일반논문 (Regular Paper)
방송공학회논문지 제31권 제2호, 2026년 3월 (JBE Vol.31, No.2, March 2026)
https://doi.org/10.5909/JBE.2026.31.2.280
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

ARM NEON 기반 셋톱박스에서의 경량 추천 모델 온디바이스 학습 시스템

김 현 수[a], 문 남 미[a]‡

On-Device Learning System for Lightweight Recommendation Models on Set-Top Box with ARM NEON Optimization

Hyunsoo Kim[a] and Nammee Moon[a]‡

요 약

본 논문은 NPU가 없는 저사양 셋톱박스(STB) 환경에서 개인화 콘텐츠 추천을 위한 온디바이스 학습 및 추론 시스템을 제안한다. 기존 추천 시스템은 서버에서 학습된 모델을 단말에 배포하여 추론만 수행함으로써 실시간 개인화에 한계가 있다. 본 연구에서는 ARM Cortex-A55 기반 Android TV STB(3GB RAM, 팬리스 설계)에서 행렬분해 기반 추천 모델의 증분 학습과 INT8 양자화 추론을 동시에 수행하는 경량 시스템을 구현하였다. ARM NEON SIMD 기반 행렬 연산 최적화와 CPU 온도·메모리 사용률 기반 적응형 리소스 제어를 적용하여 제한된 자원 환경에서도 안정적인 학습을 실현하였다. 실험 결과, 순수 C++ 대비 최대 2.2배의 추론 속도 향상과 모델 크기 75% 감소를 달성하였으며, Hit Rate@10 46.88%, End-to-End 추천 지연시간 0.86ms를 기록하였다. HNSW 기반 Top-K 검색으로 최대 11.2배의 속도 향상을 확인하였다.

Abstract

This paper presents an on-device learning system that enables both incremental training and real-time recommendation on low-spec set-top boxes (STBs) without neural processing units (NPUs). Unlike conventional recommendation systems that rely on server-trained models and perform only inference on devices, the proposed system supports on-device incremental training and INT8 quantized inference of Matrix Factorization models on ARM Cortex-A55 based Android TV STBs with 3GB RAM and fanless design. Key techniques include ARM NEON SIMD acceleration for matrix operations, adaptive resource control based on CPU temperature and memory utilization during on-device training, and Post-Training Quantization (PTQ) INT8 for inference optimization. Experimental results demonstrate up to 2.2× inference speedup over a plain C++ implementation through NEON optimization, along with a 75% reduction in model size via INT8 quantization while maintaining recommendation accuracy degradation within 0.1 percentage points. On the MovieLens 1M dataset, the proposed system achieves a Hit Rate@10 of 46.88% with an end-to-end recommendation latency of 0.86 ms, and HNSW-based approximate nearest neighbor search provides up to 11.2× speedup for Top-K retrieval.

Keyword : On-Device Learning, ARM NEON, Lightweight Recommendation System, Matrix Factorization, INT8 Quantization

Ⅰ. 서 론

1. 연구 배경

IPTV 및 OTT 서비스 이용자가 급증하면서 개인화된 콘텐츠 추천의 중요성이 커지고 있다. 방송통신위원회 통계에 따르면 2023년 12월 말 기준 국내 IPTV 가입자는 2,098만 단자로 전년 대비 1.5% 증가하여 유료 방송 중 유일하게 성장세를 유지하고 있다[1]. 채널 수는 수백 개에 달하지만 실제 선택할 수 있는 시간과 주의는 한정적이기 때문에, 사용자가 원하는 콘텐츠를 빠르게 찾을 수 있는 초저지연 개인화 추천 기술의 필요성이 더욱 커지고 있다.

기존 추천 시스템은 주로 서버 기반 학습(Server-Side Learning) 방식을 채택하며[2], 다음과 같은 한계를 갖는다: 첫째, 프라이버시 문제로 사용자의 상세 시청 이력이 서버로 전송되어 개인정보 유출 위험이 존재한다. 둘째, 실시간 개인화 제약으로 서버 모델 업데이트 주기(일/주 단위)로 인해 사용자의 최신 선호도 변화를 즉각 반영하기 어렵다. 셋째, 네트워크 의존성으로 지속적인 서버 통신이 필요하다.

2. 연구 목적 및 기여

본 연구는 NPU(Neural Processing Unit) 없는 저사양 Android TV STB에서 MF(Matrix Factorization) 기반 추천 모델의 온디바이스 학습 및 추론 시스템을 설계·구현한다. 기존 연구가 추론에 집중한 반면, 본 연구는 3GB RAM, 팬리스 설계의 극한 제약 하에서 학습까지 수행한다는 점에서 차별화된다. 핵심 기여는 1) ARM NEON SIMD (Single Instruction, Multiple Data) 기반 행렬 연산 고속화, 2) 증분 학습 기반 실시간 사용자 임베딩 업데이트, 3) CPU 온도/메모리 기반 적응형 리소스 제어, 4) PTQ(Post-Training Quantization) 기반 INT8 양자화를 통한 추론 최적화, 5) 상용 STB 기반 실증 평가이다.

a) 호서대학교 벤처대학원 융합공학과(Dept. of Convergence Engineering, Hoseo University)
‡ Corresponding Author : 문남미(Nammee Moon)
E-mail: mnm@hoseo.edu
Tel: +82-2-2059-2310
ORCID: https://orcid.org/0000-0003-2229-4217
· Manuscript January 22, 2026; Revised March 7, 2026; Accepted March 9, 2026.

