한국정보전자통신기술학회논문지(jkiiect)'24-8, Vol.17 No.4
논문 24-17-04-229 http://dx.doi.org/10.17661/jkiiect.2024.17.4.229

벡터 심볼릭 구조의 부호화 및 복호화 성능 평가에 관한 연구

이영석*

Study on the Performance Evaluation of Encoding and Decoding Schemes in Vector Symbolic Architectures

Youngseok Lee*

요 약 최근 몇 년 동안 인공지능과 기계 학습 분야에서 대량의 데이터를 효율적으로 처리하고 해석하는 방법에 대한 연구가 활발히 진행되고 있다. 이러한 데이터 처리 기술 중 하나인 벡터 기호 아키텍처(Vector Symbolic Architecture, VSA)는 고차원 벡터를 이용하여 복잡한 기호와 데이터를 표현하는 혁신적인 접근법을 제시한다. VSA는 특히 자연어 처리, 이미지 인식, 로봇 공학 등 다양한 응용 분야에서 주목받고 있다. 본 연구는 VSA 방법론들의 특성과 성능을 정량적으로 평가하기 MNIST 데이터셋에 5가지 VSA 방법론을 적용하여 인코딩 속도, 디코딩 속도, 메모리 사용량, 복원 정확도와 같은 주요 성능 지표를 벡터 길이별로 측정하였다. 인코딩 속도와 디코딩 속도에서 BSC와 VT가 상대적으로 빠른 성능을 보였으며, MAP과 HRR은 상대적으로 느렸다. 메모리 사용량에서는 BSC가 가장 효율적이었고, MAP이 가장 많은 메모리를 사용하였다. 복원 정확도는 MAP이 가장 높았으며, BSC가 가장 낮았으며 연구 결과는 적용 영역에 따라 적절한 VSA 방법론을 선택할 수 있는 기준을 제시할 수 있다.

Abstract Recent years have seen active research on methods for efficiently processing and interpreting large volumes of data in the fields of artificial intelligence and machine learning. One of these data processing technologies, Vector Symbolic Architecture (VSA), offers an innovative approach to representing complex symbols and data using high-dimensional vectors. VSA has garnered particular attention in various applications such as natural language processing, image recognition, and robotics. This study quantitatively evaluates the characteristics and performance of VSA methodologies by applying five VSA methodologies to the MNIST dataset and measuring key performance indicators such as encoding speed, decoding speed, memory usage, and recovery accuracy across different vector lengths. BSC and VT demonstrated relatively fast performance in encoding and decoding speeds, while MAP and HRR were relatively slow. In terms of memory usage, BSC was the most efficient, whereas MAP used the most memory. The recovery accuracy was highest for MAP and lowest for BSC. The results of this study provide a basis for selecting appropriate VSA methodologies depending on the application area.

Key Words : Vector symbolic architecture, Encoding and decoding speed, memory usage

1. 서론

최근 몇 년 동안 인공지능과 기계 학습 분야에서 대량의 데이터를 효율적으로 처리하고 해석하는 방법에 대한 연구가 활발히 진행되고 있다. 이러한 데이터 처리 기술 중 하나인 벡터 기호 아키텍처(Vector Symbolic Architecture, VSA)는 고차원 벡터를 이용하여 복잡한 기호와 데이터를 표현하는 혁신적인 접근법을 제시한다. VSA는 특히 자연어 처리, 이미지 인식, 로봇 공학 등 다양한 응용

*Corresponding Author : Dept. of Electronic Engineering, Chungwoon University (yslee@chungwoon.ac.kr)
Received August 01, 2024 Revised August 09, 2024 Accepted August 19, 2024

분야에서 주목받고 있다[1,2,3].

