https://doi.org/10.7840/kics.2023.48.12.1559

콘텐츠 캐싱 서비스를 위한 딥러닝 기반 조회수 예측 기법

박 용 문◆, 김 영 진°

Deep Learning-Based View Count Prediction for Content Caching Services

Yongmoon Park◆, Yeongjin Kim°

요 약

미디어 데이터의 소비가 늘어나면서 저지연 서비스를 위한 콘텐츠 캐싱 기술이 중요해지고 있다. 콘텐츠 캐싱의 효과를 극대화하기 위해서는 향후 높은 조회수를 가질 것으로 예상되는 콘텐츠를 찾을 수 있어야 한다. 하지만, 콘텐츠별 조회수는 시간대에 따라 변화하며 유행의 변화, 콘텐츠 간 상관관계 등과 복잡하게 얽혀있기 때문에 정확한 예측이 쉽지 않다. 본 연구에서는 향후 콘텐츠들의 조회수를 높은 정확도로 예측하기 위한 딥러닝 기반의 학습 기법을 제안한다. 특히, 학습모델 전후에 데이터 매핑/역매핑 방법을 새롭게 제안하여, 학습모델이 콘텐츠 조회수의 크기를 제외한 트렌드 학습에 집중할 수 있게 한다. 또한, 1DCNN-LSTM-Dense 계층 기반의 딥러닝 모델을 제안하여 다중 속성 데이터 간의 상관관계, 시간 축에서의 데이터의 상관관계를 학습할 수 있도록 한다. YouTube 데이터셋에 기반하여 제안 학습 기법과 기존에 제안되었던 다양한 휴리스틱 알고리즘, 정규화 기법, 데이터 매핑 기법 등과 성능을 비교 평가한다. 평가 결과, 제안 학습 기법은 어떠한 미래 정보를 사전에 필요로 하지 않으면서도 가장 높은 캐싱 정답률을 달성하는 것을 확인한다.

키워드 : 콘텐츠 캐싱, 딥러닝, 장단기 메모리, 콘텐츠 인기도, 조회수 예측.
Key Words : Content caching, deep learning, LSTM, content popularity, view count prediction

ABSTRACT

As media data consumption increases, content caching becomes important for low-latency services. To maximize the effect of content caching, we need to find the contents expected to be viewed frequently in the near future. However, the number of expected views for each content is hard to predict because it changes over time, and is intertwined with view trends and correlations with other contents. In this study, we propose a deep learning technique to predict the number of views of contents with high accuracy. In particular, we propose a new data mapping/demapping method before and after the learning model, allowing the learning model to focus on learning trends of view counts excluding the magnitude of view counts. In addition, we propose a deep learning model based on the 1DCNN-LSTM-Dense layers to learn the correlation among multi-attribute data and

※ 본 연구는 2023년도 정보통신기획평가원 (No.RS-2022-00155915 인공지능융합혁신인재양성(인하대학교), 2021-0-02201 사용자 프라이버시를 보존하는 비디오 캐싱을 위한 연합 학습 시스템, 2022-0-00448 인간처럼 회상이 가능한 인공 신경망 지속학습 플랫폼 개발), 및 한국연구재단 (No. RS-2023-00240019)의 지원을 받아 수행된 연구임.
◆ First Author : Inha University Department of Electrical and Computing Engineering, sky123pq@inha.edu, 학생회원
° Corresponding Author : Inha University Department of Electrical and Computing Engineering, yj.kim@inha.ac.kr, 정회원
논문번호 : 202309-077-B-RU, Received September 11, 2023; Revised September 13, 2023; Accepted September 15, 2023

the correlation of data on the time axis. Based on the YouTube dataset, we evaluate the performance of the proposed learning technique and various previously proposed ones, including heuristic algorithms, normalization techniques, and other data mapping techniques. The results shows that the proposed learning technique achieves the highest caching performance without requiring any future information in advance.

Ⅰ. 서 론

스마트 기기의 대중화에 따라 SNS 및 Over The Top (OTT) 서비스에서 발생하는 미디어 데이터 트래픽이 급격히 증가하고 있다. 이러한 데이터 트래픽으로 인한 백홀 네트워크의 부담을 완화 시키고, 미디어 서비스를 낮은 지연 시간으로 제공하는 방법으로써 콘텐츠 캐싱 기술이 주목받고 있다. 콘텐츠 캐싱이란, 콘텐츠를 사용자와 가깝게 위치한 모바일 에지 컴퓨팅(MEC) 서버에 복제하여, 해당 콘텐츠 요청이 발생 시 기존 콘텐츠 서버가 아닌 MEC 서버에서 직접 서비스하는 기술이다. 그러나 MEC 서버의 캐싱 자원의 양은 제한적이고 고비용이기 때문에 극소수의 콘텐츠만 캐싱할 수 있다.

일반적으로, 사용자들로부터 많은 수요가 있을 것으로 예상되는 (인기도가 높은) 콘텐츠들을 위주로 MEC 서버에 캐싱한다. 이러한 방식을 따르는 이유는 많은 수의 콘텐츠 요청에 대하여 서비스 지연 시간을 줄일 수 있기 때문이다. 이를 위해서는 향후 어떠한 콘텐츠 요청이 많이 발생할지 정확하게 예측하는 것이 중요하다. 과거 연구들에서는 콘텐츠 캐싱 문제를 단순화하기 위하여 콘텐츠의 인기도가 변하지 않는다고 가정하거나[1-3], 변화하더라도 사전에 알 수 있다고 가정하였다[4-6]. 그러나 실제 환경에서의 콘텐츠의 인기도는 미리 주어지지 않으며 다양한 요소로 인하여 시간에 따라 변화한다[7].