Ⅱ. 관련 연구

1. 온디바이스 AI 및 경량화 기법

온디바이스 AI는 클라우드 의존성을 줄이고 지연시간 감소 및 프라이버시 보호를 목적으로 단말 기기에서 직접 AI 모델 추론 또는 학습을 수행하는 기술이다. TensorFlow Lite[3], Core ML[4] 등 주요 프레임워크가 모바일 추론을 지원하며, 양자화(Quantization), 프루닝(Pruning), 지식 증류(Knowledge Distillation) 등 경량화 기법이 연구되고 있다[5]. Jacob et al.[6]은 INT8 양자화로 Snapdragon 835에서 약 50%의 지연시간 감소를 달성하였다. 온디바이스 학습은 추론 대비 연산량이 높아 연구 사례가 제한적이며, McMahan et al.[7]의 FedAvg 알고리즘을 통한 Federated Learning이 대표적이다.

2. 추천 시스템 및 협업 필터링

협업 필터링(Collaborative Filtering, CF)은 사용자-아이템 상호작용 행렬을 기반으로 추천하는 기법이다[8]. Matrix Factorization(MF)[9]은 희소 상호작용 행렬을 저차원 임베딩의 곱으로 근사하며, Netflix Prize에서 우수한 성능을 입증하였다. Hu et al.[10]은 TV STB의 시청 로그를 활용한 Implicit Feedback 기반 MF에서 인기도 기반 추천 대비 50% 성능 향상을 달성하였다. Kim et al.[11]은 IPTV 환경에서 협업 필터링 기반 추천 시스템을 구현하여 사용자 시청 패턴 분석의 유효성을 검증하였으며, Oh et al.[12]은 시청 이력 기반 방송 콘텐츠 추천 기법을 제안하여 사용자 선호도 예측 정확도를 개선하였다.

NCF[13], BERT4Rec[14] 등 딥러닝 기반 추천 모델은 높은 정확도를 보이나, 수백만 파라미터와 역전파 연산으로 인해 STB급 디바이스에서의 실시간 학습에는 부적합하다.

반면, MF는 파라미터 수가 (사용자 수 + 아이템 수) × 임베딩 차원으로 제한되고, SGD 업데이트가 단순 벡터 연산으로 구성되어 3GB RAM 환경에서도 증분 학습이 가능하다. 따라서 본 연구는 연산 효율성과 학습 가능성을 고려하여 MF 모델을 채택하였다.

3. ARM NEON 및 SIMD 최적화

ARM NEON은 128비트 SIMD 확장 명령어로, 4개의 float32 또는 16개의 int8 값을 동시에 처리 가능하다[15]. Lee et al.[16]은 ARM NEON SIMD 벡터 레지스터의 활용을 극대화하여 CNN 추론 속도를 2.66배 향상시켰으며, XNNPACK[17]은 ARM NEON 기반 고효율 추론 연산자를 제공한다. 그러나 기존 연구는 추론 최적화에 집중되어 있으며, 온디바이스 학습의 NEON 최적화 사례는 드물다.

4. 기존 온디바이스 프레임워크와의 비교

표 1은 기존 온디바이스 AI 프레임워크와 본 연구의 제안 시스템을 STB급 저사양 환경에서의 경량 추천 모델 학습 적합성 관점에서 비교한 결과이다. TFLite(LiteRT)와 ONNX Runtime은 범용 온디바이스 학습을 지원하나, 런타임 오버헤드가 크고 STB급 저사양 환경에 대한 특화 최적화가 부족하다. 특히, 표 7의 TFLite 비교 실험에서 확인되듯이 TFLite는 동일 STB에서 기준 대비 0.62배로 오히려 느린 성능을 보였으며, 바이너리 크기도 16배 증가하였다. 본 시스템은 STB 환경의 ARM NEON SIMD를 직접 활용한 학습·추론 통합 최적화를 제공한다는 점에서 차별화된다.

Ⅲ. 시스템 설계

1. 대상 하드웨어 환경 분석

본 연구의 대상 플랫폼은 상용 Android TV STB (K1200UA)로, 주요 하드웨어 사양은 표 2와 같다[18].

표 2. 대상 STB 하드웨어 사양
Table 2. Target STB Hardware Specifications

Category	Specification
Processor	Amlogic S905X4, ARM Cortex-A55 Quad, NEON 128-bit, NPU: None
Memory	DDR4 64bit, 3GB
Cooling	Fanless, Natural convection

ARM Cortex-A55는 저전력 고효율 코어로, NEON SIMD 128비트 벡터 레지스터를 지원한다. 총 3GB RAM 중 Android TV OS 및 미들웨어가 약 1.52GB를 점유하므로, AI 학습에 할당 가능한 메모리는 500MB~1GB 수준이다. 팬리스 설계로 CPU 온도가 65°C를 초과하면 DVFS

