https://doi.org/10.5392/JKCA.2020.20.10.715

DNN 모델을 이용한 기계 학습 기반 k-최근접 질의 처리 최적화 기법

k-NN Query Optimization Scheme Based on Machine Learning Using a DNN Model

위지원*, 최도진*, 이현병*, 임종태*, 임헌진*, 복경수**, 유재수*
충북대학교 정보통신공학부*, 원광대학교 SW융합학과**

Ji-Won We(upsidejiwon@cbnu.ac.kr)*, Do-Jin Choi(mycdj91@cbnu.ac.kr)*,
Hyeon-Byeong Lee(lhb@cbnu.ac.kr)*, Jong-Tae Lim(jtlim@cbnu.ac.kr)*,
Hun-Jin Lim(limhunjin@netsgo.com)*, Kyoung-Soo Bok(ksbok@wku.ac.kr)**,
Jae-Soo Yoo(yjs@cbnu.ac.kr)*

요약

본 논문에서는 고차원의 특징 벡터에서 질의와 가장 가까운 k개의 데이터를 찾는 k-최근접 질의 최적화 방법을 제안한다. k-최근접 질의는 k개의 데이터를 포함할 가능성이 있는 범위를 기반으로 범위 질의로 변환되어 처리하는 기법이다. 본 논문에서는 처리 비용을 감소시키고 검색 속도를 가속화 할 수 있는 최적의 범위를 도출하기 위해 k-최근접 질의 처리 시 DNN 모델을 이용한 최적화 기법을 제안한다. 제안하는 기법은 온라인 모듈과 오프라인 모듈로 구성된다. 온라인 모듈에서는 클라이언트로부터 요청을 받아 실제 질의를 처리한다. 오프라인 모듈에서는 과거 최적화 기법의 결과를 학습 로그로 사용한 DNN 모델로 최적의 범위를 도출하고 온라인 모듈로 전달한다. 제안하는 기법의 우수성 및 타당성의 입증을 위하여 다양한 성능 평가를 수행한다.

■ 중심어 : | 기계 학습 | K-최근접 질의 처리 | DNN 모델 | 분산 처리 | Spark |

Abstract

In this paper, we propose an optimization scheme for a k-Nearest Neighbor(k-NN) query, which finds k objects closest to the query in the high dimensional feature vectors. The k-NN query is converted and processed into a range query based on the range that is likely to contain k data. In this paper, we propose an optimization scheme using DNN model to derive an optimal range that can reduce processing cost and accelerate search speed. The entire system of the proposed scheme is composed of online and offline modules. In the online module, a query is actually processed when it is issued from a client. In the offline module, an optimal range is derived for the query by using the DNN model and is delivered to the online module. It is shown through various performance evaluations that the proposed scheme outperforms the existing schemes.

■ keyword : | Machine learning | K-Recent Query Processing | DNN Model | Distributed Processing | Spark |

* 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원과(No. 2019R1A2C2084257) 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원과(No.B0101-15-0266, 실시간 대규모 영상 데이터 이해·예측을 위한 고성능 비주얼 디스커버리 플랫폼 개발) 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술 개발사업의 지원을 받아 수행된 연구임(No. NRF-2017M3C4A7069432).

접수일자 : 2020년 07월 30일
수정일자 : 2020년 08월 31일
심사완료일 : 2020년 08월 31일
교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

I. 서론

최근 실시간 영상 처리 기술의 발달로 영상 내의 객체를 인식하고 검색하는 기술이 연구되고 있다. 이러한 기술들은 영상의 객체를 수집하여 처리하고 다양한 목적을 가지고 사용된다. 정확한 객체 인식을 위해서는 객체가 가지는 특징 데이터로 이미지를 찾거나 두 이미지 간의 유사도를 검색하는 기술이 효과적으로 이루어져야 한다. 이미지 검색은 크게 의미 기반(Semantic-based) 검색과 내용 기반(Content-based) 검색으로 나뉜다. 의미 기반 이미지 검색은 태그의 키워드나 웹 사이트의 텍스트를 이용하여 검색한다. 내용 기반 이미지 검색은 객체가 가지고 있는 특징들(예를 들어 색상, 굴곡 등)을 이용하여 객체를 검색하는 기술을 일컫는다. 특징은 한 가지만 사용하여 객체를 검색할 수도 있고 여러 가지 특징을 혼합하여 사용할 수도 있다. 두 이미지의 유사도 검색은 한 객체를 지속해서 추적하기 위해서도 필요하다. 본 논문에서는 내용 기반 이미지 검색에서의 k-최근접 질의를 처리한다. 이미지는 N차원의 벡터 데이터로 변환되어 거리공간에 표현된다. 질의점으로 N차원의 데이터가 한 개가 들어왔을 경우, k-최근접 질의 처리는 질의점과 거리 계산을 통해 가장 가까운 k개의 결과를 찾는 질의이다. k-최근접 처리는 k개가 포함될 수 있을법한 특정 범위를 갖는 범위 질의로 변환되어 처리된다. 범위 질의는 주어진 범위 내에 존재하는 데이터를 찾는 질의이다. k-최근접 질의는 질의점과 범위 내에 존재하는 데이터 후보군과 거리를 계산하여 가장 가까이 존재하는 k를 모두 찾을 때까지 처리를 진행한다. 만약 범위 내에 k개를 모두 찾지 못한 경우 범위를 특정 방법에 따라 확장한 후 질의를 진행한다. 빠른 질의 처리와 처리 비용의 감소를 위해서는 k-최근접 질의를 위한 최적의 범위를 도출해내는 최적화 방법이 필요하다. 최적 범위의 도출은 검색을 수행해야 하는 후보군의 개수를 줄여 질의의 효율성을 증대시켜준다.

