학습하는 바이오분자기계

(Biomolecular Learning Machines)

▌장병탁 / 서울대학교 컴퓨터공학부 교수

생물학은 학습하는 기계에 대한 여러가지 모델을 제공하였으며 신경망과 유전자 알고리즘이 그 대표적인 예이다. 그런데 이러한 학습 방식은 자연계로부터 영감만을 제공받아 현재의 실리콘 컴퓨터상에서 구현할 뿐, 자연계 물질 본유의 물리적인 특성을 활용하지는 못하였다. 본고에서는 자연계의 바이오분자를 직접 정보저장 및 계산 매체로 이용용함으로써 그 물리 화학적인 특성에 기반하여 성능이 향상되는 학습기계를 만들기 위한 분자학습기술을 소개한다. 이러한 기계학습 장치는 액체상에서 3차원 생화학반응에 의하여 동작하며 최신 분자생물학 실험기술을 사용하여 초병렬적으로 열역학적인 계산을 수행함으로써 기존의 기계학습 방식에서의 여러 기술적인 문제점들을 해결할 수 있는 새로운 돌파구를 제공한다. 또한 생체분자가 지닌 초소형 초고밀도의 정보저장 능력, 초병렬적 공간 탐색 능력, 생화학반응에 의한 자발적 일반화 능력, wet 상태의 생체 데이터를 직접 다룰 수 있는 능력이 Lab-on-a-Chip 등의 자동화 기술과 결합될 때 파생될 수 있는 새로운 무한한 응용 가능성과 경제 산업적인 파급 효과에 대하여 알아본다.

서 론

학습 능력은 지능적인 개체의 가장 대표적인 특징 중의 하나로서 학습하는 기계를 만들려는 시도가 오래전부터 있어 왔다. 생물학은 특히 새로운 지능형 계산 모델을 제시해 주었으며 신경망연산(neural computation)과 진화연산(evolutionary computation)이 그 대표적인 예이다(그림 1). 신경망은 신경계의 가소성에 기반한 적응성을 모방한 학습 방식이며 진화연산(또는 유전자 알고리즘)은 자연의 진화현상을 모델링한 적응적 병렬 탐색 방법이다. 그런데 이러한 연산 방식은 자연계의 정보처리 기작을 실리콘 컴퓨터상에서 소프트웨어적으로 시뮬레이션한 것이다. 보다 최근에는 자연계에 존재하는 물질의 물리적인 특성을 그대로 살려서 계산을 하는 자연계산(natural computation) 방식들이 연구되고 있으며, 그 대표적인 예가 분자컴퓨팅(mole-cular computing)이다.

기계학습 관점에서 분자컴퓨팅 기술은 여러 가지 시사점을 제시한다. 첫째, 분자컴퓨팅이 제공하는 막강한 공간적 병렬 탐색 특성은 현재 기계학습 기술에서의 한계를 극복하기 위한 방안을 제시해 준다. 둘째, 시험관안에서(in vitro)의 액체상태 생화학반응에 의한 연산은 기계학습에서 필요한 자발적인 일반화(spontaneous generalization)에 대한 자연스러운 기작을 제공한다. 이것은 현재 실리콘 반도체 기반의 컴퓨터기술로서 획득하기 어려운 자연의 능력이다. 셋째, 생체분자기반의 기계가 지닌 초소형 대용량 정보저장 및 초병렬 연상 기억 능력은 기계학습기술의 새로운 산업적 응용 지평을 열어줄 가능성을 지닌다. 넷째, 분자컴퓨팅의 wet 데이터 처리 능력 등은 기존의 실리콘 기술을 사용할 때의 데이터 변환 문제를 제거함으로써 특히 바이오데이터 처리에 있어서 새로운 응용을 생성할 가능성이 크다.

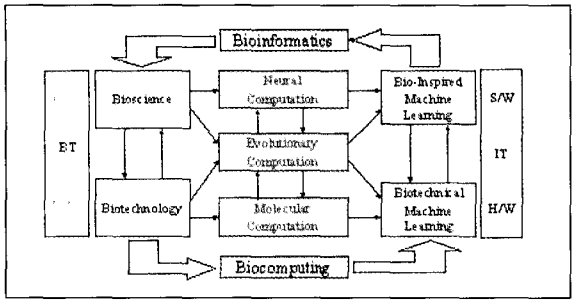

그림 1 생물학과 컴퓨터의 상호관계: 기계학습을 중심으로

본 고에서는 먼저 2절에서 기계학습 분야의 현재 기술 수준을 간략히 살펴 보고, 지속적인 발전에 대한 기술적인 장애 요인을 분석한다. 3절에서는 이에 대한 해결책의 하나로서 분자컴퓨팅 기술을 소개하고 그 특성을 실리콘기반의 컴퓨터 기술과 비교하여 살펴본다. 4절에서는 이를 활용한 새로운 기계학습 기술의 개념을 소개하고, 이동로봇의 학습 시나리오를 사용한 구체적인 예를 통해 이 문제를 분자컴퓨팅으로 해결하는 실험 결과를 제시한다. 5절에서는 학습하는 바이오분자장치에 대한 원천기술 개발의 필요성과 파급 효과 및 향후 연구 방향에 대해 토론한다.

