(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0137637
(43) 공개일자 2023년10월05일

(51) 국제특허분류(Int. Cl.)
G06Q 30/02 (2023.01) *G06N 3/08* (2023.01)
(52) CPC특허분류
G06Q 30/0247 (2013.01)
G06N 3/08 (2023.01)
(21) 출원번호 10-2022-0035327
(22) 출원일자 2022년03월22일
심사청구일자 2022년03월22일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
임희석
서울특별시 성북구 안암로 145 자연계캠퍼스애기능생활관 311호 NLP&AI연구실
문현석
서울특별시 강남구 자곡로 260, 422동 903호
(뒷면에 계속)
(74) 대리인
김등용

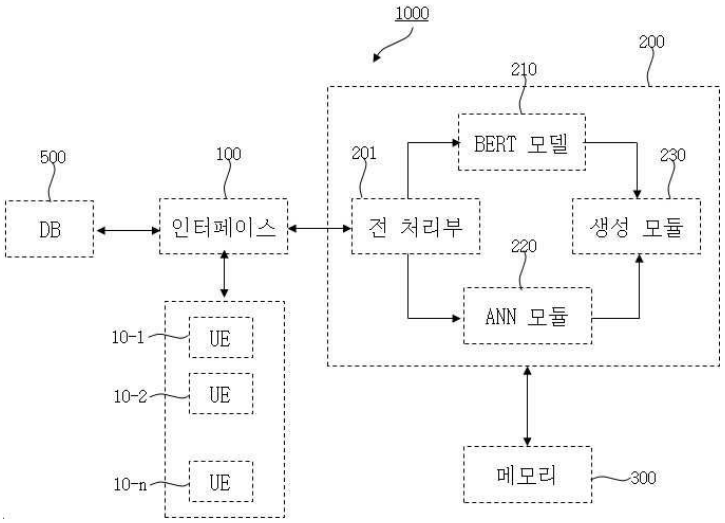
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법 및 이를 수행하는 장치**

(57) 요 약

본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법을 수행하는 장치는 사용자 단말로부터 광고 투입과 연관된 사용자의 행동 데이터를 수신하도록 구성된 인터페이스; 및 상기 인터페이스와 동작 가능하게 결합되고, 상기 행동 데이터 및 예측 모델을 이용하여 광고 수익률(Return on Advertising Spend, ROAS)를 예측하도록 구성된 프로세서를 포함하고, 상기 프로세서는 입력 특징을 상기 예측 모델에 입력하여 날짜 별로 광고 투입에 따른 광고 수익 발생 여부와 향후 특정 날짜에 대한 ROAS를 예측할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
G06Q 30/0246 (2013.01)
(72) 발명자
박기남
서울특별시 강서구 마곡중앙로 33, 1406동 101호

이태민
경기도 광명시 가림로 38, 504동 1404호

이 발명을 지원한 국가연구개발사업

과제고유번호	1711134382
과제번호	2021-0-01081-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	ICTR&D혁신바우처지원(R&D)
연구과제명	AI 기반 실시간 온라인 마케팅 성과 예측 시스템
기 여 율	1/1
과제수행기관명	(주)비즈스프링
연구기간	2021.04.01 ~ 2022.03.31

명 세 서

청구범위

청구항 1

광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법을 수행하는 장치에 있어서,

사용자 단말로부터 광고 투입과 연관된 사용자의 행동 데이터를 수신하도록 구성된 인터페이스; 및

상기 인터페이스와 동작 가능하게 결합되고, 상기 행동 데이터 및 예측 모델을 이용하여 광고 수익률(Return on Advertising Spend, ROAS)를 예측하도록 구성된 프로세서를 포함하고,

상기 프로세서는,

상기 행동 데이터 및 쿼리를 광고 그룹별 및 날짜 별로 클러스터링된 입력 특징(input feature)을 획득하고,

상기 입력 특징을 상기 예측 모델에 입력하여 날짜 별로 광고 투입에 따른 광고 수익 발생 여부와 향후 특정 날짜에 대한 ROAS를 예측하는, 장치.

청구항 2

제1 항에 있어서,

상기 행동 데이터 및 쿼리와 연관된 데이터는 상기 입력 특징을 구성하고,

상기 입력 특징은,

상기 데이터의 데이터 수집 일자, 상기 쿼리가 속한 광고 그룹, 광고 플랫폼, 광고 프로그램 및 상기 쿼리가 발생한 기기에 대한 정보를 포함하는 스트링 타입의 제1 타입 입력 특징; 및

사용자의 광고 체류 시간, 회원 가입 횟수, 주문 횟수, 상기 사용자에 대한 광고를 통해 발생한 비용, 물품 구매에 따른 수익 및 상기 사용자가 광고에 접근하는데 사용된 키워드를 포함하는 정수 타입의 제2 타입 입력 특징을 포함하는, 장치.

청구항 3

제2 항에 있어서,

상기 프로세서는,

상기 데이터의 타입이 상기 스트링 타입의 제1 타입 입력 특징이면, 클러스터 내에서 가장 높은 빈도를 가지는 클래스를 그룹을 대표하는 특징으로 선택하고,

상기 데이터의 타입이 상기 정수 타입의 제2 타입 입력 특징이면, 연속 값으로 이루어진 상기 입력 특징은 쿼리 별 발생값을 더하여 그룹별 특징을 생성하고,

상기 제1 타입 입력 특징을 갖는 데이터가 키워드 특징이면, 쿼리 별 키워드들을 중복 없이 하나의 스트링으로 연접하여 상기 그룹에 대한 키워드로 설정하는, 장치.

청구항 4

제1 항에 있어서,

상기 프로세서는,

상기 행동 데이터 및 쿼리와 연관된 키워드 정보와 수치 특징을 각각 사전 학습 언어 모델인 BERT 모델과 인공신경망(artificial neural network, ANN)에 입력하는, 장치.

청구항 5

제4 항에 있어서,

상기 프로세서는,

상기 BERT 모델의 제1 출력과 상기 ANN의 제2 출력을 결합하여 상기 날짜 별로 상기 입력 특징을 추출하고,

상기 추출된 입력 특징을 LSTM 모델 구조의 상기 예측 모델에 입력하여, 상기 향후 특정 날짜에 대한 ROAS를 예측하는, 장치.

청구항 6

제1 항에 있어서,

상기 프로세서는,

훈련 데이터 내의 입력데이터 x와 ROAS 값 y를 이용하여 상기 예측 모델을 학습하고,

상기 학습된 예측 모델에 기초하여 상기 향후 특정 날짜에 대하여 ROAS를 예측하고,

상기 향후 특정 날짜에 대하여 상기 예측된 ROAS와 실제 ROAS 간의 오차가 최소화되도록 상기 예측 모델을 다시 학습하는, 장치.

청구항 7

제1 항에 있어서,

상기 프로세서는,

상기 날짜 별로 광고 투입에 따른 수익이 발생하는지 예측하는 발생 예측 모델(Occurrence Prediction Model)에 따른 제1 학습 단계를 수행하고,

상기 수익이 발생한 경우 광고 수익률인 ROAS를 정량적으로 예측하는 발생 ROAS 회귀 모델(Occurred ROAS Regression Model)에 따른 제2 학습 단계를 수행하는, 장치.

청구항 8

제7 항에 있어서,

상기 프로세서는,

상기 제1 학습 단계 및 상기 제2 학습 단계를 수행한 이후, 추론 단계(inference phase)를 수행하고,

상기 추론 단계에서 상기 발생 예측 모델의 제1 출력 및 상기 발생 ROAS 회귀 모델의 제2 출력을 결합하여, 상기 향후 특정 날짜에 대한 ROAS와 연관된 최종 출력을 산출하는, 장치.

청구항 9

제7 항에 있어서,

상기 프로세서는,

상기 광고 그룹별 및 날짜 별로 ROAS가 nonzero인지 판단하는 바이너리 분류 모델(binary classification model)을 생성하고,

상기 바이너리 분류 모델을 반영하여 상기 발생 예측 모델에 따른 상기 제1 학습 단계를 수행하고,

상기 입력 특징에 기초하여 상기 ROAS에 따라 광고 투입에 따른 향후 수익이 증가할지 여부를 예측하는, 장치.

청구항 10

제9 항에 있어서,

상기 프로세서는,

상기 발생 ROAS 회귀 모델에 따른 상기 제2 학습 단계와 상기 제1 학습 단계 및 상기 제2 학습 단계에 따른 추론 단계를 수행하고,

상기 추론 단계에서, nonzero로 판별된 해당 ROAS값이 정량적으로 어떤 값을 가지는지 예측하는, 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 예측에 기초한 태스크 분리 방법에 관한 것이다. 보다 상세하게는 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법 및 이를 수행하는 장치에 관한 것이다.