[9]에서는 kd-트리와 iDistance를 이용하여 하이브리드(Hybrid) 색인을 구축하였다. kd-트리는 k차원 의 데이터를 분할하는 기법으로 이진 탐색 트리를 다차원 공간으로 확장한 것이다. iDistance는 거리 기반 색인 기법으로 고차원 데이터를 1차원인 거리로 표현한 뒤 B+-Tree에 색인하는 기법이다. [9]에서는 두 자료구조의 장단점을 고려하여 결합한 하이브리드 인덱스를 구축하였다. 구축된 색인에서 k-최근접 질의 처리의 최적화를 위해 밀집도 및 질의 처리 비용을 기반으로 k-최근접을 위한 최적의 범위를 계산하는 방법을 제안하였다. 밀집도 기반의 최적화 기법은 데이터의 통계를 반영한다. 데이터 분포를 기반으로 한 밀집도 계산 결과를 초기 탐색 범위로 지정하고 질의를 진행한다. 탐색 비용 기반의 최적화 기법은 질의 처리 시간이 통계를 반영한다. 탐색 비용 기반은 질의 탐색의 횟수와 탐색 범위에 존재하는 후보군의 개수를 기반으로 질의처리 범위를 지속해서 변경하고 일정 값에 수렴하여 최적화한다.

본 논문에서는 효율적인 내용 기반 이미지 검색을 위해 k-최근접 질의 처리를 수행한다. 앞서 설명한 대로 내용 기반 이미지 검색은 객체의 특징으로 객체를 검색한다. N차원의 특징 벡터 간 거리 계산을 하여 가장 가까운 k개를 이미지 검색 결과로 출력한다. 거리가 가깝다는 것은 유사한 이미지라는 것을 의미한다. 질의 처리의 수행을 위하여 [9]에서 구축된 것과 같은 색인을 사용하고, 기계 학습을 적용한 k-최근접 최적화 기법을 제안한다. [9]과 같은 방법으로 스파크(Spark)를 이용하여 분산으로 하이브리드 색인을 구축한다. 제안하는 기법은 DNN(Deep Neural Network)[12] 모델을 이용하여 k-최근접 질의의 최적 범위를 도출하는 방법을 제안한다.

제안하는 방법은 데이터를 학습하여 범위를 도출하고 있기 때문에 데이터의 양이 적은 경우에도 효과적으로 k-최근접 질의를 수행할 수 있다. 지속적인 학습이 가능하기 때문에 다양한 이미지에 대해서도 효과적으로 검색을 수행할 수 있다. k-최근접 질의를 통한 이미지 검색은 이미지 검색뿐만 아니라 이미지를 k개의 그룹으로 분류하는데 추가 활용이 가능하다. 또한, 벡터 데이터를 분류하므로 이미지뿐만 아니라 일반 데이터도 가능하여 다양한 분야에서 활용도가 높다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 설명한다. 3절에서는 제안하는 기법의 특징 및 처리 방법에 관해 기술한다. 4절에서는 제안하는 기법의 타당성 및 우수성 검증을 위한 성능 평가를 수행한다. 마

지막으로 5절에서는 본 논문의 결론 및 향후 연구에 대해 제시한다.

II. 관련 연구

근접 데이터를 찾는 것은 다양한 방법을 이용하여 연구되고 있다[1]. 그 중 k-NN은 k개의 값을 주어 이웃한 데이터를 k를 찾아내는 것으로 활용 분야가 매우 다양한 기본적인 알고리즘이다[2-5]. k-NN은 굉장히 효과적인 방법이지만 데이터양 또는 차원의 수에 따라 비례하여 높아지는 처리 비용이 발생하는 문제점이 존재한다. k-최근접의 속도 및 정확도를 개선하기 위한 연구는 활발히 이루어지고 있다.

[6]에서는 k-최근접의 속도를 증가시키기 위하여 “jump method"를 제안하였다. 이를 위해 k-평균 알고리즘을 데이터에 적용하고 논문에서 제안하는 수식을 이용하여 클러스터를 생성한다. k-평균 알고리즘은 사용자가 k를 부여하면 k개의 중심을 가지게 되고, 각 데이터는 가장 가까운 중심에 할당된다.

[7]에서는 k-최근접과 DNN을 함께 이용하여 k-최근접의 문제점을 해결하였다. [7]에서는 Wi-Fi RSSI(Received Signal Strength Indicator) 기반의 지문을 이용한 실내 위치 파악의 개선이 목적으로 하고 있다. RSSI는 AP(Acees Poing)와 지문 학습 단계에 대한 의존성이 크다. 또한, 기존의 k-NN은 이 의존성을 해결하기 위하여 k-최근접 및 DNN을 이용한다. DNN은 지문 데이터 세트를 분류하는 데에 사용된다. 향상된 k-최근접 알고리즘은 인접한 AP의 수에 따라 가중치를 부여하여 기존 k-최근접이 인접 AP의 영향력을 무시하던 문제점을 해결하였다. 또한, 향상된 K-최근접 알고리즘은 분류된 데이터 세트를 기반으로 최종 위치를 결정한다.

[8]에서는 존재하는 침입 탐지 시스테믜 침입 감지 정확도를 해결하기 위하여 k-NN과 DNN을 이용하였다. “CICIDS-2017” 라는 네트워크 공격이 포함된 데이터세트에 대하여 분류 및 라벨링 작업을 통해 네트워크 공격 데이터를 분류한다. 기계학습은 k-최근접 알고리즘으로 수행하고 딥러닝은 DNN을 수행한 뒤 두 방법의 결과를 비교한다. 결과를 통해 일반 k-최근접 질의와 비교했을 때 DNN의 우수함을 알 수 있다. 본 논문 또한 k-최근접 질의의 성능을 증대시키기 위하여 DNN을 이용한다.

[9]에서는 k-최근접 질의 처리를 위해 밀집도와 처리 비용을 기반으로 한 두 가지 최적화 기법을 제안하였다. 밀집도 기반의 최적화 기법은 데이터의 분포 통계를 반영한다. 이는 k의 개수를 반영할 수 있다. 처리 비용 기반의 최적화 기법은 탐색 횟수와 탐색 후보군의 수를 반영하여 지속해서 범위를 갱신하고 가장 최적의 값에 수렴할 수 있도록 한다.