최근, 향후의 콘텐츠 인기도를 예측하기 위해 딥러닝 기반의 학습모델 연구가 진행되고 있다. 예를 들어, 콘텐츠의 인기도를 예측하기 위해 합성곱 신경망(CNN)을 통하여 시간에 따른 과거의 콘텐츠 요청 패턴을 분석하거나[8], 소셜 네트워크와 전이학습(Transfer Learning)을 함께 이용하여 콘텐츠별 인기도 학습 시간을 단축한 사례가 있다[9].

콘텐츠의 요청은 시간에 따라 수집되는 시퀀스 데이터이기 때문에, 시퀀스 데이터를 학습하는데 효과적인 순환신경망(RNN)을 기반으로 하는 딥러닝 모델(예: Long Short Term Memory (LSTM), Gated Recurrent Unit(GRU))을 적용한 연구들이 제안되어왔다. 예를 들어, 학습을 위한 입력 시퀀스의 각 요소는 시간 순서에 따른 각 콘텐츠의 one-hot 인코딩된 인덱스가 될 수 있으며[10], 출력은 콘텐츠들의 향후 인기도 벡터나[11,12], 앞으로 요청될 것으로 예상되는 콘텐츠의 인덱스가 된다[13]. 이러한 입력 방식은 시간 축에서 요청이 인접한 콘텐츠 간의 상관관계를 포착하는데 효과적이며 유사한 콘텐츠 소비 취향을 가진 개별 사용자 또는 사용자 집단을 학습하는데 유리하다.

또 다른 학습 방법은 단위 시간 동안의 콘텐츠별 조회수를 이용하는 것이다. 이때 입력 시퀀스의 각 요소는 해당 시간대 동안 각 콘텐츠에 대한 조회수로 구성된 벡터로 정의된다[14-17]. 이 방법은 시간에 따른 콘텐츠별 조회수 변화 패턴을 효과적으로 포착하여 미래의 콘텐츠 인기도를 예측한다. 특히, LSTM 모델의 입력 차원을 줄이고 학습 시간을 단축하기 위하여 오토인코더에 기반한 입력 데이터 임베딩을 적용하거나[14], LSTM의 학습 성능을 높이기 위하여 입력값의 범위를 정규화하는 연구가 있었다[15]. 이와 더불어, BiLSTM 모델과 어텐션 계층을 사용하여 인접한 시간대 이외의 시간대의 콘텐츠 요청 간 상관관계를 포착하고자 하는 연구와[17] 콘텐츠 요청 패턴이 하루 동안에도 시간대별로 다르다는 점을 이용하여 시간대별로 학습모델을 별도로 구성하여 학습한 연구가 있었다[16]. 다른 분야에 적용된 사례로는 시공간에 따른 사용자들의 트래픽 요청량을 예측하기 위하여 CNN과 LSTM을 함께 사용하거나[18], 주식 시장을 예측하기 위하여 시간에 따른 주식별 시세 변화 데이터를 LSTM으로 학습한 경우가 있었다[19]. 그러나 앞서 언급한 학습모델 기반의 콘텐츠 인기도 예측 연구들은 특정 콘텐츠(예: 상위 인기 콘텐츠)에 치중되어 학습되거나, 미리 잘 학습된 임베딩 모델을 필요로 하거나, 정규화를 위한 콘텐츠별 사전 정보(예: 모든 시간에 대한 조회수의 평균/표준편차 또는 최댓값/최솟값)를 필요로 한다는 한계가 존재한다.

본 연구에서는 콘텐츠 조회수 히스토리를 이용하여 학습모델이 조회수의 변화 트렌드에만 집중할 수 있는 데이터 변환 모듈과 이를 기반으로 향후 콘텐츠 조회수를 예측할 수 있는 1-dimensional CNN (1DCNN)-LSTM-Dense 계층 기반의 학습모델을 제안한다. 제안한 데이터 변환 모듈은 콘텐츠별 조회수 크기와 상관없이 입력 데이터를 같은 영역으로 매핑하여 특정 콘텐츠에 치우치지 않고 학습할 수 있으면서도 콘텐츠에 대한

사전 정보가 필요하지 않다. 성능 평가를 위하여 YouTube 콘텐츠 요청 데이터셋을 사용하였으며 기존에 제안된 기법들과 비교하여 예측 성능이 상회함을 보인다.

Ⅱ. 시스템 모델

2.1 서비스 모델

그림 1은 본 연구에서 제안하는 콘텐츠 조회수 예측 시스템 구조이다. 대상 지역의 콘텐츠 요청을 관리하는 MEC 서버가 존재하며, 이 서버는 캐싱할 콘텐츠를 결정하기 위한 딥러닝 모델과 캐싱 공간을 보유하고 있다. 전체 콘텐츠의 집합은 C로 정의하며, 단위 시간대 t마다(예: 1시간 간격) 해당 시간대 동안의 콘텐츠 $c \in C$의 실제 조회수는 $a_c(t)$로 정의한다. 시간대 t의 모든 콘텐츠에 대한 조회수 벡터는 $\boldsymbol{a}(t) = (a_c(t), \forall c \in C)$로 정의한다. 본 시스템의 목표는 시간대 t가 시작되는 시점마다 과거 시간대의 콘텐츠 조회수 히스토리 $\boldsymbol{a}(:t-1) = (\boldsymbol{a}(\tau), \tau < t)$를 입력 데이터로 사용하여 시간대 t의 미래 콘텐츠 조회수 $\boldsymbol{a}(t)$를 예측하는 것이다. 시나리오에 따라, 입력으로 주어지는 데이터는 조회수 히스토리 외에도 다른 속성값(예: 댓글 수, 좋아요 수 등)이 있을 수 있는 상황을 가정한다.

