DBpia

논문 18-43-08-14 The Journal of Korean Institute of Communications and Information Sciences '18-08 Vol.43 No.08
https://doi.org/10.7840/kics.2018.43.8.1347

TensorFlow를 활용한 검색 키워드의 최적 순위 예측 알고리즘

지 민 준◆, 박 현 희°

Prediction Algorithm for Optimal Ranking of Search Keyword by TensorFlow

Minjun Ji◆, Hyunhee Park°

요 약

검색 키워드의 가치는 대체로 검색어의 빈도수에 기반하여 산정되지만 검색 엔진에서는 검색 키워드에 대한 가격을 실시간으로 공개하지 않고 블라인드 경매 방식을 통하여 입찰하도록 구성한다. 결국 검색 키워드의 가격 예측을 위하여 수동적인 통계 방식으로 가격을 선정함으로써 원하는 최적의 순위에 도달하기가 어렵다. 또한 쇼핑 의도를 포함하고 있을 가능성이 높을 것으로 추정되는 검색 키워드를 추출하여 순위를 집계하고 다시 가격을 예측하는 것은 자동 추출 방식이 아닌 이상 많은 인력의 필요와 비효율적인 작업을 하도록 만든다. 이 과정을 빠르고 효율적으로 수행하기 위한 자동화 방법론으로써 본 논문에서는 국내 최대의 검색 엔진에서 발생하는 검색 키워드를 수집하여 검색 키워드에 대한 최적의 순위를 예측하는 모델링 기법을 제안한다. 특히 머신러닝 알고리즘을 적용하여 각 기법별로 최적의 순위에 도달하는지에 대한 예측의 정확도를 비교 분석함으로써 자동화된 검색 키워드의 순위 예측 시스템의 구축을 위한 기법을 제시한다.

Key Words : Prediction, Machine learning, keyword, search engine, Tensor Flow

ABSTRACT

Although the value of the search keyword is generally calculated based on the frequency of the search word, the search engine configures the price of the search keyword to be bid through the blind auction method without disclosing the price in real time. As a result, it is difficult to reach a desired optimal ranking by selecting a price with a statistical method that is passive in order to predict the price of the search keyword. In this paper, we propose a modeling algorithm to predict the optimal ranking of search keywords by collecting search keywords from the largest search engine in Korea. In particular, we propose a mechanism for constructing an automated ranking prediction system for a search keyword by comparing and analyzing the prediction accuracy whether the optimal ranking is reached for each algorithm by applying a machine learning algorithm.

※ 본 연구는 2017년 중소벤처기업부 기술개발사업 (C0563763) 지원으로 수행되었습니다.
◆ First Author : (ORCID:0000-0002-4168-7895)e-Glue Comminications, inciojs@gmail.com, 정회원
° Corresponding Author : (ORCID:0000-0003-3810-7367)Korean Bible University Department of Computer Software, parkhyunhee@gmail.com, 종신회원
논문번호 : 201808-243-C-RN, Received August 13, 2018; Revised August 17, 2018; Accepted August 21, 2018

www.dbpia.co.kr

Ⅰ. 서 론

인터넷 광고는 과거 마케팅의 한 부분을 차지했던 것과는 달리 마케팅 영역의 핵심으로 부상하고 있으며, 그 방법과 방식이 매우 다양해지고 있다. 특히 인터넷 광고 시장에서 검색 광고가 차지하는 비율은 점점 늘어나고 있는 추세이다[1]. 또한 전체 광고 집행비 중 온라인 광고가 49.7%를 차지하고 있으며 그 중 45.2%가 검색 광고를 위해 사용되고 있다. 따라서 광고 집행을 위한 입찰 솔루션으로써 RTB (Real Time Bidding) system의 도입이 증가하고 있는 추세이다. 검색 광고를 통한 광고 집행을 위하여, 글로벌 기업인 구글은 GoogleAD와 같이 검색 엔진에서 제공하는 RTB system을 통해 광고를 개시할 수 있도록 한다[2,3]. 또한 국내의 경우 대표적으로 네이버의 검색엔진을 통해 검색 광고를 개시하고자 하는 경우 네이버 서치애드 RTB system을 통하여 광고 입찰을 시행한다. 예를 들어, 네이버의 검색 광고를 이용하여 광고를 집행하는 경우, 1개의 검색 키워드에 대하여 n개의 광고가 집행될 수 있다. 검색 키워드의 순위는 입찰 가격을 비롯한 여러 요인에 따라 달라질 수 있다. 예를 들어, "보험"이라는 키워드를 검색 광고에 개시하는 경우 n개의 광고가 개시된다. 이 경우 검색 키워드의 순위는 입찰 가격을 비롯하여 여러 가지 요소에 의해 달라질 수 있다[4]. 그러나 광고 입찰 과정은 비공개로 진행되기 때문에 원하는 순위에 광고를 개시하기 위해 얼마의 금액을 입찰해야 하는지 알기 어렵다. 또한 입찰가격은 시간대별로 다른 가격을 형성할 수 있기 때문에 입찰 가격의 예측이 더욱 어려워진다. 이러한 문제점을 해결하고자 본 논문에서는 키워드 검색 광고의 입찰 가격을 예측하는 feature engineering 기법과 머신러닝 알고리즘을 이용한 순위 예측 방법론을 제시하고자 한다.

본 논문에서는 검색 광고의 데이터 셋을 수집하기 위한 전처리 과정과 머신러닝 알고리즘을 이용한 모델링 과정까지의 일련의 과정을 기술한다. 특히 검색 광고의 가격 예측을 위하여 Tensor Flow에서 linear regression, logistic regression, softmax, ANN 기법을 적용한 가격 예측 결과를 제시하고 최종적으로 검색 광고의 가격을 실시간적으로 예측하기 위한 가장 정확도가 높은 적합한 모델을 제시한다. 본 연구의 구성은 제2장에서 관련된 선행연구를 살펴보고 제3장에서 본 연구의 연구방법과 데이터 수집 방법 및 데이터 전처리 과정에 대하여 기술한다. 제 4장에서는 다양한 모델링 기법을 적용한 모델의 비교 분석 및 검증을 제시한다. 제 5장 결론에서는 연구 결과의 요약과 시사점 및 향후 연구 방향을 기술한다.