현재 기계학습 기술의 문제점

학습시스템은 환경으로부터의 경험을 통해서 새로운 데이터를 관측함에 따라 스스로 성능이 향상되는 장치로 정의된다. 학습 기술은 여러 가지 기준에서 분류될 수 있다. 한 가지는 환경이 학습자에게 제공하는 피드백 정보의 종류에 따른 분류이다(그림 2). 이에 따르면 환경이 문제 x와 원하는 응답(desired output) d를 모두 제공해주는 감독학습(supervised learning), 환경이 문제 x만을 제공하고 답은 제시해 주지 않는 무감독학습(unsupervised learning), 문제를 제공하고 학습자가 출력(actual output) y를 제시하면 이에 대한 보상치(reward) r을 제시해 주는 강화학습(reinforcement learning)이다.

학습 기술은 또한 습득된 지식에 대한 표상(representation)에 따라 나눌 수 있다. 일부 학습 방식은 이산적인 구조를 사용하여 학습 결과를 표현한다. 결정리스트(decision list)는 속성-속성값(attribute-value) 쌍들의 조합이나 if-then 규칙의 집합으로 지식을 표현하며, 결정트리(decision tree)는 속성값들을 그 값에 따라 계층적인 트리 형태로 표현함으로써 의사결정 등에 필요한 지식을 표현한다. 다른 학습 방식은 연속값을 갖는 함수 형태로 지식을 표현한다. 다층신경망(multilayer perceptron)의 경우 단순한 함수(function)를 계산하는 인공뉴런들이 적응적인 파라메터로 연결된 계층망 구조이다. 확률그래프모델(probabilistic graphical model)은 확률변수에 해당하는 노드들이 연결선으로 구성된 일반적인 그래프 구조를 하고 있다. 노드들은 조건부 확률에 따라 상관관계를 가진다. 확률그래프모델의 장점은 임의의 변수의 값들을 관측하였을 때 다른 임의의 변수의 값들을 확률적 추론에 의해 계산할 수 있다는 것이다.

학습 기술을 구별하는 또 다른 기준은 표상을 수정하는 추론 방법이다. 경험으로부터의 학습은 구체적인 사례들로부터 일반적인 규칙을 발견해 내는 귀납적인 추론에 기반하는데, 이 때 사용되는 추론 방식이 학습의 효과와 효율을 결정하는데 중요하

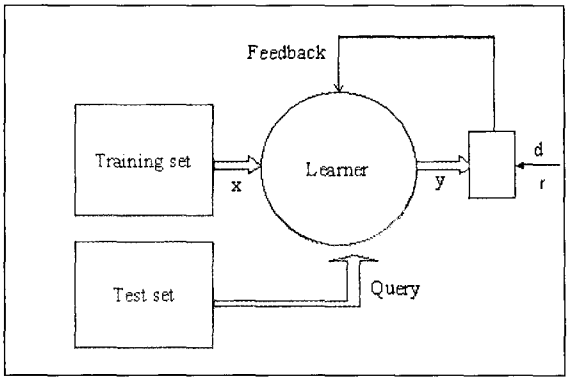

그림 2 감독 학습, 무감독 학습, 강화 학습. x: 입력(문제), y: 학습자의 출력(actual output), d: 목표 응답(desired output), r: 보상치(reward). 세 경우 모두 입력 x가 주어지며, 감독학습의 경우 d가 강화학습의 경우는 r이 추가로 주어지나 무감독학습의 경우에는 아무 feedback 정보도 추가로 주어지지 않는다.

다. 이산적인 구조를 사용하는 결정리스트(decision list)의 경우 generalization operator와 specialization operator 등과 같은 연산자를 써서 일반화 또는 특수화를 통해 이를 수행한다. 예를 들면, 상수 대신에 변수를 대치함으로써 일반화를 실현할 수 있다. 결정트리(decision tree)의 경우 하나의 속성을 가진 트리 즉 일반적인 표상으로부터 출발하여 주어진 학습 예를 반복적으로 사용하면서 속성값의 개수를 늘려나감으로써 즉 트리를 성장시킴으로써 점점 더 특수한 표현으로 변경한다. 이때 트리에 추가시킬 속성을 선택할 기준으로 엔트로피에 기반한 목적 함수를 사용한다. 다층신경망의 경우 망에 존재하는 모든 시냅스 가중치들의 벡터공간상에서 기울기강하(gradient-descent)에 의한 탐색을 함으로써 표현을 변경한다. 이 때 주어진 학습예를 반복적으로 사용하여 에러가 최소화하는 방향으로 탐색이 진행된다. 확률그래프 모델의 경우 목적함수가 확률분포로 주어지며 역시 주어진 학습 예들을 반복적으로 사용함으로써 관측데이터의 확률 분포와 그래프모델의 확률 분포가 일치하도록 모델의 파라메터를 변경한다.

