딥러닝 기반 데이터 압축 알고리즘 설계 및 구현*

김선형 O1**, 전재호 1**, 심윤성 1**, 김정균 2, 진권휴 2, 이성진 2***

1 대구경북과학기술원 기초학부, 2 대구경북과학기술원 정보통신융합전공

ksh981028@dgist.ac.kr, dgwogh@dgist.ac.kr, sllove2@dgist.ac.kr,
sslktong@dgist.ac.kr, overclocked@dgist.ac.kr, sungjin.lee@dgist.ac.kr

Design and Implementation of Deep-learning-assisted Data Compression

Seonhyeong Kim O1**, Jaeho Jeon1**, Yunseong Shim1**,

Jeonggyun Kim2, Gwonhyu Jin2, Sungjin Lee2***

1Department of School of Undergraduate Studies, DGIST
2Information and Communication Engineering, DGIST

요 약

최근 대용량 데이터 생성이 빠르게 증가함에 따라 데이터를 유지하기 위해 많은 비용과 서버가 필요하다. 이에 많은 선행 연구에서 다양한 데이터 감소 기술을 제안했다. 그 중 활발하게 연구되고 있는 Delta Compression은 디스크 내에 존재하는 유사한 패턴을 가진 데이터 블록을 찾아 입력된 블록과 동시에 XOR을 수행하고 XOR 연산의 결과인 델타 부분만 압축을 진행하여 데이터를 감소시킨다. 그러나 기존 Delta Compression은 유사한 데이터 블록을 찾지 못해 대부분의 경우 최적 성능을 보이지 못한다. 본 논문에서는 가장 유사한 블록을 찾아 Delta Compression을 수행하는 딥러닝 기반의 새로운 압축 알고리즘인 DeepComp를 제안한다. DeepComp을 사용해 압축을 진행하여 기존 Delta Compression만을 사용한 압축보다 5.79% 압축률의 개선을 보인다.

1. 서론

매일 많은 양의 디지털 데이터가 생성되고 저장된다. 이러한 방대한 양의 데이터를 유지하기 위해서는 많은 비용과 서버가 필요하다. 데이터 센터에서 저장된 데이터의 양을 줄이고 더 나은 TCO를 실현하기 위해 많은 선행 연구에서 다양한 데이터 감소 기술을 제안했다. 그 중 원본을 인코딩하여 디스크에 물리적으로 저장되는 데이터의 크기를 줄이는 데이터 압축 방법[1]과, 중복 블록을 제거하여 효과적으로 데이터를 저장하는 데이터 중복 제거 방법[1]이 주로 연구되었다. 그러나 데이터 압축 방법은 이미 압축된 이미지 및 비디오와 같이 엔트로피가 높은 데이터의 크기를 줄이는 데 효과적이지 않으며, 데이터 중복 제거 방법은 입력 데이터의 엔트로피에 의존하지 않지만 동일한 블록이 존재할 경우에만 효과적이라는 한계가 존재한다. 상기 한계점을 극복하기 위해 본 논문에서의 데이터 압축과 중복 제거 기법의 한계를 극복한, DeepComp라고 하는 새로운 딥러닝 기반 압축 기술을 제안한다. 제안된 DeepComp 기법은 데이터 압축률 향상을 통해 실제 저장장치에 기록되는 데이터를 약 5.79% 감소시킬 수 있다.

2. 선행 연구

디지털 데이터를 관리함에 있어 데이터의 물리적인 크기를 줄이기 위해 고안된 데이터 압축(Compression) 방식은 원본 데이터를 인코딩하여 하드 디스크에 저장되는 데이터의 크기를 줄인다[2]. 하지만 이는 이미지나 비디오와 같이, 이미 압축되어 엔트로피가 높은 데이터의 크기를 줄이는 데에는 효과적이지 못하다. 또한 중복되는 데이터를 제거하는 중복 제거(Deduplication) 방식은 입력 데이터의 엔트로피에 의존하지는 않지만, 동일한 블록이 존재할 경우에만 효과적이다. 즉, 데이터 간의 작은 차이가 존재하더라도 중복되는 패턴을 제거하지 못하는 한계점을 가지고 있다. Delta Compression은 데이터 압축 기술과 데이터 중복 제거 기술의 한계 및 문제점을 해결하기 위해서 고안되었다[3]. 새로운 데이터 블록이 입력되면 델타 압축 기술은 우선적으로 디스크 내에 존재하는 유사한 패턴을 가진 참조 블록을 찾아 입력된 블록과 동시에 XOR을 수행하고 XOR 연산의 결과인 델타 부분만 압축을 진행한다. 이러한 방식은 디스크 내 증가하는 엔트로피와 독립적으로 작용하여 높은 비율의 압축 결과를 얻을 수 있고, 입력 데이터와 기존 참조 블록이 정확하게 일치하지 않아도 두 블록에 존재하는 중복 패턴을 효율적으로 제거하면서 압축을 진행할 수 있다. Delta Compression과 관련한 주요 기술적 과제는 가장 유사한 데이터 블록들을 찾는 것이다. 하지만 기존 Delta Compression 기술들은 디스크에서 유사한 데이터 블록을 찾지 못해 대부분의 경우 최적의 성능을 제공하지 못한다.