Ⅱ. 관련 연구

인터넷 키워드 광고는 인터넷 사용자가 검색 엔진에서 검색을 통해 나타나는 결과에 연관된 광고를 노출시켜 진행하는 형태의 광고이다. 특히 키워드 검색 광고는 광고와 밀접한 관계가 있는 유사 키워드를 사용자가 검색했을 때 나타남으로써 광고 기업의 제품을 구매할 가능성이 높은 잠재고객에게 노출시킬 수 있는 장점이 있다. 이러한 키워드 검색 광고는 특정 제품에 대한 직접적인 정보가 검색 되었는지 알 수 있기 때문에 효과적인 광고 모델로 인정되고 있다. 이러한 키워드 검색 광고의 광고 비용은 일반적으로 클릭당 비용을 지불하는 종량제 형태의 CPC (Cost Per Click) 광고 방식이 이용된다[5,6].

키워드 검색 광고의 효과에 대한 기존 연구를 보면, 키워드 검색광고와 인적 특성 중심의 이메일 광고를 타켓팅 한 관점에서 비교한 연구 결과가 있다[7]. 인터넷 광고 효과를 측정하기 위해 방문 횟수, 방문자 수, 시간당 접속자 수, 노출 수, 클릭 수를 제시한 결과도 제시되었다[8]. 인터넷 키워드 광고는 광고 비용 산정 기준에 따라 앞서 기술한 CPC 방식이 있고 또한 정액제 방식인 CPM (Cost Per Mile)과 행위 당 과금 검색 광고 방식인 CPA (Cost Per Action)이 제안되어 있는데 이러한 검색 광고의 광고 비용 산정 기준을 비교한 연구 결과도 제시되었다[9]. 일반적으로 인터넷 검색 사용자가 검색 매체에서 일정 키워드를 클릭한 후 클릭 당 과금되는 방식의 CPC 키워드 검색 광고는 국내 대표적인 사례이다. 각각의 검색 키워드에 대하여 검색이 일어났을 때 입찰가의 내림차순으로 링크된 결과가 검색결과에 표시된다. 검색 결과의 노출은 일반적으로 5위까지의 순위로 표현되고, 실시간 경매 방식으로 순위가 결정되게 된다. 이 광고 방식의 경우 광고비 지출이 사용자의 클릭이 발생한 경우에만 광고비의 지출이 발생함으로 수많은 키워드를 저렴한 비용으로 사용할 수 있다는 장점이 있다[10]. 하지만 CPC 광고 방식은 관련 검색 키워드를 모두 유추해야 하고 관련 검색 키워드에 대한 가격 예측을 모두 해야 한다는 점에서 가격 예측을 더욱 어렵게 한다. 예를 들어, "보험" 이라는 검색 키워드 광고를 위해서는 "자동차 보험", "상해보험", "1위보험" 등 다양한 연관 검색 키워드를 유추하여 검색 키워드를 선정하고 각 키워드에 대하여 가격을 예측해야 한다. 이러한 점에서 검

www.dbpia.co.kr

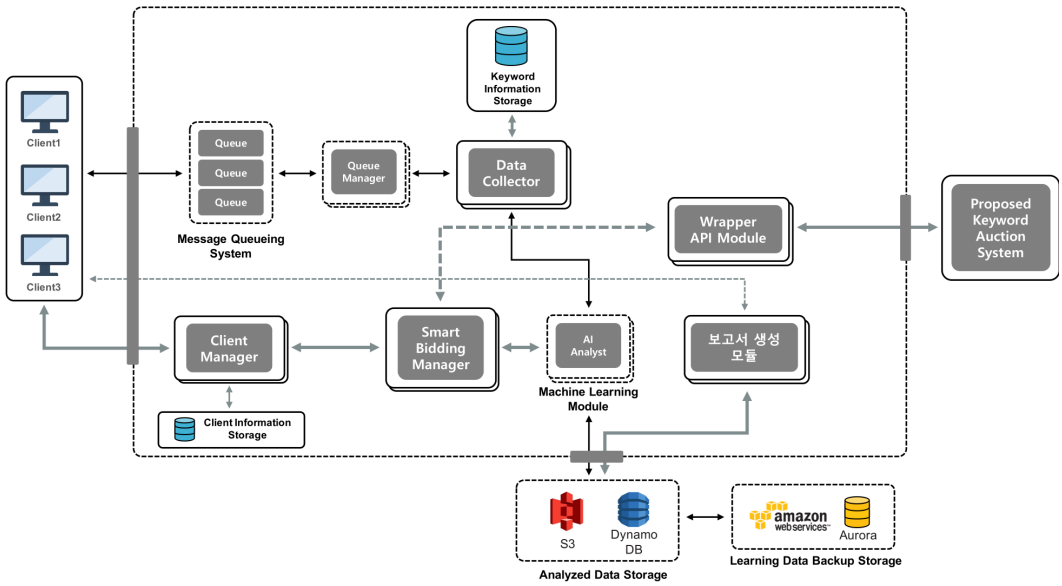


그림 1. 시스템 구성도
Fig. 1. System Architecture.

색 키워드에 대한 가격 예측은 수동적인 방식이 아닌 자동화된 예측 시스템을 필요로 하고 있다. 본 연구에서는 이러한 점을 해결하고자 머신러닝 알고리즘을 적용한 키워드 검색 광고의 가격을 예측하는 모델링 기법을 제시하고자 한다.

