# VAE 와 연합 학습 활용 콘텐츠 캐싱 전략의 Jetson Boards 기반 성능 평가

임혜리, 김유노, 최민석
경희대학교

mary@khu.ac.kr, rladbsh456@naver.com, choims@khu.ac.kr

# Implementation of VAE-Federated Learning-based Content Caching using Jetson Boards

Hyelee Lim, Yunoh Kim, Minseok Choi
Kyung Hee Univ.

요 약

본 연구는 VAE 와 연합 학습을 결합한 캐싱 전략을 통해 에지 서버와 디바이스 간의 캐싱 효율을 개선하고 지연 시간을 줄이는 방법을 제안했다. Heuristic 1 은 글로벌 인기도와 개인 인기도를 각각 독립적으로 고려하여 캐싱을 결정하며, Heuristic 2 는 글로벌 인기도를 우선적으로 고려한 후 남은 파일을 디바이스에서 캐싱하는 전략을 따른다.

## 1. 서론

비디오 트래픽의 급격한 증가로 인해 네트워크 자원의 효율적 관리와 신속한 콘텐츠 전달을 위한 콘텐츠 캐싱 및 인기도 예측이 더욱 중요해지고 있다. 그러나 비디오 콘텐츠의 인기는 시간과 지역에 따라 크게 변동하며, 이러한 콘텐츠 요청 데이터는 개인정보이며 신중한 처리가 필요하다. 특히, 사용자의 요청 데이터를 중앙 서버로 직접 전송하지 않고도 이를 분석하고 예측할 수 있는 기술적 접근이 요구된다. 이러한 문제를 해결하기 위해 연합 학습[1]이 주목받고 있으며, 이는 사용자의 콘텐츠 요청 기록을 개별적으로 학습한 후, 이를 공유하여 전체적인 콘텐츠 인기도를 정확하게 예측할 수 있는 방법을 제공한다 [2]. 또한, Variational Autoencoder (VAE)는 입력 데이터를 압축하여 잠재 변수를 학습하고 이를 바탕으로 데이터를 재구성하는 방식으로 동작한다. [2]에 따르면 VAE 와 연합 학습을 결합하여 개별적으로 학습된 사용자 데이터를 기반으로 하여 전체적인 콘텐츠 인기도를 정확하게 예측할 수 있다. 이때 연합 학습을 통해 사용자 디바이스에 분산된 데이터를 중앙 서버에 직접 공유하지 않으면서도 프라이버시를 보호할 수 있으며, 지속적으로 모델을 업데이트할 수 있다.

따라서 본 연구에서는 VAE 와 연합 학습을 결합한 시나리오를 기반으로 에지 서버와 사용자 디바이스의 콘텐츠 캐싱을 수행하고, 캐싱 방법에 따른 콘텐츠 전송 지연 시간을 NVIDIA 사의 Jetson board 기반의 테스트베드에서 측정하여 VAE 활용 연합 학습 기반 캐싱 기술을 실증한다.

## 2. 캐싱 네트워크 모델

본 논문은 $M$ 개의 에지 서버가 서로 겹치지 않는 고유의 커버리지 영역을 가지고, 각 에지 서버는 $K$ 명의 사용자로 구성된 시나리오를 고려한다. 사용자는 온라인 비디오 서비스를 통해 개별적으로 콘텐츠를 요청한다. 이때, 에지 서버와 사용자 디바이스는 일부 인기 콘텐츠를 사전 캐싱할 수 있고, 각각 캐시 크기를 $C_E$ 와 $C_D$ 라 하자. 사용자가 콘텐츠를 요청할 때, 해당 콘텐츠가 사용자 기기에 캐싱되어 있으면 지연 시간 없이 바로 이용할 수 있다. 그렇지 않다면, 에지 서버에게 콘텐츠 요청을 하고, 에지 서버가 해당 파일을 캐싱하고 있다면 바로 사용자에게 전달한다. 반면, 캐싱되지 않은 콘텐츠에 대한 요청은 중앙 서버로 전달되며, 중앙 서버는 파일 라이브러리에서 해당 콘텐츠를 검색해 에지 서버를 통해 사용자에게 전달한다. 이때, 사용자마다 다른 개인 선호도를 반영하여 개인별로 다른 콘텐츠 인기도 분포를 갖는다고 가정하며, 인기도 분포는 BBC dataset 을 기반으로 콘텐츠 인기도를 모델링한 결과를 가정하였다 [3].

