대비 성능 향상이 있음이 밝혀졌다[4]. 하지만, FM은 특징들의 2차(second order) 상호 작용만 모델링함에 따라 실제 데이터의 모델링에 있어 성능상의 열화가 내재하는 한계를 지닌다.

최근 영상, 이미지 인식 분야에서 성공을 거두고 있는 심층 신경망(deep neural network: DNN)의 강력한 패턴 학습 능력을 활용하여 FM이 반영할 수 없는 고차 특징 상호 작용을 모델링하기 위한 딥러닝 모델이 일부 제안되었다[5][6]. 이러한 모델은 원시(raw) 데이터를 심층 신경망에 공급하여 특징 간 상호 작용을 명시적 또는 암시적으로 학습한다. 하지만, CTR 데이터는 특징 간 전체 조합에 비하여 성능에 유의미한 특징 조합은 일반적으로 드물기 때문에 신경망과 같이 학습에 많은 매개 변수를 필요로 하는 모델은 매우 비효율적이다[5]. 아울러, 심층 신경망 모델들은 여전히 전문가에 의한 feature engineering에 의존한다. 쉽게 이해할 수 있는 일부 특징 상호 작용은 전문가가 설계할 수 있겠지만 대부분의 특징 상호 작용은 데이터 안에 숨겨져 있어 선험적으로 식별하기가 어렵다(예: 클래식한 연관 규칙 사례인 '기저귀와 맥주' 규칙은 전문가가 아닌 데이터로부터 얻어졌음). 이해하기 쉬운 상호 작용에 대해서도 특징들의 수가 많을 때는 전문가가 모두 모델링하는 것은 불가능하다.

본 논문에서는 CNN(Convolutional Neural Network)의 CTR 데이터 로컬 정보 획득의 한계를 극복하여 예측 성능을 향상시킬 수 있는 CNN 설계 방안에 대한 연구를 수행한다. 구체적으로, 2장에서는 CNN에 대한 소개와 CTR 예측을 위해 적용되고 있는 CNN 구조에 대해 알아본다. 3장에서는 특징 정보 수집 구간이 주어졌을 때 특징들의 모든 조합 정보를 획득할 수 있는 배열 생성 방안과 이를 활용한 새로운 CNN 구조를 제안한다. 4장에서는 제안된 방법론의 성능을 실제 온라인 거래 데이터를 사용하여 비교 모델과의 성능 분석을 수행한다. 마지막으로, 시사점 및 연구의 한계점과 추후 연구 방향을 5장에서 논의한다.

II. 관련 연구

심층 신경망을 활용한 CTR 예측의 한계를 극복하고자 최근 CNN을 활용한 연구가 시도되고 있다[7-9]. CNN은 고급 신경망 구조로서 컴퓨터 비전 및 자연어 처리 영역에서 큰 성공을 거두었다[10][11]. CNN은 컨볼루션과 풀링 레이어를 반복적으로 적층하여 중요한 로컬 패턴을 찾는 방식으로 학습에 사용되는 매개 변수의 가중치를 공유하여 필요한 매개 변수 수를 크게 줄일 수 있고 특징에 대한 학습이 자동적으로 이루어져 후반부의 심층 신경망 구조의 최적화 어려움을 완화시킬 수 있는 장점이 있다[그림 1].

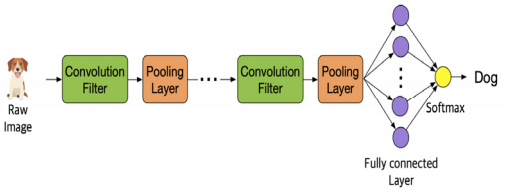

그림 1. 이미지 처리를 위한 CNN 아키텍쳐

CTR 예측을 위해 적용된 CNN의 일반적인 구조는 [그림 2]와 같다. 입력으로 들어가는 원시 데이터에는 사용자 정보(성별, 직업 등), 컨텍스트 정보(상품 정보, 구매 이력 등) 등의 필드(혹은 특징)을 포함하는데 대부분 이산적(discrete)이며 범주형(categorical)이다. 특징은 필드를 포함하는 개념으로 필드 간의 관계를 통해 잠재된 특징이 생성될 수 있는데, 본 논문에서는 설명

의 간략화를 위해 두 개념을 혼용하여 사용한다.