Ⅲ. 검색 키워드의 순위 예측을 위한 모델링 기법

3.1 시스템 구성 및 전처리 과정

키워드 검색 데이터를 수집하기 위하여 클라이언트 노드에서 발생되는 키워드 검색 결과를 크롤링 하는 방식을 사용한다. 본 연구에서는 국내 검색 엔진인 "N"사에서 검색되는 결과를 기반으로 키워드 검색 데이터를 수집하였다. 검색 키워드와 고객 정보는 정형화된 정보이므로 이에 대한 관리의 용이성을 위해 RDBMS (Relational Database Management System) 기반의 MySQL (Structured Query Language)을 데이터베이스로 이용한다. 검색 키워드의 입찰을 위한 분석 데이터는 JSON (JaveScript Object Notation) 파일로 작성되며, 이는 클라우드 시스템으로 구성된 Amazon AWS의 S3 (Amazon Simple Storage Service)에 업로드 되어 머신러닝 알고리즘의 학습 데이터로 사용될 수 있도록 구성하였다. 만일 학습이 끝난 검색키워드 데이터들은 보관 가격의 최적화를 위해서 클라우드 아카이브인 Amazon Glacier로 이전되어 관리되도록 구성하였고, 로그 데이터의 경우 양이 많고 각 서비스들과 유기적으로 연결되어야 하기 때문에 Amazon AWS Aurora Database (Managed Cloud Database)를 사용하여 RDBMS에 로그 정보들을 저장하도록 구성한다. 여러 개의 클라이언트 노드에서 동일한 키워드 검색 결과가 발생하는 경우 하나의 키워드 데이터 값으로 저장하게 된다. 이때 동일한 키워드의 발생 횟수는 카운팅 하여 저장한다. 모든 수집 데이터는 메인 서버를 통해 취합하게 된다. 그림 1은 시스템으로 구성될 수 있는 모듈화된 키워드 예측 시스템의 구성도를 보여준다.

검색 키워드 데이터는 동일한 검색 키워드라 할지라도 시간에 따라 가격 차이가 발생한다. 특히 키워드 데이터는 주중과 주말에 따라 그 가격이 달라진다. 또한 24시간을 기준으로 보면 동일한 검색 키워드라 할지라도 시간에 따른 가격 차이가 크게 발생함을 알 수 있다. 그림 2를 보면 키워드 데이터의 가격이 24시간 동안 변화하는 것을 볼 수 있다. 결국, 예측 모델에 큰 영향을 미치는 요인은 시간임을 확인할 수 있었다. 따라서 본 논문에서는 24시간을 30분 단위로 나누어 48개의 시간 단위로 나누어 시간 데이터 셋을 구성한다. 이를 위하여 48개의 시간 데이터 셋은 one-hot encoder의 형태로 구현한다.

키워드 데이터의 경우 한국어로 된 언어를 연산 처리해야 하기 때문에 word2vec 라이브러리를 사용하여 키워드 데이터를 수치 값으로 변환한다. 이 과정을 통하여 데이터 셋을 독립 변수와 종속 변수의 조합으

www.dbpia.co.kr

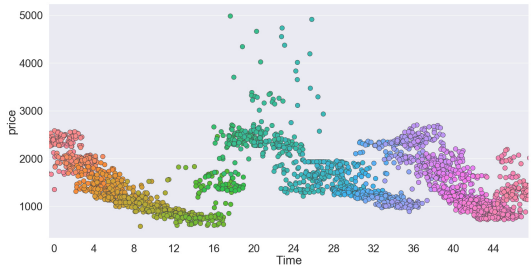


그림 2. 24시간 기준으로 본 특정 검색 키워드의 가격 분포 (기준: 주중 데이터)
Fig. 2. Time series result of specific keyword depends on the price.

로 구성한다. 독립 변수로 구성되는 x의 데이터 셋 (예, 입력)은 키워드 명 (word2vec의 100차원의 vector 값), 가격, 시간 (48의 단위 값, 주중 혹은 주말로 표현된 단일 값)의 값이 들어가게 되고, 종속 변수로 구성되는 y의 데이터 셋 (예, 출력)은 데이터의 순위 값이 들어가게 된다. 크롤링을 통해 수집된 내용 중 키워드 명, 키워드 카테고리, 평균 노출 수, 평균 클릭 수, 평균 입찰가격, 광고 지면 평균 입찰 가격, 평균 입찰가 차액, 현재 광고 순위, 목표 광고 순위, 품질 지수, 노출 중인 광고 수, 광고 노출 위치, 요일, 시간 총 14개의 feature들에 대한 feature engineering을 수행하기 위해 (Waikato Environment for Knowledge Analysis) WEKA를 통해 검증하였다. WEKA는 의사결정트리를 위한 분류 분석을 수행하는 대표적인 데이터 마이닝 프로그램으로써 크롤링을 통해 수집된 다양한 내용 중 위에 언급한 14개의 feature의 적합성을 class 결과로써 확인한다. 최종적으로 입찰 가격, 시간, 요일, 현재 광고 순위, 목표 광고 순위 총 5개의 feature를 PCA (Principal Component Analysis)의 결과로써 추출하였다.

3.2 순위 예측 알고리즘

본 논문에서는 기본적으로 머신러닝 모델링을 수행하기 위하여 Optimizer로 Adam, Adagrad, Momentum, RMSProp, GradientDescent 최적화 알고리즘을 사용하여 결과를 분석하였고 Learning rate는 0.0001 ~ 1까지 적용한 후 가장 최상의 결과를 내는 모델을 사용하였다. 기본적으로 머신러닝 모델링을 수행하기 위하여 hypothesis, loss function (혹은 cost function), optimizer를 적용한다. 기본적으로 머신러닝에서 최적화 알고리즘은 데이터를 학습할 때 실제 결과와 예측 결과 값의 차이를 최소화하는 방법이다. 일반적으로 gradient descent optimizer가 기반인 다양한 최적화 알고리즘들이 존재한다. 그 중에서 본 논문에서는 Adam 알고리즘을 최적화 기법에 사용하고 그 결과를 분석한다. 또한 learning rate은 loss function이 minimum이 되는 최적의 해를 찾아가는 과정을 반복하는 방식이다. Learning rate이 너무 크게 되면, 최적의 값으로 수렴하지 않고 발산하는 overshooting이 발생하게 되고, learning rate이 너무 작게 되면 수렴하는 속도가 너무 느리고 local minimum에 빠질 확률이 증가하게 된다. 따라서 본 논문에서는 적합한 learning rate을 찾기 위한 과정을 제시하고 그 결과를 분석한다.

