

제134회 정보관리기술사 해설집

2024.07.27

국가기술자격 기술사 시험문제

기술사 제 134 회

제 4 교시 (시험시간: 100 분)

분야	정보통신	자격종목	정보관리기술사	수검 번호		성 명	
----	------	------	---------	----------	--	--------	--

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 10 점)

1. IT 프로젝트 관리에서 리스크 대응에 대하여 설명하시오.

- 가. 리스크 대응 계획 수립 절차
- 나. 위협에 대한 대응 전략
- 다. 기회에 대한 대응 전략

2. 딥러닝에서 대규모 신경망을 효율적으로 훈련하기 위한 멀티 GPU 기술에 대하여 설명하시오.

- 가. 멀티 GPU 기술의 개념과 장점
- 나. 멀티 GPU 환경 구축 시 고려사항

3. AI 시스템에 대한 법적 이슈, 윤리적 문제, 기술적 문제에 대하여 설명하고 해결 방안을 제시하시오.

4. 개방형 API(Open API)에 대하여 설명하시오.

- 가. 정의 및 특징
- 나. SOAP 및 REST 구성요소
- 다. 취약점 및 대응 방안

5. 클라우드 전환 사업의 단계별 감리 방법과 검토 항목에 대하여 설명하시오.

6. 군집분석 기법인 SOM(Self Organizing Map)에 대하여 설명하시오.

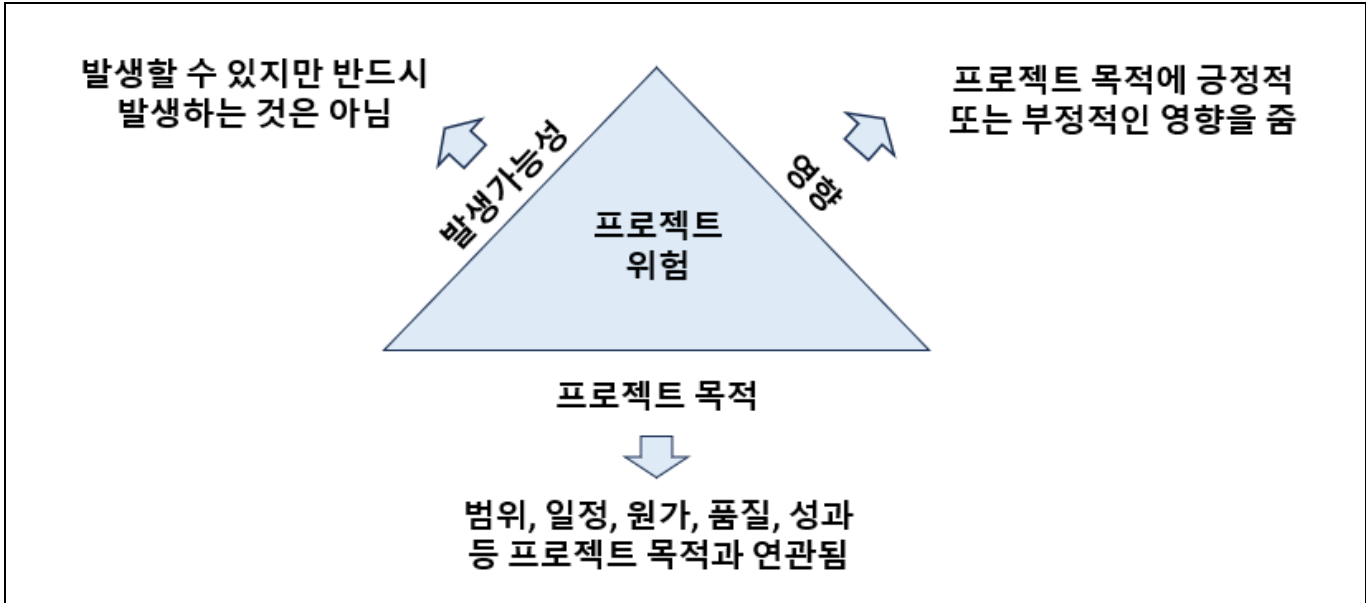
가. SOM 정의 및 특징

나. SOM 구성요소

다. SOM 과 신경망 분석기법의 차이점

01	리스크 관리		
문제	IT 프로젝트 관리에서 리스크 대응에 대하여 설명하시오. 가. 리스크 대응 계획 수립 절차. 나. 위협에 대한 대응 전략 다. 기회에 대한 대응 전략		
도메인	프로젝트관리	난이도	하(상/중/하)
키워드	위험, 회피, 전가, 완화, 수용, 에스컬레이션(Esclation)		
출제배경	위험관리 토픽이 지속적으로 기출되고 있어 위험관리의 전반에 대한 개념 이해 점검		
참고문헌	ITPE 서브노트		
해설자	강남평일야간반 전일 기술사(제 114회 정보관리기술사 / nikki6@hanmail.net)		