범주적 변수를 사용하여 예측 모델을 구축하기 위해 일차적으로 one-hot 인코딩을 통해 이를 이진(binary) 특징 집합(벡터)으로 변환하는데, 이로 인해 범주형 변수의 특징 벡터의 차원은 매우 높으며 동시에 벡터의 대부분의 값이 0으로 희소(sparse)한 특성을 가진다. 임베딩 레이어는 이러한 고차원의 희소 특징을 저차원의 잠재(latent) 공간에 삽입(embedding)함으로써, 데이터를 조밀한(dense) 입력 공간에 매핑하여 학습이 효과적으로 수행되도록 전처리한다. 임베딩 레이어는 이러한 CTR 데이터의 특징으로 인해 기존 CNN에 일반적으로 추가되는 레이어이다[12-14]. 임베딩 특징 벡터는 특징 정보를 학습하도록 컨볼루션 및 풀링 레이어에 제공된다. 학습된 모든 잠재 특징은 심층 연결망에서 처리되어 모델의 마지막 단계에서 CTR을 예측한다.

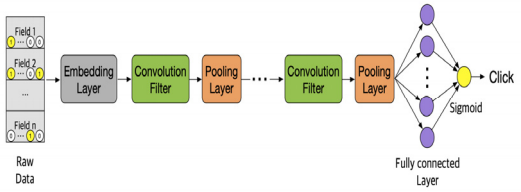

그림 2. CTR 예측을 위한 CNN 아키텍쳐

하지만, CTR 데이터 특성과 CNN의 학습 동작 원리로 인해 CTR 예측을 위한 naive한 CNN의 적용은 성능의 저하를 피할 수 없다. 비디오나 텍스트 데이터와는 달리 CTR 데이터의 특징들은 정렬 순서가 의미를 갖지 않는다. 예를 들어 (이름, 연령, 키, 성별) 또는 (나이, 이름, 키, 성별) 인 특징의 정렬 순서는 데이터의 의미를 설명하는 데 차이가 없다. 따라서, 특징 벡터는 임의의 순서로 배열될 수 있다. 나아가, CNN은 컨볼루션 및 풀링 계층에서 필터 사이즈에 해당하는 영역의 로컬 정보만을 캡처하므로 필터 안에 포함되지 않은 특징 간의 상호 작용에 대한 학습이 불가하다. 이는 CNN 기반 CTR 예측 모델의 성능이 특징 벡터의 배열 선택에 따라 차이가 존재함을 의미하지만, 이 문제에 대한 심층적인 연구는 거의 이루어지지 않았다.

III. 제안한 방법론

CTR 예측에서 필드 데이터의 배열은 예측 정확도에 크게 영향을 미치기 때문에 이를 해결하기 위한 접근이 필요하다. 많은 배열 중 하나의 최적 배열을 선택하는 방안은 많은 시간이 소요될 뿐만 아니라 선택되지 못한 배열에 의해 제공되는 잠재적 정보를 활용할 수 없어 예측 결과의 정확성을 높일 기회를 잃게 된다. 이런 배경에서, 본 연구는 다중 배열을 사용하여 특징 간 연관성 정보를 최대한 추출하여 활용할 수 있는 모델을 제안한다. 먼저, 특징의 조합된 모든 영향을 학습할 수 있게 하는 데이터 배열 생성 방법을 1절에서 논의한다. 생성된 배열들을 활용할 수 있는 다중 학습 모듈을 탑재한 새로운 CNN 모델은 2절에서 제안한다.

1. 배열 생성 방법

이 절에서는 CNN에서 로컬 정보를 효과적으로 탐색할 수 있는 필드 배열을 생성하는 방법을 소개한다. 배열 생성은 학습을 위한 서로 다른 로컬 정보를 제공하여 학습 모듈이 가능한 많은 특징들의 조합을 학습하여 예측 능력을 높이는 것을 목표로 한다.