현재 기계학습 기술에서의 한 가지 문제점은 모델의 학습에 많은 계산 시간이 소요되어 학습할 수 있는 모델의 크기가 제한된다는 것이다. 이는 학습할 모델의 포함된 변수의 수가 늘어남에 따라 학습데이터와 일치하는 모델을 선택하고 파라메터를 수정하는데 걸리는 계산 시간이 급격히 증가하기 때문이다. 예를 들어, 신경망의 경우 현재의 실리콘 컴퓨터 기술로 백만개(10^6) 정도의 뉴런을 가진 임의의 망 구조의 학습을 시도하는 것은 실제로는 불가능하다. 그러나 백만개의 노드로 구성된 망은 인간의 두뇌에 존재하는 10^11개의 뉴런수에 비하면 십만분의 1의 크기 즉 0.001 %이다. 확률 그래프 모델의 경우에도 계산상의 제약은 마찬가지이다. 확률분포를 추정하기 위하여 sampling 등의 방법이 사용되는데 이러한 계산은 보통 많은 시간을 요하여 현재로서는 워크스테이션상에서 10000개 정도의 노드를 가진 그래프모델을 학습하는 것이 가능한 정도이다.

분자컴퓨팅의 특성

DNA 기반 분자컴퓨팅은 기본적으로 DNA 분자를 합성함으로써 정보를 코딩하여 저장하고, DNA가 가진 화학적 특성을 이용하여 정보를 처리하는 새로운 컴퓨팅 방식이다. 약 1 그램의 DNA는 10^{21}개의 DNA 염기를 가지며 따라서 10억 terabits의 정보저장 능력을 지닌다. 1 Mole의 DNA 수용액에는 아보가드로수 만큼의 즉 6×10^{23}개의 분자를 가지고 있으며 이들은 용액상에서의 화학 반응에 의해 초병렬적 정보처리가 가능하다. 분자정보처리기술은 이러한 많은 수의 초미세구조의 연산 소자가 초고집적도로 모여서 정보를 저장하고 초병렬적으로 처리함으로써 기존의 실리콘 기술로서 불가능한 정보처리 능력을 발휘할 수 있는 잠재력을 지니고 있다. 〈그림 3〉은 DNA 분자컴퓨팅의 특성을 요약한 슬라이드이다.

DNA는 뉴클레오티드(nucleotide)로 구성되어 있으며, 단일가닥(single strand)의 DNA 조각을 올리고라고 하며 현재 바이오텍 기술로 임의의 올리고 서열을 합성할 수 있다. 단일가닥 DNA들은 보통 Watson-Crick 상보성에 의해 상호 결합함으로써 이중사슬구조(double strand) DNA를 형성한다. 이 반응은 용액상에서 초병렬적으로 일어나며 이를 혼성

Why DNA Computing?

- 6.022×10^{23} molecules/mole
- Immense, brute force search of all possibilities
  - Desktop: 10^9 operations/sec
  - Supercomputer: 10^{12} operations/sec
  - 1*m* mol of DNA: 10^{26} reactions
- Favorable energetics: Gibbs' free energy
  $\Delta G = -8kcal\ mol^{-1}$
- 1 *J* for 2×10^{19} operations
- Storage capacity: 1 bit per cubic nanometer

그림 3 DNA 분자컴퓨팅의 특성

화(hybridization)라 한다. 또한 온도를 (예를 들어, 섭씨 90도 정도로) 높여주면 두 가닥으로 구성된 DNA가 다시 한 가닥으로 분리되며 이 반응을 dehybridization 또는 denaturation 또는 annealing이라고 한다.

〈그림 4〉은 혼성화 반응을 통해 세 조각의 단일 가닥이 두 조각의 이중사슬 구조가 되는 예를 보여준다. 이때 하나의 조각은 다른 두 개의 조각을 이어주는 다리 역할을 하고 있다.

병렬적인 정보처리의 한 예로 〈그림 5〉는 비드를 이용한 Affinity Separation 과정을 보여준다. 이는 보통 많은 수의 DNA 분자들이 혼합되어 있는 시험관내에서 원하는 DNA 분자만을 분리하기 위하여 사용된다. 자화된 비드에 원하는 분자 서열의 상보 서열을 부착하고 이를 시험관에 넣어 혼성화 반응을 시킨 후 자기장을 걸어서 비드를 회수함으로써 원하는 분자들을 한꺼번에 추출한다.

현재 사용되는 대부분의 DNA 컴퓨팅 방식의 기

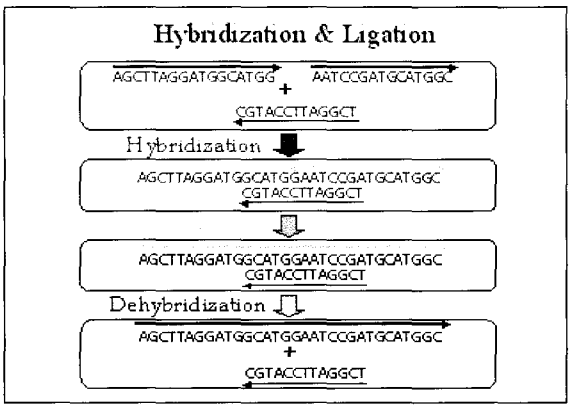

그림 4 DNA 혼성화 반응

본 원리는 다음과 같다.

(Step 1) 풀고자 하는 문제에 대한 가능한 해답을 DNA 코드로 표현한다.

(Step 2) 이들 코드를 올리고 합성 기술을 사용하여 다량(보통 10^15 이상) 합성한다.