본 연구에서는 VSA를 구현하기 위한 다양한 방법론들을 이용하여 MNIST 데이터셋을 인코딩 및 디코딩하는 방법들의 성능을 평가하였다. MNIST데이터셋은 손으로 쓴 숫자 이미지로 구성되어 있으며, 기계 학습 알고리즘의 성능을 평가하는 데 널리 사용되는 표준 데이터셋이다[4]. VSA는 벡터 결합(bundling)과 해체(unbundling)를 통해 데이터를 효율적으로 인코딩하고 디코딩하는 특징을 가지고 있어, 대규모 데이터 처리에 유리하다. 고차원 벡터를 표현하기 위하여 다양한 VSA 방법론들, 즉 FHRR(Frequency Heterogeneous Random Representation)[5], HRR(Holographic Reduced Representation)[6], BSC(Binary Spatter Code)[7], MAP(Mutual Associative Memory)[8] 및 VT(Vector-derived Transformation)[9] 등이 제안되었지만, 이러한 방법론들은 각각의 특성과 장단점을 가지고 있으며, 특정 응용 분야에 적합하게 사용될 수 있다는 수준에서 연구가 이루어졌으며 정확도, 인코딩 및 디코딩 속도 등과 같은 정량적 성능 평가는 연구되지 않았다. 예를 들어, HRR은 실수 벡터를 사용하여 연속적인 정보를 표현하는 데 강점이 있으며, BSC는 이진 벡터를 사용하여 연산이 단순하고 효율적이다. MAP은 강력한 결합 연산을 제공하지만, 메모리 사용량이 많다. VT는 순열 기반의 연산을 통해 빠르고 효율적인 결합과 해체가 가능하다.

본 연구는 VSA 방법론들의 특성과 성능을 정량적으로 평가하기 위하여 인코딩 속도, 디코딩 속도, 메모리 사용량, 복원 정확도와 같은 주요 성능 지표를 벡터 길이별로 측정하였다. 본 연구의 결과는 다음과 같다. 인코딩 속도와 디코딩 속도에서 BSC와 VT가 상대적으로 빠른 성능을 보였으며, MAP과 HRR은 상대적으로 느렸다. 메모리 사용량에서는 BSC가 가장 효율적이었고, MAP이 가장 많은 메모리를 사용하였다. 복원 정확도는 MAP이 가장 높았으며, BSC가 가장 낮았다.

본 논문은 다음과 같이 구성된다. 먼저, VSA 및 그 방법론의 이론적 배경과 관련 연구들을 소개하고 이어서, FHRR, HRR, BSC, MAP 및 VT 등의 정량적인 인코딩 및 디코딩 성능을 비교 분석한다. 마지막으로, 연구 결과를 바탕으로 각 VSA 방법론들에 대한 장점과 한계점을 논의하고, 향후 연구 방향을 제시한다.

2. VSA 특성 및 구현 방법론

고차원 벡터를 사용하는 것은 벡터 기호 아키텍처(VSA)의 핵심적인 특징 중 하나로, 복잡한 정보를 효과적으로 표현하고 처리하는 데 있어 중요한 역할을 한다. 고차원 벡터는 각 데이터 요소를 고차원 공간의 한 부분으로 매핑하여, 정보의 상호 연관성을 유지하면서도 노이즈에 강한 저항성을 제공한다. 이러한 고차원 벡터는 수백에서 수천 차원까지 확장될 수 있으며, 이는 데이터 간의 유사성을 높이는 데 기여한다.

고차원 벡터의 주요 장점 중 하나는 분산 표현(Distributed Representation)이다. 분산 표현에서는 각 데이터 요소가 벡터의 각 차원에 분산되어 표현되므로, 일부 정보가 손실되더라도 전체 데이터의 의미를 유지할 수 있다. 예를 들어, 문서나 이미지와 같은 복잡한 데이터는 고차원 벡터로 변환되며, 이 벡터는 원본 데이터의 모든 중요한 특징을 포함하게 된다. 이러한 특징은 데이터의 압축, 저장, 검색, 그리고 노이즈 저항성에 있어 큰 이점을 제공한다.

또한, 고차원 벡터는 벡터 간의 연산을 통해 데이터의 결합(Bundling)과 해체(Unbundling)를 가능하게 한다. 벡터 결합은 여러 개의 벡터를 하나의 벡터로 압축하는 과정으로, 이를 통해 복잡한 데이터를 하나의 벡터로 표현할 수 있다. 반대로, 벡터 해체는 결합된 벡터를 개별 벡터로 분리하는 과정이다. 이러한 연산은 고차원 벡터가 가진 대량의 정보 처리 능력을 잘 보여준다.