데이터의 로컬 연관성이 큰 이미지(인물 사진의 눈, 코, 입 등)나 텍스트 데이터와는 달리 CTR 데이터는 컨볼루션 필터 크기와 데이터의 필드 배열에 따라 필드 간 로컬 연관성이 존재할 수도 그렇지 않을 수도 있다. 이는 CTR 예측을 위해 CNN을 활용하는 경우 특징 파악을 위해 데이터의 필드 배열의 조작이 필요함을 시사한다. 전통적인 CNN 모델은 하나의 데이터 배열을 입력으로 수용하기 때문에 필터 크기 내에 포함된 필드들과 이와 관련한 특징 간 연관 관계는 파악할 수 있으나 학습을 위해 필터가 이동하게 되면 이동 전과 이동 후 수집된 필드들간 연관성을 파악할 수 없는 경우가 발생한다. 예를 들어, (F1, F2, F3, F4) 순으로 총 4개의 필드가 존재하고 필터 크기가 3이라고 할 때, 필터는 먼저 (F1, F2, F3)의 로컬 정보에 대한 컨볼루션을 수행한다. 필터 stride를 1로 설정할 경우 필터는 1자리 옆으로 이동하여 (F2, F3, F4)의 로컬 정보를 수집하는데 이 경우 첫 번째 필터에서 수집된 F1과 두 번째 필터에서 수집된 F4 간의 관계는 파악할 수 없게 된다.

이처럼, 필터 사이즈가 주어진 상황에서 특징 간 관계를 완전히 추출하기 위해서는 전체 필드 간의 모든 관계를 학습하는 방법이 필요하다. 이 문제를 해결하기 위한 아이디어를 얻기 위해 상기에서 언급된 예제를 고려하자면, (F1, F2, F3, F4)에서 중복과 순서를 고려하지 않는 크기 3인 조합은 총 4개로 (F1, F2, F3), (F1, F2, F4), (F1, F3, F4), 그리고 (F2, F3, F4)이다. 각 조합은 필터가 학습할 수 있는 로컬 정보로 모든 필드 간의 연관성을 파악하기 위해서는 위의 조합을 모두 포함하는 필드 배열들이 필요하다. 아울러, 생성된 배열들의 수는 최소화되어야 하는데, 그렇지 않으면 중복된 특징 조합이 발생하고 이는 CNN 모델의 학습 매개 변수의 수를 증가시켜 계산의 부담이 커지기 때문이다. 이러한 조건을 만족시키는 배열은 (F4, F1, F2, F3)과 (F2, F3, F4, F1)이다.

Algorithm Field Array Configuration

Notation :
Δ: set of features
B: set of candidate sequence
Ω: set of completed sequence
α: working sequence
κ: size of filter
λ: # of empty slot in **α**
π: set of common features

Input : Δ, κ
output : Ω

```
(1)  while |B| > 0
(2)      if |B| == 1
(3)          Add the element of B in Ω and Exit algorithm
(4)      Choose a candidate β in B
(5)      while λ > 0
(6)          π ← Forward_Search(α, β, κ)
(7)          if | (β−π) ∩ α| == 0
(8)              if |β| − |π| < λ
(9)                  Append elements of π in front of α
(10)                 λ ← λ − |π|
(11)                 if λ == 0
(12)                     Add α in Ω and Exit while
(13)         else
(14)             π ← Backward_Search(α, β, κ)
(15)             if | (β−π) ∩ α| == 0
(16)                 if |β| − |π| < λ
(17)                   Append elements of π subsequent to α
(18)                     λ ← λ − |π|
(19)                     if λ == 0
(20)                         Add α in Ω and Exit while
(21)             else
(22)                 Exit while
```

그림 3. Field Array Configuration 알고리즘 의사코드

이 문제를 일반화시키면 총 n개의 필드와 필터 크기가 κ로 주어졌을 때(stride는 1로 가정), $CLSUBn_{\kappa}$ 개의 특징 조합들을 모두 포함하는 길이 n의 배열들의 개수를 최소화하는 문제가 된다. 이 문제는 NP-hard 문제로 솔루션을 찾기 위해 본 논문에서는 Field Array Configuration 알고리즘을 제안한다.