(Step 3) 각각의 성분 분자들을 합성기에 넣고 화학반응을 시킴으로써 가능한 모든 해를 생성한다.

(Step 4) 생성된 분자들 중에 찾는 해가 포함되었는지를 검사하여 답을 제시한다.

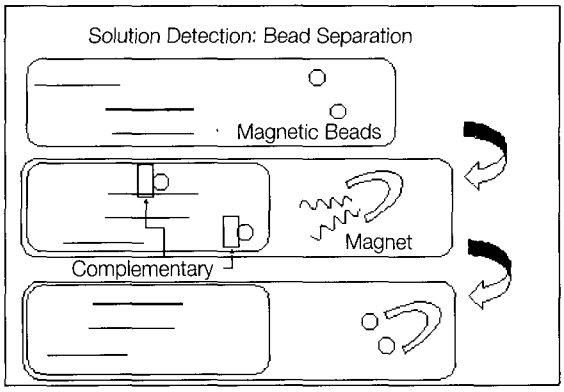

그림 5 비드를 사용한 특정 DNA 분자의 분리

이러한 원리를 이용하여 NP-complete 문제인 해밀톤 경로 문제를 해결할 수 있음이 1994년에 Adleman에 의해 처음으로 실험적으로 증명되었고, 그 이후 여러 가지 계산 문제 및 다양한 응용에 대해 DNA 컴퓨팅이 연구되고 있다.

바이오분자컴퓨터의 특징을 기존의 실리콘 컴퓨터 기술과 비교하여 요약하면 다음의 표 1과 같다.

분자컴퓨팅기반 기계학습

앞에서 정의한 바와 같이 학습은 새로운 데이터가 관측됨에 따라서 기존의 지식표현을 변경함으로써 성능을 향상시켜 나가는 과정이며, 여기서는 이 과정이 DNA 분자를 사용함으로써 어떻게 개선될 수 있는지를 살펴본다. 설명의 편의를 위해서 간단한 학습 시나리오를 하나 들어서 이 문제를 DNA 컴퓨팅으로 해결하는 절차와 실험 결과를 제시한다. 보다 구체적으로 다음과 같은 방식으로 기술한다.

(1) 개념 학습 문제를 정의하고 이것이 집합 연산 문제로 정형화될 수 있음을 보인다.

(2) DNA 컴퓨팅을 적용하여 한 번의 실험 단계로 병렬적 집합 연산을 수행한다.

표 1 실리콘 컴퓨터와 바이오분자 컴퓨터의 특성 비교

특성	실리콘 컴퓨터	바이오분자 컴퓨터
소자의 속도	빠름	느림
연산의 병렬성	제한적	초병렬성
소자 물성	고체	액체
회로의 형태	2차원 회로	3차원
회로 구조	고정됨	무정형성
연산의 재현성	결정적인 재현	확률적인 재현
연산 정확도	극히 정확	비교적 정확
계산 절차	순차적	초병렬적
데이터 전달 방식	전자적	화학적
프로그램 가능성	범용성	특수성
데이터 입력	dry 전자 신호	wet 상태 가능
데이터 출력	모니터	Gel 사진
집적도	우수	극히 우수
정보저장능력	우수	극히 우수
생체적합성	부적합	적합
소형화가능성	우수	극히 우수
재사용성	쉬운문제의 반복 해결	고난도 문제 일회적 해결
연산 방식	결정적	확률적
열역학적 요인	열역학적 변화 독립적	열역학적 변화 활용
HW/SW의 분리	차이가 큼	차이가 적음
발전의 역사	50년 이상	10년 미만
연산 절차	복잡	단순
기본 명령어의 수	많음(수백)	적음(수십)

(3) 이는 하나의 예제 학습에 필요한 연산이 상수시간에 수행될 수 있음을 보이는 것이다.

(4) 질의에 대한 답을 하는데는 두 번의 집합 연산이 필요하면 역시 상수시간에 수행된다.

(5) 총 n개의 학습 예제가 주어지는 문제의 경우 O(n)의 집합 연산이 필요하다.

결과적으로 전형적인 기계학습 문제인 귀납적인 개념학습 문제가 분자컴퓨팅을 적용함으로써 O(n)의 집합 연산을 수행함으로써 해결될 수 있음을 보인다.

이동로봇의 개념학습: 시나리오

건물의 사무실을 방문하며 쓰레기통을 비우는 자율이동 로봇을 생각해 보자. 이 로봇의 임무는 4층과 5층에 있는 컴퓨터공학부와 전기공학부 교수 및 행정직원들의 방을 매일 돌아다니며 쓰레기통에 있는 재활용 캔을 수거하는 것이다(이 예는 Dean의 인공지능 교과서의 예제를 변형한 것임). 지능형 로봇이라면 방을 방문할 때마다 경험으로부터 학습을 통하여 "재활용 캔이 있는 사무실"이라는 개념을 계속 변경하여 갈 것이다. 이는 개념 학습의 문제이다. 이 개념을 표시하기 위해 다음과 같은 세 개의 속성을 생각해 보자.

Dept = {cs, ee}

Status = {prof, staff}

Floor = {four, five}

여기서 Dept, Status, Floor는 속성의 이름이며 각각 두 가지씩의 속성값을 갖는다. 이 문제에 대해 로봇이 학습할 모든 가능한 개념의 종류는 〈그림 6〉과 같은 개념 그래프로 나타낼 수 있다.