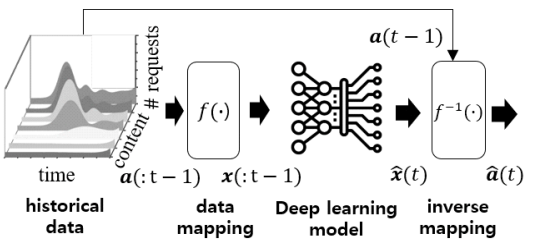

그림. 1. 과거 데이터를 이용한 미래의 콘텐츠 조회수 예측 시스템.
Fig. 1. System architecture of future content request prediction using historical data.

Ⅲ. 제안 기법

3.1 데이터 매핑 및 역매핑 모듈

콘텐츠 조회수 $a_c(t)$는 콘텐츠마다 크게 다르다. 예를 들어, 인기도가 낮은 콘텐츠는 단위 시간대 동안의 조회수가 100회 미만일 수 있지만, 인기도가 높은 콘텐츠는 100만 회 이상의 조회수를 기록 할 수 있다. 이해를 돕기 위하여, 시간대에 따른 두 콘텐츠 c_1과 c_2의 조회수가 각각 $a_{c_1}(t) = 10000\sin(t)$와 $a_{c_2}(t) = 10\sin(t)$의 형태를 보이고, MEC 서버는 각 콘텐츠가 이러한 함수의 조회수를 가진다는 것을 모르는 상태에서 조회수 히스토리를 기반으로 향후 조회수를 예측한다고 가정하자. 두 콘텐츠의 조회수는 서로 다른 *크기*(예: 10000, 10)를 가지지만 시간에 대하여 동일한 *트렌드*(예: $\sin(t)$)를 가지고 있는 상황이다. 이때 시간대에 따른 조회수 $a_{c_1}(t)$, $a_{c_2}(t)$를 어떠한 가공 없이 그대로 학습하는 경우 학습모델은 제한된 모델의 용량으로 콘텐츠별 조회수의 *크기*와 트렌드 정보를 모두 학습해야 한다. 하지만, *크기* 정보는 조회수 히스토리에서 쉽게 추출할 수 있기 때문에 *크기* 정보를 제외한 트렌드 정보만을 모델에 학습시킨다면 더 높은 예측 정확도를 기대할 수 있다. 이러한 이유로 본 연구에서는 모델에 $a_c(t)$를 직접 학습하지 않고 다음과 같이 $a_c(t-1)$와 $a_c(t)$ 사이의 관계를 학습한다.

$$x_c(t) = \begin{cases} -\dfrac{a_c(t-1)+d}{a_c(t)+d} + 1, & \text{if } a_c(t) \geq a_c(t-1), \\ \dfrac{a_c(t)+d}{a_c(t-1)+d} - 1, & \text{if } a_c(t) < a_c(t-1). \end{cases} \quad (1)$$

식 (1)에서 상수 d는 오프셋으로 분모가 0이 되는 것을 방지하고 비인기 콘텐츠(예: 시간대별 조회수가 0~10 사이의 콘텐츠)에 대해 $x_c(t)$가 급격하게 변화하는 것을 완충해주는 역할을 담당한다. $x_c(t)$는 콘텐츠의 조회수 $a_c(t)$가 직전 시간대 $a_c(t-1)$ 대비 증가하는 경우엔 $[0,1)$ 범위의 값으로 매핑되고, 반대로 감소하는 경우엔 $(-1,0)$ 범위의 값으로 매핑된다. 증가하는 경우와 감소하는 경우 모두 동일한 *크기*의 범위로 매핑되기 때문에 학습모델은 증가 트렌드와 감소 트렌드를 편향 없이 학습할 수 있다. 또한 $x_c(t)$는 모든 범위의 $a_c(t)$에 대해 증가 및 연속함수이며 미분 가능하다. 즉, 식 (1)은 $a_c(t)$에서 *크기* 정보를 제거하면서 트렌드 정보는 왜곡을 최소화하면서 보존하는 함수이다. 식 (1)은 그림 1에서 $f(\cdot)$으로 표현되어있다.

매핑된 히스토리 데이터 $\boldsymbol{x}(:t-1)$를 기반으로 딥러닝 모델을 학습 후 예측값 $\hat{\boldsymbol{x}}(t) = (\hat{x_c}(t), \forall c \in C)$가 출력하면, 아래와 같이 데이터 역매핑 과정을 거쳐 콘텐츠 조회수 형태의 예측값 $\hat{\boldsymbol{a}}(t) = (\hat{a_c}(t), \forall c \in C)$으로 복원한다.

$$\hat{a}_c(t)=\begin{cases}\dfrac{a_c(t-1)+d}{1-\hat{x}_c(t)}-d & \text{if } 0\le \hat{x}_c(t)<1,\\ (\hat{x}_c(t)+1)(a_c(t-1)+d)-d & \text{if } -1<\hat{x}_c(t)<0.\end{cases} \quad (2)$$