표 1. 온디바이스 AI 프레임워크 비교
Table 1. Comparison of On-Device AI Frameworks

Comparison Item	TFLite (LiteRT)	ONNX Runtime	Proposed System
On-Device Training	Supported (general-purpose); High runtime overhead for lightweight MF	Supported (since 2023, artifact-based); High runtime overhead for lightweight MF	Fully Supported
ARM NEON Optimization	Automatic (via XNNPACK delegate)	Automatic (built-in kernels)	Fully Supported (BPR-MF incremental training)
Quantization Scheme	PTQ / QAT (INT8/FP16)	PTQ / QAT inference (INT8/FP16)	PTQ INT8 (symmetric, NEON-optimized)
Memory Footprint	Medium (~50MB binary+runtime)	High (~80MB binary+runtime)	Minimal (~2MB engine)
STB-Specific Optimization	Not specialized (general mobile)	Not specialized (general purpose)	Specialized (thermal and memory control)
Recommendation Model Training	Unsupported	Unsupported	BPR-MF-specific optimization

(Dynamic Voltage and Frequency Scaling)에 의한 성능 저하가 발생한다.

2. 시스템 아키텍처

그림 1은 Android TV Framework와 OnDevice AI Framework 간의 연동 구조를 나타내며, 주요 컴포넌트로 Adaptive Resource Manager, Inference Optimizer, On-device Learner가 있다. 데이터 처리 흐름은 DATA → LEARNING → QUANTIZATION → INFERENCE의 순차적 파이프라인을 따른다.

3. 데이터 파이프라인

온디바이스 학습을 위한 데이터 파이프라인은 시청 로그 수집, 피처 엔지니어링, 학습 데이터 생성의 3단계로 구성된다. BPR(Bayesian Personalized Ranking) 학습을 위한 트리플렛 (user, positive_item, negative_item) 데이터를 생성하며, 부정 샘플 비율은 긍정 샘플당 5개로 설정하였다. 모든 처리 과정은 STB 내부에서 수행되어 사용자 데이터가 외부로 전송되지 않는다.

Ⅳ. NEON 최적화 기반 경량 학습 엔진

1. Matrix Factorization 알고리즘

1.1 BPR 손실 함수

본 연구에서는 암시적 피드백 데이터에 적합한 BPR 손실 함수를 사용한다[19]. BPR 손실 함수는 식 (1)과 같이 정의된다:

$$L_{BPR} = -\sum_{(u,i,j)\in D} \ln \sigma(\hat{x_{ui}} - \hat{x_{uj}}) + \lambda(\|Uvert|^2 + \|Vvert|^2) \quad (1)$$

여기서 $\hat{x_{ui}} = U_u \cdot V_i$는 사용자 u와 아이템 i의 예측 점수(내적), D는 트리플렛 집합, σ는 시그모이드 함수, λ는 L2 정규화 계수, U와 V는 사용자 및 아이템 임베딩 행렬이다.

1.2 SGD(Stochastic Gradient Descent) 기반 학습 알고리즘

확률적 경사 하강법(SGD)을 사용하여 임베딩을 업데이트한다. 각 트리플렛 (u, i, j)에 대해 식 (2)~(4)와 같이 업데이트한다:

$$U_u \leftarrow U_u + \eta \cdot (e_{uij} \cdot (V_i - V_j) - \lambda \cdot U_u) \quad (2)$$

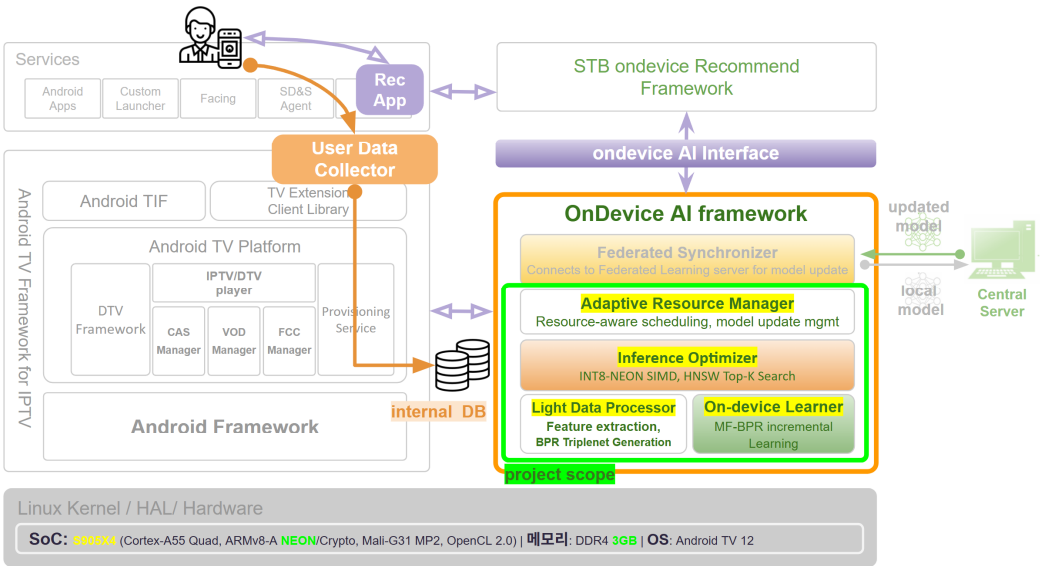

그림 1. STB 온디바이스 추천 프레임워크 아키텍처
Fig. 1. STB On-Device Recommendation Framework Architecture

$$V_i \leftarrow V_i + \eta \cdot (e_{uij} \cdot U_u - \lambda \cdot V_i) \quad (3)$$

$$V_j \leftarrow V_j + \eta \cdot (-e_{uij} \cdot U_u - \lambda \cdot V_j) \quad (4)$$

여기서 $e_{uij} = \sigma(\widehat{x_{uj}} - \widehat{x_{ui}})$이며, η는 학습률로 Step Decay 스케줄링을 적용한다.