위의 계층적 그래프 구조에서 상위의 노드는 "보다 일반적인"(more general) 개념을 나타내고 하위의 노드는 "보다 특수한"(more specific) 개념을 나타낸다. 예를 들어, 〈cs〉는 〈cs, faculty〉 보다 일반적이고 〈cs, faculty〉는 〈cs, faculty, four〉보다 더 일반적이다. 반대로 〈cs, faculty, four〉는 〈cs, faculty〉 보다 더 특수하고 이는 다시 〈cs〉 보다 더 특수하다. 개념 표현을 정형화시키기 위해서 아래에서는 〈cs〉는 〈cs, *, *〉로 〈cs, faculty〉는 〈cs,

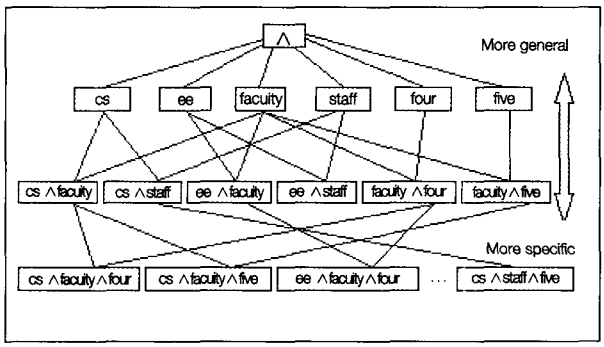

그림 6 로봇학습 문제에 대한 개념 그래프

faculty, *〉로 don' t care 기호 *를 사용하여 표시하기로 한다.

개념 학습의 절차를 설명하기 위해서 다음과 같은 학습 시나리오를 하나 예를 들어보자.

1. 정례 〈cs, faculty, four〉+ 가 주어진다.
2. 반례 〈cs, staff, five〉- 가 주어진다.
3. 질의 〈cs, staff, four〉? 가 주어진다.

위에서 〈...〉의 오른쪽에 부가된 + 표시는 정례(positive example) 즉 이 예가 학습하는 개념에 속한다는 것을 표시한다. 마찬가지로 - 표시는 반례(negative example) 즉 이 예는 학습하고자 하는 개념에 속하지 않음을 나타낸다. 그리고 ? 표시는 질의query)에 해당함을 나타내는 기호이다. 따라서 예제들이 의미하는 바는 다음과 같다.

〈cs, faculty, four〉+ : "4층에 있는 컴퓨터과 교수의 방에는 재활용 캔이 있다"

〈cs, staff, five〉- : "5층에 있는 컴퓨터과 직원의 방에는 재활용 캔이 없다"

〈cs, staff, four〉? : "4층에 있는 컴퓨터과 직원의 방에는 재활용 캔이 있는가?"

이 문제를 해결하는 한 가지 방법은 예가 주어질 때마다 지금까지 관측한 예들과 일치되는 개념만을 유지하는 즉 버전공간(version space)를 유지하는 것이다. 즉 버전 공간은 매순간 정례는 모두 포함하고 반례는 하나도 포함되지 않아야 한다. 이 절차를 요약하면 다음과 같다.

(Step 1) 모든 가능한 개념들을 생성한다. 이들의 집합을 현재의 버전공간 A라 하자.

(Step 2) 새로운 예 x가 주어지면 이와 일치하는 모든 개념의 집합 B를 생성하고 x의 종류에 따라 다음과 같이 처리한다.

(Step 2.1) x가 정례이면 새로운 버전공간 A를 $A \leftarrow A \cap B$로 계산한다.

(Step 2.2) x가 반례이면 새로운 버전공간 A를 $A \leftarrow A - B$로 계산한다.

(Step 2.3) x가 질의이면 A와 B로부터 $Y \leftarrow A \cap B$와 $N \leftarrow A - B$계산하고 $|Y| > |N|$이면 Yes, 아니면 No라고 대답한다.

(Step 3) 위의 단계 (2)로 돌아가서 계속한다.

위의 절차를 좀 더 자세히 살펴보자면, 처음에 속성들로 표현될 수 있는 모든 가능한 개념들을 포함하는 버전 공간 A로 출발한다. 만약 정례가 들어오면 (이는 버전공간상에서 포함되어야 하므로) 현재 버전공간상에서 이것과 일치하는 개념만을 남기고 나머지 개념들을 제거하며(Step 2.1), 반례가 들어오면 (이는 버전공간상에 포함되면 안되므로) 현재 버전공간상에서 이것과 일치하는 개념들을 모두 제거한다(Step 2.2). 이 연산은 현재 버전 공간 A와 새로운 예 x를 포함하여 이 보다 더 일반적인 모든 개념들로 구성된 개념집합 B 사이의 집합 연산으로 형식화될 수 있으며, 각각 A와 B의 교집합과 차집합을 계산하는 것과 동등하다. 질의에 대해서는 A와 B의 교집합(현재 버전공간과 일치하는 개념들)과 차집합(현재 버전공간과 일치하지 않는 개념들)을 모두 구한 후 이들의 크기를 비교하여 그 개념에 속하는 예인지를 결정한다(Step 2.3). 아래에서는 위의 연산이 DNA 컴퓨팅을 이용하여 초병렬로 수행될 수 있음을 보인다.