## 3. VAE-연합학습 기반 디바이스 및 에지 서버 캐싱

본 연구에서는 VAE 모델을 장르별로 여러 서브네트워크로 나누어 설계하였다. 각 사용자는 VAE 모델을 기반으로 로컬 학습을 수행한 후, 학습된 로컬 모델을 에지 서버에 업로드한다. 에지 서버는 사용자들로부터 수집한 로컬 모델을 통합하여 글로벌 모델을 형성하고, 이를 다시 사용자들에게 전달한다. 사용자들은 수신한 글로벌 모델과 자신이 학습한 개인화 모델을 활용해 요청 데이터에 대한 샘플을 생성한다. 이 과정에서 장르별 샘플을 집계하여 장르별 샘플 비율, 즉 장르 인기도를 계산하며, 각 장르의 상대적 인기도에 비례한 샘플을 생성하여 새로운 학습 데이터를 구성한다.

이에 따라 본 연구에서는 연합 학습 기반 VAE[2] 알고리즘을 활용하여 캐싱 순서 및 방법에 따른 Heuristic 1 과 Heuristic 2 를 제안한다. 두 Heuristic 은 글로벌 인기도와 개인 인기도를 고려하여 에지 서버와 디바이스의 캐싱 전략을 다르게 설정하는 방식이다. 여기서 글로벌 인기도 $q = [q_1, \cdots, q_F]$ 는 전체 사용자들이 요청한 콘텐츠의 인기도를 나타내며, 각 파일 $F$의 인기도 값을 포함한다. 개인 인기도 $x_k = [x_{k,1}, \cdots, x_{k,F}]$는 각 사용자 k 의 요청 패턴을 반영한 인기도를 나타내며, 특정 사용자가 자주 요청하는 파일들의 인기도 값을 포함한다.

1) Heuristic 1: 글로벌인기도 $q$ 와 개인인기도 $x_k$ 를 각각 독립적으로 고려하여 에지 서버와 디바이스 캐싱을 결정한다.

2) Heuristic 2: 글로벌 인기도 $q$ 에 따라 에지 서버 캐싱을 결정한 후, 캐싱되지 않은 남은 파일들에 대해 디바이스 캐싱을 결정한다.

## 4. 실험결과

Testbed 실험을 위해 에지 서버는 Windows 11, Python 3.10.9, PyTorch 2.0.1, CUDA 11.7, CuDNN 8.5.0 의 소프트웨어 환경과 GTX 1660 SUPER GPU, Intel i5-11660k CPU 를 갖춘 데스크탑을 사용했다. 사용자로는 Nvidia Jetson Nano 3 대를 사용했으며, 소프트웨어 환경은 Ubuntu 18.04, Python 3.6.9, PyTorch 1.8.0, CUDA 10.2.89, CuDNN 8.0.0 으로 설정되었다. Jetson Nano 의 하드웨어는 128-core NVIDIA Maxwell GPU 와 Quad-Core ARM A57 CPU 로 구성되어 있다. 무선 네트워크 환경에서는 ipTIME A2004MU 공유기를 통해 5GHz 대역에서 Intel 8265NGW 무선 랜카드를 사용하여 최대 867Mbps 속도를 지원하였다. Jetson Nano 의 모든 하드웨어 성능을 최대로 설정하여 실험을 진행했다.

실험 설정은 글로벌 라운드 100, 로컬 에포크 10 으로, 총 3 명의 유저가 2000 개의 컨텐츠 중 1000 번씩 요청을 수행했다. 실시간 통신을 통해 Jetson Nano 와 데스크탑이 학습을 진행하며, 이후 각 Heuristic 방법에 따라 각 유저별로 테스트를 진행하였다. 유저는 특정 장르에 치우치지 않도록 다양한 장르를 선호하는 사용자들로 선택되었다. Table 1 에서는 실험 환경 확인을 위하여 대역폭 평가를 진행했다. 총 10 번의 대역폭 실험을 하였으며, 각 실험은 10 번의 송수신 트래픽을 발생시켜 평균을 낸 값이다.

| Maximum | Minimum | Average |
|---|---|---|
| 253 | 143 | 220 |

Table 1. 대역폭 평가 실험(Mbit/sec).