먼저 Linear regression 알고리즘을 이용하기 위하여 hypothesis와 loss function을 적용한다. 그리고 linear regression의 함수를 Sigmoid 함수로 적용하여 hypothesis와 loss function을 구성한 Logistic regression 알고리즘을 적용한다. Logistic regression은 원칙적으로 하나의 클래스를 구분한 것에 최적화되어있다. 이를 multinomial classification 형태로 구현하여 softmax regression 알고리즘을 적용할 수 있다. Softmax regression의 hypothesis와 loss function은 기본적으로 logistic regression의 함수와 동일하다. 단, loss function을 위하여 cross entropy를 사용한다. 이 3가지 regression 알고리즘과 달리 ANN 알고리즘은 딥 러닝 모델을 기반으로 한다. 하나의 layer로 구성된 perceptron에서 layer를 여러 개 만든 multilayer perceptron으로 적용된다. 기본적으로 ANN은 perceptron에 트레이닝 데이터 셋을 input으로 넣어 전방향 연산(forward propagation)을 수행한다. 이때 결과로 나온 Neural networks의 prediction의 값과 target value의 값 차이로 에러 값으로 계산하게 되고 이를 back propagation (chain rule) 알고리즘의 적용이라 할 수 있다. MLP에서는 hidden layer를 증가시킴으로써 정확도가 향상될 것을 기대하였으나, back propagation에서 이전 기울기 값을 잃어버리는 vanishing gradient problem이 발생할 수 있다. 이를 위해 RELU 함수를 적용하며 vanishing gradient problem을 해결한다.

Ⅳ. 실험 환경 및 결과

4.1 실험 환경 및 구성

키워드 auction 데이터 상에서 검색 키워드의 가격 예측을 하기 위한 모델의 성능 평가를 위해 사용한 실험 환경은 표 1과 같다. 키워드 입찰 데이터 셋은 약 9만4천 개의 데이터로 구성되고 이 중 training data를 위하여 70%를 사용하고 test data를 위하여

www.dbpia.co.kr

표 1. 시뮬레이션 환경
Table 1. Simulation environment

Category	Value
OS	Windows 10 Pro
CPU	Intel Core i5-8400 2.80GHz 64bits
GPU	GeForce GTX1070 8GB
Memory	16GB
Language	Python 3.7
Library	Tensor Flow 1.9

30%를 사용한다. 그림 3은 training data를 위하여 input으로 적용된 데이터의 sample을 보여준다.

그림 4는 본 논문에서 만든 모델의 전체적인 블록도 이며 본 논문에서는 키워드 명에 대하여 Word2Vector를 적용하였고, 100개의 클래스와 1개의 입찰 가격을 MinMaxScaler에 적용하였으며, 48개의 시간 클래스와 주중과 주말을 나타내는 2개 클래스에 대하여 One-Hot Encoder 사용함으로써 총 151개 클래스 형태의 Input(독립 변수) 데이터를 구성하였으며 검색 엔진에 게시되는 광고 순위를 Output(종속 변수)으로 구성하였다. ANN 알고리즘을 적용하기 위해 input 데이터의 클래스 값을 151로 설정한다. 이 데이터가 input layer를 나와 첫 번째 hidden layer로 들어갈 때 256가지의 클래스로 나오게 되고, 이는 다시 두 번째 hidden layer로 들어가게 된다. 마찬가지로 세 번째 hidden layer로 256개의 클래스로 들어가도록 설정한 후 세 번째 hidden layer를 걸쳐 나온 output layer에서 15개의 클래스 값으로 나오도록 neural networks를 구성한다.

기본적으로 cost function을 정의하기 위하여 다음과 같이 정보이론의 cross entropy cost function을 사용한다:

$$J=-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}y_j^{(i)}\log\left(\widehat{y_j^{(i)}}\right)+\left(1-y_j^{(i)}\right)\log\left(1-\widehat{y_j^{(i)}}\right)$$

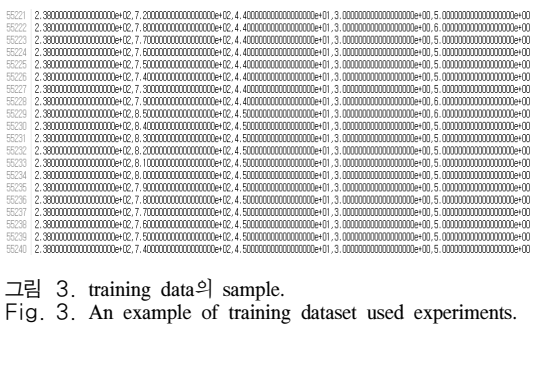


그림 3. training data의 sample.
Fig. 3. An example of training dataset used experiments.

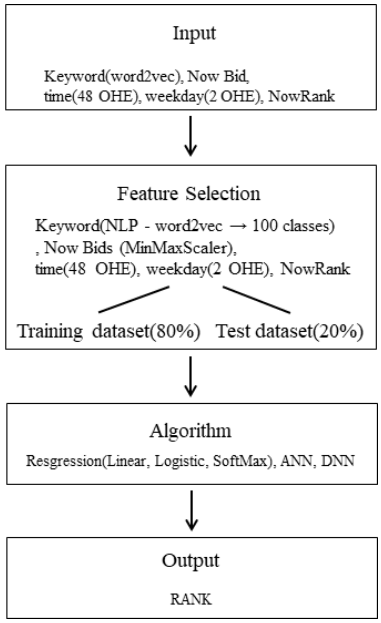


그림 4. 전체적인 모델링의 블록도
Fig. 4. Block diagram of overall modeling

여기서 $y_j^{(i)}$는 output node j를 위한 ith training label을 나타내고, $\widehat{y_j^{(i)}}$는 output node j를 위한 ith predicted label을 나타낸다. 또한 n은 batch sample을 training 수로 나눈 값을 나타낸다. 정확도 계산을 위하여 스텝 당 100개의 단위로 샘플링하여 1000번을 진행한 경우 stochastic training을 통해 계산된 레이블 중 가장 높은 점수를 사용하여 tf.argmax 함수로 계산된다. 그리고 tf.equal을 통해 예측 값과 정답이 같은 경우 true를 반환하고 아니면 false 값을 반환하는 과정을 통해 정확도를 계산하게 된다. 정확도 계산을 위한 식은 다음과 같이 사용된다:

$$Acc=\frac{(TP+TN)}{N},$$

여기서 N은 수행 횟수, TP는 True positive, TN은 True Negative를 나타낸다. 본 논문에서는 정확도의 결과를 통해 본 알고리즘을 예측 결과를 판단하였으나, 정확도 외에 precision과 recall을 통해 결과를 추출하는 F1 score나 검색 시스템의 평가를 위해 사용되는 nDCG와 같이 ranking 시스템에서 사용되는 또 다른 계산을 통해 추가적인 고찰을 진행할 수 있을 것으로 본다.

www.dbpia.co.kr

4.2 실험 결과

그림 5는 ANN에 Adam 최적화 알고리즘을 적용한 training data의 정확도를 나타낸 결과이다. Adam 최적화 알고리즘을 적용했을 때 값의 변동이 적도 epoch이 증가함에 따라 정확도가 단계적으로 증가하는 것을 볼 수 있다.

그러나 adam 최적화 알고리즘에 적절한 learning rate이 적용되어야 최적의 결과를 찾을 수 있음을 알 수 있다. 그림 5의 training data의 정확도를 위해 learning rate이 0.001 값일 때 가장 정확도가 높음을 알 수 있다. 특히 learning rate이 0.01 값인 경우 빠르게 정확도에 도달하지만 learning rate가 0.001인 경우 좀 더 높은 정확도의 결과가 도출되었다. 그림 6는 ANN에 Adam 최적화 알고리즘을 적용한 경우 cost function의 값을 도출한 결과이다. Epoch이 증가함에 따라 점점 cost 이 minimum 값에 도달함을 알 수 있다. Training accuracy의 learning rate이 0.001인 경우 가장 높은 정확도를 보인 것과 같이 cost 결과도 learning rate이 0.001 값인 경우 가장 minimum에 도달하는 것을 알 수 있다. 그러나 그림 6의 결과 그래프에 따라 learning rate이 0.01인 경우 더욱 빠르게 minimum에 수렴하는 것을 알 수 있다.

그림 7은 모델의 정확도를 판단하기 위하여 test 데이터 셋의 정확도를 확인한 결과이다. 그림 5와 동일하게 adam 최적화 알고리즘을 적용하였고, 결과에 따라 learning rate이 0.01일 경우와 0.001일 경우에 가장 높은 정확도를 보였다. 특히 learning rate이 0.01일 때 정확도 값에 빠르게 도달하는 것을 알 수 있다. 그림 8은 모델의 정확도에 따른 cost function을 확인하기 위한 결과이다. 그림 6에서 확인한 test data set의

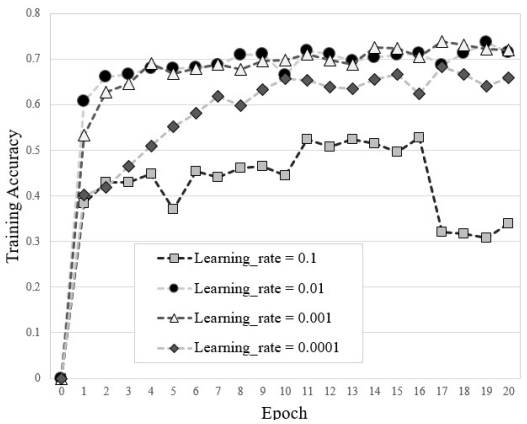


그림 5. ANN 알고리즘을 적용한 training accuracy의 결과.
Fig. 5. Simulation result of training accuracy in ANN.

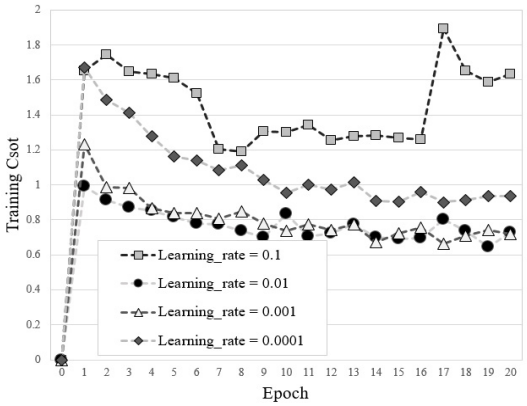


그림 6. ANN 알고리즘을 적용한 training cost의 결과.
Fig. 6. Simulation result of training cost in ANN.

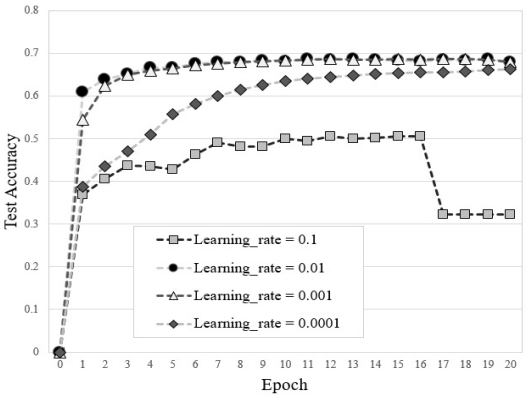


그림 7. ANN 알고리즘을 적용한 test accuracy의 결과.
Fig. 7. Simulation result of test accuracy in ANN.