배 경 기 술

[0002] Return on Advertising Spend (ROAS) 란, 광고를 위해 투입한 금액 대비, 발생한 수익의 비율을 의미한다. 이는 특정 광고에 대한 수익성을 정량화한 것으로, advertising project의 성과를 보여주는 지표로서 현재 매우 활발하게 활용되고 있으며, 해당 지표를 산출하기 위한 연구 역시 구글을 비롯한 여러 업계에서 활발하게 진행되고 있다.

[0003] 현재 ROAS를 계산하기 위한 연구는, geo-experiment와 같이, 광고를 노출시키는 그룹(treatment group)과 광고를 노출시키지 않는 그룹(control group)에서 발생하는 수익과 비용의 차이를 분석하는 방법을 통해 이루어지고 있다. 이는 advertising exposure를 제외한 다른 변인이 통제된 대조 실험을 통하여 데이터를 수집하고, 광고를 통해 발생하는 실제 수익성에 대해서 예측하는 연구를 의미한다. 하지만 이러한 연구들은, 광고를 통해 기존에 발생한 성과들에 대해 분석할 수 있으나, 해당 광고를 통해 향후 발생하게 될 광고 효과에 대해서는 예측할 수 없다는 한계가 있다.

[0004] 이는 실제 기업이 수집된 정보를 기반으로 industrial 관점에서 유용한 정보를 추출해 내려 할 때, 활용에 있어 제한점이 있다. 특히, 향후 발생할 수익이나 비용에 대해 ROAS를 예측하는 연구는 대부분 casual한 수준의 통계 기반 모델을 활용하는 접근방법에 의존하고 있다. 이와 관련하여, 딥러닝 프레임워크(deep learning framework)를 적용한 ROAS 예측은 전혀 이루어지지 않고 있다.

[0005] 또한, 기존 연구들은 수익이나 비용을 예측하는 데 활용될 수 있는 부가적인 요인들을 배제한 채, 오직 발생한 수익과 발생한 비용만을 ROAS 분석 연구에 활용했다. 이는 사용자의 광고 클릭 수, 광고 체류시간 등, ROAS 예측 및 분석에 활용될 여지가 있는 유용한 정보들이 활용되지 않고 있음을 의미한다. 추가적으로, 이렇게 분석 및 예측 모델 학습을 위한 데이터를 습득하기 위해서는 대부분 controlled group과 treatment group으로 구분된 대조실험환경에서 수집된 데이터가 요구된다. 이는 기존 연구들을 활용하여 광고 프로젝트에 대한 효과를 확인하기 위해서는, controlled group과 treatment group으로 광고 대상군을 재설정한 이후, 광고 이외의 요인들을 통제한 채, 실험 목적에 맞는 데이터를 다시 수집해야 함을 의미한다. 이는 확인하고자 하는 광고마다 데이터를 새롭게 수집해야 함을 보여주며, 이에 따라 데이터 수집 비용이 높아짐을 암시한다.

[0006] 이에 따라, 본 특허에서는 이러한 기존 연구들의 한계점을 극복할 수 있는,딥러닝 기반의 ROAS 예측 시스템을 제안한다. 해당 모델은 훈련 데이터 수집을 위해 추가적인 대조 실험을 진행할 필요 없이, 광고를 진행하며 자동으로 수집되는 사용자 데이터를 활용하여 향후 ROAS를 예측할 수 있다는 장점이 있다. 추가적으로, 광고 프로젝트를 통해 수집되는 데이터베이스 내에는 수익이 발생한 데이터 query보다, 수익이 발생하지 않은 query가 월등히 더 많다는 것을 고려하여, 태스크 분리(task decomposition)에 기반한 딥러닝 framework를 제안한다. 이는 수익 예측 모델을 end-to-end manner으로 생성하는 것이 아닌, 해당 task를 수익 발생을 예측하는 task와, 발생한 수익을 정량적으로 예측하는 task로 분리하여 두 개의 task를 수행하는 모델을 설계하는 방법을 의미한다. 본 명세서에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법을 통하여 end-to-end causal model에서보다 더 월등히 뛰어난 성능의 모델을 생성할 수 있다.

발명의 내용

해결하려는 과제

[0007] 본 발명의 목적은 전술한 문제를 해결하기 위해, 딥러닝 기반의 Return on Advertising Spend (ROAS) 예측 시스템을 제안하기 위한 것이다.

[0008] 또한, 본 발명의 목적은 태스크 분리(task decomposition)에 기반한 딥러닝 프레임워크(framework)를 제안하기 위한 것이다.

[0009] 또한, 본 발명의 목적은 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법 및 이를 수행하는 장치를

제공하기 위한 것이다.

과제의 해결 수단

[0010] 상기와 같은 과제를 해결하기 위한 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법을 수행하는 장치는 사용자 단말로부터 광고 투입과 연관된 사용자의 행동 데이터를 수신하도록 구성된 인터페이스; 및 상기 인터페이스와 동작 가능하게 결합되고, 상기 행동 데이터 및 예측 모델을 이용하여 광고 수익률(Return on Advertising Spend, ROAS)를 예측하도록 구성된 프로세서를 포함하고, 상기 프로세서는 입력 특징을 상기 예측 모델에 입력하여 날짜 별로 광고 투입에 따른 광고 수익 발생 여부와 향후 특정 날짜에 대한 ROAS를 예측할 수 있다.

[0011] 일 실시 예에 따르면, 상기 프로세서는 상기 행동 데이터 및 쿼리를 광고 그룹별 및 날짜 별로 클러스터링된 입력 특징(input feature)을 획득하고, 상기 입력 특징을 상기 예측 모델에 입력하여 날짜 별로 광고 투입에 따른 광고 수익 발생 여부와 향후 특정 날짜에 대한 ROAS를 예측할 수 있다.

[0012] 일 실시 예에 따르면, 상기 행동 데이터 및 쿼리와 연관된 데이터는 상기 입력 특징을 구성한다. 상기 입력 특징은 상기 데이터의 데이터 수집 일자, 상기 쿼리가 속한 광고 그룹, 광고 플랫폼, 광고 프로그램 및 상기 쿼리가 발생한 기기에 대한 정보를 포함하는 스트링 타입의 제1 타입 입력 특징; 및 사용자의 광고 체류 시간, 회원 가입 횟수, 주문 횟수, 상기 사용자에 대한 광고를 통해 발생한 비용, 물품 구매에 따른 수익 및 상기 사용자가 광고에 접근하는데 사용된 키워드를 포함하는 정수 타입의 제2 타입 입력 특징을 포함할 수 있다.

[0013] 일 실시 예에 따르면, 상기 프로세서는 상기 데이터의 타입이 상기 스트링 타입의 제1 타입 입력 특징이면, 클러스터 내에서 가장 높은 빈도를 가지는 클래스를 그룹을 대표하는 특징으로 선택하고, 상기 데이터의 타입이 상기 정수 타입의 제2 타입 입력 특징이면, 연속 값으로 이루어진 상기 입력 특징은 쿼리 별 발생값을 더하여 그룹별 특징을 생성하고, 상기 제1 타입 입력 특징을 갖는 데이터가 키워드 특징이면, 쿼리 별 키워드들을 중복 없이 하나의 스트링으로 연접하여 상기 그룹에 대한 키워드로 설정할 수 있다.

[0014] 일 실시 예에 따르면, 상기 프로세서는 상기 행동 데이터 및 쿼리와 연관된 키워드 정보와 수치 특징을 각각 사전 학습 언어 모델인 BERT 모델과 인공 신경망(artificial neural network, ANN)에 입력되도록 제어할 수 있다.

[0015] 일 실시 예에 따르면, 상기 프로세서는 상기 BERT 모델의 제1 출력과 상기 ANN의 제2 출력을 결합하여 상기 날짜 별로 상기 입력 특징을 추출하고, 상기 추출된 입력 특징을 LSTM 모델 구조의 상기 예측 모델에 입력하여, 상기 향후 특정 날짜에 대한 ROAS를 예측할 수 있다.

[0016] 일 실시 예에 따르면, 상기 프로세서는 훈련 데이터 내의 입력데이터 x와 ROAS 값 y를 이용하여 상기 예측 모델을 학습하고, 상기 학습된 예측 모델에 기초하여 상기 향후 특정 날짜에 대하여 ROAS를 예측하고, 상기 향후 특정 날짜에 대하여 상기 예측된 ROAS와 실제 ROAS 간의 오차가 최소화되도록 상기 예측 모델을 다시 학습할 수 있다.