DNA 분자컴퓨팅에 의한 기계학습

위의 로봇 학습 문제와 같이 작은 문제의 경우 전체 버전공간을 집합으로 표시할 수 있고 이 과정을 예로 들어 기술한다. 먼저 DNA 구현을 위해서 개념들을 DNA 서열에 코드화 한다. 각각의 속성은 속성이름을 표시하기 위해 20bp를 가지며 속성간의 연결을 위해 추가의 sticky end를 가지도록 설계하였다(그림 7). 이는 3개의 속성이 순서대로 즉 〈Dept, Status, Floor〉와 같이 생성되도록 하기 위함이며 가운데에 있는 속성인 Status의 경우 양쪽에 10mer 씩의 sticky end가 있고 다른 속성들은 Status와 연결부위 쪽에만 sticky end를 갖도록 설계되었다. 이 코딩 방식에 의하면 모든 개념은 20+10+20+10+20 = 80bp로 표시된다.

〈그림 8〉은 DNA 코드 설계 소프트웨어인 NACST

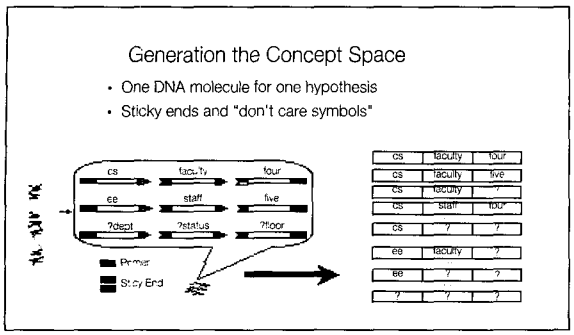

그림 7 DNA 코딩 방식 및 초기 개념 공간의 생성

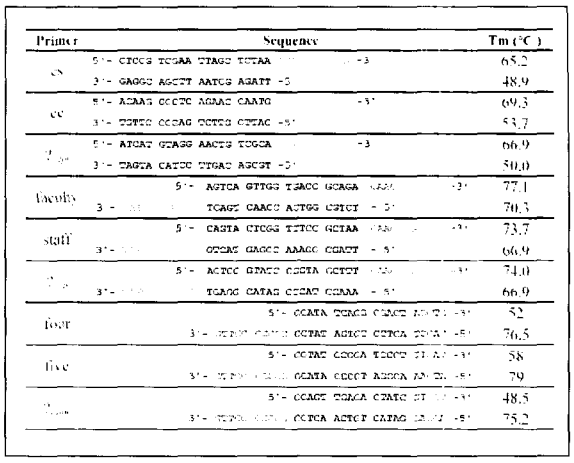

그림 8 NACST를 사용하여 설계된 코드

를 사용하여 결정한 실제 실험에 사용한 서열을 보여준다. 여기서 ?dept, ?stat, ?floor는 각각 해당 속성에 대한 don't care 변수를 나타낸다. 표의 오른쪽 끝에 기술된 Tm은 melting temperature이며 이는 이중가닥 DNA의 50%가 단일가닥으로 분리될 때의 온도를 표시한다.

Step 1: 초기에 모든 가능한 개념들의 공간은 다음과 같이 구성된다.

A = {⟨*, *, *⟩, ⟨cs, *, *⟩, ⟨ee, *, *⟩, ⟨*, faculty, *⟩, ⟨*, staff, *⟩, ⟨*, *, four⟩, ⟨*, *, five⟩, ⟨cs, faculty, *⟩, ⟨cs, staff, *⟩, ⟨cs, *, four⟩, ⟨cs, *, five⟩, ⟨ee, faculty, *⟩, ⟨ee, staff, *⟩, ⟨ee, *, four⟩, ⟨ee, *, five⟩, ⟨*, faculty, four⟩, ⟨*, faculty, five⟩, ⟨*, staff, four⟩, ⟨*, staff, five⟩, ⟨cs, faculty, four⟩, ⟨cs, faculty, five⟩, ⟨cs, staff, four⟩, ⟨cs, staff, five⟩, ⟨ee, faculty, four⟩, ⟨ee, faculty, five⟩, ⟨ee, staff, four⟩, ⟨ee, staff, five⟩}

⟨그림 9⟩는 초기의 버전 공간이 올바로 생성되었는지를 실험적으로 검증하기 위한 Gel 사진이다. 사진의 제일 왼쪽 레인(lane)은 DNA의 길이를 표시하는 마커이며, 오른쪽의 레인에서 80bp에 해당하는 부분의 색이 가장 밝은 것으로부터 생성된 코드들의 길이가 원하는 바와 같이 80bp가 대부분임을 확인할 수 있다.

Step 2: 정례 ⟨cs, faculty, four⟩+ 가 주어지면 이 예와 일치하는 개념 B를 계산한다.

B = {⟨cs, *, *⟩, ⟨*, faculty, *⟩, ⟨*, *, four⟩, ⟨cs, faculty, *⟩, ⟨cs, *, four⟩, ⟨*, faculty, four⟩, ⟨cs, faculty, four⟩}

현재의 버전공간 A와 위의 B로부터 갱신된 새로운 버전공간은 A←A∩B로 계산될 수 있으며 그 결과는 다음과 같다.