* 이 논문은 DGIST UGRP 프로그램의 지원을 받아 수행된 연구임
** 이 저자들은 논문에 대한 기여도가 같음
*** 교신 저자

3. DeepComp: 딥러닝 기반 Delta Compression 기술

본 논문에서는 딥러닝 알고리즘을 활용하여 유사성이 높은 데이터 블록들을 찾아 압축률을 향상시키는 새로운 압축 기술인 DeepComp를 제안한다. DeepComp는 파일의 압축률을 위해 다음 세 단계의 과정을 거친다. 첫번째는 파일 내부의 데이터 블록들을 군집화 하여 유사한 데이터 블록들을 찾는 군집화 단계, 다음은 군집화 된 데이터 블록들의 유사성을 찾기 위한 CNN Algorithm을 기반의 학습 단계, 마지막은 입력된 데이터와 유사성이 높은 데이터 블록들 학습 데이터를 통해 검색하는 추론 단계이다. 그림 1은 제안된 DeepComp의 각 단계와 전체적인 동작 과정을 보여준다.

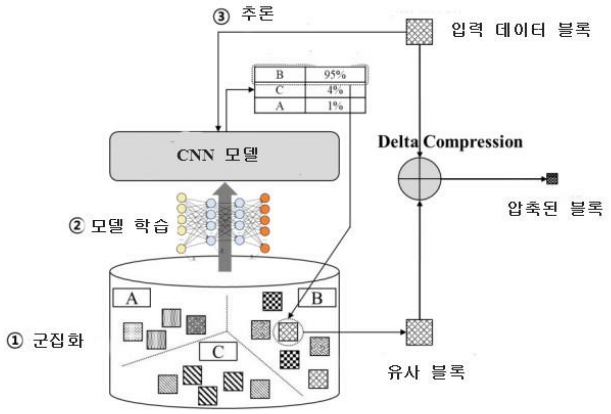

그림 1. DeepComp 모식도

3-1. 군집화

유사한 패턴을 가진 블록을 찾기 위해 바이너리 데이터로 구성된 블록들은 거리를 계산할 다른 블록과 XOR을 수행하고, 수행한 결과에 대한 엔트로피 값을 클러스터 거리로 정의한다. 이후 계산된 거리에 따라 군집화 알고리즘을 수행한다. 본 논문에서 사용한 군집화 알고리즘 DBSCAN(Density Based Spatial Clustering of Applications with Noise)[4]은, 밀도 기반의 군집화 알고리즘으로 데이터가 세밀하게 몰려 있어 밀도가 높은 부분을 군집화 한다. DBSCAN에는 두 개의 매개변수 Minpt와 Eps가 있다. 한 데이터 포인트에서 Eps 거리 안에 데이터가 Minpt 개수만큼 포함되면, 이 데이터 포인트를 핵심 포인트로 분류한다. 핵심 포인트를 중심으로 Eps반경 내에 Minpts 이상의 수의 데이터가 존재하면, 그 기준점을 중심으로 하나의 군집을 생성한다. 어떠한 클러스터에도 속하지 않는 데이터는 Noise Point로 처리한다.

3-2. 모델 학습

본 논문에서는 1000개의 폴더 내 총 27155개의 4096 바이트 단위로 존재하는 군집화 된 데이터에 대해서 학습을 진행한다. 폴더 내 27155개의 파일 중 50%를 학습 데이터, 나머지를 테스트 데이터로 설정하여 입력데이터로 사용하였으며 군집화 된 데이터들을 분석하기 위해 CNN (Convolutional Neural Network)을 사용하여 학습한다. CNN은 Convolution Layer와 Maxpooling Layer가 반복적으로 쌓인 구조에 마지막 출력 층에 활성화 함수가 추가된 모습이다. 2차원 형태의 3*3 커널 크기의 Convolution layer 필터 8개, 16개, 32개를 사용하였고, 각 필터 사이에 2차원 형태의 2*2 커널 크기의 Maxpooling Layer를 삽입한다. 이후 Dense 층을 추가하는 단계에서 Flatten Layer를 삽입하고, 활성화 함수로는 ReLU와 Softmax를 사용하여 구성한다. 마지막으로 모델을 컴파일하기 위해서 Optimizer를 지정하고, 앞서 사용한 활성화 함수 Softmax에 대한 손실함수로 CrossEntropy를 사용한다. 군집화 된 입력 데이터가 위와 같은 방식으로 형성된 합성곱 신경망을 통과하면서 군집들 간의 유사성 정확도를 평가한다. 유사성 정확도는 Root Mean Square Propagation (RMSProp), Adam, Adagrad, Stochastic Gradient Descent (SGD)의 4종류의 Optimizer를 각각 사용하여 평가한다.