첫 번째 방법은 밀집도 기반의 최적화 기법이다. 데이터를 거리공간에 표시했을 때 두 데이터 사이의 최대 거리를 계산하고, 영역에 존재하는 데이터 수를 최대 거리로 나누어주면 DPR(Date Per Range)을 얻을 수 있다. 초기 탐색 범위 값을 얻기 위해서 k를 DPR로 나누어 주면 초기 탐색 범위를 얻을 수 있다. 이를 이용하여 k 값을 반영한 질의 범위를 얻을 수 있다.

두 번째 방법은 탐색 비용 기반의 최적화 기법이다. 범위 확장 및 축소를 위하여 임곗값을 지정한다. 초기에 범위는 이전 질의의 처리 시간을 기반을 두어 계속해서 변경되고 특정 값에 수렴한다. 처음에는 기존 iDistance의 질의 범위 할당과 같이 1%로 진행한다. 다음 질의부터는 임곗값을 이용하여 범위를 조절한다. 임곗값은 상승 지수 α(Increase factor)과 하강지수 β(Decrease factor)을 이용한다. 상승 지수 α는 질의 처리 탐색의 횟수를 반영한다. 질의 처리가 완료된 이후 1-NN의 탐색 범위를 2%로 지정한다. 이후 같은 1-NN이 입력되었을 때 탐색 범위 2%로 질의처리를 수행한다. 만약 탐색 횟수가 두 번 이상이라면 탐색 횟수만큼 상승 값을 추가 적용한다. 예를 들어 2% 범위 탐색에 두 번 추가 탐색 범위가 증가하였다면 다음 탐색 범위는 $2\%*(2*\alpha)$가 된다. 하강 값은 탐색 후보군의 개수를 반영한다. 탐색 범위가 처음 지정했던 2%로 끝났을 경우 탐색 후보군의 개수를 확인한다. 만약 후보군의 개수가 이전보다 많을 경우는 더 많은 탐색을 수행하였다는 의미이기 때문에 하강 값만큼 탐색 범위를 줄인다. 만약 2%의 탐색에 후보군이 이전보다 증가하였을 경우 $2\%*\beta$가 된다. 반복적으로 상승 지수와 하강

지수를 적절하게 이용하여 질의 범위를 최적의 값으로 수렴시킨다.

그러나 [9]에서 제안한 최적화 방법은 최적화 기법별로 문제점을 가지고 있다. 첫째, 밀집도 기반의 최적화 기법은 밀집도가 일치하는 경우에만 좋은 성능을 보이는 결과를 보였으며 평균적인 처리 시간은 매우 나쁜 것으로 나타났다. 둘째, 탐색 비용을 기반으로 하는 최적화 기법은 일정 최적화 값에 수렴하는 데 시간이 걸린다. 본 논문에서는 해당 문제점을 해결하기 위하여 기계 학습 모델을 이용한다. 이를 위해 DNN모델을 이용하며, [9]에서의 처리 과정과 비교했을 때, 비교적 빠르고 안정적인 결과를 도출하는 방법을 제안한다. 또한, [9]에서는 기존에 존재하는 데이터만 가지고 최적의 범위를 도출하기 때문에 새로운 데이터에 대한 추가 작업이 없다. 이에 반해 기계 학습을 이용한 방법은 지속해서 데이터를 학습할 수 있으므로 새로운 데이터에 대해서도 효과적으로 최적 범위를 도출할 수 있다.

Ⅲ. 제안하는 k-최근접 질의 처리 최적화 기법

1. 전체 구조

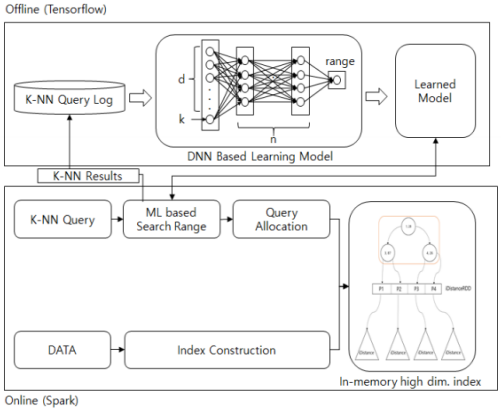

그림 1. 기계 학습을 이용한 질의 처리의 전체 구조

[그림 1]과 [그림 2]는 제안하는 기법의 전체 구조를 나타낸다. 전체 구조는 온라인과 오프라인으로 나뉜다. 온라인 구조에서는 분산 인-메모리 플랫폼인 스파크를 이용하여 [9]에서 구축한 인덱스 구조와 같게 kd-트리와 iDistance를 이용한 하이브리드 색인을 구축한다.

오프라인에서는 K-최근접 질의의 범위 최적화를 위한 DNN 모델의 학습이 진행된다. 학습을 위한 데이터를 저장할 로그를 작성하고 기계학습 모델을 구현한다. 기계 학습 모델은 기계학습 오픈소스 플랫폼 텐서플로우(Tensorflow)를 이용해 구현한다. 로그는 질의와 해당 질의에 대한 기존 최적화 기법으로 도출된 질의 범위가 정답 세트로 구성되어있다. 학습 모델은 오프라인에서 로그에 기록된 K-최근접 질의 처리 로그를 이용하여 충분히 학습한다.