A = {⟨cs, *, *⟩, ⟨*, faculty, *⟩, ⟨*, *, four⟩, ⟨cs, faculty, *⟩, ⟨cs, *, four⟩, ⟨*, faculty, four⟩, ⟨cs, faculty, four⟩}

결과적으로 주어진 정례와 일치하는 개념만 남고 나머지는 버전공간에서 제거되었다. 이 연산에 대한 실험 결과는 ⟨그림 10⟩에 나와 있다.

Step 3: 다음으로 반례 ⟨cs, staff, five⟩- 가 주어지면 이와 관련된 개념의 집합 B가 다음과 같이 계산된다.

B = ⟨cs, *, *⟩, ⟨*, staff, *⟩, ⟨*, *, five⟩, ⟨cs, staff, *⟩, ⟨cs, *, five⟩, ⟨*, staff, five⟩, ⟨cs, staff, five⟩

현재의 버전공간 A와 위의 B로부터 학습된 새로운 버전공간 A←A—B로 계산되며 결과는 다음과 같다.

A = {⟨*, faculty, *⟩, ⟨*, *, four⟩, ⟨cs, faculty, *⟩, ⟨cs, *, four⟩, ⟨*, faculty, four⟩, ⟨cs, faculty, four⟩}

Step 4: 이제 질의어 ⟨cs, staff, four⟩? ("4층에 있는 컴퓨터공학과 직원의 방에는 재활용 캔이 있는가?") 가 주어진다. 이 질의와 일치하는 개념들은 다음의 B와 같다.

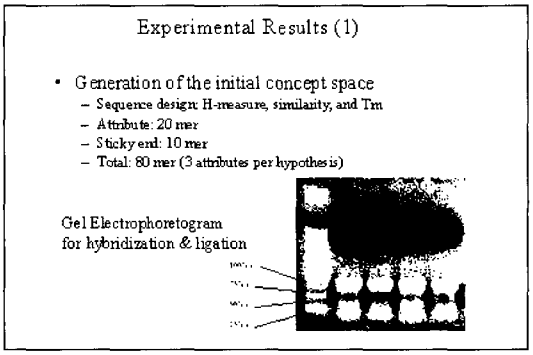

그림 9 초기에 생성된 개념 공간에 대한 Gel 사진

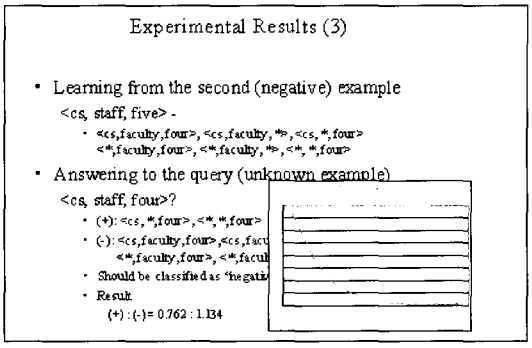

그림 11 두 번째 예제를 학습한 후의 결과

B = {⟨cs, *, *⟩, ⟨*, staff, *⟩, ⟨*, *, four⟩, ⟨cs, staff, *⟩, ⟨cs, *, four⟩, ⟨*, staff, four⟩, ⟨cs, staff, four⟩}

이에 대한 해답을 알기 위해 현재 버전공간 A 내의 개념들 중에서 B를 정례로 생각하는 개념의 갯수와 B를 반례로 생각하는 개념의 갯수가 얼마나 있는가를 상대적으로 비교하여 더 많은쪽을 답이라고 결정하는 것이다. 이를 위해 먼저 Y←A∩B와 N←A−B를 계산하며 그 결과는 다음과 같다.

Y = {⟨*, *, four⟩, ⟨cs, *, four⟩}

N = {⟨cs, *, *⟩, ⟨*, staff, *⟩, ⟨cs, staff, *⟩, ⟨*, staff, four⟩, ⟨cs, staff, four⟩}

이제 두 개의 집합 Y와 N의 크기를 비교하여 |Y| 〈 |N|이므로 No 라고 대답한다. 즉 질의 예제 ⟨cs, staff, four⟩는 재활용 캔이 없는 방으로 로봇이 판단한다. 이에 대한 실험 결과는 ⟨그림 12⟩에 제시되어 있다.

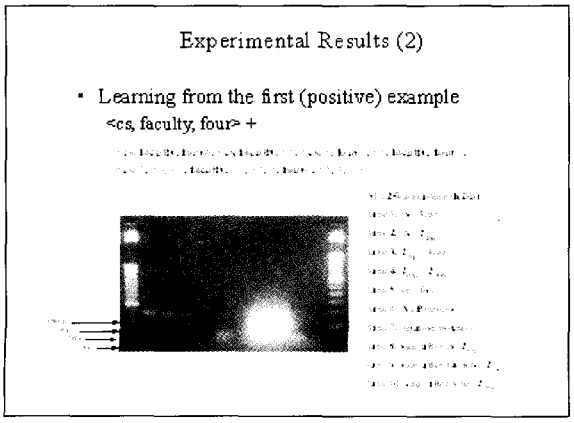

그림 10 첫 번째 예제를 학습한 후의 개념 공간에 대한 Gel 사진

결론 및 전망

기계학습 기술의 현재 문제점을 살펴보았으며 DNA 분자컴퓨팅 기술이 이 문제에 대해 어떤 해결책을 제시해 줄 수 있는지를 구체적인 실험 결과를 통해서 알아보았다. 지금까지의 기계학습 연구는 생물학으로부터 모델만을 제공받았으나 본고에서 살펴본 분자기반의 기계학습 기술은 생물학을 테크놀러지로 사용한 혁신적인 생화학적 학습기술이다. 기계학습과 관련된 바이오분자 컴퓨팅 연구는 향후 크게 세 가지 방향으로 진행될 수 있을 것으로 보인다.