1.3 BPR-MF 학습 알고리즘

그림 2는 NEON 최적화가 적용된 BPR-MF 학습 알고리즘의 의사코드를 나타낸다.

2. NEON SIMD 최적화

ARM NEON은 128비트 SIMD 확장으로, float32는 4개, INT8은 16개 값을 동시 처리한다. 임베딩 벡터 간 내적 연산은 vld1q_f32, vmlaq_f32 등 NEON intrinsics를 활용하여 구현하였으며, SGD 업데이트 연산에도 NEON을 적용하여 4개 요소를 동시에 업데이트한다. 임베딩 차원은 64로 설정하여 16바이트 경계 정렬 및 NEON 벡터 연산 효율을 극대화하였다.

3. 리소스 모니터링 및 적응형 제어

팬리스 STB에서 장시간 학습 시 발열 관리가 중요하다. CPU 온도는 Linux sysfs 인터페이스를 통해 모니터링하며, 학습 루프에서 리소스 상태를 주기적(1,000 샘플마다)으로 확인한다. 온도가 임계값(65°C)을 초과하거나 가용 메모리가 임계값(200MB) 미만인 경우 학습을 일시 중지하고, 온도가 재개 임계값(55°C)으로 하락하면 학습을 재개한다. 학습 중단에 대비하여 에폭마다 체크포인트를 저장한다.

V. INT8 양자화 추론 파이프라인

1. Post-Training Quantization (PTQ)

1.1 PTQ INT8 변환 절차

학습 완료된 Float32 임베딩을 INT8로 양자화하여 메모리 사용량 75% 감소 및 추론 속도 향상을 달성한다. 본 연구에서는 대칭 양자화(symmetric quantization)를 적용하여 zero-point를 0으로 고정함으로써 NEON SIMD 연산 시 offset 연산을 제거하였다. 양자화 절차는 최대 절대값 수집, Scale 계산, 양자화 순으로 진행되며, 식 (5)~(6)과 같이 정의된다:

$$scale = \frac{max\,|x_i|}{127} \quad (5)$$

$$x_q = clamp(round(\frac{x}{scale}), -127, 127) \quad (6)$$

INT8 양자화된 임베딩 간 내적은 정수 연산으로 수행되

$Input: D = \{(u,i,j)\}, \eta, \lambda, E, d = 64 \mid Output: U, V$

1 : $Initialize\ U, V (Xavier, 16-byte\ aligned)$
2 : for $epoch = 1\ to\ E\ do$
3 : for $each\ (u,i,j) \in D\ do$
4 : $\widehat{x_{ui}},\ \widehat{x_{uj}} \leftarrow Dot\,Product\ NEON(U_u,\ V_i),\ Dot\,Product\ NEON(U_u,\ V_j)$
5 : $e_{uij} \leftarrow \sigma(\widehat{x_{uj}} - \widehat{x_{ui}})$
6 : $Update\ U_u, V_i, V_j\ via\ Vector\,Update\ NEON$
7 : end for
8 : $Check\ resource\ status\ (pause\ if\ T > 65°C)$
9 : end for
10 : $return\ U, V\}$

Note: NEON acceleration indicates SIMD-optimized implementation of the corresponding operations.

그림 2. BPR-MF 학습 알고리즘 (ARM NEON SIMD 최적화)
Fig. 2. BPR-MF Training with ARM NEON SIMD Optimization

며, NEON INT8 내적으로 16개 값을 동시에 처리하여 추가 속도 향상을 달성한다.

2. Top-K 추천 생성

대규모 아이템 집합에서 Top-K 검색 시, HNSW(Hierarchical Navigable Small World) 인덱스를 사용하여 검색 속도를 향상시킨다[20]. HNSW 파라미터로 M=16, ef_construction=200, ef_search=50을 설정하였다. 아이템 수 5,000개 미만은 Brute-force, 50,000개 이상은 HNSW를 권장한다.

Ⅵ. 실험 및 평가

1. 실험 환경

1.1 하드웨어 및 소프트웨어

본 연구의 모든 실험은 실제 상용 STB 환경에서 수행되었다. 표 3은 실험에 사용된 하드웨어 및 소프트웨어 환경을 나타낸다.

표 3. 실험 환경
Table 3. Experimental Environment

Category	Specification
Hardware	IPTV STB, Cortex-A55 Quad (ARMv8.2-A), NEON 128-bit, 3GB DDR4
Software	Android TV 12, NDK r23 (armeabi-v7a), Clang -O3

본 실험에서는 추천 시스템 연구에서 널리 사용되는 MovieLens 데이터셋[21]을 사용하였다. ML-1M은 6,040명의 사용자가 3,706개 영화에 대해 부여한 1,000,209개의 평점으로 구성되며(GroupLens, 2003), ML-10M은 69,878명의 사용자가 10,681개 영화에 대해 부여한 10,000,054개의 평점으로 구성된다(GroupLens, 2009). 표 4는 데이터셋의 통계를 나타낸다.