[0017] 일 실시 예에 따르면, 상기 프로세서는 상기 날짜 별로 광고 투입에 따른 수익이 발생하는지 예측하는 발생 예측 모델(Occurrence Prediction Model)에 따른 제1 학습 단계를 수행하고, 상기 수익이 발생한 경우 광고 수익률인 ROAS를 정량적으로 예측하는 발생 ROAS 회귀 모델(Occurred ROAS Regression Model)에 따른 제2 학습 단계를 수행할 수 있다.

[0018] 일 실시 예에 따르면, 상기 프로세서는 상기 제1 학습 단계 및 상기 제2 학습 단계를 수행한 이후, 추론 단계(inference phase)를 수행하고, 상기 추론 단계에서 상기 발생 예측 모델의 제1 출력 및 상기 발생 ROAS 회귀 모델의 제2 출력을 결합하여, 상기 향후 특정 날짜에 대한 ROAS와 연관된 최종 출력을 산출할 수 있다.

[0019] 일 실시 예에 따르면, 상기 프로세서는 상기 광고 그룹별 및 날짜 별로 ROAS가 nonzero인지 판단하는 바이너리 분류 모델(binary classification model)을 생성하고, 상기 바이너리 분류 모델을 반영하여 상기 발생 예측 모델에 따른 상기 제1 학습 단계를 수행하고, 상기 입력 특징에 기초하여 상기 ROAS에 따라 광고 투입에 따른 향후 수익이 증가할지 여부를 예측할 수 있다.

[0020] 일 실시 예에 따르면, 상기 프로세서는 상기 발생 ROAS 회귀 모델에 따른 상기 제2 학습 단계와 상기 제1 학습 단계 및 상기 제2 학습 단계에 따른 추론 단계를 수행하고, 상기 추론 단계에서, nonzero로 판별된 해당 ROAS값이 정량적으로 어떤 값을 가지는지 예측할 수 있다.

발명의 효과

[0021] 본 발명에 따르면, 딥러닝 기반의 Return on Advertising Spend (ROAS) 예측 시스템을 제공하여, 훈련 데이터 수집을 위해 추가적인 대조 실험을 진행할 필요 없이, 광고를 진행하며 자동으로 수집되는 사용자 데이터를 활용하여 향후 ROAS를 예측할 수 있다.

[0022] 본 발명에 따르면, 태스크 분리(task decomposition)에 기반한 딥러닝 프레임워크(framework)를 통해, 해당 task를 수익 발생을 예측하는 task와 발생한 수익을 정량적으로 예측하는 task로 분리하여 두 개의 task를 수행하는 모델을 설계할 수 있다.

[0023] 본 발명에 따르면, 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법을 통하여 end-to-end causal model에서보다 더 월등히 뛰어난 성능의 모델을 생성할 수 있다.

[0024] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.

도면의 간단한 설명

[0025] 도 1은 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법을 수행하는 장치의 구성도를 나타낸다.

도 2는 본 발명에 따른 광고 수익률(ROAS) 예측과 관련하여 범주형 데이터, 수치형 데이터 및 자연어 데이터로 클러스터링되는 구성을 나타낸다.

도 3은 본 발명에 따른 ROAS 예측 방법과 관련하여 BERT과 인공 신경망(artificial neural network, ANN)의 출력을 결합하여 날짜 별 특징을 추출하는 구성을 나타낸다.

도 4는 본 발명에 따른 날짜 별 특징 데이터에 기초하여 향후 발생할 ROAS를 예측하는 모델의 구성을 나타낸다.

도 5는 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법과 관련하여 여러 기법들에 따른 프레임워크의 성능 검증 결과를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0026] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.

[0027] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 상세한 설명에 구체적으로 설명하고자 한다. 그러나 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0028] 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용한다.

[0029] 제1, 제2등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다.

[0030] 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0031] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다.

[0032] 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않아야 한다.

[0033] 이하의 설명에서 사용되는 구성요소에 대한 접미사 모듈, 블록 및 부는 명세서 작성의 용이함만이 고려되어 부여되거나 혼용되는 것으로서, 그 자체로 서로 구별되는 의미 또는 역할을 갖는 것은 아니다.

[0034] 이하, 본 발명의 바람직한 실시 예를 첨부한 도면을 참조하여 당해 분야에 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 설명한다. 하기에서 본 발명의 실시 예를 설명함에 있어, 관련된 공지의 기능 또는 공지의 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다.

[0035] 이하에서는, 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법 및 이를 수행하는 장치에 대해 설명한다. 이와 관련하여, 도 1은 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법을 수행하는 장치의 구성도를 나타낸다. 도 1을 참조하면, 장치(1000)는 인터페이스(100), 프로세서(200) 및 메모리(300)를 포함하도록 구성될 수 있다.

[0036] 인터페이스(100)는 광고 투입과 연관된 사용자의 행동 데이터를 수신하도록 여러 사용자 단말들(10-1, 10-2, ..., 10-n)로부터 수신할 수 있다. 인터페이스(100)는 사용자 단말들(10-1, 10-2, ..., 10-n)로부터 수신된 행동 데이터를 데이터베이스(DB)(500) 또는 서버로부터 수신할 수도 있다. 인터페이스(100)는 유선/무선 통신을 통해 여러 사용자 단말들(10-1, 10-2, ..., 10-n) 및/또는 데이터베이스(DB)(500)와 연동되도록 구성될 수 있다.

[0037] 프로세서(200)는 인터페이스(100)와 동작 가능하게 결합될 수 있다. 프로세서(200)는 복수의 프로그램 모듈 및 구성이 실행되도록 구성될 수 있다. 딥러닝 기반의 Return on Advertising Spend (ROAS) 예측 시스템을 제공하기 위해 프로세서(200)는 복수의 프로그램 모듈 및 구성이 실행되도록 구성될 수 있다. 프로세서(200)는 사전 학습 언어 모델인 BERT 모델(210)과 인공 신경망(artificial neural network)인 ANN 모듈(220)이 실행되도록 이들을 포함하여 구성될 수 있다. 프로세서(200)는 예측 모델(230)을 더 포함하도록 구성될 수 있다. 프로세서(200)는 입력된 데이터들의 타입을 고려하여 데이터들을 분류하고 전처리하도록 구성된 전처리부(201)를 더 포함하도록 구성될 수 있다.

[0038] 프로세서(200)는 수집된 사용자의 행동 데이터에 기초하여 예측 모델을 생성하도록 구성될 수 있다. 프로세서(200)는 행동 데이터 및 예측 모델(230)을 이용하여 광고 수익률(Return on Advertising Spend, ROAS)를 예측하도록 구성될 수 있다.

[0039] 프로세서(200)는 사용자의 행동 데이터를 수집하고 전처리하여 예측 모델의 입력 특징(feature)을 이용하여 특징 추출, 데이터 클러스터링 및 입력 데이터 인코딩을 수행할 수 있다. 프로세서(200)는 예측 모델(230)의 구성을 이용하여 Return on Advertising Spend (ROAS)를 예측할 수 있다. 이를 위해, 프로세서(200)는 행동 데이터 및 쿼리를 광고 그룹별 및 날짜 별로 클러스터링된 입력 특징(input feature)을 획득할 수 있다. 또한, 프로세서(200)는 날짜 별로 상기 입력 특징에 따른 입력 데이터를 예측 모델에 입력하여 향후 특정 날짜에 대한 ROAS를 예측할 수 있다.

[0040] 이와 관련하여, 프로세서(200)는 입력 특징을 예측 모델(230)에 입력하여 날짜 별로 광고 투입에 따른 광고 수익 발생 여부와 향후 특정 날짜에 대한 ROAS를 예측할 수 있다. 메모리(300)는 행동 데이터, 예측 모델 및 예측된 ROAS 및 이와 연관된 정보들을 저장하도록 구성될 수 있다.

[0041] 프로세서(200)는 설계된 예측 모델을 엔드-투-엔드 프레임워크(end-to-end framework)를 적용하여 학습하여 1단계 프레임워크를 구성할 수 있다. 프로세서(200)는 예측 프로세스(process)를 두 가지로 분리하는 태스크 분리(task decomposition)을 기반으로 한 2단계 프레임워크를 구성할 수 있다.

[0042] 전술한 본 발명의 기술적 특징에 따른 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법에 대해 상세하게 설명한다. 전술한 바와 같이, 프로세서(200)는 사용자의 행동 데이터를 수집하고 전처리하여 예측 모델의 입력 특징(feature)을 이용하여 특징 추출, 데이터 클러스터링 및 입력 데이터 인코딩을 수행할 수 있다. 프로세서(200)는 예측 모델 구성을 이용하여 Return on Advertising Spend (ROAS)를 예측할 수 있다.