이때 $a_c(t-1)$는 시간대 t에서 이미 관측된 값이므로 실제 조회수 값을 사용한다. 식 (2)는 그림 1에서 $f^{-1}(\cdot)$으로 표현되어있다. 제안 매핑 방법 외에도 minmax 정규화와 z-score 정규화 방법이 기존에 존재한다. 그러나 minmax 정규화는 콘텐츠별 전체 시간대에 대한 조회수 최댓값/최솟값 정보가 필요하며, z-score 정규화는 전체 시간대에 대한 콘텐츠별 조회수의 평균과 표준편차 정보가 필요하다. 이 정보들은 사전에 알기 힘든 경우가 많으며, 실시간 학습 시나리오에 적용하는 것이 어렵다. 또한, 훈련 데이터 내에 정상값과 매우 다른 큰 스케일의 이상치가 들어오는 경우, 기존 정규화 방법으로는 전체 시간대에 대한 이상적인 정규화가 이루어지지 않아 학습 성능이 떨어질 수 있다. 반면, 제안하는 데이터 매핑은 사전에 데이터의 통계 정보가 필요하지 않으며, 훈련 데이터에 이상치가 존재하는 상황에도 강인하다는 장점이 있다. 마지막으로, 조회수 뿐만 아니라 시간대에 따른 기타 속성 데이터들에 대해서도 각 속성 간의 크기 차이와 상관없이 모두 같은 범위로 데이터를 매핑하여 학습시킬 수 있다는 장점이 있다.

3.2 딥러닝 모듈

그림 2는 매핑된 데이터를 학습하는 딥러닝 모델이며, 크게 1DCNN, LSTM, Dense 계층으로 구성된다. 학습모델의 입력은 3차원이며, (조회수를 포함한 다중 속성 인덱스, 콘텐츠 인덱스, 시간대 인덱스)에 따른 수치 데이터로 구성된다. 학습모델의 입력부엔 1DCNN 계층이 존재하며, 동일한 시간대에 대하여 주어진 다중 속성 데이터들을 하나의 합성곱 특징으로 매핑한다. 이를 통해 다중 속성 데이터간의 선형적인 상관관계를 학습할 수 있으며, 입력의 차원을 감소시켜 모델의 학습

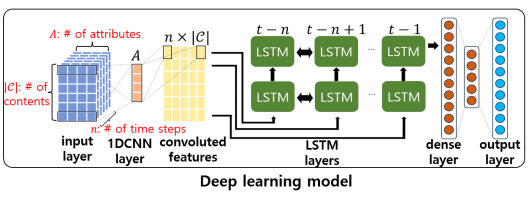

그림. 2. 1DCNN-LSTM-Dense 계층 기반 딥러닝 모델 구조
Fig. 2. Architecture of deep learning model based on 1DCNN-LSTM-Dense layers.

연산 시간을 단축할 수 있다. 1DCNN 계층의 파라미터 수는 입력에 정의된 속성의 개수와 동일하게 설정하며, 활성화함수는 사용하지 않는다.

1DCNN을 통해 차원이 축소된 시계열 데이터를 학습하기 위하여 본 연구에서는 RNN 기반의 학습모델 중 하나인 LSTM을 사용한다. LSTM은 과거부터 시간 순으로 입력된 데이터들을 누적하여 cell state에 저장한다. 이 과정을 통해 cell state의 누적된 입력값과 현재의 입력을 함께 활용하여 향후의 정답값을 예측할 수 있기 때문에, 과거 시간대의 정보들을 효율적으로 활용할 수 있다. 본 연구에서는 시계열 데이터의 복잡한 특징을 더욱 잘 학습할 수 있도록 2층의 LSTM 계층을 구성하며, 활성화함수는 tanh를 사용한다.

마지막으로 LSTM의 출력은 3층으로 구성된 Dense 계층을 통해 분석한다. Dense 계층의 최종 출력값은 각 요소가 (-1, 1) 범위를 갖는 시간대 t에서의 예측값 $\hat{\boldsymbol{x}}(t)$이다. 따라서, Dense 계층의 1, 2층의 활성화함수는 Rectified Linear Unit (ReLU)을 사용하고 3층의 활성화함수는 Leaky ReLU를 변형하여 다음과 같은 Customized Leaky ReLU (CL ReLU)를 사용한다.

$$\mathrm{CL\,ReLU}(x)=\begin{cases}0.01(x-0.9)+0.9, & \text{if } x>0.9,\\ x, & \text{if } |x|\le 0.9,\\ 0.01(x+0.9)-0.9, & \text{if } x<0.9.\end{cases} \quad (3)$$

CL ReLU를 통해 학습모델의 최종 출력 $\hat{\boldsymbol{x}}(t)$의 각 요소가 (-1, 1) 범위에 최대한 매핑될 수 있으며, $\hat{x}_c(t)>|0.9|$인 경우에도 학습이 멈추지 않도록 일반 ReLU 대신 Leaky ReLU를 사용한다. 학습모델의 최종 출력 $\hat{\boldsymbol{x}}(t)$는 앞서 설명한 데이터 역매핑 모듈 식 (2)를 거쳐 콘텐츠 조회수의 예측값 $\hat{\boldsymbol{a}}(t)$가 생성된다.

3.3 모델 학습

본 연구에서 제안한 딥러닝 모델을 학습할 때 사용한 손실함수는 mean squared error (MSE)로 다음과 같은 식을 지닌다.

$$\mathrm{loss}=\sum_{c\in C}(x_c(t)-\hat{x}_c(t))^2. \quad (4)$$

이때 $x_c(t)$는 시간대 t에서의 콘텐츠 c의 실제 조회수 매핑 값, $\hat{x}_c(t)$는 학습모델이 예측한 매핑 값이다.

Ⅳ. 시뮬레이션 기반 평가

4.1 데이터셋

본 시뮬레이션에서는 YouTube에 업로드된 임의의 비디오 콘텐츠 중 2018년 5월 동안 정상적으로 기록된 1574개의 비디오 콘텐츠에 대한 데이터셋을 사용한다[20]. 데이터셋은 1시간 단위로 총 695시간의 데이터가 기록되어 있으며, 콘텐츠별 네 가지 속성(조회수, 댓글 수, 좋아요 수, 싫어요 수)이 기록되어 있다. 또한, 단위 시간대를 3시간으로 가정하고 695시간의 데이터를 3시간 단위로 합산하여 총 226개의 데이터 샘플을 제작한다. 제작한 데이터 샘플은 5:5의 비율로, 시간순으로 훈련 데이터셋과 테스트 데이터셋으로 나눈다.