3-3. 추론 및 압축

학습과정을 마친 모델을 불러와 유사한 데이터끼리 분류한다. 모델에 데이터를 통과시키면 각 데이터 마다 256-bit의 해시 값을 갖게 된다. 이 때, 서로 유사한 데이터는 동일한 해시 값을 갖게 되며, 각 데이터의 해시 값을 비교하여 데이터들의 유사성 여부를 판단할 수 있게 된다. 유사하다고 분류된 데이터는 XOR 과정을 수행하여 Delta Compression 방법으로 파일을 압축한다. 유사한 입력 데이터를 사용하여 압축을 진행했기 때문에 압축을 마친 데이터의 엔트로피가 낮아지고, 따라서 압축률은 XOR 과정을 마치기 전보다 높아진다. 데이터 압축 이후 각 데이터가 어떠한 데이터와 Delta Compression을 수행했는지에 대한 참고 문서가 생성된다. 참고 문서에는 압축된 데이터들의 크기, XOR 압축 여부, 어떠한 데이터와 XOR 했는지의 정보가 담겨있으며, 추후 압축을 해제할 때 다시 사용된다.

4. 실험 결과

4-1. 군집화 정확도 평가

DBCSAN 알고리즘을 수행하여 데이터 블록의 군집화 결과가 최적의 값을 가질 수 있도록 Eps와 Minpt값을 달리하여 최적의 Eps와 Minpt 값을 도출한다(그림 2). 데이터 블록의 Entropy 값이 0-8 사이의 값을 가지므로 Eps 값을 1.5부터 2.3까지 달리하여 최적 Eps값을 계산한다. 군집화 결과, 각 클러스터 안에서 원소들 간의 거리가 가장 적은 대표 원소를 찾고, 대표 원소와 다른 원소들의 XOR Entropy 값과 대표 원소의 Entropy 값의 합의 평균이 가장 작은 Eps가 가장 작은 메모리 값을 가진다고 판단한다. 이때, Minpt 값을 모두 10으로 하여 계산하고, Noise값은 48555에서 48736 값 사이로 전체 Entropy 값과 유사한 양상을 보이지만, 결과 값에 유의미한 차이를 주지 못한다. Minpt 값이 커지면 생성되는 군집의 개수는 줄어들고, Noise값은 커진다. 학습 과정에 사용할 군집의 개수와, Noise 개수의 적절한 비율을 찾아 최적 Minpt

값을 도출한다. 도출한 최적의 Eps는 2.0, Minpt는 10이다. 이 경우 전체 데이터 블록 102342개, 클러스터 개수 122개, Noise 개수 48356개의 결과를 얻는다.