표 4. 데이터셋 통계
Table 4. Dataset Statistics

Dataset	Users	Items	Interactions	Train Samples	Test Users
ML-1M	6,040	3,706	1,000,209	517,205	6,038
ML-10M	69,878	10,681	10,000,054	5,116,172	68,368

데이터 전처리 과정에서 암시적 피드백 형태로 변환하였으며, 평점 4.0 이상을 긍정적 상호작용으로 간주하였다. 학습/테스트 분할은 시간순 80:20 비율로 수행하였다.

2. 학습 성능 평가

표 5는 STB에서의 온디바이스 학습 성능 및 메모리 사용량을 나타낸다.

표 5. 온디바이스 학습 성능 및 메모리 사용량
Table 5. On-Device Training Performance and Memory Usage

Dataset	Epochs	Time/Epoch	Total	Throughput	VmRSS
ML-1M	40	18sec	12min	28.7K/sec	38MB
ML-10M	20	165sec	55min	31.0K/sec	711MB

ML-1M 데이터셋의 경우 약 12분, ML-10M의 경우 약 55분의 학습 시간이 소요되었다. 이는 STB의 대기 모드 시간에 충분히 수행 가능한 수준이다.

3. 추론 성능 평가

표 6은 다양한 최적화 기법의 추론 및 검색 성능을 비교한 결과이다. 표에서 '-'로 표기된 항목은 해당 실험이 다른 단계의 성능을 측정하기 위한 것임을 나타낸다.

표 6의 속도 향상(Speedup) 배율은 NEON SIMD를 사용하지 않은 순수 C++ Float32 구현(Float32(base))을 기준(1.00×)으로 측정하였다. INT8 양자화와 NEON SIMD 최적화를 결합한 경우, ML-1M에서 1.83배, ML-10M에서 2.19배의 속도 향상을 달성하였다.

추론 성능 측면에서, INT8+NEON 방식은 ML-1M에서 0.56ms(1.83배), ML-10M에서 1.69ms(2.19배)를 달성하여, INT8 양자화와 NEON SIMD의 결합이 두 데이터셋 모두에서 가장 우수한 추론 성능을 보였다. 검색 성능 측면에서, HNSW는 ML-1M에서 0.30ms(3.80배), ML-10M에서

표 6. 추론 및 검색 성능 비교
Table 6. Inference and Search Performance Comparison

Data	Method	Infer. (ms)	Speedup	Search (ms)	S. Speedup
ML-1M	Float32(base)	1.03	1.00x	-	-
ML-1M	INT8 Seq.	0.96	1.07x	-	-
ML-1M	INT8+NEON	0.56	1.83x	-	-
ML-1M	Brute-force	-	-	1.15	1.00x
ML-1M	HNSW	-	-	0.30	3.80x
ML-10M	Float32(base)	3.70	1.00x	-	-
ML-10M	INT8 Seq.	3.07	1.20x	-	-
ML-10M	INT8+NEON	1.69	2.19x	-	-
ML-10M	Brute-force	-	-	4.14	1.00x
ML-10M	HNSW	-	-	0.37	11.22x

0.37ms(11.22배)를 달성하였으며, 특히 아이템 수가 약 3배 많은 ML-10M에서 Brute-force 대비 11배 이상의 속도 향상을 보여 대규모 아이템 환경에서의 효용성을 입증하였다.

전체 추천 파이프라인의 End-to-End(E2E) 지연시간은 사용자 임베딩 조회, INT8 양자화 내적 기반 추론, Top-K 검색의 3단계를 포함하며, 측정 시작 지점은 사용자 ID 입력 시점, 종료 지점은 Top-K 추천 결과 반환 시점이다. 측정은 warm-up 10회를 제외한 100회 반복의 평균값으로 산출하였다. E2E 지연시간은 ML-1M에서 0.86ms, ML-10M에서 2.06ms로 실시간 서비스에 적합한 성능을 확인하였다.

3.1 TFLite 프레임워크 비교

동일 STB 환경에서 TFLite를 활용한 배치 MatMul 가속 효과를 비교 검증하였다. 표 7은 동일 모델(1,000 users × 4,867 items, dim=16)에 대한 500회 반복 측정 결과이다.

TFLite는 STB 환경에서 기준 대비 0.62배로 오히려 느린 성능을 보였다. 이는 (1) TFLite Interpreter의 API 호출 오버헤드, (2) STB의 32-bit ARM 모드에서 64-bit 최적화 경로 미사용, (3) 임베딩 차원이 작아 TFLite 오버헤드가 연산

표 7. TFLite 비교 실험 결과 (동일 STB)
Table 7. TFLite Comparison Results (Same STB)

Method	Inference Time (ms)	Speedup	Binary Size
Float32 (base)	0.332	1.00×	104KB
INT8+NEON (Proposed)	0.319	1.04×	104KB
Batch Sequential (Proposed)	0.229	1.45×	104KB
TFLite Batch MatMul	0.534	0.62×	1.7MB

시간을 초과, (4) 바이너리 크기 16배 증가(104KB → 1.7MB)에 기인한다. 반면, 제안 시스템의 Batch Sequential 방식은 1.45배 속도 향상을 달성하여 경량 직접 구현의 우수성을 확인하였다.