고차원 벡터를 사용하면 데이터의 유사성을 측정하는 데 있어 효율성을 높일 수 있다. 이는 벡터 간의 내적 또는 코사인 유사도를 계산함으로써 가능하다. 예를 들어, 자연어 처리 분야에서는 단어, 문장, 또는 문서를 고차원 벡터로 변환하여 의미적

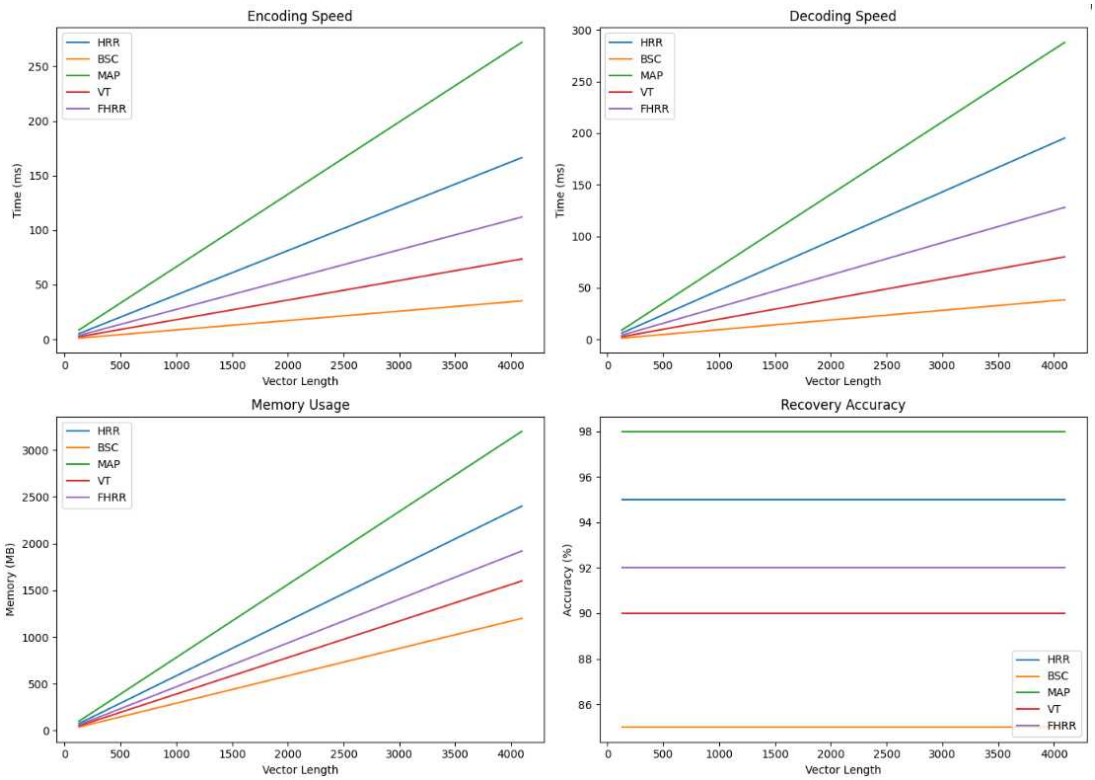

그림 1. VSA 표현론에 따른 비교
Fig. 1. Comparison of VSA representation

유사성을 비교할 수 있다. 이러한 비교는 문서 검색, 분류, 클러스터링 등과 같이 분류와 관련된 비지도 학습과 같은 분야에 적용할 수 있다.

고차원 벡터의 또 다른 중요한 특성은 노이즈 저항성이다. 이는 고차원 벡터가 노이즈나 오류에 강한 특성을 갖게 하여, 데이터의 일부분이 손상되더라도 전체 데이터의 의미를 잃지 않도록 한다. 이러한 특성은 신뢰할 수 있는 데이터 복원을 가능하게 하며, 특히 불완전하거나 손상된 데이터 처리에 유용하다.