4.2 시뮬레이션 세팅

LSTM 계층은 2층 모두 cell state의 크기를 150으로 설정한다. Optimizer는 Adaptive Moment Estimation (Adam)을 사용하였으며 학습률은 0.001로 설정한다. Dense 계층은 각각 1024, 512, 1574 개의 노드를 가지며, 1574개의 노드가 있는 계층이 출력층이다. 앞서 소개한 네 가지 속성 데이터를 모두 입력 데이터로 활용한다. 시나리오별로 고정된 네 개의 seed를 기반으로 학습 모델의 파라미터를 초기화하고 훈련 데이터셋에 대한 학습 epoch는 100회로 설정한다. 테스트 데이터셋에 대해서도 매 시간대가 지날 때마다 최신 트렌드를 지속적으로 학습하기 위하여 fine tuning을 적용한다. 매 시간대 t에서 fine tuning은 $\hat{\boldsymbol{a}}(t)$ 예측 이후에 진행되며, 데이터 $\boldsymbol{a}(t-47:t)$에 대하여 2 epoch 씩 진행한다. 데이터 매핑을 진행할 때 오프셋 d는 조회수 속성에 대해서는 100을 적용하고, 나머지 세 속성에는 10을 적용한다.

4.3 성능 평가 지표

본 연구에서는 콘텐츠 캐싱 시나리오를 고려하고 캐싱 가능한 콘텐츠의 총개수를 캐싱 용량으로 가정한다. 같은 캐싱 용량에 대해서 예측 조회수가 높은 순으로 캐싱 된 콘텐츠의 집합을 C^{pred}, 실제 조회수가 높은 순으로 캐싱 된 콘텐츠의 집합을 C^{opt}라 할 때, 다음과 같은 지표로 성능을 평가한다.

$$\text{캐싱 정답률} = \frac{n(C^{\text{pred}} \cap C^{\text{opt}})}{n(C^{\text{opt}})} \tag{5}$$

캐싱 용량은 고려하는 총 콘텐츠 개수의 1%에 해당하는 15개에서 4%에 해당하는 60개까지 범위에 대하여 평가를 진행한다.

4.4 성능 평가 결과

그림 3은 제안 학습 기법과 학습모델에 의존하지 않는 두 휴리스틱 알고리즘의 성능 비교 결과이다. 비교 알고리즘의 동작 방식은 다음과 같다. 1) heuristic_{3h}: 바로 이전 시간대 $t-1$에서 관측된 콘텐츠 조회수가 t에서도 유지된다고 가정했을 때 콘텐츠를 캐싱하는 방식. 2) heuristic_{24h}: 최근 과거 24시간 동안의($t-8$부터 $t-1$까지) 조회수 평균값을 기준으로 콘텐츠를 캐싱하는 방식. 그림 3을 통해 heuristic_{3h}와 heuristic_{24h} 모두 0.8 이상의 캐싱 정답률을 달성하는 것을 확인할 수 있다. 이는 현재 사용한 YouTube 데이터셋이 인접한 시간대 사이에 조회수 상관관계가 매우 높다는 것을 의미한다. 또한, 제안 학습 기법은 heuristic_{3h}, heuristic_{24h}보다 캐싱 정답률이 상회하는 것을 볼 수 있다. 예를 들어, 캐싱 용량이 20개일 때, 각 휴리스틱 알고리즘보다 캐싱 정답률이 각각 9.95%, 14.70% 더 높다. 이는 학습모델을 통하여 시간에 따른 콘텐츠 조회수의 숨겨진 패턴을 학습하고, 이에 기반하여 향후 조회수를 예측하는 것이 콘텐츠 캐싱에 효과적임을 의미한다.

그림 4는 제안 학습 기법과 기존의 다른 정규화 기법을 적용한 경우의 성능 비교 결과이다. 비교 대상은 다음과 같다. 1) $\text{minmax}_{\text{norm}}$: 제안 학습 기법에서 데이터 매핑 및 역매핑 모듈 대신 0-1 minmax 정규화 기법을 적용. 2) $\text{z-score}_{\text{norm}}$: 제안 학습 기법에서 데이터 매핑 및 역매핑 모듈 대신 z-score 정규화 기법을 적용. 3) count: 제안 학습 기법에서 데이터 매핑 및 역매핑 모듈을 사용하지 않고 조회수를 직접 학습. 두 정규화 기법 1), 2)을 데이터 매핑 및 역매핑으로 표현하면 다

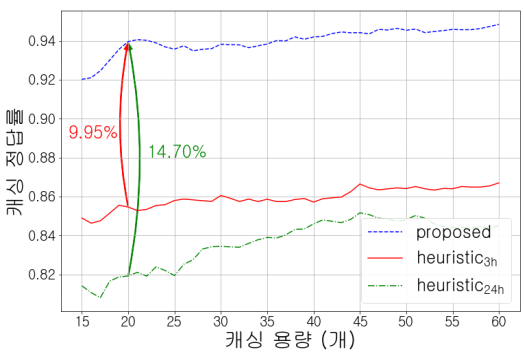

그림. 3. heuristic_{3h} 및 heuristic_{24h}와 성능 비교
Fig. 3. Performance comparison with heuristic_{3h} and heuristic_{24h}