[0043] 이하, 본 발명에 따른 사용자의 행동 데이터를 이용한 딥러닝 기반의 광고 수익률(Return on Advertising Spend, ROAS)에 대해 설명한다. ROAS는 광고로부터의 이득(Gain from Ad)를 광고에 대한 지출(Spend on Ad)로 나눈 값으로 정의될 수 있다. 즉, ROAS는 지출(Spend)에 대한 응답(Response)으로 정의될 수 있다.

[0044] ROAS는 광고에 투입한 비용 대비 얻을 수 있는 광고 효과를 나타내고, 광고 프로젝트의 효과를 측정하는 정량적 지표로서 활용될 수 있다. 전통적인 통계기반 ROAS 예측은 세그먼트의 시장 크기, 광고 반응도, 마케팅 단가의

반응도 등 다양한 변인을 측정한다. 구체적으로, 해당 광고로 벌어들일 수익의 상한선을 예측하고, 단가 대비 수익을 모델링한다. 이와 관련하여, ROAS 계산에 사용되는 변인들을 측정하기 빅테크 커머스 기업의 경우 다양한 방식으로 시장을 수치화하려 하고 있다.

[0045] ROAS를 측정하기 위해서는 해당 광고로 벌어들일 수익의 상한선을 예측할 수 있어야 하며, 시장의 가치, 세그먼트의 가치, 상품의 가치, 상품의 반응도를 계량화 하여야 한다. 하지만, 이러한 정보를 수집하는 것은 커머스 전체에 대한 거시적인 관점이 필요한 영역 인력 및 자본의 과대한 투여 없이 달성하기 어렵다.

[0046] 따라서, 본 발명에서는 딥러닝 기반의 Return on Advertising Spend (ROAS) 예측 시스템을 제공하고자 한다. 이와 관련하여, 훈련 데이터 수집을 위해 추가적인 대조 실험을 진행할 필요 없이, 광고를 진행하며 자동으로 수집되는 사용자 데이터를 활용하여 향후 ROAS를 예측할 수 있다.

[0047] 딥러닝 기반의 ROAS 예측 방법과 관련하여, ROAS 예측 시스템에서 데이터 전처리(Data Pre-processing)에 대해 설명한다. 데이터 전처리와 관련하여 특징 추출(Feature Extraction)에 대해 먼저 설명한다. 이와 관련하여, 본 명세서에서 이용되는 입력 feature들과, 이에 대한 설명은 표 1과 같다.

[0048] [표 1]

Feature Name	Description	Data Type	Class
stat_date	데이터 수집 일자	string	cluster
adgroup	데이터 쿼리가 속해 있는 ad group	string	cluster
ad_platform	광고 플랫폼	string	categorical
ad_program	광고 프로그램명	string	categorical
device	데이터 쿼리가 발생한 device	string	categorical
impr	사용자의 광고 체류 시간	integer	numeric
click	사용자의 광고 체류 시간	integer	numeric
rgr	사용자의 회원가입 횟수	integer	numeric
odr	사용자의 주문 횟수	integer	numeric
cart	사용자의 회원가입 횟수	integer	numeric
conv	사용자의 회원가입 횟수	integer	numeric
cost	사용자를 통해 발생한 비용	integer	numeric
rvn	사용자를 통해 발생한 수익	integer	numeric
keyword	사용자가 광고에 접근하는 데 사용된 키워드	string	keyword

[0049]

[0050] 여기서 각 특징 별 클래스는, 이후 데이터 클러스터링(data clustering)에서 전처리 방법을 결정하는 하나의 기준으로 사용된다. 본 발명은 해당 입력 특징을 기반으로 향후에 발생하게 될 ROAS를 출력하려는 목적이 있다. 본 발명에서 ROAS는, 수익(revenue)과 비용(cost)간의 비율로써 측정될 수 있다. 이는 수집된 데이터 내에 광고를 통해 발생된 수익과 비용이 계상되어 있기 때문에, 이들의 비율이 곧 해당 광고의 ROAS를 의미한다. 만약 수익이 발생하였으나 비용이 0인 경우는, 광고를 통해 발생한 효과가 아닌 것으로 간주하고 ROAS 값을 0으로 설정할 수 있다.

[0051] 이하, 데이터 클러스터링(Data Clustering)에 대해 설명한다. 광고 프로젝트를 진행함에 따라서, 데이터베이스(DB)에는 특정 사용자로 인해 발생하는 데이터들이 쿼리로 구축되게 된다. 이는 특정 사용자로 인해 발생한 수익 및 비용, 특정 사용자의 광고 클릭수 등이 축적되는 것을 의미한다. 단, 이를 직접 훈련에 사용하는 경우, 사용자에 따른 편차가 매우 크게 발생할 우려가 존재한다. ROAS 예측의 목적 자체가, 광고 프로그램 각각에 대한 효과를 확인하는 것이기 때문이다. 따라서, 사용자별로 발생한 쿼리를 광고 그룹별로 클러스터링하는 과정이 요구된다. 또한 본 발명의 목적이 날짜별로 발생하는 데이터들을 통해, 향후에 발생하게 될 ROAS를 예측이다. 따라서, 이러한 데이터들을 날짜별로 클러스터링하여 입력 특징(input feature)을 생성하는 과정을 거친다. 이러한 과정을 통하여, 사용자별로 수집된 다량의 데이터들은, 광고그룹별 순차 데이터(sequential data)로 전처리 된다. 이렇게 클러스터링의 기준이 되는 특징들을, 클러스터 클래스라고 지칭될 수 있다.

[0052] 본 발명에서 데이터를 클러스터링하는 과정에서, 데이터 타입이 정수(integer)의 연속 값(continuous value)으로 이루어진 특징은 데이터 타입에 쿼리 별 발생 값을 더하여 그룹별 특징을 생성할 수 있다. 이러한 방식으로 전처리(pre-processing)되는 특징을 수치 클래스(numeric class)라고 지칭할 수 있다. 수치 클래스의 값들은 광고 그룹별 편차가 매우 크게 발생하기 때문에, 이들을 모두 log10값을 취하여 재 스케일링(rescaling)한다. 데이터 타입이 스트링으로, 값이 정해진 입력 특징들은 범주형 클래스(categorical class)로 정의하며, 이 특징들을 클러스터링하는 경우 해당 특징은 클러스터 내에서 가장 높은 빈도를 가지는 클래스 하나가 그룹을 대표하는 특징으로 선택되게 된다. 키워드 특징(keyword feature)은 쿼리 별로 사용자가 광고에 접근하는 데 사용된 키워

드를 의미하며, 클러스터링되는 경우 쿼리별 키워드들이 중복 없이 하나의 스트링으로 연접(concatenated)되어 그룹에 대한 키워드로 설정된다. 이러한 특징을 키워드 클래스로 지칭할 수 있다.

[0053] 도 1 및 표 1을 참조하면, 행동 데이터 및 쿼리와 연관된 데이터는 입력 특징을 구성할 수 있다. 입력 특징은 스트링 타입의 제1 타입 입력 특징 및 정수 타입의 제2 타입 입력 특징으로 분류될 수 있다. 제1 타입 입력 특징은 데이터의 데이터 수집 일자, 상기 쿼리가 속한 광고 그룹, 광고 플랫폼, 광고 프로그램 및 상기 쿼리가 발생한 기기에 대한 정보를 포함할 수 있다. 제2 타입 입력 특징은 사용자의 광고 체류 시간, 회원 가입 횟수, 주문 횟수, 상기 사용자에 대한 광고를 통해 발생한 비용, 물품 구매에 따른 수익 및 상기 사용자가 광고에 접근하는데 사용된 키워드를 포함할 수 있다.

[0054] 프로세서(200)는 데이터의 타입이 스트링 타입의 제1 타입 입력 특징이면, 클러스터 내에서 가장 높은 빈도를 가지는 클래스를 그룹을 대표하는 특징으로 선택할 수 있다. 프로세서(200)는 데이터의 타입이 정수 타입의 제2 타입 입력 특징이면, 연속 값으로 이루어진 상기 입력 특징은 쿼리 별 발생값을 더하여 그룹별 특징을 생성할 수 있다. 프로세서(200)는 제1 타입 입력 특징을 갖는 데이터가 키워드 특징이면, 쿼리 별 키워드들을 중복 없이 하나의 스트링으로 연접하여 그룹에 대한 키워드로 설정할 수 있다.

[0055] 이하, 입력 데이터 인코딩(Input data encoding)에 대해 설명한다. 예측 모델의 입력 특징을 생성하기 위하여, 그룹 특징(grouped feature)들을 다음과 같은 과정을 통해 인코딩된다. 이는 다층 퍼셉트론(multi-layer perceptron)을 활용하여 수치형 클래스(numeric class)에 속하는 특징들과 범주형 클래스(categorical class)에 속하는 특징들을 인코딩하는 과정과, 언어 모델을 활용하여 키워드 특징을 인코딩하는 과정을 의미한다.