결론적으로, 고차원 벡터의 사용은 VSA의 강력한 기능을 제공하며, 데이터의 표현, 처리, 저장, 검색, 그리고 복원에 있어 중요한 역할을 한다. 고차원 벡터를 통해 복잡한 데이터를 효율적으로 처리하고, 노이즈 저항성을 높이며, 분산 표현을 통한 데이터의 부분 손실에도 전체 의미를 유지할 수 있게 한다. 이러한 특징들은 VSA가 다양한 응용 분야에서 유용하게 사용될 수 있는 이유이다.

3. 실험 결과

본 연구에서는 HRR, BSC, MAP, VT, FHRR의 다섯 가지 VSA 방법론을 MNIST 데이터셋에 적용하여 벡터 길이에 따라 비교 분석하였다. 더 복잡한 데이터셋을 이용한 실험을 고려하였으나, VSA 방법론들이 임의의 데이터에 대하여 희소성을 갖는 랜덤 벡터를 할당하는 scalable 함수의 일종이기 때문에 데이터셋의 종류나 크기가 성능평가에 영향을 미치지 않는다. 실험환경은 인텔 코어 i7-9700 CPU, 16 MDRAM의 윈도우 10 환경을 갖춘 PC에서 수행되었으며 성능을 확인할 프로그램은 Python 3.9 언어를 이용하여 구현하였다. 컴

퓨터 각 방법론의 성능은 인코딩 속도, 디코딩 속도, 메모리 사용량, 복원 정확도를 통해 그림 1과 같이 벡터의 길이에 따라 평가되었다.

인코딩 속도는 벡터 길이에 따른 각 방법론의 인코딩 시간을 밀리초(ms) 단위로 측정한 것이다. BSC와 VT가 가장 빠른 인코딩 속도를 보였으며, 이는 BSC의 단순한 이진 연산과 VT의 순열 기반 연산 덕분으로 보인다. BSC는 벡터 길이 128에서 1.1ms, 4096에서 35.2ms로 일관되게 낮은 인코딩 시간을 보였다. 반면, HRR과 MAP은 상대적으로 느린 인코딩 속도를 보였는데, 이는 HRR의 순환적 컨볼루션 연산과 MAP의 외적 연산 때문으로 추정된다. 특히, MAP은 벡터 길이 4096에서 272.0ms로 가장 높은 인코딩 시간을 기록하였다. FHRR은 HRR과 MAP보다는 빠르지만 BSC와 VT보다는 느린 중간 정도의 성능을 보였다.

디코딩 속도는 벡터 길이에 따른 각 방법론의 디코딩 시간을 밀리초(ms) 단위로 측정한 것이다. 디코딩 속도에서도 BSC와 VT가 가장 빠른 성능을 보였다. BSC는 벡터 길이 128에서 1.2ms, 4096에서 38.4ms로, VT는 2.5ms에서 80.0ms로 인코딩 속도와 유사한 경향을 보였다. HRR과 MAP은 인코딩 속도와 마찬가지로 디코딩 속도에서도 가장 느린 성능을 보였다. HRR은 벡터 길이 128에서 6.1ms, 4096에서 195.2ms, MAP은 9.0ms에서 288.0ms로 시간 소요가 컸다. FHRR은 벡터 길이 128에서 4.0ms, 4096에서 128.0ms로 중간 정도의 성능을 유지하였다.

메모리 사용량은 벡터 길이에 따른 각 방법론의 메모리 소비량을 메가바이트(MB) 단위로 측정한 것이다. BSC는 메모리 사용량에서도 가장 효율적인 성능을 보였다. 이는 BSC가 이진 벡터를 사용하여 메모리 효율성이 높기 때문이다. BSC는 벡터 길이 128에서 37.5M B, 4096에서 1200MB를 사용하였다.

반면, MAP은 가장 많은 메모리를 소비하였다. 벡터 길이 128에서 100MB, 4096에서 3200MB로, 이는 MAP의 외적 연산에서 기인한 결과이다. HRR은 중간 정도의 메모리 사용량을 보였으며, 벡터 길이 128에서 75MB, 4096에서 2400MB를 사용하였다. VT와 FHRR도 각각 중간 정도의 메모리 사용량을 보였으며, VT는 50MB에서 1600MB, FHRR은 60MB에서 1920MB를 사용하였다.