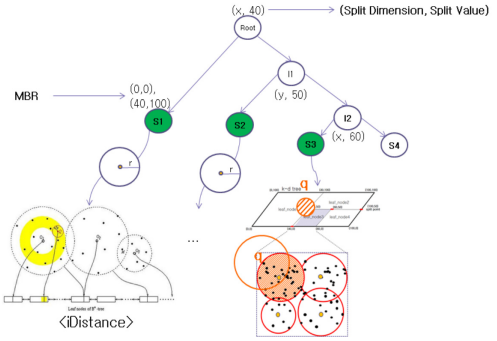

그림 2. [9]에서의 인덱스 구조

k-최근접 질의는 다음과 같이 처리된다. 모든 데이터는 데이터 간 거리 계산을 위해 거리 공간에 표현된다. [그림 3]은 설명을 위해 2차원 공간으로 표현하였다. 본 논문의 실제 질의점은 128차원의 벡터이다. 질의점 q와 k가 10으로 들어왔다고 가정할 때, 사전에 학습된 모델은 q와 k를 가지고 범위 100을 출력한다. 출력된

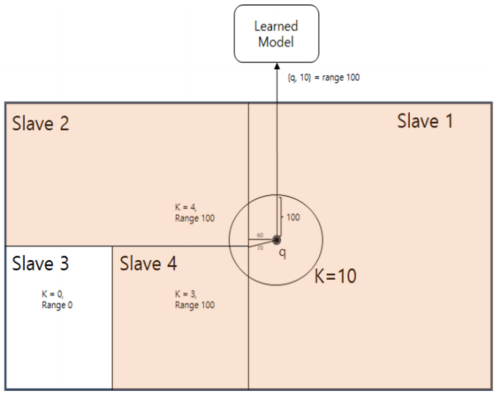

그림 3. k-최근접 질의 방법

범위 내에 존재하는 슬레이브 모델을 필터링하여 각 슬레이브에게 질의점과 k를 전달하여 결과 후보군을 도출할 수 있도록 한다. 또한 본 방법에서는 처리 비용을 줄이기 위해 각 노드마다 다른 k값을 부여한다. 거리가 먼 노드일수록 질의 결과를 포함할 확률이 적기 때문이다.

기계 학습 기반의 k-최근접 질의 처리 순서는 다음과 같다. K-최근접 질의처리가 들어왔을 때, 온라인에서는 DNN 모델을 기반으로 범위질의를 진행한다. k-최근접 질의는 k개가 포함될법한 범위를 계산하여 범위질의를 진행한다. DNN은 k-최근접 질의 요청 시 온라인이 클라이언트로부터 전달받은 k 값과 질의점을 입력 값으로 받는다. 입력받은 데이터와 DNN모델을 이용해 최적의 범위를 예측하여 결과로 전달한다. 온라인 모듈에서는 전달받은 범위를 이용하여 구축된 색인에 질의를 할당하고 최종 질의 결과를 도출한다.

2. DNN 모델을 사용한 범위 최적화

k-최근접 질의를 위한 범위 도출 최적화를 위해 학습 모델로는 심층 신경망인 DNN모델을 사용한다. DNN 모델은 입력층(input Layer)과 출력층(output Layer) 사이에 여러 개의 은닉층(hidden Layer)이 존재하는 모델이다. DNN 모델의 깊이는 N개이며 각 계층 간에 ReLu(Rectified Linear Unit)[13] 함수를 통해 가중치 값을 학습한다. ReLu함수는 기계학습의 은닉층을 활성화하는 함수로 사용된다. 이진 분류 활성화 함수 시그모이드(sigmoid)는 0 〈 n 〈 1의 사이 값만 다루던 것과 달리 ReLu함수는 0보다 작은 값은 0으로, 0보다 큰 값은 그대로 반환한다. 스파크를 이용하여 오프라인에서는 사전에 k-최근접 질의를 수행하면서 얻어낸 최적의 범위의 질의점, 탐색 범위, k 값 등을 기반으로 지도(supervised)학습을 진행한다. 학습을 위해 정답 세트와 예측 세트의 차이를 평균 제곱 오차인 MSE (Mean Squared Error)을 학습을 통해 지속해서 줄여나가는 것을 목표로 한다. 정답 세트는 기존의 처리 비용 및 밀집도 기반으로 도출한 최적 범위를 기록한 k-최근접 질의 로그를 이용한다. MSE는 식 (1)과 같이 계산되며, 값이 작을수록 정답 세트와 예측 세트의 오차가 적다는 것을 의미한다. 모델은 가장 작은 값의 MSE를 갖는 깊이 N을 학습모델의 깊이로 선택한다.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_y)^2 \quad (1)$$

Ⅳ. 성능 평가

1. 성능 평가 환경

본 절에서는 기계 학습을 위한 파라미터 실험과 제안하는 최적화 기법과 기존 최적화 기법을 비교하여 타당성을 제시하고 우수성을 검증한다. 성능 평가 환경은 [표 1]과 같다. Intel(R) Core(TM) i5-6400, 2.7GHz 4 Core 프로세서와 48G의 메인 메모리를 가진 컴퓨터 4개로 구성하여 분산 환경을 구현하였다. 운영체제는 CentOs7을 사용하였다. 파티션은 서버별로 2개를 할당하여 총 8개의 파티션을 생성하였다. 스파크는 2.3 버전을 사용하였으며 DNN 모델은 Tensorflow 2.0에서 구현되었다. 타당한 비교를 위하여 데이터는 [9]에서 사용한 데이터를 그대로 사용하였다. 데이터 집합은 SIFT(Scale-Invariant Feature Transform)[14] 알고리즘을 사용하여 이미지의 특징 벡터를 추출하였다. SIFT는 이미지의 크기나 회전에 불변하는 특징을 추출한다. 데이터의 각 차원은 0에서 255까지의 값을 가지며 128차원 데이터 100만 개의 데이터 집합을 사용하였다.

표 1. 성능 평가 환경

이름	값
CPU	Intel(R) Core(TM) i5-6400 CPU @ 2.7GHz x 4
Memory	8GB
Partitions	8
Platform	Spark 2.3, Tensorflow 2.0
# of Data	1,000,000(Skewed)