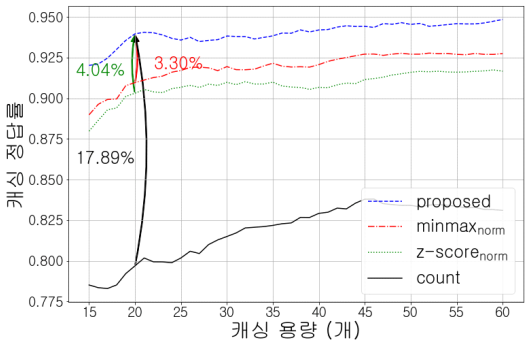

그림. 4. minmax$_{norm}$, z−score$_{norm}$, count와 성능 비교
Fig. 4. Performance comparison with minmax$_{norm}$, z−score$_{norm}$ and count

음의 수식과 같다.

$$\text{minmax}_{\text{norm}}: \begin{cases} x_c(t) = \dfrac{a_c(t) - a_c^{\min}}{a_c^{\max} - a_c^{\min}}, & (6) \\ \hat{a_c}(t) = \hat{x_c}(t)(a_c^{\max} - a_c^{\min}) + a_c^{\min}, & (7) \end{cases}$$

$$\text{z-score}_{\text{norm}}: \begin{cases} x_c(t) = \dfrac{a_c(t) - a_c^{\text{mean}}}{a_c^{\text{std}}}, & (8) \\ \hat{a_c}(t) = \hat{x_c}(t)(a_c^{\text{std}}) + a_c^{\text{mean}}. & (9) \end{cases}$$

이때, $a_c^{\min}$와 $a_c^{\max}$는 전체 시간대에서 콘텐츠 c의 조회수의 최댓값과 최솟값을 의미하고, a_c^{mean}, a_c^{std}는 전체 시간대에서 콘텐츠 c의 조회수의 평균과 표준편차를 의미한다. 그림 4를 통해 정규화를 거치지 않은 count가 가장 낮은 성능을 보이는 것을 확인할 수 있다. 이는, 콘텐츠마다 조회수가 제각각이고, 주어진 LSTM cell state에 콘텐츠별 조회수의 크기와 트렌드 정보를 모두 담아야 하기 때문에 예측 조회수 정확도가 떨어지면서 발생하는 결과이다. 정규화를 적용한 minmax$_{norm}$와 z−score$_{norm}$의 경우, count보다 더 높은 캐싱 정답률을 얻을 수 있지만 minmax$_{norm}$은 콘텐츠별 ($a_c^{\min}$, $a_c^{\max}$) 정보를, z−score$_{norm}$은 콘텐츠별 (a_c^{mean}, a_c^{std}) 정보를 사전에 요구한다는 현실적인 한계가 존재한다. 반면에 제안 학습 기법은 어떠한 사전 정보 없이도 두 정규화 기법을 상회하는 것을 볼 수 있다. 예를 들어 캐싱 용량이 20개일 때, 제안 학습 기법의 캐싱 정답률은 count보다 17.89%, minmax$_{norm}$보다 3.30%, z−score$_{norm}$보다 4.04% 더 높다. 이는 제안 학습 기법이 기존 정규화 기법의 현실적 한계를 극복하면서도 더 효과적인 조회수 예측이 가능하다는 것을 의미한다.

그림 5는 제안 학습 기법과 유사한 다른 매핑 및 역매핑 기법을 적용한 경우의 성능 비교 결과이다. 비교 대상은 다음과 같다. 1) ascend$_{map}$: 매 시간대 t마다 $a_c(t-1)$ 대비 $a_c(t)$의 증가 비율을 학습[19]. 2) descend$_{map}$: 매 시간대 t마다 $a_c(t)$ 대비 $a_c(t-1)$의 증가 비율을 학습. 두 기법 1), 2)을 데이터 매핑 및 역매핑으로 표현하면 다음의 수식과 같다.

$$\text{ascend}_{\text{map}}: \begin{cases} x_c(t) = \dfrac{a_c(t)}{a_c(t-1)}, & (10) \\ \hat{a_c}(t) = \hat{x_c}(t)a_c(t-1), & (11) \end{cases}$$

$$\text{descend}_{\text{map}}: \begin{cases} x_c(t) = \dfrac{a_c(t-1)}{a_c(t)}, & (11) \\ \hat{a_c}(t) = \dfrac{a_c(t-1)}{\hat{x_c}(t)}. & (12) \end{cases}$$

ascend$_{map}$를 적용한다면, $x_c(t)$는 $[0,\infty)$의 범위를 가질 수 있다. 특히, 조회수가 증가하는 경향 ($a_c(t) \geq a_c(t-1)$)은 $[1,\infty)$의 범위에 매핑되고 감소하는 경향($a_c(t) < a_c(t-1)$)은 $(0,1)$의 범위로 매핑된다. 즉, ascend$_{map}$는 조회수 감소보다 증가 경향에 더 초점을 맞추는 매핑 방식이다. descend$_{map}$는 ascend$_{map}$와 정확히 역수 관계에 있으므로, 동일한 논리에 따라 조회수 증가보다 감소 경향에 더 초점을 맞추는 매핑 방식이다. 그림 5을 통해 제안 학습 기법은

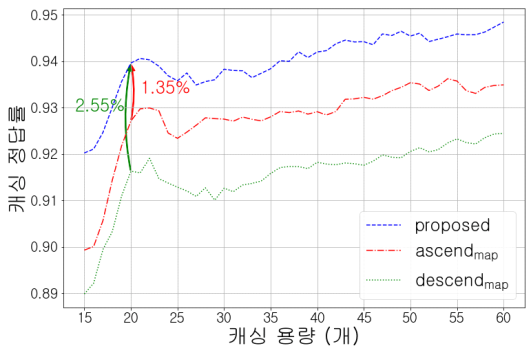

그림. 5. ascend$_{map}$ 및 descend$_{map}$와 성능 비교
Fig. 5. Performance comparison with ascend$_{map}$ and descend$_{map}$