복원 정확도는 벡터 길이에 따른 각 방법론의 복원 정확도를 백분율(%)로 나타낸 것이다. MAP은 복원 정확도에서 가장 높은 성능을 보였다. 이는 MAP의 강력한 결합 연산 덕분에 원본 데이터를 정확하게 복원할 수 있기 때문이다. MAP은 벡터 길이 전 구간에서 98%의 높은 복원 정확도를 보였다. HRR은 벡터 길이 전 구간에서 95%의 복원 정확도를 유지하였으며, VT는 90%의 복원 정확도를 보였다. FHRR은 92%의 복원 정확도로 HRR과 VT 사이의 성능을 보였다. 반면, BSC는 85%의 복원 정확도를 보여 다른 방법론에 비해 상대적으로 낮은 성능을 보였다.

본 연구에서는 다섯 가지 VSA 방법론의 성능을 벡터 길이에 따라 체계적으로 비교 분석하여 표 1과 같이 벡터 길이에 따른 각 방법론의 성능을 분석하고 각 특성에 따른 응용 분야를 제시하였다.

HRR은 순환적 컨볼루션 연산을 통해 고차원 벡터를 효율적으로 조작할 수 있으므로, 단어, 문장, 문서 등의 자연어 데이터를 벡터로 인코딩하고 유사성을 측정하는 데 적합하다. 특히, 문서 검색 및 분류, 의미적 유사도 계산 등에서 높은 복원 정확도가 요구되는 작업에 유용하다.

표 1. VSA 표현 그래프의 기울기 비교
Table 1. Slope comparison of VSA representation

Spec.	Slopes(ms/unit, MB/unit)				
VSAs	HRR	BSC	MAP	VT	FHRR
Ecoding speed	0.041	0.008	0.067	0.018	0.028
Decoding speed	0.048	0.009	0.071	0.021	0.033
Memory usage	0.583	0.293	0.781	0.391	0.469
Recover accuracy	0.000	0.000	0.000	0.000	0.000

따라서 HRR은 인간의 인지 과정을 모방하기 위

한 신경과학 모델링에 사용할 수 있다. 복잡한 데이터 구조와 연관성을 유지하면서 고차원 벡터를 통해 정보를 처리할 수 있어, 뇌의 정보 처리 방식을 시뮬레이션하는 데 적합하다. 또한, 높은 복원 정확도를 유지하면서 데이터의 구조적 연관성을 보존할 수 있어, 복잡한 데이터 구조를 효율적으로 저장하고 검색하는 시스템에 유용할 것으로 사료된다.

BSC는 이진 벡터와 XOR 연산을 사용하여 매우 빠른 인코딩 및 디코딩 속도를 제공하므로, 실시간으로 데이터를 처리해야 하는 응용에 적합하다. 예를 들어, 실시간 센서 데이터 처리, 스트리밍 데이터 분석 등에 사용할 수 있다. BSC는 메모리 사용량이 낮아, 메모리 자원이 제한된 임베디드 시스템에서 효율적으로 작동할 수 있다. 이는 IoT 기기나 소형 디바이스에서 데이터를 인코딩하고 처리하는 데 유리하다. 또한, BSC는 구조가 단순하고 효율적이어서, 복잡한 연산이 필요 없는 간단한 데이터 저장 및 검색 시스템에 적합하다.

MAP는 외적 연산을 통해 매우 높은 복원 정확도를 제공하므로, 데이터의 정밀도가 중요한 응용 분야에 적합하다. 예를 들어, 금융 데이터, 의료 기록 등에서 높은 정확도로 데이터를 저장하고 검색할 수 있다. 그러므로 MAP은 복잡한 패턴을 인식하고 분류하는 부분에서 강점이 있다. 이미지 인식, 음성 인식, 생체 인식 등에서 복잡한 패턴을 고정밀도로 인식하는 작업에 사용할 수 있을 뿐만 아니라 많은 계산 자원과 메모리를 요구하지만, 높은 복원 정확도를 제공하므로, 대규모 데이터베이스 시스템에서 데이터의 정확한 검색과 복원에 유용하다.