2. DNN 모델의 MSE 계산

기계 학습을 이용한 타당성 입증하기 위해 MSE를 DNN 모델의 깊이별로 평가하였다. 텐서플로우를 이용

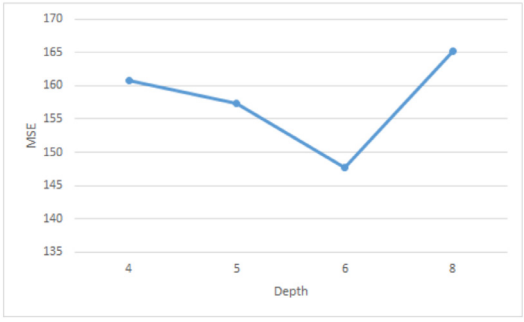

그림 4. MSE 계산 결과

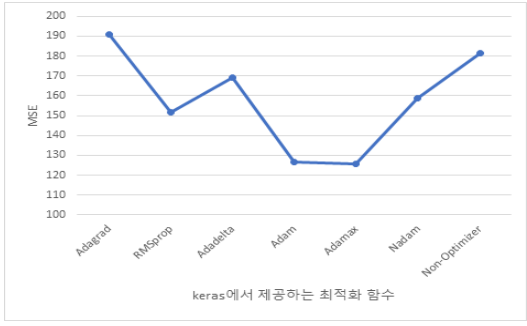

그림 5. 최적화 함수별 MSE 계산 결과

하여 DNN 모델을 구현하였다. 전체 데이터에서 80%는 학습데이터로 사용하고 20%는 정답 데이터로 사용하였다. 그림 4는 MSE 계산 결과를 보여준다. 깊이 값 n이 커질수록 MSE가 감소하는 경향을 보인다. 그러나 깊이가 8일 때, MSE가 증가하는 현상이 보이는데 이는 기계 학습 모델에서 문제점 중 하나인 과적합(overfitting)이다. 이는 학습데이터에 대한 학습이 과하게 진행된 것을 의미한다. 학습된 데이터에 대해서는 확연하게 오차가 감소하는 현상을 보이지만 전체 실험데이터에 대해서는 오차가 증가하는 현상이 발생한다. 본 논문에서는 과적합 문제를 해결하기 위해서 적절한 깊이인 6을 선택하여 평가를 진행하였다. 그 결과 효과적인 범위 도출을 위한 모델을 생성할 수 있었다.

파이썬으로 구현된 오픈 소스 신경망 네트워크 API인 keras는 기계 학습 모델을 최적화시킬 수 있는 다양한 함수들을 제공하고 있다. [그림 5]는 keras에서 제공하는 최적화 함수를 이용하여 MSE를 측정한 결과를 보여준다. 실험 평가에서 사용된 모델의 깊이는 이전 실험을 바탕으로 6으로 설정하였다. 그림에서 알 수 있듯이 ADAMAX[15]가 가장 작은 값의 MSE를 갖는다. 따라서 본 논문에서는 ADAMAX를 최적화 함수로 사용하여 모델을 생성하였다.

[그림 6]은 epoch에 따른 학습 곡선을 나타낸다. epoch는 기계학습에서 전체 데이터 세트에 대한 학습 횟수를 의미한다. 최적의 MSE를 도출할 때까지 걸리는 학습 횟수를 측정하기 위해 다음과 같은 실험을 진행하였다. 본 논문에서는 총 100 epoch를 진행하였다. 실험 결과, 약 5 epoch부터 MSE가 감소세를 띄고 있다.

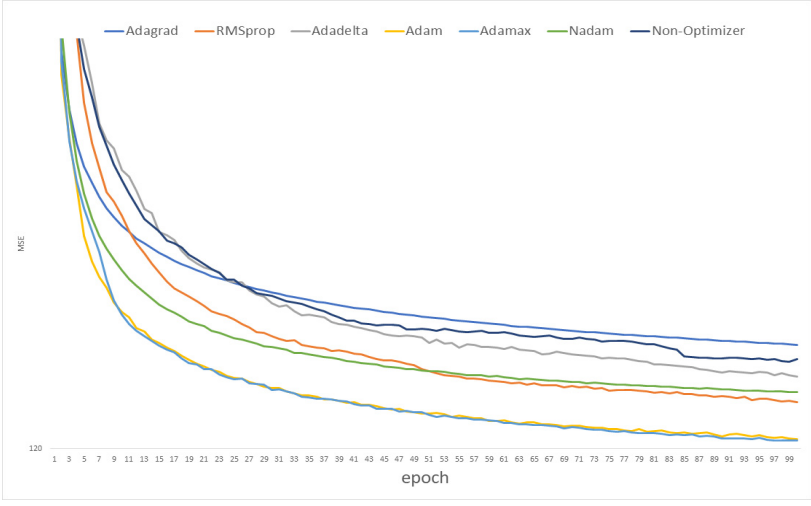

그림 6. epoc에 따른 MSE 감소율

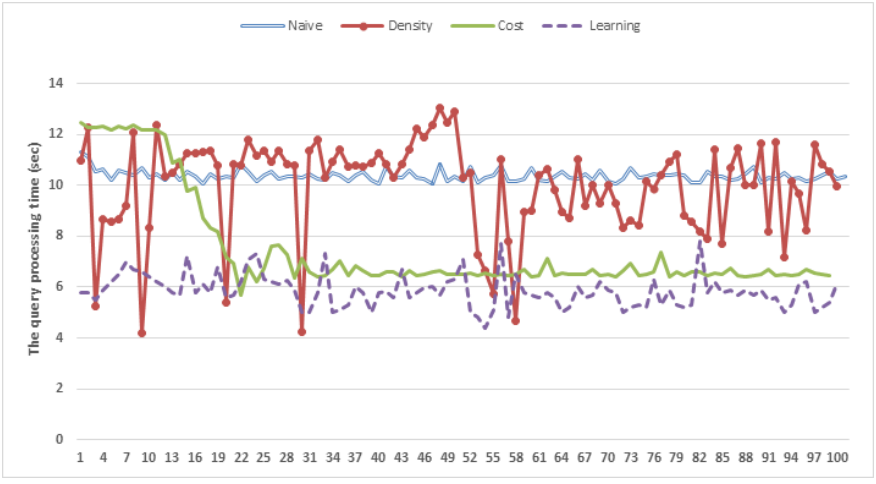

그림 7. 최적화 기법별 질의 처리시간 경과

이는 적은 횟수의 반복으로도 빠른 학습이 가능하다는 것을 의미한다. 학습을 위한 파라미터를 설정할 때, 일정 이상 값의 변동이 존재하지 않을 때 학습을 그만들 수 있다(early stop). 본 실험을 통해 데이터를 학습하는데 큰 반복이 필요하지 않음을 알 수 있었다.