$ascend_{map}$와 $descend_{map}$보다 높은 캐싱 정답률을 달성한 것을 볼 수 있다. 예를 들어, 캐싱 용량이 20개일 때, 제안 학습 기법의 캐싱 정답률은 $ascend_{map}$보다 1.35%, $descend_{map}$보다 2.55% 더 높다. 이는 제안 학습 기법이 식 (1)에 따라 조회수가 증가하는 경우와 감소하는 경우를 동일한 크기의 범위로 매핑 함으로써 조회수 증가와 감소 경향 모두를 편향되지 않고 학습하기 때문에 더욱 효과적인 조회수 예측이 가능하다는 것을 의미한다.

그림 6은 제안 학습 기법에 대하여, 테스트 데이터셋을 대상으로 실시간 fine tuning을 적용했을 때(proposed)와 적용하지 않았을 때(non－ftune)의 성능 비교 결과이다. 해당 비교에서도 유의미한 캐싱 정답률 차이를 확인하였으며, 이는 콘텐츠의 인기도는 시간이 흘러감에 따라 과거와 다른 양상으로 변화할 수 있다는 것을 의미한다. 따라서, 최신 히스토리를 바탕으로 지속적인 온라인 학습이 동반되어야 효율적인 콘텐츠 캐싱이 가능함을 알 수 있다.

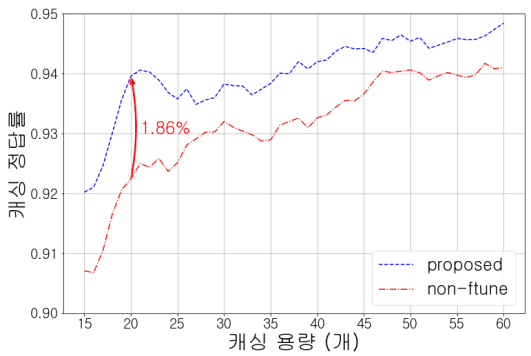

그림. 6. non－ftune과 성능 비교
Fig. 6. Performance comparison with non－ftune.

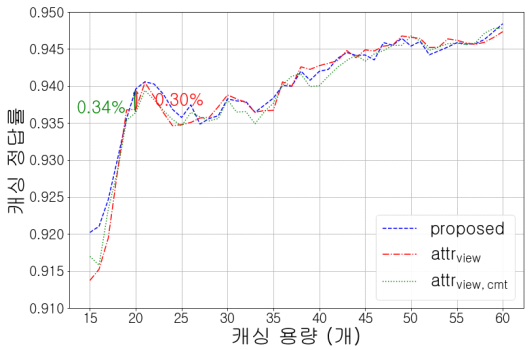

그림. 7. $attr_{view}$ 및 $attr_{view,cmt}$와 성능 비교
Fig. 7. Performance comparison with $attr_{view}$ and $attr_{view,cmt}$

그림 7는 제안 학습 기법에 대하여, 모델의 입력으로 사용한 다중 속성 개수를 달리했을 때의 성능 비교 결과이다. 비교 대상은 다음과 같다. 1) $attr_{view}$: 콘텐츠 조회수만 활용한 경우. 2) $attr_{view,cmt}$: 콘텐츠 조회수와 댓글 수를 활용한 경우. 그림 7을 통해 네 가지 속성을 모두 활용하는 proposed가 아주 근소한 차이로만 $attr_{view}$와 $attr_{view,cmt}$를 상회하는 것을 볼 수 있다. 이는 조회수 예측에는 다른 속성의 히스토리보다 조회수 히스토리가 압도적인 상관관계가 있음을 의미한다.

V. 결 론

본 연구에서는 시간대마다 관측되는 사용자의 콘텐츠 요청 히스토리를 이용하여 미래의 콘텐츠 조회수를 예측하기 위한 딥러닝 기법을 제안하였다. 특히, 예측 성능을 높이기 위하여, 조회수의 크기 정보를 제외하고 트렌드에만 초점을 맞추어 학습할 수 있는 데이터 매핑 기법을 제안하였고, 다중 속성 데이터 간의 상관관계, 시간 축에서의 데이터의 상관관계를 학습할 수 있는 1DCNN-LSTM-Dense 계층 기반의 딥러닝 모델을 제안하였다. 제안 학습 기법은 기존에 제시되어왔던 다양한 휴리스틱, 정규화 기법, 데이터 매핑 기법 등과 비교하여 미래 정보가 사전에 필요하지 않으면서도 가장 높은 캐싱 정답률을 달성하는 것을 확인하였다. 제안 학습 기법은 콘텐츠 캐싱뿐만 아니라 수치 데이터의 변화를 학습 및 예측해야 해야 하는 다양한 시나리오에서도 높은 예측 정확도를 보일 것으로 기대된다.