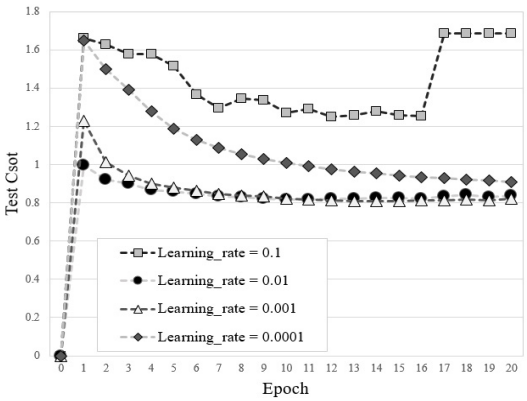


그림 8. ANN 알고리즘을 적용한 test cost의 결과.
Fig. 8. Simulation result of test cost in ANN.

정확도와 동일하게 learning rate이 0.01 값인 경우 가장 빠르게 minimum 값에 수렴하는 것을 알 수 있다.

표 2를 통해 좀 더 구체적인 결과 값을 확인할 수

www.dbpia.co.kr

표 2. Adam optimizer를 적용한 ANN 알고리즘의 시뮬레이션 결과
Table 2. Simulation results of the ANN algorithm with the Adam optimizer

learning rate	Epoch	training		test	
		accuracy	cost	accuracy	cost
0.1	5	0.461	1.523	0.463	1.365
	10	0.473	1.343	0.495	1.292
	15	0.520	1.259	0.505	1.255
	20	0.328	1.781	0.322	1.688
0.01	5	0.682	0.779	0.675	0.846
	10	0.719	0.706	0.686	0.819
	15	0.713	0.696	0.682	0.822
	20	0.738	0.678	0.685	0.833
0.001	5	0.679	0.837	0.671	0.863
	10	0.710	0.772	0.685	0.818
	15	0.705	0.756	0.685	0.812
	20	0.729	0.697	0.678	0.827
0.0001	5	0.581	1.140	0.581	1.130
	10	0.654	1.000	0.640	0.991
	15	0.623	0.961	0.655	0.936
	20	0.665	0.925	0.663	0.905

있다. 앞서 말한 바와 같이, learning rate에 따라 정확도와 cost 값이 변하는 것을 알 수 있다. 또한 learning rate이 0.01 값일 경우 training data의 정확도와 test data의 정확도가 가장 높은 것을 확인할 수 있다. 특히 test cost 값의 경우 learning rate 이 0.01이면서 epochs이 10일 경우 가장 minimum 값에 도달한다. 그리고 epochs이 증가함에 따라 다시 cost 값이 증가함을 확인할 수 있다.

본 논문에서는 linear regression, logistic regression, multinomial classification (Softmax regression)을 적용한 결과를 추가적으로 기술한다. 3가지 모델에 대한 결과 비교를 위하여 본 논문에서는

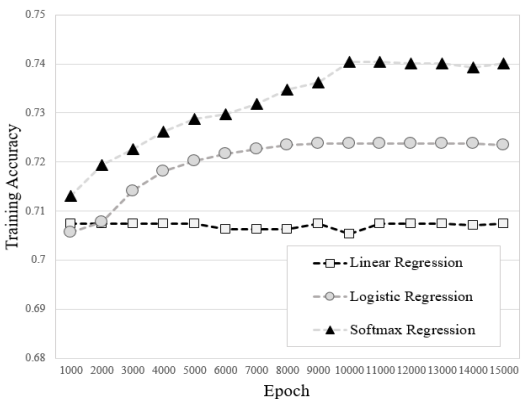


그림 9. Regression 알고리즘을 적용한 test accuracy의 결과.
Fig. 9. Simulation result of test accuracy in regression.

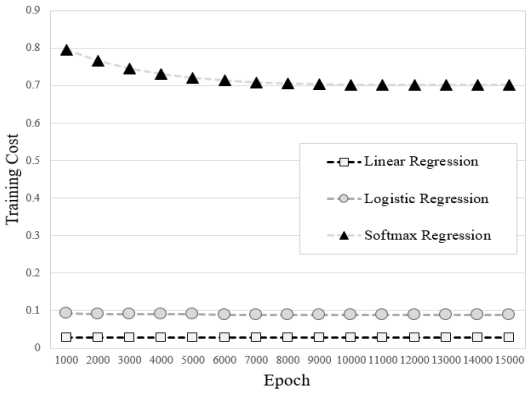


그림 10. Regression 알고리즘을 적용한 test cost의 결과.
Fig. 10. Simulation result of test cost in regression.

Adam optimization algorithm을 적용하고 learning rate은 0.01로 동일하게 적용하였다. 그림 9는 linear regression, logistic regression, Softmax regression에 따른 training data의 정확도를 비교한 결과이다. 그림 8의 결과에 따르면 training data의 정확도는 multinomial classification 알고리즘이 가장 높은 정확도를 보임을 알 수 있다. 그림 10은 3가지 모델에 대한 test data의 정확도를 비교한 결과이다. 그림 9의 결과에 따르면 test data의 정확도 또한 multinomial classification 알고리즘이 가장 높은 정확도를 보임을 알 수 있다.

그림 11과 그림 12는 3가지 모델에 대한 training data의 cost 값과 test data의 cost 값을 각각 보여준다. 결과적으로 각 3가지 모델 모두 minimum 값을 찾은 후 step이 증가함에 따라 cost가 다시 증가함을 알 수 있다. 그림 11과 그림 12에서 볼 수 있는 cost 값은 각 모델에 따라 독립적인 결과로 비교할 수 있다. 예를 들어, multinomial classification의 test data에 대한 cost값의 경우 8000 정도의 step에서 minimum에 도달하였고 이 후 다시 cost가 증가하는 것을 알 수 있다. 또한 logistic regression의 test data에 대한 cost 값의 경우 670 정도의 step에서 minimum에 도달하는 것을 확인 할 수 있다.