[그림 3]은 제안된 알고리즘의 의사 코드(pseudo code)이다. B는 n개의 특징에 대해 중복을 허용하지 않고 κ(필터 크기)를 선택했을 때 생성 가능한 특징 조합들의 집합이다. 먼저 B에서 하나의 원소 α를 선택한 후 B의 나머지 원소들에 대해 (1) Forward_Search(6행)과 (2) Backward_Search(14행)를 α의 길이가 n이 될 때까지 반복 수행한다. 이 과정은 B에 적어도 두 개 이상의 원소가 있을 때까지 계속된다(2-3행).

Forward_Search는 input으로 (α, β, κ)를 입력받아 α의 앞에서 κ-1까지의 위치 내에서 후보 원소 β와의 공통된 특징을 찾는다. 찾게 된 공동 특징을 β에서 제외하고 α에 할당 가능한 특징을 찾아낸다. 만약, 할당 가능한 특징이 이미 α에 할당되어 있으면 탐색을 중단하고 Backward_Search로 이동한다(7행). 할당 가능한 특징 중 α에 할당되지 않았고 할당할 특징들의 숫자가 남은 자릿수보다 같거나 작다면 할당할 특징들을 α의 앞부분에 추가하여 배열시킨다(9행). 이 과정 후 할당할 수 있는 자릿수가 0이 되면 α에 대한 할당이 완료되고 α를 Ω에 추가한다(12행). Backward_Search은 기본적으로 Forward_Search와 동작은 동일하나 특징의 할당을 α의 끝에 추가한다는 점이 다르다(17행). 의사 코드에는 간결함을 위해 Forward_Search와 Backward_Search의 자세한 수행 과정은 생략하여 표기하였다.

2. 제안한 CNN 모델

이 절에서는 1절에서 소개된 다중 배열들을 학습하는 CNN 기반 모델을 소개한다. [그림 4]는 제안된 모델의 구조를 나타낸다. 모델에 주어진 raw data를 입력으로 받아 1절에서 소개된 특징 배열 생성을 Field Array Layer에서 수행한다. n개의 데이터 필드와 필터 사이즈 κ가 주어졌을 때 총 $CLSUBn_{\kappa}$ 개의 CNN 학습 모듈이 필요하나 n이 증가할 때 그 수가 기하급수적으로 커지므로 학습에 소요되는 시간이 매우 크다. 따라서, 실제 구현은 사용할 학습 모듈의 수를 정하면 알고리즘

을 통해 생성한 결과물 중에서 임의로 모듈의 수에 해당하는 배열들을 선택하여 각 학습 모델의 입력으로 사용되게 한다. 서로 다른 필드 배열로 생성된 임베딩 특징 벡터는 각 CNN 학습 모듈에 의해 개별적으로 학습되는데, 학습 모듈은 컨볼루션과 풀링 레이어의 반복을 통해 특징을 학습한다.

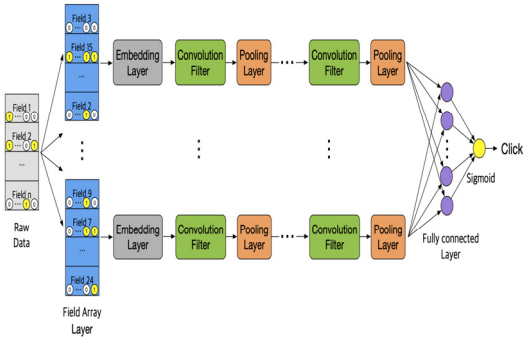

그림 4. 다중 배열 CNN 아키텍처

그런 다음 모든 학습 모듈의 출력은 fully connected layer의 입력으로 전달된다. 개별 학습 모듈에서 학습한 특징들이 일렬로 전달되어 예측의 정확성을 향상시키기 위해 각 배열의 정보를 보다 효과적으로 학습한다. 제안된 구조는 다중 배열들을 학습에 활용한다는 점에서 단일 배열만을 고려한 기존 연구[7][8][15]와는 차별된다. 또한, [9]에서 다중 임베딩 벡터의 배열을 생성하는 방법을 제안하였으나 필터 사이즈에 따른 로컬 정보 획득 범위의 영향은 고려되지 못한 점에서 본 연구는 차별성을 갖는다.