I. IT 프로젝트의 위험관리(Risk Management)의 개념



- 프로젝트 위험 식별, 분석 이에 대한 대응책 마련하여 프로젝트를 성공적으로 완료하기 위한 관리 활동

II. 리스크 대응 계획 수립 절차

프로세스	절차	설명	산출물
계획	위험관리계획수립	위험관리 기준과 활동 정의하고 계획	위험관리 계획서
	위험식별	프로젝트 영향주는 위험 식별과 특성 문서화	위험 관리대장, 이슈로그
	정성적 위험 분석	위험 발생확률과 영향 평가하여 우선순위 결정	PJT 문서 갱신
	정량적 위험 분석	위험이 프로젝트 목표에 미치는 영향 수치적분석	위험 보고서
	위험 대응 계획 수립	긍정적 위험 증대, 부정적 위험 최소화 위한 대응, 처리방안	변경 요청, PJT관리계획서 갱신, PJT 문서 갱신

실행	위험 대응 실행	합의된 위험 대응 계획 실행	변경 요청, PJT문서갱신
감시 및 통제	위험 감시 및 통제	위험 프로세스 효율성 평가	작업성과정보, PJT문서갱신

III. 위험과 기회에 대한 대응 전략

가. 위험에 대한 대응 전략

대응 수위	대응방안	설명
<div> <div>↑</div> <div>↓</div> </div> 적극적 대응	에스컬레이션	- 과업 밖에 있거나, PM 권한 밖에 있는 사항으로 서로 판단 사항
	회피 (Avoid)	- 프로젝트 목표를 위험의 영향권에서 고립 혹은 변경
	전가 (Transfer)	- 위험 영향력 및 대응 주체를 제3자에게 책임을 이동
	완화 (Mitigate)	- 수용 가능 한계선까지 위험 발생가능성, 영향 낮추는 방법
소극적 대응	수용 (Accept)	- 위험을 그대로 두거나 받아들이는 것

나. 기회에 대한 대응 전략

대응 수위	대응방안	설명
<div> <div>↑</div> <div>↓</div> </div> 적극적 대응	에스컬레이션	- 과업 밖에 있거나, PM 권한 밖에 있는 사항으로 서로 판단 사항
	활용 (Exploit)	- 기회 실현 위해 긍정적 영향 갖는 리스크 선택 전략
	공유 (Share)	- 제3자에게 유익한 기회를 공유(분담)하는 방법
	증대 (Enhance)	- 긍정적 영향 리스크 식별하여 극대화
소극적 대응	수용 (Accept)	- 기회 수용 수반되면 활용하지만 적극적 기회 추구 않는 방법

- 프로젝트 발생가능한 위험을 사전에 분석하고 대응하기 위한 프로세스와 도구 필요.