[0056] 이와 관련하여, 도 2는 본 발명에 따른 광고 수익률(ROAS) 예측과 관련하여 범주형 데이터, 수치형 데이터 및 자연어 데이터로 클러스터링되는 구성을 나타낸다. 도 2를 참조하면, 사용자 쿼리가 속해 있는 광고 그룹과 수집된 날짜를 기준으로 데이터를 클러스터링할 수 있다. 광고 수익률(ROAS) 예측과 관련하여 범주형 데이터, 수치형 데이터 및 자연어 데이터로 클러스터링될 수 있다.

[0057] 예측 모델의 입력 특징을 생성하기 위하여, 그룹 특징(grouped feature)들을 도 2와 같이 구조화된 데이터 클러스터링 과정을 통해 인코딩된다. 전술한 바와 같이, 다층 퍼셉트론을 활용하여 정수 인덱싱을 갖는 범주형 데이터, log 정규화 형태의 수치형 데이터 및 언어모델로 인코딩된 자연어 데이터가 다른 방식으로 인코딩될 수 있다. 이와 관련하여, 수치형 클래스에 속하는 특징들과 범주형 클래스에 속하는 특징들을 인코딩하는 과정 및 언어 모델을 활용하여 키워드 특징을 인코딩하는 과정이 별도로 수행될 수 있다. 이러한 인코딩 과정들에 대해서는 도 3에서 상세하게 설명한다.

[0058] 이와 관련하여, 도 3은 본 발명에 따른 ROAS 예측 방법과 관련하여 BERT과 인공 신경망(artificial neural network, ANN)의 출력을 결합하여 날짜 별 특징을 추출하는 구성을 나타낸다. 한편, 도 4는 본 발명에 따른 날짜 별 특징 데이터에 기초하여 향후 발생할 ROAS를 예측하는 모델의 구성을 나타낸다.

[0059] 이러한 데이터들의 상세한 인코딩 과정을 도 3을 참조하여 설명하면 다음과 같다. 도 3을 참조하면, 행동 데이터 및 쿼리와 연관된 키워드 정보와 수치 특징이 각각 사전 학습 언어 모델인 BERT 모델(210)과 인공 신경망(artificial neural network, ANN, 220)에 입력될 수 있다. BERT 모델(210)의 제1 출력과 ANN(220)의 제2 출력을 결합하여 날짜 별로 입력 특징을 추출될 수 있다.

[0060] 이와 관련하여, 수치형 클래스 및 범주형 클래스는 각각 NUM 및 CAT으로 표시되고, 대응하는 특징들은 $x_{num} \in Z$의 |NUM| 및 $x_{cat} \in 0, 1^{|CAT|}$이고, 여기서 |NUM| 및 |CAT|은 각 클래스의 특징들의 수를 나타낸다. x_{cat}을 생성하는 데 있어, x_{cat}의 각 특징에 대해, 대응하는 특징의 모든 카테고리가 정수 값으로 매핑된다. 이후, 이러한 정수 인덱스들은 one-hot representations으로 인코딩되어 즉 $x_{cat} \in 0, 1^{|CAT|}$이 성립한다. 두 번째로, 사전 학습된 언어모델인 BERT(Bidirectional Encoder Representations from Transformers)(210)를 활용하여 키워드 특징을 인코딩한다. BERT(210)는 Transformer의 인코더 구조만을 활용한 언어 모델로 대용량 말뭉치를 활용한 자기 지도 학습(self supervised Learning)을 통해 일반적인 언어정보를 담고 있다. 본 발명에서는 이렇게 언어에 대한 이해를 담고 있는 언어 모델을 활용하여 그룹별 키워드 정보를 인코딩한다.

[0061] 특히, 키워드 클래스 및 대응하는 특징들을 각각 KEY 및 $x_{keyword} \in Z^{maxlen}$로 나타낼 수 있고, 여기서 연접된 키워드 스트링은 토큰화되어 정수 인덱스로 매핑되고, 이후 모델의 최대 시퀀스 길이 maxlen으로 한정된다. BERT를 통해 얻은 xkeyword의 hidden representation을 $b_{keyword} \in R^{maxlen \times dbert}$ 라고 할 수 있다. 여기서, dbert는 BERT

모델의 디멘젼(dimension)을 나타내고 해당 표현에서 입력 시퀀스의 가장 첫번째 토큰에 해당하는 $b^{0}_{keyword} \in R^{dbert}$을 키워드 표현(keyword representation)이라고 정의하며, 이를 $x_{keyword}$라고 나타낸다.

[0062] x_{num}, x_{cat} 및 x_{keword}가 주어진 경우, 각 클러스터된 그룹 및 각 날짜에 대한 출력 특징 h가 수학식 1과 같이 생성될 수 있다.

[0063] [수학식 1]

$$h_{num} = W_{num} \cdot \log(x_{num})$$
$$h_{cat} = W_{cat} \cdot x_{cat}$$
$$h_{keyword} = W_{keyword} \cdot x_{keyword}$$
$$h = W_h \cdot [h_{num}; h_{cat}; h_{keyword}]$$

[0064]

[0065] 여기서 $Wnum \in R^{hidden \times |NUM|}$, $Wcat \in R^{hidden \times |CAT|}$, $Wkeyword \in {}^{Rhidden \times dbert}$ 및 $Wnum \in R^{hidden \times 3*hidden}$ 은 학습 가능한 파라미터(trainable parameters)를 의미한다. NUM의 특징들은, 로그 스케일(log scaling)을 적용하여 특징 데이터 내의 편차를 줄이는 방법을 활용할 수 있다. hidden은 이후 활용할 예측 모델의 hidden size를 의미한다. 수학식 1과 같은 과정을 통해 광고 그룹별, 날짜별로 클러스터링된 입력 특징을 얻게 된다. 이렇게 생성된 데이터는, 20일치 데이터를 연접함으로써 예측 모델의 입력을 구성한다. 결론적으로, 20일치 입력 특징 h1, h2, ..., h20에 대하여, 예측 모델의 입력 특징 h는 [h1; h2; ...; h20] $\in R^{20*hidden}$으로 정의된다.

[0066] 이하, 예측 모델 구조(Prediction Model Structure)에 대해 설명한다. 본 발명에서는 Long Short-Term Memory (LSTM)을 예측 모델의 기본 모델 프레임워크(baseline model framework)로 활용할 수 있다. LSTM은 순차 데이터를 다루기 위한 순환 신경망 모델 구조의 유형(type of recurrent neural network model structure)으로, conventional Recurrent Neural Network (RNN)에서 발생하는 장기 의존성(long term dependency) 문제를 해소하는 모델 구조로 활용될 수 있다.

[0067] 본 발명에서는 해당 모델 구조를 활용하여 과거 20 일치 데이터를 통해 향후 발생할 ROAS를 예측하는 모델을 설계할 수 있다.

[0068] 도 1 및 도 3을 참조하면, 프로세서(200)는 행동 데이터 및 쿼리와 연관된 키워드 정보와 수치 특징이 각각 사전 학습 언어 모델인 BERT 모델(210)과 인공 신경망(artificial neural network, ANN, 220)에 입력되도록 제어할 수 있다. 이와 관련하여, 프로세서(200)는 BERT 모델(210)과 ANN(220)이 실행되도록 메모리(300) 및 데이터베이스(500) (또는 다른 사용자 단말들)과 연동될 수 있다. 프로세서(200)는 BERT 모델(210)의 제1 출력과 ANN의 제2 출력(220)을 결합하여 날짜 별로 입력 특징을 추출하도록 구성될 수 있다.

[0069] 도 1, 도 3 및 도 4를 참조하면, 프로세서(200)는 날짜 별로 추출된 입력 특징을 LSTM 모델 구조의 예측 모델(230)에 입력하여, 향후 특정 날짜에 대한 ROAS를 예측할 수 있다.

[0070] 따라서, 본 발명에서는 데이터베이스(DB)에 수집된 사용자 행동 데이터를 기반으로, 향후 발생하게 될 ROAS를 예측할 수 있다. 이를 위해, 과거 20일치 특징들을 파악하여 다음날에 해당하는 ROAS 예측을 수행할 수 있다. 데이터베이스(DB)에 수집된 사용자 행동 데이터를 모델이 이해하기 쉬운 형태로 전처리하여 날짜별 데이터를 생성할 수 있다. 이와 관련하여, 텍스트상의 특징과 함께, 범주형/수치형 특징(feature)을 함께 활용하여 예측을 위한 특징을 생성할 수 있다. 이를 위해 도 4와 같은 LSTM 모델 구조를 활용하여 예측 모델을 설계할 수 있다.