IV. 성능 분석

이 장에서는 제안된 모델의 성능을 대규모 데이터 세트를 통해 검증한다. 먼저 1절에서는 실험 설정이 소개되고, 제안된 모델의 성능 평가는 2절에서 수행된다.

1. 실험 설정

1.1 데이터 세트

성능 평가를 위해 사용할 데이터는 Criteo 데이트 세트로서 Criteo Display Advertising Challenge 2013[16]에서 공개되었는데, 디스플레이 광고에 대한 특징 값들과 클릭 피드백 정보가 담긴 수백만 건의 샘플을 포함하고 있다. 구체적으로 특징 값들은 정수 값을 가지는 13개의 특징(대부분 카운트 특징)과 범주(categorical) 값을 가지는 26개의 특징으로 구성되는데, 일반적으로 범주 값을 가지는 특징은 one-hot encoding과 embedding 처리가 되는 대상이다. 따라서, 본 실험에서는 희소 데이터에 대한 CNN 성능 평가에 초점을 맞추기 위해 13개의 정수 값을 갖는 dense 특징들을 제외한 26개의 범주 값을 가지는 특징들만을 다룬다. 또한 막대한 데이터량과 심각한 클래스 불균형(3% 샘플만 양의 클릭)으로 인해 양의 클릭과 음의 클릭(클릭 하지 않음) 비율을 1 : 1로 조정한 총 4만 개의 데이터를 평가에 사용한다.

1.2 Base 모델과 평가 지표

제안된 모델과의 성능 비교를 위해 고려한 모델은 CNN과 CNN-RD이다. CNN는 전통적인 CNN 모델로서 단일 학습 모듈을 가지며 특징 배열에 대한 조작이 없는 모델이다. CNN-RD는 제안된 모델과 동일한 학습 모듈 구조를 가지는 모델로서 각 학습 모듈에서 사용하는 배열이 무작위로 생성되는 점만 제안된 모델과 다르다. 본 비교 평가의 목적은 희소 데이터에 대한 CNN 기능 향상 분석에만 초점을 두기 때문에 기존에 제안된 희소 데이터와 CNN을 활용하지 않는 다른 딥러닝 모델들은 고려하지 않는다. 모델들의 성능 평가를 위해 사용할 지표로는 클래스 분포가 균일한 분류 문제에서 일반적으로 사용되는 ROC (Receiver Operating Characteristic) AUC (Area Under Curve)를 사용한다.

1.3 파라미터 설정

실험을 위해 사용한 embedding의 사이즈는 12로, 이를 통해 데이터 세트의 특징 필드는 크기가 12인 임베딩 특징 벡터로 매핑된다. 모든 모델의 CNN은 3쌍의 컨볼루션 및 풀링 레이어로 구현된다. 커널의 크기는 7×1이고 각 컨볼루션 레이어에서 사용된 커널의 수

는 (20, 25, 30)이다. 매 컨볼루션 후에는 그래디언트 소실을 방지하기 위해 배치 정규화를 수행한다. 정규화 결과에 대한 활성화 함수로 ReLU를 사용한다. 풀링 레이어에서는 Max-pooling을 사용하고, 풀링 사이즈는 2×2, 풀링 보폭은 1을 사용한다. CNN의 결과는 하나의 fully-connected 층에 입력으로 연결되는데 256개의 유닛을 사용한다. 학습을 위한 optimizer로는 Adam을 사용한다. 실험은 텐서플로우 플랫폼(버전 2)에서 수행되었으며 제안된 기법과 CNN-RD에서 사용되는 특징 배열들은 각각 3장에서 제안한 알고리즘과 텐서플로우의 랜덤 모듈을 활용하여 CNN 학습 모듈의 수에 해당하는 개수만큼 생성하여 사용한다.