한 가지는 분자컴퓨팅의 초병렬적 공간 탐색 능력을 이용하여 기존의 기계학습에서 필요한 컴퓨팅 파워를 보강하는 문제이다. 많은 기계학습 문제들이 계산 복잡도의 벽에 부딪혀 있으며 분자컴퓨팅은 이를 해결할 수 있는 한 가지 새로운 대안을 제시해 준다. 특히 최근에 연구되는 많은 기계학습 방식들이 확률 모델과 통계적 샘플링 기술을 필요로 하는 점은 통계 열역학적 원리에 기반한 확률적인 분자컴퓨팅 기술이 자연스러운 계산 방법으로 활용될 수 있음을 시사한다. 또한 분자컴퓨팅에서 사용되는 생화학반응이 자발적인 일반화 특성을 가지고 있는 점

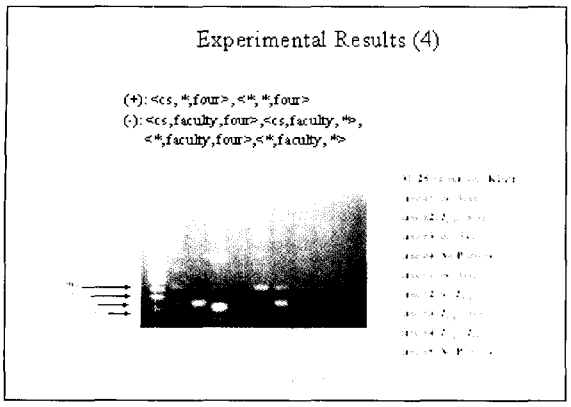

그림 12 질의에 대한 결과를 보여주는 Gel 사진

을 잘 활용할 경우 자연스러운 귀납적 학습을 수행할 수 있다.

다른 한 가지의 연구 방향은 분자기반 기계학습 기술을 생명공학 분야에 응용하는 것이다. Wet 상태의 DNA 분자로부터 직접 데이터를 입력 받아 처리할 수 있는 분자컴퓨팅의 특성을 활용할 경우, 기존의 실리콘 기반의 학습 기술로서는 불가능한 새로운 응용과 산업적 효과를 가져올 수 있을 것이다. 특히 생명공학 연구, 분석화학, 의료제약 분야 등에 응용될 수 있는 새로운 지능형 IT 장비 기술로서의 가능성은 우선 순위를 가질 만하다.

또 다른 연구 방향은 전통적인 기계학습 적용 분야에서 기존의 학습장치를 바이오분자기반의 학습장치로 대체하는 연구이다. 분자소자의 정보집적도를 활용하고 실험장치를 Lab-on-a-Chip 기술 등으로 자동화함으로써 초소형 학습장치를 실현할 수 있을 것이다. 이러한 연구는 또한 관련 BT, NT 기술 분야에 새로운 응용 및 벤치마킹 문제를 제공해 줌으로써 이들의 발전을 촉진하는 시너지 효과를 보게 될 것이다.

학습 능력은 인간을 비롯한 생명체의 대표적인 특성이다. 스스로 성능을 개선할 수 있는 학습시스템 기술은 다른 어떤 지능 기술보다 원천적이며 부가가치가 높은 첨단 기술로서 고도 지식정보화 사회에 있어서 경제 산업적인 파급 효과가 뛰어난 필수적인 기술이다. 특히 21세기의 선진화된 복지사회에서의 보건의료 산업의 비중이 커지는 점을 감안할 때 바이오분자 학습장치는 현재로서 우리가 상상하기 어려운 새로운 응용 영역들을 창출할 수 있을 것이다. 예를 들면, 세포내의 유전자 발현 및 분자활동을 감지하여 세포의 건강 상태를 예측하고 조절하는 학습능력을 갖춘 나노분자기계의 출현이 가능할 수도 있다. 이는 마치 반도체기반의 컴퓨터 기술이 40년 전에는 아무도 상상하지 못했던 분야에 현재 적용되고 있는 것과 유사한 상황이 될 것이다.

(감사의 글) 본연구는 산업자원부 차세대신기술사업 수퍼지능칩과제에 의하여 지원되었으며, 서울대 바이오지능 & 인공지능 연구실 및 한양대 프로테오믹스 연구실의 공동연구로 수행되었다. 본 고에 사용된 일부 실험 결과는 임희웅, 장해만, 채영규, 유석인, 장병탁 공저의 Eighth International Conference on DNA Based Computers (DNA8) 논문에서 발최하였다.