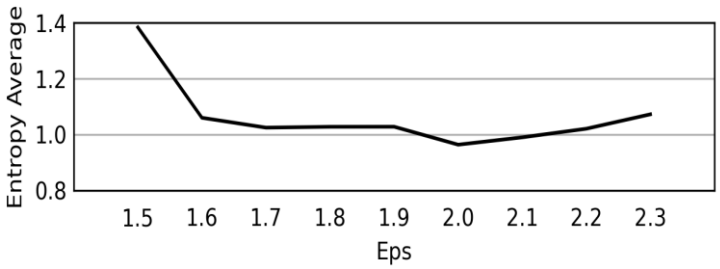

그림 2. Eps 값에 따른 데이터 블록 Entropy 평균

4-2. 모델 학습 평가

Optimizer를 각각 Root Mean Square Propagation (RMSProp), Adam, Adagrad, Stochastic Gradient Descent (SGD)로 바꿔가면서 데이터에 대한 학습률을 평가하였다. 학습은 최대 정확도에 대한 최소 손실 값을 얻을 때까지 진행한다(그림 3). 학습률은 0.01로 설정하여 진행하였다. 학습에 따른 정확도는 Adam, RMSProp, SGD, Adagrad 순으로 높다. Adagrad의 경우 학습을 계속 진행할수록 Step Size가 크게 줄어든다는 문제점을 가지고 있는데, 이에 따라 정확도의 증가 폭이 현저히 줄어듦을 확인할 수 있었고 최종적으로 82.99%의 가장 낮은 정확도을 보인다. 또한 SGD의 경우에는 91.50%의 정확도를 보이는데, SGD는 다른 알고리즘에 비해 계산 방식이 단순하고 이동속도가 현저히 느리기 때문에, 학습을 반복할수록 정확도가 미세하게 올라갈 수는 있어도 좋은 결과를 위해 소모되는 시간이 상당히 크다는 단점이 있다. 이러한 방식들을 보완한 RMSProp와 Adam 에서는 비교적 짧은 학습 시간 대비 높은 정확도를 얻을 수 있음을 확인했다.

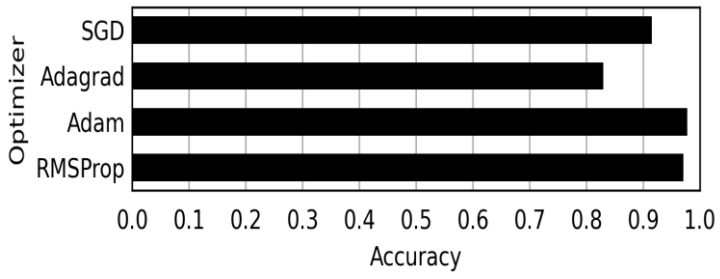

그림 3. Optimizer의 종류에 따른 모델 학습의 정확도

4-3. 추론 및 압축률 평가

학습 과정을 마친 모델로 다양한 크기의 파일을 압축했다. 참조 문서가 포함된 결과는 아래 표와 같다. 표는 DeepComp의 압축 성능을 중복 제거 + 압축 방법과 비교하여 정리한 것이다. 중복 제거 + 압축은 오직 중복 처리와 자체 압축만 사용한 방법이며, 성능향상률은 중복 제거 + 압축 방법 대비 DeepComp의 압축률이 얼마나 향상되었는지 나타낸다. DeepComp의 성능은 중복 제거 + 압축 방법보다 5.79%(기하 평균) 가량 높은 압축률을 보인다. CPU만을 사용한 DeepComp의 초당 처리량은 1.82MB/s 이다. 하지만 가속을 위해 GPU를 사용할 경우, 초당 처리량은 29.53MB/s 로 향상된다.

파일명	파일크기(MB)	원본 대비 압축률(%)		성능향상률
		중복 제거 + 압축	DeepComp	
mix	4192.0	23.6	22.6	**4.38%**
pc	1568.3	38.7	36.7	**5.13%**
synth	653.0	46.9	46.3	**1.19%**
update	3731.0	40.0	38.0	**4.80%**
web	958.6	7.4	4.6	**37.63%**
dedup	920.6	28.6	26.3	**7.88%**

표 1. 중복 제거 + 압축 방법 대비 DeepComp 성능 향상률

5. 결론

본 논문에서는 파일의 압축률을 향상시키기 위한 세 단계로, 파일 내부의 존재하는 데이터 블록들을 군집화 하여 유사한 데이터 블록들을 찾는 과정, 군집화 과정을 마친 데이터 블록들 간의 유사성에 대한 판별을 높이기 위한 CNN 알고리즘 학습 과정, 알고리즘 학습을 마친 데이터 블록들을 델타 압축하는 과정을 진행하였다. 완성된 DeepComp 방법은 압축과 중복 제거 만을 사용한 방법에 비해 5.79%(기하 평균) 높은 압축률을 보였다.

6. 참고 문헌

[1] Ronald Pagani, Jr., “Advanced Data Reduction Concepts”,https://www.snia.org/sites/default/files/DSI/2016/presentations/cap_opt/RonaldPagani_Advanced_Data_Reducti on.pdf, 2016.

[2] Zhang, Y., Xia, W., Feng, D., Jiang, H., Hua, Y., & Wang, Q. (2019). Finesse: Fine-grained feature locality based fast resemblance detection for post-deduplication delta compression. In 17th USENIX Conference on File and Storage Technologies (FAST 19) (pp. 121-128).

[3] Jisung Park, Sungjin Lee, and Jihong Kim, “DAC: Dedup-Assisted Compression Scheme for Improving Lifetime of NAND Storage Systems,” in Proceedings of the Design, Automation, and Test in Europe Conference (DATE '17), 2017.

[4] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Knowledge Discovery in Database (Vol. 96, No. 34, pp. 226-231).