2. 결과 분석

2.1 제안된 모델의 예측 정확성

이 절에서는 제안된 모델과 평가 모델들의 성능 비교·분석을 수행한다. Criteo 데이터 세트에 대한 모델들의 최종 성능을 요약한 결과는 [표 1]과 같다.

표 1. 최종 AUC 결과

방법	AUC
CNN (Base)	0.6883
CNN-RD	0.7222
CNN-Proposed	0.7308

각 모델의 결과는 10회 실행의 평균값이며 표준 편차는 CNN: 6.4×10^{-4}, CNN-RD: 7.1×10^{-4}, CNN-Proposed: 6.9×10^{-4} 이다. 제안된 모델(CNN-Proposed)은 가장 높은 성능을 달성하였는데 제안된 다중 배열 생성과 이를 활용한 구조가 CNN 성능 향상에 효과적임을 보여준다. 성능 향상의 원인에 대한 자세한 분석은 다음 절에서 다루도록 한다. 본 실험은 CNN 모델의 희소 특징 탐색에 초점을 맞추기 위해 dense data를 사용하지 않았기 때문에 모든 모델의 AUC 값이 전반적으로 낮았는데 CNN의 AUC는 모델 성능의 poor와 fair 판별 기준인 0.7 보다 낮은 AUC를 달성하였다. 이는 CTR 예측 작업에 전통적인 CNN을 그대로 사용하는 것은 효과적이지 않음을 보여준다.

2.2 다중 배열 성능 분석

이 절에서는 기존 CNN 대비 다중 배열 CNN 학습 모델의 효과에 대해 설명한다. 학습 모델의 개수에 따른 제안된 모델의 AUC 성능과 relative improvement (RI)[1] 값에 대한 결과가 [표 2]에 제시되어 있다.

표 2. 제안된 다중 학습 모듈 성능 결과

# of CNN	AUC	RI(%)	Increment
1	0.6883	0	0
3	0.7026	7.5942	7.5942
6	0.7131	13.1704	5.5762
9	0.7201	16.8879	3.7174
12	0.7258	19.9150	3.0270
15	0.7296	21.9330	2.0180
18	0.7308	22.5703	0.6372

CNN 학습 모델 개수가 1인 경우는 주어진 특징 배열의 조작이 없이 그대로 활용되는 기존 CNN 모델과 동일하다. 표에서 확인할 수 있듯이 CNN 학습 모델의 개수가 증가함에 따라 AUC는 지속적으로 증가하여 18개일 때 RI가 CNN 대비 22.57%의 성능 향상을 달성하였다. 이러한 결과는 단일 특징 학습 모듈에 비해 다중 학습 모듈에 의해 추출된 로컬 패턴이 모듈의 개수가 증가함에 따라 AUC 성능 향상에 지속적으로 기여하고 있음을 나타낸다. 하지만, CNN 모듈의 개수에 따른 RI 값의 증분(표의 increment 값)은 개수가 증가할수록 감소하여 3개에서 7.59 %였던 증가분은 18개에서 0.63 %로 감소하였다. 이는 수집된 로컬 정보의 성능에 대한 한계 기여도가 모듈 수의 증가에 따라 감소함을 나타낸다.

2.3 배열 생성 효과 분석

이 절에서는 제안된 배열 생성 효과를 분석하기 위해 제안된 모델과 제안된 모델과 CNN-RD의 AUC 성능에 대한 비교를 수행한다. [표 3]에 따르면, 제안된 모델은 CNN-RD와 비교하여 학습 모듈이 18개일 때, AUC가 3.87 % 개선되었다. 실제로 오프라인 AUC의 작은 개선은 온라인 CTR의 큰 증가를 가져올 수 있는

1 Base 모델(B) 대비 비교 모델 m의 RI(m)은 $\left(\frac{AUC(m)-0.5}{AUC(B)-0.5}-1\right)\times100$ 로 정의[17]

데, [13]에서 보고된 바에 따르면 AUC의 0.275% 개선이 3.9%의 실제 온라인 CTR의 개선으로 이어졌다. 구글, 화웨이 등의 App Store의 일일 매출은 수백만 달러 규모이고 CTR의 소폭 상승은 매년 수백만 달러의 추가 수익을 발생하는 효과를 가져온다는 점에서 유의미한 성능 향상이라고 할 수 있다.