4. 추천 정확도 평가

4.1 베이스라인 비교

표 8은 제안 방법과 기존 협업 필터링 방법들의 추천 정확도 및 K값에 따른 성능을 비교한 결과이다.

제안하는 MF-BPR 모델은 모든 평가 지표에서 베이스라인 방법들을 상회하였다. Float32 대비 INT8 양자화로 인한

표 8. 추천 정확도 종합 비교 (MovieLens 1M)
Table 8. Comprehensive Recommendation Accuracy Comparison (MovieLens 1M)

Method (K=10)	Precision	Recall	NDCG	Hit Rate	MRR
Popularity	6.68%	4.17%	7.53%	34.20%	15.29%
User-CF	7.99%	6.93%	9.76%	43.77%	18.21%
Item-CF	8.01%	6.34%	9.62%	42.71%	19.01%
MF-BPR (Float32)	8.33%	7.48%	9.95%	46.94%	20.02%
MF-BPR (Proposed)	8.33%	7.47%	9.96%	46.88%	20.04%

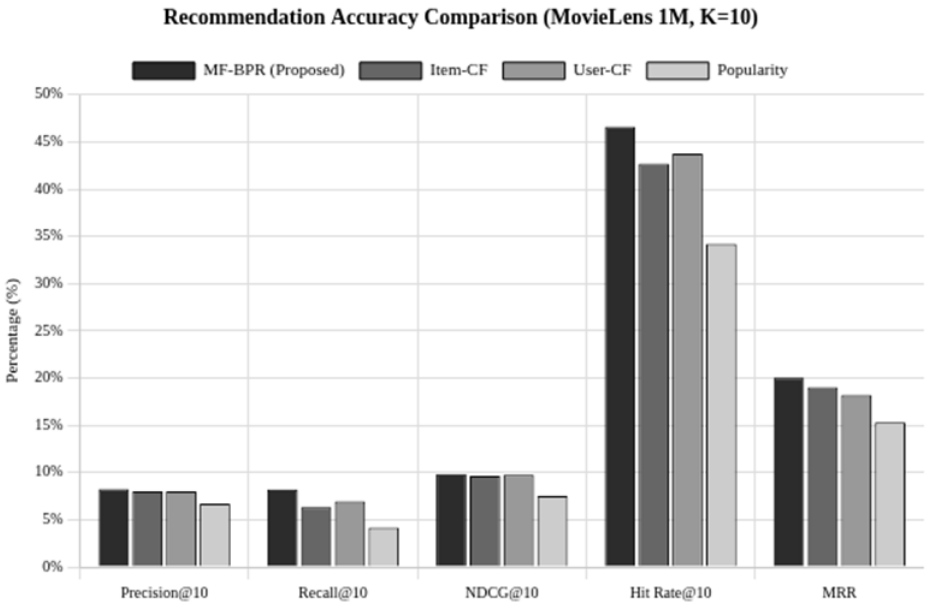

그림 3. 추천 정확도 비교 (MovieLens 1M, K=10)
Fig. 3. Recommendation Accuracy Comparison (MovieLens 1M, K=10)

평균 오차율은 0.1% 미만으로, 추천 정확도에 실질적인 영향이 없음을 확인하였다. 그림 3은 MovieLens 1M 데이터셋의 K=10 조건에서 제안 방법과 베이스라인 방법들의 주요 추천 정확도 지표를 시각적으로 비교한 결과를 보여준다.

4.2 K값에 따른 성능 변화

표 9는 다양한 K값에 따른 제안 시스템(MF-BPR)의 성능 변화를 나타낸다. 실험은 동일 STB에서 수행하였다. 표 10은 ML-10M 데이터셋에서 동일한 조건으로 측정한 K값별 추천 성능 변화를 나타낸다.

표 9. K값에 따른 추천 성능 변화 (MovieLens 1M)
Table 9. Recommendation Performance by K Value (MovieLens 1M)

Metric	@5	@10	@20
Precision	8.83%	8.33%	7.46%
Recall	4.13%	7.47%	12.87%
NDCG	9.30%	9.96%	11.42%
Hit Rate	31.22%	46.88%	63.16%
MRR	18.03%	20.04%	21.12%

K값이 증가함에 따라 Precision은 감소하고 Recall과 Hit Rate는 증가하는 전형적인 trade-off 관계를 보인다. NDCG는 K=20에서 11.42%로 가장 높았으며, K값 증가에 따라 순위 품질이 점진적으로 향상됨을 확인하였다. Hit Rate는 K=20에서 63.16%까지 증가하여 더 많은 사용자에게 관련 아이템이 노출됨을 확인하였다. 실용적 관점에서 STB UI의 추천 슬롯 수(일반적으로 5~10개)를 고려하면, K=10이 Precision과 Recall 간 균형점으로 적합하다.