References

[1] J. Kwak, Y. Kim, L. B. Le, and S. Chong, “Hybrid content caching in 5G wireless networks: Cloud versus edge caching,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3030-3045, May 2018. (https://doi.org/10.1109/twc.2018.2805893)

[2] A. Gharaibeh, A. Khreishah, and I. Khalil, “An O(1)-competitive online caching algorithm for content centric networking,” in *Proc. IEEE INFOCOM*, pp. 1-9, San Francisco, CA, USA, Apr. 2016. (https://doi.org/10.1109/infocom.2016.7524444)

[3] N. Abedini and S. Shakkottai, “Content caching and scheduling in wireless networks

with elastic and inelastic traffic," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 864-874, Jun. 2014.
(https://doi.org/10.1109/tnet.2013.2261542)

[4] Z. Zhao, M. Peng, Z. Ding, W. Wang, and H. V. Poor, "Cluster content caching: An energy-efficient approach to improve quality of service in cloud radio access networks," *IEEE J. Sel. Areas in Commun.*, vol. 34, no. 5, pp. 1207-1221, May 2016.
(https://doi.org/10.1109/jsac.2016.2545384)

[5] T. X. Tran, A. Hajisami, and D. Pompili, "Cooperative hierarchical caching in 5G cloud radio access networks," *IEEE Network*, vol. 31, no. 4, pp. 35-41, Jul 2017.
(https://doi.org/10.1109/mnet.2017.1600307)

[6] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402-8413, Dec. 2013.
(https://doi.org/10.1109/tit.2013.2281606)

[7] M. Garetto, E. Leonardi, and S. Traverso, "Efficient analysis of caching strategies under dynamic content popularity," in *Proc. IEEE INFOCOM*, pp. 2263-2271, Hong Kong, China, Apr. 2015.
(https://doi.org/10.1109/infocom.2015.7218613)

[8] X. Zhang, Z. Qi, G. Min, W. Miao, Q. Fan, and Z. Ma, "Cooperative edge caching based on temporal convolutional networks," *IEEE Trans. Parall. and Distrib. Syst.*, vol. 33, no. 9, pp. 2093-2105, Apr. 2022.
(https://doi.org/10.1109/tpds.2021.3135257)

[9] B. N. Bharath, K. G. Nagananda, and H. V. Poor, "A learning-based approach to caching in heterogenous small cell networks," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1674-1686, Apr. 2016.
(https://doi.org/10.1109/tcomm.2016.2536728)

[10] P. Rodriguez, M. A. Bautista, J. Gonzalez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Elsevier Image and Vision Computing*, vol. 75, pp. 21-31, Jul. 2018.
(https://doi.org/10.1016/j.imavis.2018.04.004)

[11] C. Zhang, H. Pang, J. Liu, S. Tang, R. Zhang, D. Wang, and L. Sun, "Toward edge-assisted video content intelligent caching with long short term memory learning," *IEEE Access*, vol. 7, pp. 152 832-152 846, Oct. 2019.
(https://doi.org/10.1109/access.2019.2947067)

[12] L. Ale, N. Zhang, H. Wu, D. Chen, and T. Han, "Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 5520-5530, Jun. 2019.
(https://doi.org/10.1109/jiot.2019.2903245)

[13] A. Narayanan, S. Verma, E. Ramadan, P. Babaie, and Z.-L. Zhang, "Making content caching policies' smart'using the deepcache framework," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 5, pp. 64-69, Jan. 2019.
(https://doi.org/10.1145/3310165.3310174)

[14] D. Li, H. Zhang, D. Yuan, and M. Zhang, "Learning-based hierarchical edge caching for cloud-aided heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 1648-1663, Mar. 2023.
(https://doi.org/10.1109/twc.2022.3206236)

[15] M. W. Kang and Y. W. Chung, "Content caching based on popularity and priority of content using seq2seq lstm in icn," *IEEE Access*, vol. 11, pp. 16 831-16 842, Feb. 2023.
(https://doi.org/10.1109/access.2023.3245803)

[16] A. Lekharu, M. Jain, A. Sur, and A. Sarkar, "Deep learning model for content aware caching at mec servers," *IEEE Trans. Netw. and Serv. Manag.*, vol. 19, no. 2, pp. 1413-1425, Jun. 2022.
(https://doi.org/10.1109/tnsm.2021.3136439)

[17] J. Liang, D. Zhu, H. Liu, H. Ping, T. Li, H. Zhang, L. Geng, and Y. Liu, "Multi-head attention based popularity prediction caching in social content-centric networking with mobile edge computing," *IEEE Commun. Lett.*, vol. 25, no. 2, pp. 508-512, Feb. 2021.
(https://doi.org/10.1109/lcomm.2020.3030329)

[18] S. H. Na, Y. J. Kim, H. M. You, H. J. Ahn, J. M. Moon, and E. K. Hong, "Clustering method for mobile traffic prediction," *J. KICS*, vol. 47, no. 2, pp. 398-407, Feb. 2022. (https://doi.org/10.7840/kics.2022.47.2.398)

[19] D. Shah, W. Campbell, and F. Zulkernine, "A comparative study of LSTM and DNN for stock market forecasting," in *Proc. IEEE BigData*, Seattle, WA, USA, Dec. 2018. (https://doi.org/10.1109/bigdata.2018.8622462)

[20] *Youtube videos view count every hour*(2018), Retrieved Feb., 20, 2023, from https://www.kaggle.com/datasets/nnqkfdjq/statistics-observation-of-random-youtube-video

박 용 문 (Yongmoon Park)

2022년 2월 : 인하대학교 전자공학과 졸업

2023년 8월~현재 : 인하대학교 전기컴퓨터공학과 석사 과정

<관심분야> 콘텐츠 캐싱 및 엣지 컴퓨팅

[ORCID:0009-0004-2607-9260]

김 영 진 (Yeongjin Kim)

2011년 2월 : 한국과학기술원 전자공학과

2013년 2월 : 한국과학기술원 전자공학과 석사

2018년 2월 : 한국과학기술원 전자공학과 박사

2018년 3월~2020년 2월 : 삼성전자 senior engineer

2020년 3월~현재 : 인하대학교 전자공학과 조교수

<관심분야> 모바일, 엣지, 클라우드 컴퓨팅

[ORCID:0000-0003-4482-2287]