표 3. 제안된 배열 생성 성능 결과

# of CNN	AUC		RI(%)
	CNN-Proposed	CNN-RD	
1	0.6883	0.6883	0
3	0.7026	0.7019	0.3467
6	0.7131	0.7117	0.6613
9	0.7201	0.7159	1.9453
12	0.7258	0.7184	3.3882
15	0.7296	0.7213	3.7505
18	0.7308	0.7222	3.8703

아울러 모든 학습 모듈의 개수에 대해 제안된 모델의 AUC 값은 CNN-RD 값을 상회하였다. 이는 제안된 배열 생성 기능이 랜덤하게 생성하는 것에 대비하여 성능상 효과가 있음을 나타낸다. 이러한 효과는 학습 모듈의 개수가 증가할수록 RI[2] 값이 증가함을 통해 알 수 있다. 이 결과는 학습 모듈의 수가 적을 때에는 생성된 배열의 조작 효과가 랜덤하게 생성된 배열 대비 작지만, 모듈의 수가 커짐에 따라 배열 조작을 통해 획득할 수 있는 중복되지 않는 특징 정보 수의 증가로 성능 향상에 기여할 수 있는 더 많은 기회가 존재함을 의미한다.

V. 결론

1. 시사점

본 논문에서는 CTR 예측에서 기존 CNN이 가지는 특징 간 내재된 연관성 추출의 한계를 극복하기 위한 다중 배열 CNN 모델을 제안한다. 제안한 모델은 각 배열에 내재된 특징간 결합 정보를 추출하고 추출된 정보를 취합하여 효과적으로 학습하기 위해 복수 개의 CNN 학습 모듈과 심층 신경망을 연결한다.

실험 결과에 따르면 AUC 측면에서 하나의 배열만 사용하는 기존 CNN 모델보다 배열 생성에 따른 특징 정보의 추가적인 제공이 정확도의 향상을 달성한다는 것을 보여준다. 아울러, 실험 결과에서 확인된 모듈 수에 따른 로컬 정보의 한계 기여도 감소는 주어진 데이터 크기에 대해 성능과 계산 비용 측면에서 최적 모듈의 개수가 존재함을 나타낸다.

특징 배열 생성 방법은 CNN 필터의 크기를 고려하여 CTR 데이터의 내재된 특징들을 최대한 추출하기 위해 제안되고 무작위 배열 생성과 비교할 때 성능 우위가 실험적으로 확인된다. 나아가 이러한 결과에 근거하여 얻을 수 있는 시사점은 제안된 모델과 유사한 성능을 얻기 위해서 CNN-RD는 더 많은 학습 모듈이 필요함을 의미하며 이는 계산 비용의 증가를 의미한다. 따라서, 제안된 방법으로 생성된 일련의 배열은 학습에 더 유용한 정보를 비용 측면에서 효과적으로 제공한다고 결론을 내릴 수 있다.

본 연구 결과물은 특징에 대한 engineering의 필요성이 없어 온라인 사이트의 다양한 소비자 특성의 활용을 용이하게 하고 소비자 행동에 대한 정교한 예측을 수행할 수 있어 다양한 온라인 추천 시스템에 활용될 수 있다. 일례로, 아이템에 대한 구매 가능성을 높은 예측값을 보이는 사용자 그룹부터 우선순위를 정하는 캠페인을 실행하여 온라인 고객의 전환율을 높이는 데 활용될 수 있다.