표 10. K값에 따른 추천 성능 변화 (MovieLens 10M)
Table 10. Recommendation Performance by K Value (MovieLens 10M)

Metric	@5	@10	@20
Precision	7.96%	7.23%	6.32%
Recall	4.31%	7.75%	13.12%
NDCG	8.65%	9.24%	10.79%
Hit Rate	28.24%	41.38%	55.54%
MRR	16.76%	18.50%	19.45%

ML-10M에서도 동일한 경향을 확인하였으며, 데이터셋 규모가 약 10배 증가했음에도 Hit Rate@10은 46.88%에서 41.38%로 5.50%p 감소(상대 감소율 11.7%)하는 데 그쳐 성능 하락이 제한적임을 확인하였다.

표 11. 적응형 리소스 제어 실험 결과
Table 11. Adaptive Resource Control Experiment Results

Category	Exp. 1	Exp. 2
Dataset / Epochs	ML-1M, 10 epochs	ML-10M, 20 epochs
Dim / Temp. Threshold	64D / 60°C (low)	128D / 65°C (default)
Training Time	493s (w/ Pause)	965s
Pause Count/Duration	2 / 465s	0 / 0s
Max CPU Temp.	60.0°C	61.5°C
Final Loss	0.166	0.089

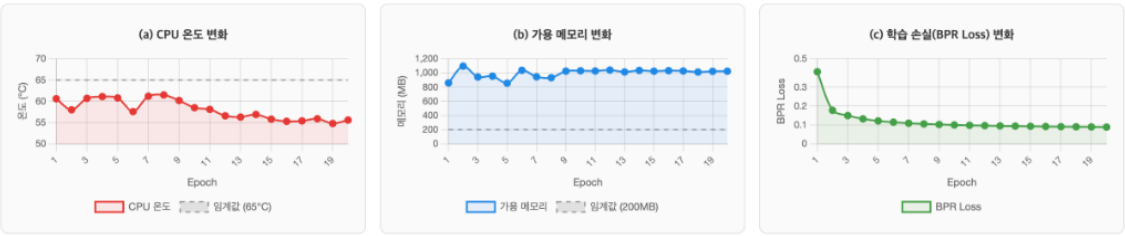

그림 4. ML-10M 학습 중 리소스 및 손실 변화
Fig. 4. Resource and Loss Changes During ML-10M Training

5. 적응형 리소스 제어 평가

표 11은 적응형 리소스 제어의 동작을 검증하기 위한 2단계 실험 결과를 나타낸다. 적응형 리소스 제어의 기능 및 효과를 검증하기 위해 2단계 실험을 수행하였다. 실험 1에서는 임계값을 의도적으로 낮추어 Pause/Resume 동작을 검증하고, 실험 2에서는 기본 임계값으로 대용량 데이터셋 학습의 안정성을 확인하였다.

실험 1에서 온도 임계값(60°C) 초과 시 학습이 자동 중지되어 약 460초간 냉각 후 재개되었으며, Pause 발생에도 Loss가 정상적으로 수렴(0.166)하였다. 실험 2에서는 기본 임계값(65°C/200MB)으로 ML-10M 규모 학습 시에도 Pause 없이 안정적으로 동작하였다. 그림 4는 ML-10M 데이터셋 학습 과정에서의 CPU 온도, 메모리 사용률 및 학습 손실(Loss)의 시계열 변화를 나타낸다.

Ⅶ. 결 론

본 논문에서는 NPU가 없는 저사양 Android TV STB 환경에서 행렬분해 기반 추천 모델의 온디바이스 학습 및 추론 시스템을 제안하고, 상용 STB(K1200UA)에서의 실증을 통해 그 실현 가능성을 검증하였다. 대상 STB는 GPU (Mali-G31)를 탑재하나 임베딩 내적 연산은 GPU 오버헤드 대비 연산량이 작아 ARM NEON SIMD가 사실상 유일한 현실적 가속 방안이다.

본 연구의 핵심 기여는 다음과 같다. 첫째, ARM NEON SIMD와 INT8 양자화를 결합하여 순수 C++ 대비 최대 2.19배의 추론 속도 향상을 달성하였다. 둘째, HNSW 기반 Top-K 검색으로 11.2배 속도 향상과 Recall 100%를 동시에 확보하였다. 셋째, 적응형 리소스 제어를 통해 팬리스 STB의 열 관리 제약 하에서도 안정적인 학습을 실현하였다. 이를 통해 End-to-End 추천 지연시간 0.86ms, Hit Rate@10 46.88%의 성능을 달성하였다.

본 시스템은 사용자 시청 데이터가 서버로 전송되지 않아 프라이버시가 보호되며, 기존 STB에서 소프트웨어 업데이트만으로 적용 가능하다. 다만, MF 기반 접근법의 콜드 스타트 문제와 단일 스레드 SGD의 한계가 존재한다. 또한, 본 연구는 단일 STB 모델(K1200UA, Cortex-A55)에서 검증되었으므로, 향후 Cortex-A73/A76 등 다양한 ARM 코어 기반 디바이스에서의 성능 검증을 통해 일반화 가능성을 확인할 필요가 있다. 향후 메타데이터 임베딩 기반 하이

브리드 모델[22], Federated Learning을 통한 분산 학습, 실제 IPTV 시청 로그를 활용한 현장 검증을 계획하고 있다.