3. 최적화 기법별 비교

[그림 7]은 k-최근접 기법에 대한 [9]에서 제안하는 최적화 기법의 성능 비교 결과를 나타내는 그래프이다. x축은 질의의 번호이며, 차례로 질의가 입력되는 상황이다. 전체 비교는 최적화가 적용되지 않은 나이브한 방법과 밀집도 그리고 비용기반과 제안하는 기법의 질의 처리 시간을 비교하였다.

나이브한 기법은 질의가 지속해서 진행되더라도, 최적화를 위해 따로 진행하는 기법이 없으므로 초반부터 후반까지 성능이 균일하다. [9]에서 제안한 밀집도 기반의 최적화 기법은 데이터 분포의 밀집도에 따라 다른 초기 탐색 값을 가지는 기법으로써 그래프에 나타난 바와 같이 특정 상황에는 좋은 성능을 보이지만 전체적으로 고른 성능이지 않다. 밀집도 값이 일치하지 않는 부분에선 탐색 범위가 매우 넓어져 불필요한 후보군의 개수를 포함하기 때문이다. 질의 탐색 비용기반의 최적화 기법은 탐색 범위를 증감하는 인자를 가져 탐색 범위를 적절하게 조정한다. 그래프의 결과를 보면 알 수 있듯이 성능이 좋아지면서 일정 값에 수렴하는 것을 알 수 있다. 그러나 일정 값에 수렴할 때까지 시간이 필요한 것을 알 수 있다. 반면에 본 논문에서 제안한 DNN 모델 기반의 최적화 기법은 처음부터 기존에 제안한 기법에 비교하여 좋은 성능으로 일정한 성능을 유지하는 것을 확인할 수 있다. 이를 통해 기존에 제안한 기법과 비교하여 제안하는 기법은 최적의 범위를 처음부터 안정성을 가지고 도출하는 것을 알 수 있었다.

k-최근접 질의의 최적화 범위는 질의점과 거리를 계산해야 하는 후보군의 개수에 따라 계산양이 달라진다. 그러므로 후보군 개수는 작을수록 좋다. [그림 8]과 [그림 9]는 k-최근접 질의를 범위 질의로 변경하였을 때 평균적으로 변환된 범위를 나타낸다.

타당한 비교를 위하여 기계 학습은 처리 비용 기반 최적화와 밀도 기반 최적화 두 가지로 도출한 최적 범

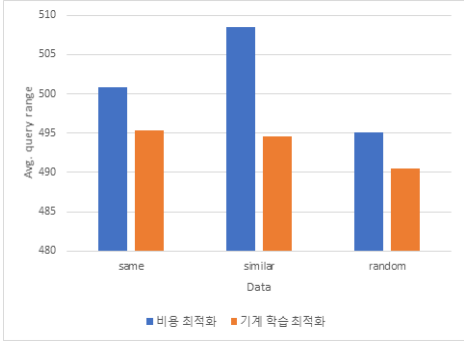

그림 8. 처리 비용 기반 최적화 기법과의 비교

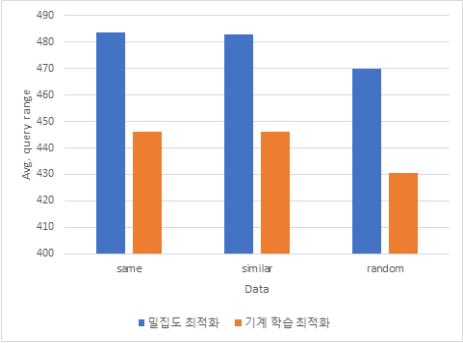

그림 9. 밀집도 기반 최적화 기법과의 비교

위 로그를 이용하여 학습하였다. [그림 6]은 임곗값을 사용하였던 처리 비용기반 최적화 기법과 비교한 결과이다. 두 가지 모두 기존 기법보다 DNN 모델을 이용하였을 때 더 작은 범위를 도출하였다. 범위가 작다는 것은 탐색해야 할 후보군의 수가 상대적으로 적다는 것을 의미하므로 처리 비용을 감소시킬 수 있다. 이를 통해 기계 학습 최적화 기법이 기존 논문에서 제안한 최적화 기법과 비교하여 우수한 성능과 안전성을 보이는 것을 확인할 수 있었다.

4. 인덱스 기법별 비교

최적화 기법의 성능을 검증하기 위하여 기존 기법의 최적화 기법과 제안하는 최적화 기법의 평균 처리 시간을 비교하였다. 또한, [9]에서 비교하였던 인덱스를 추가로 비교하였다. [그림 10]은 기존 기법에서 제안했던 기법과 k-최근접 질의 처리의 평균 시간을 비교한 결과를 보여준다. 붉은색으로 표시된 그래프는 k-최근접 질의를 위한 최적화 기법을 추가한 실험 평가 결과이다. 기존 기법에서 제안한 최적화 기법인 밀집도와 비용 기반의 최적화 기법보다 본 논문에서 제안하는 기계 학습 기법이 더 우수한 성능을 보이는 것을 알 수 있다. 비용 기반 최적화 기법과는 큰 차이가 없는 것처럼 보이지만 4.3절에서 본 것과 같이 비용 기반의 최적화 기법은 일정 값에 수렴하는 데 걸리는 시간이 소요되는 문제점을 지니고 있다는 차이가 있다.

여전히 분산 k-d 트리보다 좋지 않은 성능을 보인다. 하지만 기존 기법에서 실험한 결과 분산 k-d 트리는 범위 질의에서 하이브리드 인덱스와 비교하면 약 3배의 질의 처리 시간이 걸리는 반면, k-최근접 질의에서 비용 기반 최적화 기법을 추가하였을 때와 비교하였을 때 약 1초 정도의 시간 차이밖에 나지 않았다. 그러므로 범위 질의와 k-최근접 질의 모두를 고려하였을 때, 본 논문에서 제안하는 하이브리드 인덱스와 기계 학습 기반 최적화가 추가된 기법이 더 우수하다고 할 수 있다. 여러 가지 성능평가를 통해 제안하는 k-최근접 질의 최적화 기법의 타당성과 우수성을 입증할 수 있었다.