2. 한계점 및 향후 연구 방향

본 연구에서는 CNN의 희소 특징 추출 성능에 초점을 맞추기 위해 데이터 세트에서 범주형 데이터만을 고려하였다. 이것은 CNN 기반의 최근 연구들이 78~81%의 AUC를 달성한 것에 비해 다소 낮은 AUC를 얻은 이유가 된다. 이를 보완하기 위해 연속형 데이터와 같은 밀집 특징을 갖는 입력을 모형에 반영하는 연구가 필요하다. 제안된 모델을 확장하여 밀집 특징까지 포함한 완성된 모델은 추가적인 AUC 향상을 달성할 것으로 예상되며 모든 데이터 세트를 활용하는 CNN 기반의 선행 연구들과의 비교 연구를 통해 CTR 예측의 새로운 방향 도출도 가능할 것이다. 본 연구에서 얻은 의

2 각 CNN 개수에서 RI의 base는 해당 CNN-RD의 AUC 값으로 정의함

미있는 결과 중 하나는 주어진 데이터 크기에서 배열 증가에 따른 추가 배열의 성능 기여도가 점진적으로 감소한다는 점이다. 이는 데이터 크기에 효과적인 배열 선정 방식에 대한 연구가 필요함을 의미하며 이를 통해 네트워크 구조가 간소해져서 학습에 소요되는 시간을 크게 절감할 수 있을 것으로 기대된다.

참 고 문 헌

[1] H. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, and D. Golovin, "Ad click prediction: a view from the trenches," ACM SIGKDD, 2013.

[2] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," SIGIR, 2011.

[3] S. Rendle, "Factorization machines," ICDM, 2010.

[4] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," KDD, 2008.

[5] Y. Qu, B. Fang, W. Zhang, R. Tang, M. Niu, H. Guo, Y. Yu, and X. He, "Product-based Neural Networks for User Response Prediction over Multi-field Categorical Data," ACM Transactions on Information Systems, Vol.37, No.1, pp.1-35, 2019.

[6] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems," ACM SIGKDD, 2018.

[7] Q. Liu, F. Yu, S. Wu, and L. Wang, "A convolutional click prediction model," ACM CIKM, 2015.

[8] B. Liu, R. Tang, Y. Chen, J. Yu, H. Guo, and Y. Zhang, "Feature Generation by Convolutional Neural Network for Click-Through Rate Prediction," WWW, 2019.

[9] P. P. K. Chan, X. Hu, L. Zhao, D. S. Yeung, D. Liu, and L. Xiao, "Convolutional Neural Networks based Click-Through Rate Prediction with Multiple Feature Sequences," IJCAI, 2018.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," ICLR, 2015.

[11] G. Huang, Z. Liu, V. D. M. Laurens, and K. Weinberger, "Densely Connected Convolutional Networks," CVPR, 2017.

[12] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao, "Deep crossing: Web-scale modeling without manually crafted combinatorial features," ACM KDD, 2016

[13] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," DLRS, 2016.

[14] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data," ECIR, 2016.

[15] B. Edizel, A. Mantrach, and X Bai, "Deep character-level click-through rate prediction for sponsored search," SIGIR, 2017.

[16] http://labs.criteo.com/downloads/download-terabyte-click-logs

[17] L. Yan, W. Li, G. Xue, and D. Han, "Coupled group lasso for web-scale ctr prediction in display advertising," ICML, 2014.

저 자 소 개

김 태 석(Tae-Suk Kim) 정회원

- 1998년 2월 : 한국과학기술원 산업경영학과(공학사)
- 2000년 2월 : 한국과학기술원 산업공학과(공학석사)
- 2005년 2월 : 한국과학기술원 산업공학과(공학박사)
- 2005년 8월 ~ 2007년 8월 : UIUC Post-Doc
- 2007년 8월 ~ 2009년 8월 : UCR Post-Doc
- 2009년 10월 ~ 2016년 2월 : 삼성종합기술원 전문연구원
- 2016년 3월 ~ 현재 : 배재대학과 경영학과 교수

〈관심분야〉 : IT 경영, 시스템 최적화