참 고 문 헌 (References)

[1] Ministry of Science and ICT, Korea Communications Commission, and KISDI, “2024 Broadcasting Industry Survey Report,” 2024. https://www.kisdi.re.kr/report/view.do?key=m2101113024153&masterId=3934581&arrMasterId=3934581&artId=1812716 (accessed Mar. 11, 2026)

[2] K. Zou and A. Sun, “A Survey of Real-World Recommender Systems: Challenges, Constraints, and Industrial Perspectives,” arXiv:2509.06002, 2025.
doi: https://doi.org/10.48550/arXiv.2509.06002

[3] Google, “LiteRT (formerly TensorFlow Lite) Documentation,” https://ai.google.dev/edge/litert (accessed Mar. 11, 2026).

[4] Apple, “Core ML Documentation,” https://developer.apple.com/documentation/coreml (accessed Mar. 11, 2026)

[5] S. Han, H. Mao, and W. J. Dally, “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding,” *Proceedings of International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.
doi: https://doi.org/10.48550/arXiv.1510.00149

[6] B. Jacob et al., “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,” *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, pp. 2704-2713, June 2018.
doi: https://doi.org/10.1109/CVPR.2018.00286

[7] H. B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, USA, pp. 1273-1282, April 2017.

[8] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2nd ed., Springer, New York, pp. 1-35, 2015.
doi: https://doi.org/10.1007/978-1-4899-7637-6

[9] Y. Koren, “The BellKor Solution to the Netflix Grand Prize,” Netflix Prize Documentation, 2009, https://www2.seas.gwu.edu/~simhaweb/champalg/cf/papers/KorenBellKor2009.pdf (accessed Mar. 11, 2026).

[10] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative Filtering for Implicit Feedback Datasets,” *Proceedings of IEEE International Conference on Data Mining (ICDM)*, Pisa, Italy, pp. 263-272, December 2008.
doi: https://doi.org/10.1109/ICDM.2008.22

[11] E. Kim, S. J. Pyo, and M. Kim, “Automatic Recommendation of (IP)TV Programs Based on a Rank Model Using Collaborative Filtering,” *Journal of Broadcast Engineering*, Vol. 14, No. 2, pp. 238-252, 2009.
doi: https://doi.org/10.5909/JBE.2009.14.2.238

[12] S. Oh, Y. Oh, S. Han, and H. J. Kim, “Broadcast Content Recommender System Based on User's Viewing History,” *Journal of Broadcast Engineering*, Vol. 17, No. 1, pp. 129-139, 2012.
doi: https://doi.org/10.5909/JEB.2012.17.1.129

[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, “Neural Collaborative Filtering,” *Proceedings of International World Wide Web Conference (WWW)*, Perth, Australia, pp. 173-182, April 2017.
doi: https://doi.org/10.1145/3038912.3052569

[14] F. Sun et al., “BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer,” *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, Beijing, China, pp. 1441-1450, November 2019.
doi: https://doi.org/10.1145/3357384.3357895

[15] ARM, “ARM NEON Programmer's Guide Version 1.0,” 2013, https://developer.arm.com/documentation/den0018/latest (accessed Mar. 11, 2026).

[16] S. J. Lee, S. S. Park, and K. S. Chung, “Efficient SIMD Implementation for Accelerating Convolutional Neural Network,” *Proceedings of the 4th International Conference on Communication and Information Processing (ICCIP)*, Qingdao, China, pp. 174-178, November 2018.
doi: https://doi.org/10.1145/3290420.3290444

[17] Google, “XNNPACK: High-efficiency Floating-point Neural Network Inference Operators,” GitHub, 2024, https://github.com/google/XNNPACK (accessed Mar. 11, 2026)

[18] Amlogic, “S905X4 Datasheet Rev.03,” 2021, https://www.amlogic.com/ (accessed Mar. 11, 2026)

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian Personalized Ranking from Implicit Feedback,” *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, Montreal, Canada, pp. 452-461, June 2009.

[20] Y. Malkov and D. Yashunin, “Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 42, No. 4, pp. 824-836, April 2020.
doi: https://doi.org/10.1109/TPAMI.2018.2889473

[21] F. M. Harper and J. A. Konstan, “The MovieLens Datasets: History and Context,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, Vol. 5, No. 4, Article 19, December 2015.
doi: https://doi.org/10.1145/2827872

[22] M. Kula, “Metadata Embeddings for User and Item Cold-start Recommendations,” arXiv:1507.08439, 2015.
doi: https://doi.org/10.48550/arXiv.1507.08439

저 자 소 개

김 현 수

- 2023년 : 호서대학교 융합공학과 박사수료
- 2025년 ~ 현재 : 가온그룹 개발사업본부
- ORCID : https://orcid.org/0009-0002-8517-1353
- 주관심분야 : 추천시스템, 온디바이스 AI, 정보보안

문 남 미

- 1998년 : 이화여자대학교 컴퓨터공학부 박사
- 2008년 ~ 현재 : 호서대학교 컴퓨터공학과 교수
- ORCID : https://orcid.org/0000-0003-2229-4217
- 주관심분야 : Social Learning, 빅데이터 처리 및 분석, HCI, 메타데이터, User Centric data analysis