[0071] LSTM에서 전처리 과정을 통해 입력으로 받은 h를 통해, 최종 예측 결과물을 내는 과정은 다음과 같다. LSTM은 크게 forget gate, input gate, cell state, output gate로 구성되어있으며, 순차(sequential) 데이터의 t번째 시간 스텝(time step)에 대한 입력 특징 ht를 처리함에 있어 수학식 2와 같은 과정이 진행된다.

[0072] [수학식 2]

$$
\begin{aligned}
i_t &= \sigma(W_{xi}h_t + W_{hi}l_{t-1} + W_{ci}c_{t-1} + b_i)\\
f_t &= \sigma(W_{xf}h_t + W_{hf}l_{t-1} + W_{cf}c_{t-1} + b_f)\\
\tilde{c}_t &= tanh(W_{xc}h_t + W_{hc}l_{t-1} + b_c)\\
c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t\\
o_t &= \sigma(W_{xo}h_t + W_{ho}l_{t-1} + W_{co}c_{t-1} + b_o)\\
l_t &= o_t tanh(c_t)
\end{aligned}
$$

[0073]

[0074] 여기서 σ는 sigmoid function을 의미하며, it, ft, ct, ot각각은 forget gate, input gate, cell state, 그리고 output gate를 의미한다. LSTM 모델을 통해 인코딩된 히든 상태(hidden state)는 lt라고 표시된다. 여기서 cell state와 LSTM 출력의 초기값 c0와 l0는 제로 값(zero value)으로 정의된다. 최종적으로 20개 입력 층(input layer)을 모두 처리한 이후, 가장 마지막 입력 특징 h20에 대한 히든 상태를 활용하여 다음 일자의 ROAS를 예측하게 된다. 상세하게는, 모델을 통한 예측값은 수학식 3과 같은 과정을 통해 출력된다.

[0075] [수학식 3]

$$\hat{y} = W_2 \cdot ReLU(W_1 \cdot l_{20})$$

[0076]

[0077] 수학식 3에서, $W1 \in R^{dlstm \times dlstm}$ and $W2 \in R^{1 \times dlstm}$ 은 학습 가능한 파라미터들이고, 여기서 dlstm은 LSTM 예측 모델의 히든 크기이다. 이러한 모델의 구체적인 훈련 과정은 이하에서 상세하게 설명한다.

[0078] 이하, 1-단계 프레임워크(One-stage Framework)에 대해 상세하게 설명한다. 본 발명을 통해 설계된 예측 모델을 end-to-end framework를 적용하여 학습하는 것을 One-stage framework라고 정의한다. 이 경우, ROAS를 예측하는 모델은, 주어진 훈련 데이터 D 내의 입력데이터 x와 label ROAS값 y에 대하여, 수학식 4와 같은 학습 가능한 목표(training objective)를 갖는다. 본 발명에서는 이러한 방식으로 학습된 모델을 one-stage framework라고 정의한다.

[0079] [수학식 4]

$$\min_{\theta_{e2e}} \frac{1}{|D|} \sum_{(x,y)\in D} (\theta_{e2e}(x) - y)^2$$

[0080]

[0081] 이는 주어진 입력에 대해서, 모델은 향후 발생하게 될 ROAS를 곧바로 예측하고, 모델 훈련도, 예측한 ROAS와 실제 ROAS 간의 직접적인 mean squared loss를 최적화(optimize)하는 방향으로 이루어짐을 의미한다.

[0082] 이와 관련하여, 광고 프로젝트를 통해 수집된 데이터를 통해 예측 모델을 학습하는 경우 다음과 같은 이슈가 존재할 수 있다. 이는 일반적으로 광고 프로젝트를 통해 수집된 데이터에 수익이 발생한 데이터보다 수익이 발생하지 않은 데이터가 대부분을 차지한다는 것이다. 이는 광고를 통해 광고 사이트에 접근한 이용자들 중 대다수는 수익을 발생하지 않고, 수익 실현으로 이루어지는 소비자가 극히 일부에 해당하기 때문이다. 상세하게는, 본 발명에서 활용한 데이터에서 수익(revenue)이 0인 데이터는 일 예로 90%이상의 비율로 구성될 수 있다. 이는 훈련 데이터에서 대다수 데이터의 label이 0값을 가짐을 의미하며, 이는 곧 학습한 예측 모델이 모든 입력 데이터를 0으로 예측하게 됨을 암시한다.

[0083] 이와 관련하여, 도 1, 도 3 및 도 4를 참조하여 프로세서(200)의 동작에 대해 설명하면 다음과 같다. 프로세서(200)는 훈련 데이터 내의 입력데이터 x와 ROAS 값 y를 이용하여 예측 모델(230)을 학습할 수 있다. 프로세서(200)는 학습된 예측 모델(230)에 기초하여 상기 향후 특정 날짜에 대하여 ROAS를 예측할 수 있다. 프로세서(200)는 향후 특정 날짜에 대하여 예측된 ROAS와 실제 ROAS 간의 오차가 최소화되도록 예측 모델(230)을 다시 학습할 수 있다.

[0084] 이러한 이슈를 해결하기 위하여, 본 명세서에서는 예측 프로세스를 두 가지로 분리하는 태스크 분리를 기반으로 한 2-단계 프레임워크(Two-Stage Framework)를 제안한다. 이는 주어진 입력 데이터를 통해 향후 발생할 ROAS를 end-to-end로 예측하는 것이 아닌 다음과 같은 방식으로 수행된다. 수익이 발생하는지 예측하는 발생 예측 모델(Occurrence Prediction Model)과 발생한 ROAS를 정량적으로 예측하는 발생 ROAS 회귀 모델(Occurred ROAS

Regression Model) 두 가지를 설계하여 각각 학습하는 방법을 의미한다. 해당 방법을 적용하는 경우, 추론(inference) 단계에서는 두 모델의 출력값을 결합하여 최종 출력물을 결정하게 된다. 여기서, 수익 발생 예측 모델은 주어진 입력을 통해 바이너리 분류(binary classification)를 진행하게 되며, 정량적 예측 모델은 발생한 수익이 얼마나 될지 회귀(regression)하게 된다. 각 모델의 학습을 위한 데이터 전처리 및 학습 과정은 다음과 같다.

[0085] 이하, 발생 예측 모델(Occurrence Prediction Model)에 대해 설명한다. 먼저 ROAS가 nonzero인지 판단하는 binary classification 모델인 Θ_{cls}을 생성하기 위해서는, label ROAS의 전처리가 요구된다. 이는 주어진 데이터 0보다 큰 label ROAS 값을 1로 매핑시켜주는 과정을 의미한다. 이를 구현하기 위한 매핑 함수 맵은 수학식 5와 같이 표현할 수 있다.

[0086] [수학식 5]

[0087]
$$map(y) = \begin{cases} 1 & \text{if } y > 0 \\ 0 & \text{otherwise} \end{cases}$$

[0088] Θ_{cls}는 0 또는 1로 분류될 수 있는 확률을 지시하는 두 개의 출력을 반환한다. 상세하게는, 수학식 3과 다르게, Θ_{cls}의 모델 출력은 수학식 6에 의해 정의된다.

[0089] [수학식 6]

[0090]
$$\theta_{cls}(x) = \text{Softmax}(W_2 \cdot ReLU(W_1 \cdot l_{20})) = \begin{bmatrix} y_0^{cls} \\ y_1^{cls} \end{bmatrix}$$

[0091] 수학식 6에서, 모든 프로세스들은 $W2 \in R^{2\times hidden}$라는 것을 제외하고는 수학식 3과 유사하다. 이후, 분류 모델 Θ_{cls}는 크로스 엔트로피 손실을 통해 최적화된다. 이러한 모델에 대한 훈련 목표는 수학식 7에 의해 정의될 수 있다.

[0092] [수학식 7]

[0093]
$$\min_{\theta_{cls}} \frac{1}{|D|} \sum_{(x,y)\in D} \left(map(y) * \log(y_1^{cls}) + (1 - map(y)) * log(y_0^{cls}) \right)$$

[0094] 이는 Θcls가 map(y)를 생성하도록 훈련됨을 의미한다. 이러한 프로세스를 통해, Θ_{cls}는 입력 특징들이 주어진 경우 향후 수익이 증가할지 여부를 예측할 수 있다.