IV. 프로젝트 위험을 사전에 분석하고 대응하기 위한 프로세스와 도구

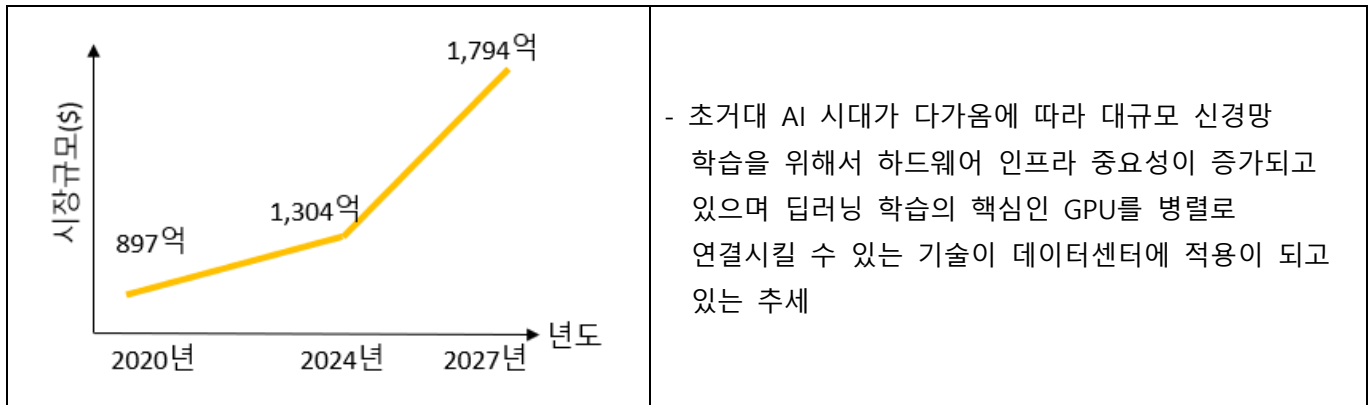
구분	항목	설명
프로세스	변경통제 절차	- 원가, 일정, 변경을 검토, 승인, 통합하는 절차
	베이스라인	- 변경기준 및 변경이력 기록 추적
도구/기법	통합관리도구	- EVM, 계획대비 작업의 성과 주기적 측정
	일정단축기법	- Crashing, Fast Tracking, 자원평준화
정책	범위 명확화	- 모호하거나 불합리한 WBS의 재분할
	예비비 확보	- Contingency Reserve, Management Reserve

- 위험은 프로젝트 초기에 식별, 분석하여 지속적으로 모니터링을 통하여 완화하여야 함.

“끝”

02	멀티 GPU 기술		
문제	딥러닝에서 대규모 신경망을 효율적으로 훈련하기 위한 멀티 GPU 기술에 대하여 설명하시오. 가. 멀티 GPU 기술의 개념과 장점 나. 멀티 GPU 환경 구축 시 고려사항		
도메인	CA/OS	난이도	중(상/중/하)
키워드	GPU, 딥러닝, 신경망, 병렬, 분산학습		
출제배경	초거대 AI 등장으로 대규모 신경망 학습이 중요하며 그 핵심 기술인 멀티 GPU 기술 이해 확인		
참고문헌	ITPE 서브노트		
해설자	NS반 멘토 백현 기술사(제 122회 정보관리기술사 / snuoo@naver.com)		

I. 딥러닝에서 대규모 신경망을 효율적으로 훈련하기 위한, 멀티 GPU 기술 개요



- 딥러닝에서 대규모 신경망을 효율적으로 학습하기 위해 병렬처리 연산이 가능한 GPU를 사용하게 되었고, 다수의 GPU를 연결하기 위한 인터커넥트 기술 발전으로 멀티 GPU 기술이 등장함

II. 멀티 GPU 기술의 개념과 장점

가. 멀티 GPU 기술의 개념

개념	- 여러 개의 그래픽 처리 장치(GPU)를 동시에 사용하여 대규모 신경망을 훈련하는 위한 다중 그래픽 장치를 병렬로 작동하기 위해 사용하는 기술	
개념도		
구성요소	GPU	- 여러 개의 GPU를 시스템에 설치하여 병렬 연산을 수행
	메인보드	- 여러 개의 GPU를 장착할 수 있는 PCIe 슬롯으로 구성된 보드
	PSU	- GPU가 소비할 전력을 안정적으로 공급하는 고출력 전원장치
	연결기술	- GPU와 GPU간, 서버와 서버간 병렬처리를 위한 인터커넥트 기술

	미들웨어	- 멀티 GPU 지원 드라이버 및 운영체제(OS)
	라이브러리	- 멀티 GPU로 딥러닝 학습을 지원할 지원 라이브러리

나. 멀티 GPU 기술의 장점

구분	장점	설명
성능	연산 속도 향상	- 여러 GPU가 동시에 병렬로 작업을 처리함으로써 학습 시간을 크게 단축
	대규모 데이터 처리	- 큰 데이터셋을 효율적으로 처리할 수 있어 더 복잡한 모델을 훈련
	확장성	- 단일 GPU로는 처리할 수 없는 큰 모델이나 데이터셋을 여러 GPU를 통해 처리할 수 있어, 모델과 데이터셋의 크기를 쉽게 확장가능
모델학습	훈련 시간 단축	- 여러 GPU가 동시에 작업을 수행함으로써, 모델 훈련에 소요되는 시간을 단축
	모델 병렬화	- 모델의 각 층을 서로 다른 GPU에 배분하여 병렬로 처리함으로써 연산 효율을 높임