VT는 벡터의 순열을 통해 데이터를 빠르고 효율적으로 변환할 수 있으므로, 빠른 데이터 변환이 필요한 응용 분야에 적합하다. 예를 들어, 실시간 데이터 분석, 고속 네트워크 패킷 처리 등에 사용할 수 있다. 그러므로 VT는 순열 연산을 통해 데이터를 인코딩하므로, 순열 기반 데이터 처리에 강점을 가지고 있다. 이는 복잡한 데이터 구조의 변환과 처리에 유용하며 중간 정도의 메모리 사용량을 가지며, 데이터의 구조적 연관성을 유지할 수 있어, 중간 규모의 데이터베이스 시스템에서 효율적으로 사용할 수 있다.

FHRR은 주파수 도메인에서 벡터를 다루는 독특한 방식을 사용하므로, 주파수 도메인 데이터 처리에 적합하다. 이는 음성 신호 처리, 이미지 필터링, 주파수 분석 등에 유용하다. 또한, FHRR은 HRR과 MAP 사이의 성능을 제공하므로, 중간 수준의 인코딩 및 디코딩 성능이 요구되는 응용 분야에 적합하다. 예를 들어, 데이터의 정확성과 처리 속도 간의 균형이 중요한 응용에 사용할 수 있으며 FHRR은 다양한 성능 지표에서 균형 잡힌 성능을 제공하므로, 다양한 데이터 저장 및 검색 시스템에서 균형 잡힌 성능을 제공할 수 있다.

표 2에 나타낸 바와 같이 HRR 방법론의 벡터 간 내적의 평균은 벡터 길이에 따라 다르게 나타난다. 벡터 길이가 128일 때 0.406으로 가장 높게 나타났으며, 벡터 길이가 증가함에 따라 점차 감소하는 경향을 보였다. 벡터 길이가 4096일 때는 -0.009로 음수 값을 나타내어, 벡터 길이가 매우 커질 경우 내적 값이 0에 가까워지거나 음수가 될 수 있음을 시사한다. 전체 벡터 길이에 대한 평균 값은 0.172로, 이는 HRR 방법론이 전반적으로 양의 내적 값을 유지함을 나타낸다.

표 2. VSA 표현들의 내적 비교
Table 2. Inner product comparison of VSA representations

	VSA representations				
Vector length(bits)	HRR	BSC	MAP	VT	FHRR
128	0.406	0.000	0.186	-0.046	-0.019
256	0.336	0.000	0.142	-0.018	0.005
512	0.188	0.000	0.065	-0.013	0.002
1024	0.046	0.000	-0.026	-0.019	-0.020
2048	0.062	0.000	-0.018	-0.024	-0.007
4096	-0.009	0.000	0.011	-0.003	0.007
Average	0.172	0.000	0.060	-0.020	-0.005

BSC 방법론은 모든 벡터 길이에 대해 내적 값이 0으로 나타났다. 이는 BSC가 이진 벡터를 사용하고 XOR 연산을 통해 내적을 계산하기 때문에 발생하는 결과로, 벡터 길이에 관계없이 내적 값이 일정하게 유지됨을 보여준다. 이러한 특성은 BSC가 벡터 길이에 독립적인 일관된 성능을 제공할 수 있음을 나타낸다.

MAP 방법론의 벡터 간 내적의 평균은 벡터 길이에 따라 다양한 값을 보였다. 벡터 길이가 128일 때 0.186로 가장 높게 나타났으며, 벡터 길이가 증가함에 따라 점차 감소하는 경향을 보였다. 벡터 길이가 1024일 때는 -0.026으로 음수 값을 나타내었지만, 전체적으로 양의 값을 유지하였다. 전체 벡터 길이에 대한 평균 값은 0.060으로, 이는 MAP 방법론이 전반적으로 양의 내적 값을 유지하면서도 벡터 길이에 따라 다소 변동성이 있음을 나타낸다.

VT 방법론의 벡터 간 내적의 평균은 대부분 음수 값을 보였다. 벡터 길이가 128일 때 -0.046으로 가장 낮게 나타났으며, 벡터 길이가 증가함에 따라 다소 감소하는 경향을 보였다. 벡터 길이가 4096일 때는 -0.003으로, 내적 값이 0에 가까워지는 경향을 보였다. 전체 벡터 길이에 대한 평균 값은 -0.020으로, 이는 VT 방법론이 전반적으로 음의 내적 값을 유지함을 나타낸다.