본 논문에서는 4가지 알고리즘을 적용하여 키워드 데이터의 가격 예측 결과를 비교하였다. 대표적으로 ANN 알고리즘을 적용한 데이터 셋의 크기는 총 94,258개이고 training data (65,980개)와 test data (28,278개)는 70%와 30%로 비중을 두었다. 또한 나머지 regression algorithms을 위하여 대표적으로 linear regression, logistic regression, multinomial classification을 적용한 가격 예측 결과를 비교하였다.

www.dbpia.co.kr

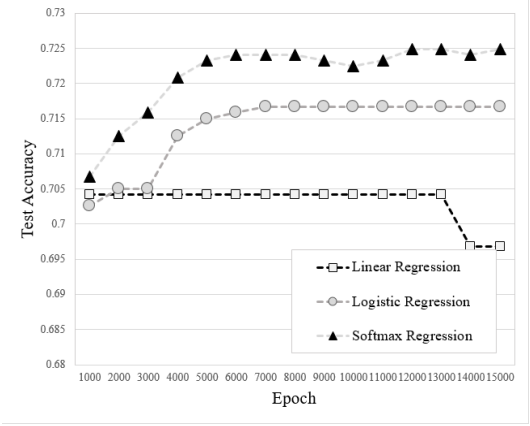


그림 11. Regression 알고리즘을 적용한 training accuracy의 결과.
Fig. 11. Simulation result of training accuracy in regression.

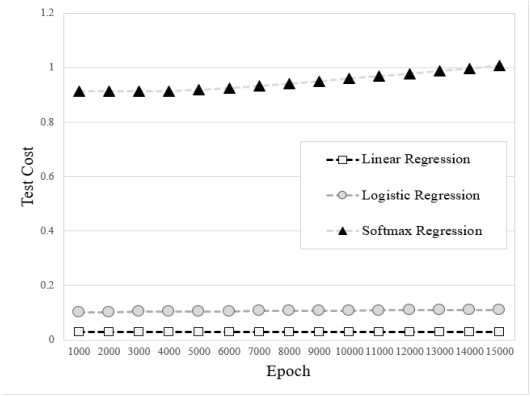


그림 12. Regression 알고리즘을 적용한 training cost의 결과.
Fig. 12. Simulation result of training cost in regression.

Regression 알고리즘을 적용한 데이터 셋의 크기는 총 4,023개이고 training data (2,816개)와 test data (1,207개)는 70%와 30%로 비중을 두었다. 결과적으로 ANN 알고리즘의 경우 training data의 정확도는 77%이고, test data의 정확도는 68%를 보였다. 그리고 regression 알고리즘의 경우 각각 알고리즘에 따라 다르지만 평균적으로 training data의 정확도는 74%이고, test data의 정확도는 71%를 보였다. 전반적으로 training data의 정확도가 70%대의 결과를 보였기 때문에test data의 정확도도 70%대의 정확도를 보인 것으로 예측된다. Training data의 정확도가 70%대의 결과를 보인 것으로 보아 데이터 셋의 부족으로 인해 과소적합 문제가 발생된 것으로 분석된다. 본 결과는 keyword data의 rank를 정확히 예측하는 결과에 따른 값이다. 만일 keyword data의 rank를 1 정도의 Gap을

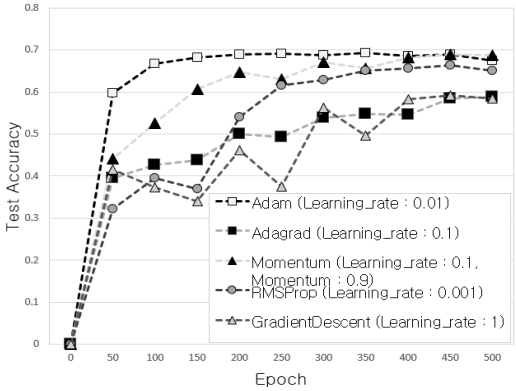


그림 13. Optimizer별 ANN의 Test Accuracy.
Fig. 13. Test Accuracy of ANN based on various optimizer.

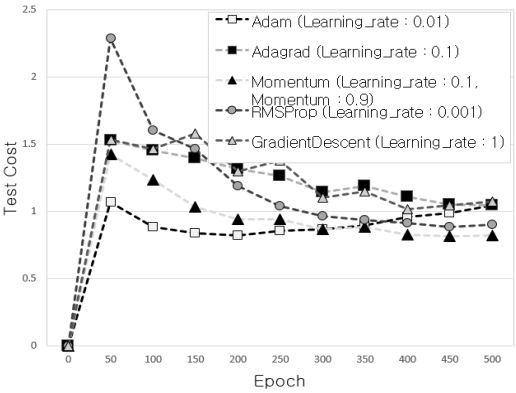


그림 14. Optimizer별 ANN의 Test Cost.
Fig. 14. Test Cost of ANN based on various optimizer.

두고 예측하는 결과의 정확도를 도출하면 ANN test data의 정확도는 91.8%, linear regression algorithms의 test data의 정확도는 92%, logistic regression algorithm과 softmax regression algorithm의 정확도는 94%의 결과를 보였다. 결과적으로 rank의 예측도에 대한 Gap을 두게 되는 경우 정확도가 향상되는 것을 알 수 있다. 본 논문의 결과를 바탕으로 추가적인 데이터 수집으로 과소적합 문제를 해결한 결과에 대한 연구로 확장할 예정이다.