Jetson 에서 에지 서버로의 요청 지연 시간은 평균 0.0948 초로 측정되었으며, 에지 서버에서 클라우드 서버로의 요청 지연 시간은 0.2 초(고정 상수)로 설정되었다. 또한, 각 요청에 따른 파일 전송 지연 시간은 1 초(고정 상수)로 설정되었다. 디바이스 캐시 히트 시 지연 시간은 0 초이며, Table 2 에 따르면 에지 서버 캐시 히트 시의 총 지연 시간은 Heuristic 1 에서 485.1 초, Heuristic 2 에서 880 초로 계산되었다. Heuristic 1 의 경우, 개인 인기도를 반영한 디바이스 캐싱 전략이 사용자의 개별 요청 패턴을 더 잘 반영하여 디바이스에서의 캐시 히트 빈도가 높았다. 반면, Heuristic 2 는 글로벌 인기도를 기준으로 에지 서버에서 우선적으로 데이터를 캐싱하므로, 에지 서버에서 캐시되지 않은 파일들만 디바이스에서 캐싱하게 되어 디바이스 캐시 히트율이 낮게 나타난 것을 확인할 수 있었다.

| | Device cache hit | Edge cache hit |
|---|---|---|
| Heuristic 1 | 206 | 441 |
| Heuristic 2 | 39 | 800 |

Table 2. 유저 1 명의 1000 번 요청 중 캐시 히트율.

Table 3 에서 ES delay 은 디바이스와 에지 서버 간의 지연 시간으로, 에지 서버에서 캐시 히트가 발생할 때의 지연 시간을 의미한다. Total delay 은 ES delay 에 더해, 에지 서버와 클라우드 서버 간의 지연 시간을 포함한 값이다. 이는 디바이스가 요청한 콘텐츠가 에지 서버에 캐싱되지 않은 경우, 클라우드 서버에 콘텐츠를 요청하는 데 소요되는 시간을 의미한다. SW 시뮬레이션 실험에서는 통신 환경을 고려하지 않고 지연 시간을 고정된 상수로 설정한 반면, testbed 실험에서는 실제 통신 환경을 반영하여 진행되었다.

| | ES delay | Total delay |
|---|---|---|
| Heuristic 1 | 531.8713 | 1386.6265 |
| Heuristic 2 | 53.9941 | 1915.3215 |

Table 3. 캐싱 전략에 따른 지연시간 속도 측정값(sec).

SW 시뮬레이션과 testbed 실험 모두에서 Heuristic 1 과 2 순서로 전체 지연 시간이 더 작은 것으로 관찰되었다. Heuristic 2 에서 전체 지연 시간이 더 크게 측정된 이유는, 에지 서버 캐시 히트가 더 많이 발생하여 디바이스 히트에 비해 추가적인 지연 시간이 소요되었기 때문이다. 디바이스에서 캐시 히트가 발생할 경우 지연 시간이 0 초로 가정되지만, 에지 서버 캐시 히트는 ES 지연 시간이 존재하므로 전체 지연 시간이 더 길어지게 되었다.

## 5. 결론

본 논문에서는 VAE-연합 학습 기반 캐싱 전략을 통해 에지 서버와 디바이스 간의 캐시 히트율 및 지연 시간을 비교 분석하였다. 실험 결과, Heuristic 1 이 글로벌 인기도와 개인 인기도를 독립적으로 고려하여 디바이스 캐시 히트율이 높아지고 전체 지연 시간이 짧아졌다. 반면, Heuristic 2 는 에지 서버 캐싱을 우선시하여 에지 서버 캐시 히트율은 높았지만, 디바이스 캐시 히트율이 낮아져 ES 지연 시간이 증가하고 전체 지연 시간도 더 길어졌다. 실제 통신 환경을 반영한 testbed 실험에서도 SW 시뮬레이션과 유사한 결과가 나타났다. 따라서, 글로벌 인기도와 개인 인기도를 균형 있게 고려하고, 디바이스에서의 캐시 히트를 극대화하는 전략이 전체 지연 시간을 줄이는 데 효과적임을 확인할 수 있었다.


## ACKNOWLEDGMENT

이 논문은 2024년도 정보(과학기술정보통신부)의 재원으로 한국연구재단과 정보통신기획평가원의 지원을 받아 수행된 연구임 (NRF-2022R1C1C1010766, No. 2022R1A4A3033401, No.2021-0-02201, 사용자 프라이버시를 보존하는 비디오 캐싱을 위한 연합 학습 시스템).


## 참 고 문 헌


[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). PMLR.

[2] Ahn, M., & Choi, M. (2023, October). Federated Learning with Variational Autoencoder for Popularity Profile Prediction. In *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 1027-1032). IEEE.

[3] M.-C. Lee, et al. “Individual preference probability modeling and parameterization for video content in wireless caching networks,” IEEE/ACM Transactions on Networking, 27.2 (2019): 676-690.