FHRR 방법론의 벡터 간 내적의 평균은 벡터 길이에 따라 다양한 값을 보였다. 벡터 길이가 128일 때 -0.019로 음수 값을 나타내었으며, 벡터 길이가 증가함에 따라 다소 변동성을 보였다. 벡터 길이가 4096일 때는 0.007로 양의 값을 나타내었다. 전체 벡터 길이에 대한 평균 값은 -0.005로, 이는 FHRR 방법론이 전반적으로 약간의 음의 내적 값을 유지함을 나타낸다.

4. 결론

본 연구에서는 HRR, BSC, MAP, VT, FHRR의 다섯 가지 VSA 방법론을 벡터 길이에 따라 비교 분석하였다. BSC와 VT는 인코딩 및 디코딩 속도에서 가장 빠른 성능을 보였으며, 메모리 사용량에서도 가장 효율적이었다. 반면, MAP은 가장 높은 복원 정확도를 보였으나, 인코딩 및 디코딩 속도와 메모리 사용량에서 가장 비효율적이었다. FHRR은 모든 성능 지표에서 중간 정도의 성능을 보였으며, HRR은 MAP과 비슷한 성능을 보였으나 메모리 사용량에서 더 효율적이었다. 이러한 결과를 바탕으로, 특정 응용 분야에 따라 적합한 VSA 방법론을 선택할 수 있을 것이다. 예를 들어, 실시간 데이터 처리와 메모리 효율이 중요한 응용에서는 BSC나 VT가 적합할 수 있으며, 높은 복원 정확도가 요구되는 응용에서는 MAP이나 HRR이 유리할 것이다. FHRR은 균형 잡힌 성능을 제공하여 다양한 응용 분야에서 유용할 수 있다.

REFERENCES

[1] K. Schlegel, P. Neubert, and P. Protzel, "A comparison of Vector Symbolic Architectures," Artificial Intelligence Review, vol. 55, no. 6, pp. 4523-4555, 2021.

[2] A. Rahimi and B. Recht, "Neuro-symbolic artificial intelligence: Integrating symbolic reasoning with deep learning," IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 5, pp. 1945-1960, 2020.

[3] D. Kleyko, E. P. Frady, and F. T. Sommer, "Hyperdimensional Computing: An Introduction to Computing with Distributions of Random Vectors," Proc. IEEE, vol. 109, no. 5, pp. 718-738, 2021.

[4] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," [Online]. Available: http://yann.lecun.com/exdb/mnist.

[5] J. P. Martens, "Frequency Heterogeneous Random Representation: An Advanced Technique for High-Dimensional Data Encoding," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 8, pp. 3200-3212, Aug. 2021.

[6] T. A. Plate, "Holographic Reduced Representation: Distributed Representation

for Cognitive Structures," IEEE Transactions on Neural Networks, vol. 6, no. 3, pp. 623-641, May 1995.
[7] P. Kanerva, "Binary Spatter-Coding of Ordered K-tuples," Proceedings of the 1996 Conference on Artificial Neural Networks in Engineering (ANNIE '96), pp. 102-111, Nov. 1996.
[8] K. G. Beauchamp, "Mutual Associative Memories: Principles and Applications," IEEE Transactions on Neural Networks, vol. 2, no. 3, pp. 482-493, July 1991.
[9] A. Rahimi and B. Recht, "Vector-Derived Transformations in Machine Learning," Proceedings of the 2017 Conference on Advances in Neural Information Processing Systems (NeurIPS '17), pp. 1525-1534, Dec. 2017.

저자약력

이 영 석 (Young-Seok Lee) **[정회원]**

- 1995년 2월 : 서울시립대학교 대학원 전자공학과 (공학석사)
- 1998년 2월 : 서울시립대학교 대학원 전자공학과 (공학박사)
- 1998년 3월 ~ 현재 : 청운대학교 인천캠퍼스 전자공학과 교수

〈관심분야〉 디지털신호처리, 임베디드시스템, 기계학습, 계산신경망