[0095] 이하, 발생 ROAS 회귀 모델(Occurred ROAS Regression Model)에 대해 설명한다. 두 번째로, ROAS값의 정량적인 예측을 위해 Θreg 모델을 학습한다. 이는 ROAS가 nonzero로 판별되었다고 가정했을 때, 해당 ROAS값은 정량적으로 어떤 값을 가지는지 예측하는 모델이다. 즉, 해당 모델의 훈련을 위해서는 오로지 non-zero label ROAS값을 가지는 훈련 데이터만이 이용된다. 구체적인 훈련과정은 다음과 같다. 가장 먼저, 기존에 수집된 원시 데이터셋(original dataset) D로부터, non-zero label만을 담고 있는 데이터셋 Dnz를 생성한다. 이는 수학식 8과 같이 정의할 수 있다.

[0096] [수학식 8]

[0097]
$$D_{nz} = \{(x,y) | (x,y) \in D \wedge y > 0\}$$

[0098] 이를 위해 가장 먼저, Θcls의 예측 결과물로부터 ind라는 값을 생성한다. 이는 예측되는 ROAS값이 0이 될지, 혹은 0보다 큰 값을 가질지 결정해주는 값으로, 수학식 9와 같은 과정을 통해 계산된다.

[0099] [수학식 9]

$$ind(x) = \begin{cases} 0 & \text{if } \hat{y}_0 > \hat{y}_1 \\ 1 & \text{if } \hat{y}_0 < \hat{y}_1 \end{cases}, \text{where } \begin{bmatrix} \hat{y}_0^{cls} \\ \hat{y}_1^{cls} \end{bmatrix} = \theta_{cls}(x)$$

[0100]

[0101] 각 x에 대한 ind(x)값은, 이후 Θreg의 결과물에 곱해짐으로써 최종 ROAS예측 값을 산출해내게 된다. 해당 값과 Θreg모델의 출력물이 합성된 최종적인 모델 예측 결과물은 수학식 10과 같이 산출된다.

[0102] [수학식 10]

$$\hat{y} = ind(x) * \theta_{reg}(x)$$

[0103]

[0104] 이러한 태스크 분리를 통해 얻을 수 있는 가장 주요한 효과 중 하나는, non-zero data의 업 스케일링(up-scaling)이 가능해진다는 점이다. 딥러닝에서 데이터 불균형을 해소하기 위한 하나의 해결방안으로 업 스케일링이 사용될 수 있다. end-to-end framework에서 non-zero data를 업 스케일링하는 경우, 회귀 모델에서 예측에 심한 bias를 부여하게 된다. 하지만 0과 1의 바이너리 분류를 진행하는 상황에서는 업 스케일링 방법이 데이터 분포(data distribution) 문제를 해결하고 zero data를 취급(dealing)하는 매우 효과적인 방법으로 작용할 수 있다. 상세하게는, 업 스케일링을 적용하는 경우 Θcls의 훈련 데이터는 수학식 11과 같이 정의된다.

[0105] [수학식 11]

$$D_{ups} = [(x,y)|(x,y) \in D \wedge y > 0] + [(x,y)|(x,y) \in D \wedge y > 1] * n$$

[0106]

[0107] 여기서 n는 스케일링 요소(scaling factor)로, nonzero 데이터를 업 스케일할 비율을 의미하며, 동일한 데이터가 중복을 허용하여 n번 반복되어 데이터셋을 구성함을 의미한다. 이렇게, upscaling이 적용된 데이터셋인 Dups을 통해 Θcls를 훈련시킴으로써, 훈련 데이터의 편향된 분포(biased distribution)를 효과적으로 해소할 수 있다. 즉, zero label이 대부분인 훈련 데이터로 학습됨에 따라 모든 예측 결과물이 0을 지시하는 것이 아닌, nonzero data도 잘 구분해낼 수 있게되는 것이, 업스케일링의 효과로 볼 수 있다.

[0108] 또한, 태스크 분리 회귀 모델의 성능을 향상시킬 수 있다는 점에서도 매우 효과적이다. Θreg모델은, 수익이 발생하여 ROAS가 nonzero라는 가정하에 모든 훈련 데이터의 label이 nonzero인 셋팅에서 훈련되었기 때문에, zero data로 인해 예측값이 0에 overly baiased되는 것이 아닌, 보다 정확한 ROAS값을 회귀할 수 있게 된다. 본 발명에서는 2단계 프레임워크에서 태스크 분리를 적용하는 경우, 단순 end-to-end framework를 적용한 모델에서보다 월등히 높은 성능을 보이는 모델을 생성할 수 있음을 제안한다.

[0109] 이와 관련하여, 도 1, 도 3 및 도 4를 참조하여 프로세서(200)의 동작에 대해 설명하면 다음과 같다. 프로세서(200)는 날짜 별로 광고 투입에 따른 수익이 발생하는지 예측하는 발생 예측 모델(Occurrence Prediction Model)에 따른 제1 학습 단계를 수행할 수 있다. 프로세서(200)는 수익이 발생한 경우 광고 수익률인 ROAS를 정량적으로 예측하는 발생 ROAS 회귀 모델(Occurred ROAS Regression Model)에 따른 제2 학습 단계를 수행할 수 있다.

[0110] 프로세서(200)는 제1 학습 단계 및 제2 학습 단계를 수행한 이후, 추론 단계(inference phase)를 수행할 수 있다. 프로세서(200)는 추론 단계에서 발생 예측 모델의 제1 출력 및 발생 ROAS 회귀 모델의 제2 출력을 결합하여, 향후 특정 날짜에 대한 ROAS와 연관된 최종 출력을 산출할 수 있다.

[0111] 프로세서(200)는 광고 그룹별 및 날짜 별로 ROAS가 nonzero인지 판단하는 바이너리 분류 모델(binary classification model)을 생성할 수 있다. 프로세서(200)는 바이너리 분류 모델을 반영하여 발생 예측 모델에 따른 제1 학습 단계를 수행할 수 있다. 프로세서(200)는 입력 특징에 기초하여 ROAS에 따라 광고 투입에 따른 향후 수익이 증가할지 여부를 예측할 수 있다.

[0112] 프로세서(200)는 발생 ROAS 회귀 모델에 따른 상기 제2 학습 단계와 상기 제1 학습 단계 및 상기 제2 학습 단계에 따른 추론 단계를 수행할 수 있다. 프로세서(200)는 추론 단계에서, ROAS가 nonzero로 판별된 해당 ROAS값이 정량적으로 어떤 값을 가지는지 예측하여, 최종적으로 향후 특정 날짜에 대한 ROAS와 연관된 최종 출력을 산출할 수 있다.

[0113] 이러한 본 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법과 기존 방법의 성능을 비

교하면 다음과 같다. 이와 관련하여, 도 5는 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법과 관련하여 여러 기법들에 따른 프레임워크의 성능 검증 결과를 나타낸다.

[0114] 도 5를 참조하면, one-stage model을 활용했을 때 모델은 향후 수익 발생 여부를 아예 예측할 수 없는 것을 확인할 수 있다. 이는 mapped F1-score가 거의 0에 수렴하는 점수를 보이는 것을 통해 확인할 수 있다. 이러한 실험 결과는, 이러한 ROAS 예측 모델을 설계함에 있어 end-to-end framework로 모델을 학습하는 방법이 유효하지 않은 방법임을 보여준다. 해당 모델에서 MSE와 MAE는 가장 낮은 모습을 보여주었으나, 이는 모든 모델의 출력물이 0과 유사한 값을 내는데, 대부분의 label ROAS값이 0이기 때문에 전체적인 MSE가 낮아진 것으로 유추된다.

[0115] 이러한 문제는 two-stage framework를 적용함으로써 어느 정도 해소할 수 있었다. 단순하게 수익 발생 예측 모델과 발생 수익 정량화 모듈을 분리하는 것만으로도 mapped f1-score를 0.27578점까지 끌어올릴 수 있었다. 이는 task를 분리하여 학습하는 해당 방법론이 각 task에 특화된 모델을 생성하는데 매우 효과적인 접근방법이며, 매우 유의미한 성능 향상을 이끌어낼 수 있음을 보여준다. 특히, non zero data의 up-scaling을 적용한 two-stage framework는 non-zero MSE, non-zero MAE, Mapped F1-score등에서 가장 뛰어난 성능을 보여줬다. 이는 up-scaling을 적함으로써 Θcls모델의 성능을 큰 폭으로 끌어올릴 수 있음을 보여주며, 수익 발생 여부에 대한 정확도가 높아짐에 따라 non-zero MSE와 non-zero MAE부분에서도 높은 성능 향상을 얻을 수 있는 것으로 이해할 수 있다.