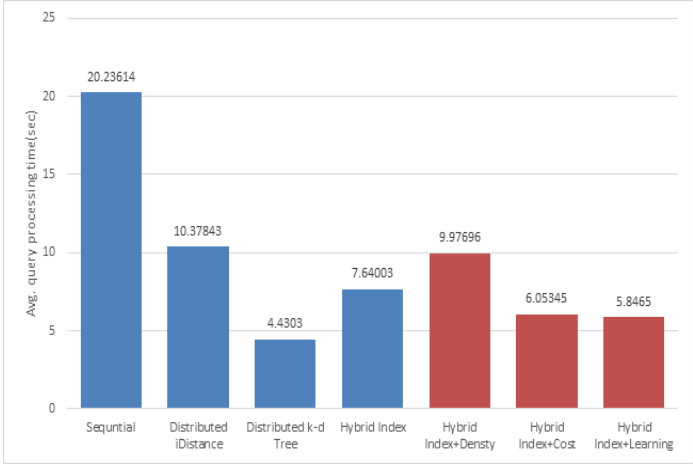

그림 10. 인덱스별 k-최근접 질의 처리 시간 비교

V. 결론 및 향후 연구

본 논문에서는 k-최근접 질의 처리에 대한 기계 학습 기반의 최적화 기법을 제안하였다. 제안한 질의 처리 최적화 기법은 기계 학습 모델 DNN을 이용하였다. 처리 비용 및 데이터 밀집도 기반의 최적화 기법을 통해 도출된 최적 범위를 학습 데이터로 사용하였다. 학습을 통한 k-최근접 질의 처리는 밀집도 및 비용 기반 최적화 기법과 비교하였다. 그 결과 처리 비용 및 밀집도 기반의 최적화 기법의 문제점을 해결할 수 있었다. 첫째, 처리 비용 기반 최적화 기법의 문제점이었던 수렴시간을 해결할 수 있었다. 둘째, 밀집도 기반의 최적화 기법에서 나타났던 문제점을 해결하여 안정적인 처리 속도를 도출할 수 있었다. 그 결과 검색 속도가 최대 5% 이상 증가했다. 또한, DNN 모델을 통해 예측한 질의 범위는 기존 기법에 비교하여 더 작은 것을 확인할 수 있었다. 그러나 밀집도 및 비용 기반 최적화와 비교하여 성능은 좋아졌지만, 여전히 분산 k-d 트리보다 질의 처리시간이 좋지 않았다. 또한, 학습을 위한 사전 데이터를 준비를 위하여 사용자가 직접 최적화 기법을 선택하여 로그를 생성해야 한다는 문제점을 여전히 내포하고 있다. 향후 연구에는 분산 k-d 트리보다 더 좋은 성능을 낼 방법을 연구하고, 데이터의 형태에 따라 최적화 기법을 선정하여 로그를 출력하고 학습할 수 있는 형태의 시스템을 구성할 예정이다.

참 고 문 헌

[1] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Wenjie Zhang, and Xuemin Lin, “Approximate nearest neighbor search on high dimensional data-experiments, analyses, and improvement,” IEEE Transactions on Knowledge and Data Engineering, 2019.

[2] Abu-Aisheh, Zeina, Romain Raveaux, and Jean-Yves Ramel, “Efficient k-nearest neighbors search in graph space,” Pattern Recognition Letters, Vol.134, pp.77-86, 2020.

[3] Yiwei Pan; Zhibin Pan, Yikun Wang, and Wei Wang, “A new fast search algorithm for exact k-nearest neighbors based on optimal triangle-inequality-based check strategy,” Knowledge-Based Systems, 189, 105088, 2020.

[4] Zhiyin Zhang, Xiaocheng Huang, Chaotang Sun, Shaolin Zheng, Bo Hu, Jagannadan Varadarajan, Yifang Yin, Roger Zimmerman, and Guanfeng Wang, “Sextant: Grab's Scalable In-Memory Spatial Data Store for Real-Time K-Nearest Neighbour Search,”, 20th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2019.

[5] Gallego, Antonio-Javier, Jorge Calvo-Zaragoza, and Juan Ramón Rico-Juan, “Insights into efficient k-Nearest Neighbor classification with Convolutional Neural Codes,” IEEE Access, 2020.

[6] Vajda, Szilárd and K. C. Santosh, “A fast k-nearest neighbor classifier using unsupervised clustering,” International conference on recent trends in image processing and pattern recognition. Springer, Singapore, 2016.

[7] Peng Dai, Yuan Yang, Manyi Wang, and Ruqiang Yan, “Combination of DNN and improved KNN for indoor location fingerprinting,” Wireless Communications and Mobile Computing, 2019.

[8] K. Atefi, H. Hashim, and M. Kassim, “Anomaly Analysis for the Classification Purpose of Intrusion Detection System with K-Nearest Neighbors and Deep Neural Network,” 2019 IEEE 7th Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia, 2019.

[9] 최도진, 박송희, 김연동, 위지원, 이현병, 임종태, 복경수, 유재수, “스파크 환경에서 내용 기반 이미지 검색을 위한 효율적인 분산 인-메모리 고차원 색인 기법,” 정보과학회논문지, 제47권, 제1호, pp. 95-108, 2020.

[10] H. Wei, Y. Du, F. Liang, C. Zhou, Z. Liu, J. Yi, and D. Wu, “A kd tree-based Algorithm to Parallelize Kriging Interpolation of Big Spatial Data,” Journal of GIScience & Remote Sensing,

Vol.52, No.1, pp.40-57, 2015.

[11] H. V. Jagadish, B. C. Ooi, K. L. Tan, C. Yu, and R. Zhang, "iDistance: An Adaptive B+-tree based Indexing Method for Nearest Neighbor Search," Journal of Transactions on Database Systems (TODS), Vol.30, No.2, pp.364-397, 2005.