본 논문에서는 ANN에 사용되는 Optimizer를 5가지로 적용하였다. 대표적으로 Adam, Adagrad, Momentum, RMSProp, GradientDescent의 optimizer를 적용하여 검색 키워드 가격 예측의 정확도를 비교하였다. 가장 좋은 성능을 보인 최적화 알고리즘은 Adam 최적화 알고리즘으로써 Training 정확도는 79.80%, Test 정확도는 69.78%를 보였다. 그에 반해

www.dbpia.co.kr

표 3. Optimizer 별 ANN의 결과에 Gap ±1 방식을 적용한 정확도
Table 3. Comparison of Accuracy based on Gap

Optimizer	Adam	Adagrad	Momentum	RMSProp	GradientDescent
Learning rate	0.01	0.1	0.1	0.001	1
Gap 0	69.78%	62.39%	69.13%	63.70%	58.50%
Gap ±1	92.36%	87.23%	91.58%	88.81%	84.90%

가장 안 좋은 성능을 보인 최적화 알고리즘은 GradientDescent 최적화 알고리즘으로써 Training 정확도는 61.87%, Test 정확도는 61.03%를 보였다. training 정확도를 분석해 본 결과 데이트 셋의 부족으로 인한 under fitting 문제가 발생했음을 알 수 있다. 따라서 본 논문에서는 해당 결과에 Gap을 두고 키워드 데이터의 순위를 추정할 수 있도록 하였다. 여기서 Gap은 하나의 순위에 대한 ±1의 불일치를 허용한다는 것이다. 이는 예를 들어 3번째 순위가 2번 째 또는 4번 째 순위로 예측되는 경우에도 정확도가 포함됨을 의미한다. 이는 실제 검색 키워드 광고에서 순위를 정도의 단위로 보기 때문에 가능 방식이다. 즉, 가격에 대한 순위는 실시간으로 변화하고 가격에 대한 예측은 10원 단위로 변화하기 때문에 그 가격과 순위에 대한 변동폭이 크기 때문에 ±1 정도의 순위 변화가 허용되는 방식이다. 표 3은 순위를 Gap ±1을 두고 계산한 정확도로써 Gap을 두고 계산할 때 정확도가 상당 부분 개선됨을 알 수 있다.

V. 결 론

본 논문에서는 온라인 검색 키워드 데이터에 대한 가격 예측 알고리즘을 적용한 예측 정확도를 위하여 다양한 머신러닝 알고리즘을 적용한 결과를 제시하였다. 또한 optimizer로써 5가지 최적화 알고리즘 (Adam, Adagrad, Momentum, PMSProp, GradientDescent)과 다양한 learning rate을 적용함으로써 검색 키워드 데이터의 입찰 가격 예측 결과를 비교한다. optimizer는 Adam optimizer를 적용했을 때 정확도가 가장 높은 결과를 보였고, 다양한 learning rate에 의해 정확도가 변동하는 결과를 제시하였다. 결과적으로 Adam 최적화 알고리즘에서 Training 정확도는 %, Test 정확도는 %로 나타났다. 결과를 분석한 결과 under fitting 문제로 인하여 낮은 정확도를 보이는 것을 알 수 있었고 이를 위하여 검색 광고의 실시간성을 반영한 Gap의 개념을 적용함으로써 결과의 정확성을 높이는 방법을 제안하였다. 향후 추가적으로 데이터를 수집하여 under fitting 문제를 해결하고 보다 다양한 머신러닝 알고리즘을 적용함으로써 가격 예측에 대한 정확성을 높이는 연구를 추가적으로 진행할 예정이다. 또한 본 논문에서는 정확도의 결과를 통해 본 알고리즘을 예측 결과를 판단하였으나, 정확도 외에 F1 score나 nDCG와 같이 ranking 시스템에서 사용되는 계산을 통해 추가적인 고찰을 진행할 수 있을 것으로 본다.

References

[1] K. Ren, W. Zhang, K. Chang, Y. Rong, Y. Yu, and J. Wang, “Bidding machine: Learning to bid for directly optimizing profits in display advertising,” *IEEE Trans. Knowledge and Data Eng.*, vol. 30, no. 4, pp. 645-659, 2018.

[2] D. Agarwal, S. Ghosh, K. Wei, and S. You, “Budget pacing for targeted online advertisements at Linkedin,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pp. 1613-1619, 2014.

[3] Google, *The arrival of real-time bidding and what it means for media buyers*, Google White Paper, 2012.

[4] H. Cai, et al., “Real-time bidding by reinforcement learning in display advertising,” in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, pp. 661-670, 2017.

[5] O. Chapelle, “Offline evaluation of response prediction in online advertising auctions,” in *Proc. 24th Int. Conf. World Wide Web Companion*, pp. 919-922, 2015.

[6] X. Li and D. Guan, “Programmatic buying bidding strategies within rate and winning price estimation in real time mobile advertising,” in *Proc. Pacific-Asia Conf. Advances Knowl. Discovery Data Mining*, pp.

www.dbpia.co.kr

447-460, 2014.

[7] S. More and S. A. Kulkarni, "Data mining with machine learning applied for email deception," *Int. Conf. Optical Imaging Sensor and Security*, pp. 1-4, 2013.

[8] B. Edelman, "Accountable? The problems and solutions of online ad optimization," *IEEE Security&Privacy*, vol. 12, no. 6, pp. 102-107, 2014.

[9] C. Oh, "A study of method for the measurement of product in placement effects recognition, recall, intention of selection based on MCQ scale," *The Korean J. Advertising and Public Relations*, vol. 16, no. 2, pp. 261-302, 2014.

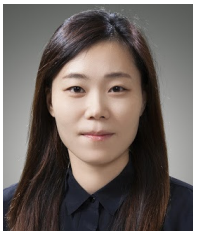
[10] W. Hwang, Y. Chen, and T. Jiang, "Personalized internet advertisement recommendation service based on keyword similarity," in *Proc. IEEE Computer Software and Appl. Conf.*, pp. 29-33, 2015.

지 민 준 (Minjun Ji)

2017년 2월 : 한국성서대학교 컴퓨터소프트웨어학과 졸업

2018년 2월~현재 : 이글루 커뮤니케이션즈

박 현 희 (Hyunhee Park)



2011년 8월 : 고려대학교 전자컴퓨터공학과 박사 졸업

2017년 3월~현재 : 한국성서대학교 컴퓨터소프트웨어학과 조교수

<관심분야> 전자공학, 통신공학, 머신러닝, 빅데이터

www.dbpia.co.kr