[0116] 하지만, up-scaling기법을 단순하게 one-stage framework에 적용하는 경우에는 유의미한 성능 향상을 기대할 수 없는 것으로 확인되었다. up-scaling을 적용한 데이터로 학습한 one-stage framework모델의 경우, mapped f1-score는 0.06116점에 그치며 여전히 zero label data를 제대로 분별해낼 수 없는 모습을 보여줬다. 특히, mapped recall이 1.0점임을 확인할 수 있는데, 이는 모든 모델의 출력값이 positive nonzero값이 되도록 예측모델이 학습된 것임을 의미한다. 이러한 실험 결과는 본 연구에서 제안한 task decomposition 기반의 two-stage framework와 up-scaling이 ROAS 예측 모델을 생성하는데 매우 유의미한 접근방법임을 보여준다. 특히 기존 데이터만을 활용하여 향후 발생할 ROAS를 예측함에 있어, 준수한 성능의 예측 모델을 생성하는데 활용될 수 있음을 보여준다.

[0117] 이하에서는 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법 및 이를 수행하는 장치에 대해 도 1을 참조하여 상세하게 설명한다. 네트워크 시스템은 제1 호스트(100), 제2 호스트(200) 및 네트워크 장치(1000)를 포함하도록 구성될

[0118] 이상에서는 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법 및 이를 수행하는 장치에 대해 설명하였다. 본 발명에 따른 본 발명에 따른 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법 및 이를 수행하는 장치의 기술적 효과는 다음과 같이 요약될 수 있지만, 이에 한정되는 것은 아니다.

[0119] 본 발명에 따르면, 딥러닝 기반의 Return on Advertising Spend (ROAS) 예측 시스템을 제공하여, 훈련 데이터 수집을 위해 추가적인 대조 실험을 진행할 필요 없이, 광고를 진행하며 자동으로 수집되는 사용자 데이터를 활용하여 향후 ROAS를 예측할 수 있다.

[0120] 본 발명에 따르면, 태스크 분리(task decomposition)에 기반한 딥러닝 프레임워크(framework)를 통해, 해당 task를 수익 발생을 예측하는 task와 발생한 수익을 정량적으로 예측하는 task로 분리하여 두 개의 task를 수행하는 모델을 설계할 수 있다.

[0121] 본 발명에 따르면, 광고 투입에 관한 수익의 예측에 기초한 태스크 분리 방법을 통하여 end-to-end causal model에서보다 더 월등히 뛰어난 성능의 모델을 생성할 수 있다.

[0122] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.

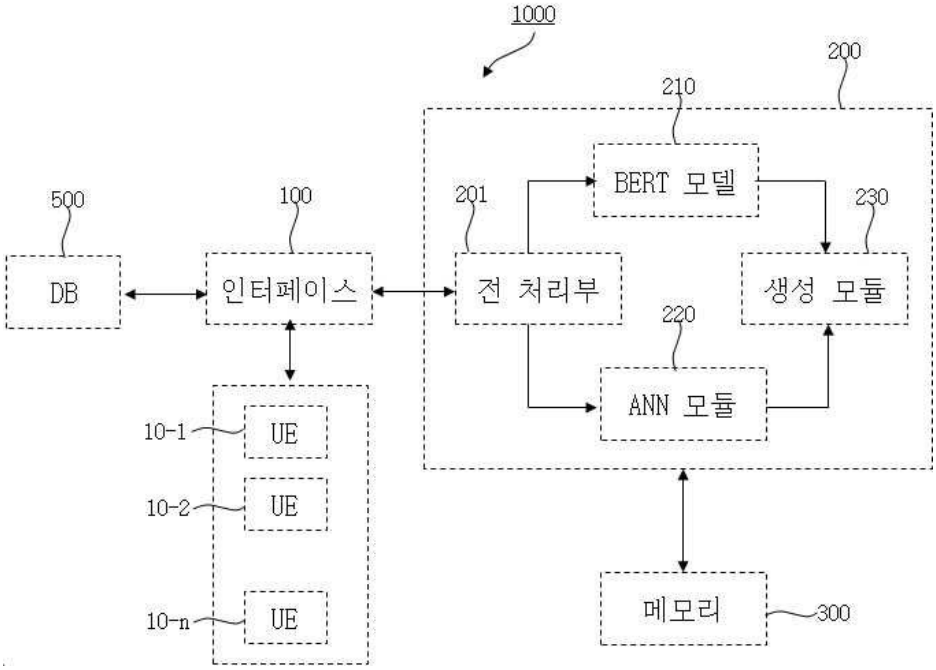
[0123] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 상세한 설명에 구체적으로 설명하고자 한다. 그러나 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0124] 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능뿐만 아니라 각각의 구성 요소들에 대한 설계 및 파라미터 최적화는 별도의 소프트웨어 모듈로도 구현될 수 있다. 적절한 프로그램 언어로 쓰여진 소프

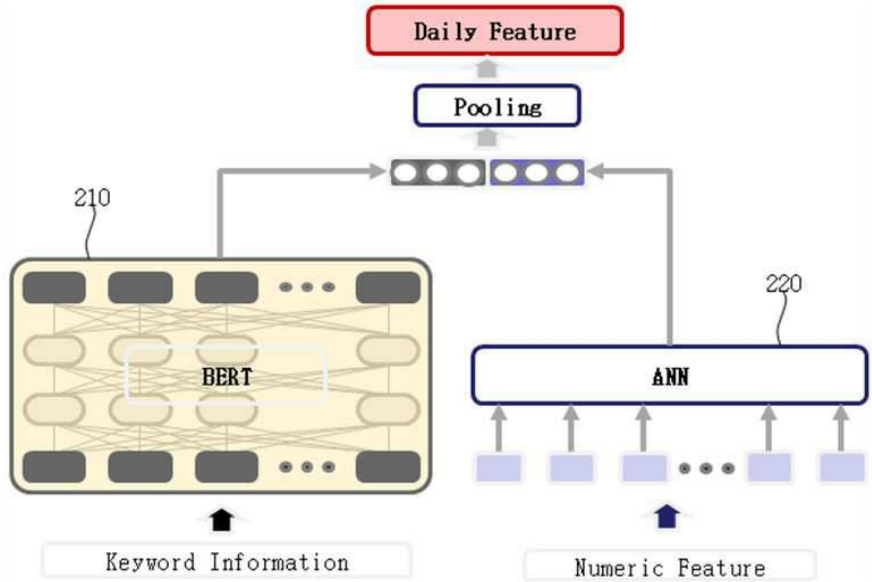
트웨어 어플리케이션으로 소프트웨어 코드가 구현될 수 있다. 상기 소프트웨어 코드는 메모리에 저장되고, 제어부(controller) 또는 프로세서(processor)에 의해 실행될 수 있다.

도면

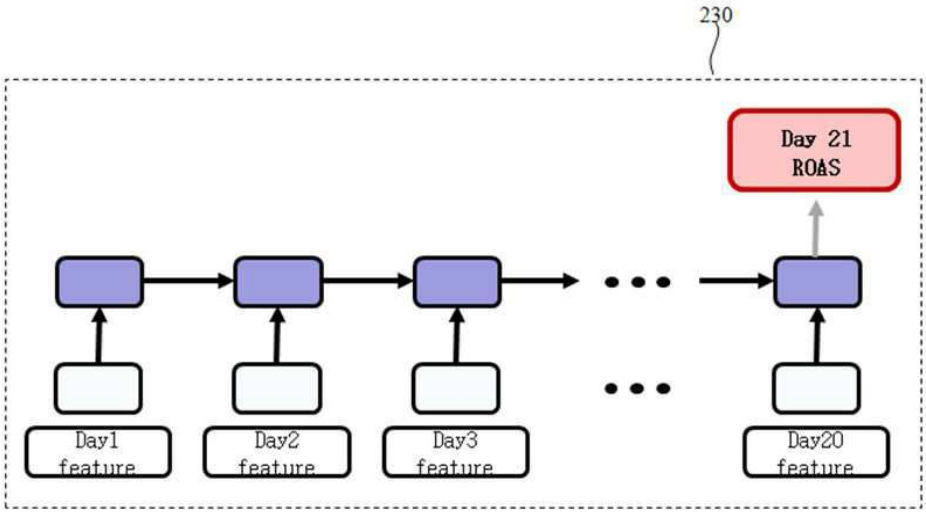
도면1



도면2



도면3



도면4



도면5

	MSE	MAE	Non-zero MSE	Non-zero MAE	Mapped F1-score	Mapped Precision	Mapped Recall
One-stage	**0.68400**	**0.11545**	21.64975	2.81465	0.00513	0.00271	0.04782
One-stage + Up-scaling	0.72601	0.39744	19.30089	2.32553	0.06116	0.03155	**1.00000**
Two-stage	0.68830	0.12501	19.66112	3.23538	0.27578	0.17062	0.71875
Two-stage + Up-scaling	0.81438	0.26652	**11.72011**	**2.02405**	**0.40642**	**0.25850**	0.95000