[12] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," Neural networks, Vol.61, pp.85-117, 2015.

[13] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital Selection and Analogue Amplification Coexist in aCortex-Inspired Silicon Circuit," Nature, Vol.405, pp.947-951,2000.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, Vol.60, No.2, pp.91-110, 2004.

[15] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412-6980, 2014.

[16] J. Maillo, S. García, J. Luengo, F. Herrera, and I. Triguero, "Fast and Scalable Approaches to Accelerate the Fuzzy k-Nearest Neighbors Classifier for Big Data," in IEEE Transactions on Fuzzy Systems, Vol.28, No.5, pp.874-886, 2020.

[17] J. Maillo, S. García, J. Luengo, F. Herrera, and I. Triguero, "Fast and Scalable Approaches to Accelerate the Fuzzy k-Nearest Neighbors Classifier for Big Data," in IEEE Transactions on Fuzzy Systems, Vol.28, No.5, pp.874-886, 2020.

[18] J. M. Lee, "Fast k-nearest neighbor searching in static objects," Wireless Personal Communications, Vol.93, No.1, pp.147-160, 2017.

[19] Utsav Sheth, Sanghamitra Dutta, Malhar Chaudhari, Haewon Jeong, Yaoqing Yang, Jukka Kohonen,Teemu Roos, Pulkit Grover, "An Application of Storage-Optimal MatDot Codes for Coded Matrix Multiplication: Fast k-Nearest Neighbors Estimation," in IEEE International Conference on Big Data, Seattle, WA, USA, pp.1113-1120, 2018.

[20] K. Li and Jitendra Malik, "Fast k-nearest neighbour search via prioritized DCI," arXiv preprint arXiv:1703.00440, 2017.

[21] H. C. V. Ngu and J. H. Huh, "B+-Tree Construction on Massive Data with Hadoop," Journal of the Cluster computing, Vol.22, No.1, pp.1011-1021, 2019.

[22] Mishra, Gaurav, and Sraban Kumar Mohanty, "A fast hybrid clustering technique based on local nearest neighbor using minimum spanning tree," Expert Systems with Applications 132 ,28-43, 2019.

[23] H. J. Jang et al. "Nearest base-neighbor search on spatial datasets," Knowledge and Information Systems, Vol.62, No.3, pp.867-897, 2020.

[24] D. H. Yan et al, "K-nearest Neighbors Search by Random Projection Forests," IEEE Transactions on Big Data, 2019.

저 자 소 개

위 지 원(Ji-Won Wee) 정회원

- 2018년 2월 : 한국교통대학교 컴퓨터 공학과(공학사)
- 2020년 8월 : 충북대학교 정보통신공학과(공학석사)
- 2019년 8월 ~ 현재 : 충북대학교 정보통신공학과(박사과정)

〈관심분야〉 : 빅데이터, 그래프 스트림, 이상 그래프, 데이터베이스 시스템

최 도 진(Do-Jin Choi) 정회원

- 2014년 2월 : 한국교통대학교 컴퓨터 공학과(공학사)
- 2016년 2월 : 한국교통대학교 컴퓨터 공학과(공학석사)
- 2020년 2월 : 충북대학교 정보통신공학과(공학박사)
- 2020년3월 ~ 현재 : 충북대학교 Postdoc.

〈관심분야〉 : 연속 질의 처리, 그래프스트림, 빅데이터, 데이터베이스 시스템

이 현 병(Hyeon-Byeong Lee) 정회원

- 2016년 8월 : 한국교통대학교 컴퓨터 공학과(공학사)
- 2018년 8월 : 한국교통대학교 컴퓨터 공학과(공학석사)
- 2019년 3월 ~ 현재 : 충북대학교 정보통신공학과(박사과정)

〈관심분야〉 : 그래프 스트림, 빅데이터, 데이터베이스 시스템

임 종 태(Jong-Tea Lim) 정회원

- 2009년 2월 : 충북대학교 정보통신공학과(공학사)
- 2011년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2015년 8월 : 충북대학교 정보통신공학과(공학박사)
- 2015년 9월 ~ 2019년 8월 : 충북대학교 정보통신공학과 Postdoc.
- 2019년 10월 ~ 현재 : 충북대학교 전자정보대학 정보통신 공학부 초빙 조교수

〈관심분야〉 : 소셜 미디어, 빅데이터, 시공간 데이터베이스, 위치기반 서비스 등

임 헌 진(Hun-Jin Lim) 정회원

- 2001년 2월 : 충주대학교 컴퓨터공학과(공학사)
- 2012년 8월 : 충북대학교 정보통신공학과(공학석사)
- 2017년 2월 : 충북대학교 빅데이터 협동과정(박사수료)

〈관심분야〉 : 빅데이터, 정보보안, 개인정보보안

복 경 수(Kyoung-Soo Bok) 종신회원

- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2005년 8월 : 충북대학교 정보통신공학과(공학박사)
- 2005년 3월 ~ 2008년 2월 : 한국과학기술원 정보전자연구소 Postdoc.
- 2008년 3월 ~ 2011년 2월 : 가인정보기술 연구소 연구원
- 2011년 3월 ~ 2019년 8월 : 충북대학교 전자정보대학 정보통신공학부 초빙 교수
- 2019년 9월 ~ 현재 : 원광대학교 SW융합학과 조교수

〈관심분야〉 : 데이터베이스 시스템, 이동 객체 데이터베이스, 이동 P2P 네트워크, 소셜 네트워크 서비스, 빅데이터 등

유 재 수(Jae-Soo Yoo) 종신회원

- 1991년 2월 : 한국과학기술원 전산학과(공학석사)
- 1995년 2월 : 한국과학기술원 전산학과(공학박사)
- 1995년 2월 ~ 1996년 8월 : 목포대학교 전산통계학과 전임강사
- 1996년 8월 ~ 현재 : 충북대학교 전자정보대학 정보통신공학부 정교수

〈관심분야〉 : 데이터베이스 시스템, 멀티미디어 데이터베이스, 센서 네트워크, 바이오인포매틱스, 빅데이터 등