III. 멀티 GPU 환경 구축 시 고려사항

가. 하드웨어 및 소프트웨어 고려사항

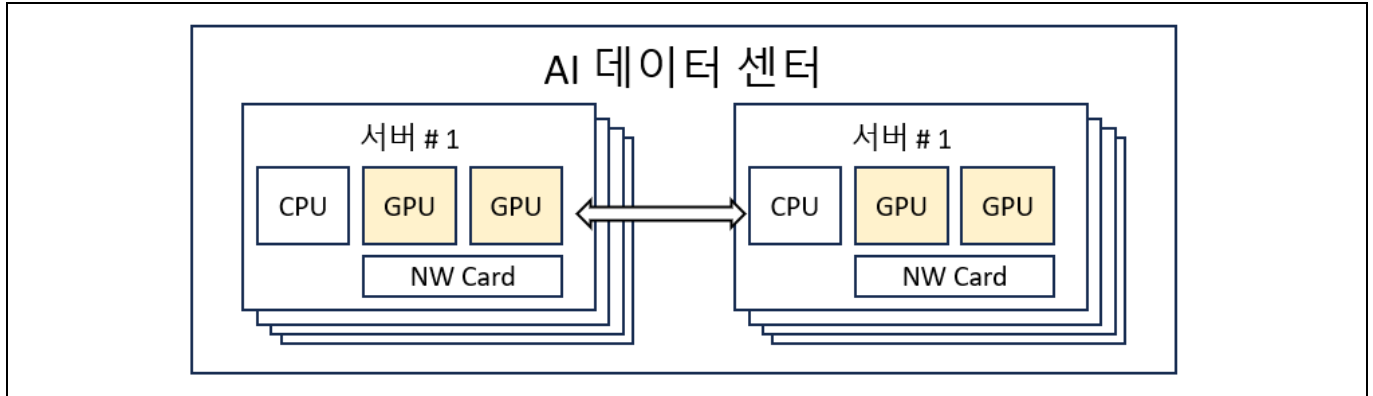
구분	고려사항	설명
하드웨어	GPU 수	- 필요한 연산 능력과 예산에 맞춰 GPU 수를 결정 고려
	GPU 메모리	- 데이터셋과 모델의 크기에 따라 충분한 메모리를 가진 GPU를 선택
	인터커넥트	- GPU 간의 데이터 전송 속도를 높이기 위해 고속 인터커넥트 기술사용
소프트웨어	프레임워크 지원	- TensorFlow, PyTorch와 같은 딥러닝 프레임워크가 멀티 GPU를 지원하는지 확인
	분산 학습	- 데이터 병렬화, 모델 병렬화와 같은 학습 전략 필요 1) 데이터 병렬화 : 데이터셋을 다수의 GPU에 나누어 병렬로 훈련고려 2) 모델 병렬화 : 모델을 여러 GPU에 나누어 각 GPU가 모델의 다른 부분을 훈련
	동기화 통신	- 멀티 GPU 간의 동기화와 통신을 효율적으로 처리하기 위한 기술고려

나. 프로그래밍 및 성능최적화 고려사항

구분	고려사항	설명
프로그래밍	메모리 관리	- 각 GPU의 메모리 사용을 효율적으로 관리
	데이터 로드 및 전처리	- 데이터 로드와 전처리 과정에서 병목 현상이 발생하지 않도록 최적화 고려
	모델 분할 및 병렬 처리	- 모델을 적절히 분할하고, 병렬 처리를 효과적으로 구현하기 위한 코드 작성 고려
성능최적화	프로파일링 도구	- 훈련 과정에서 성능 병목을 찾고 최적화하기 위해 프로파일링 도구 사용을 고려

	하이퍼파라미터 튜닝	- 멀티 GPU 환경에서 최적의 성능을 얻기 위해 하이퍼파라미터를 튜닝을 고려
--	------------	---

IV. 멀티 GPU 기술을 핵심기술로 사용, AI 데이터 센터

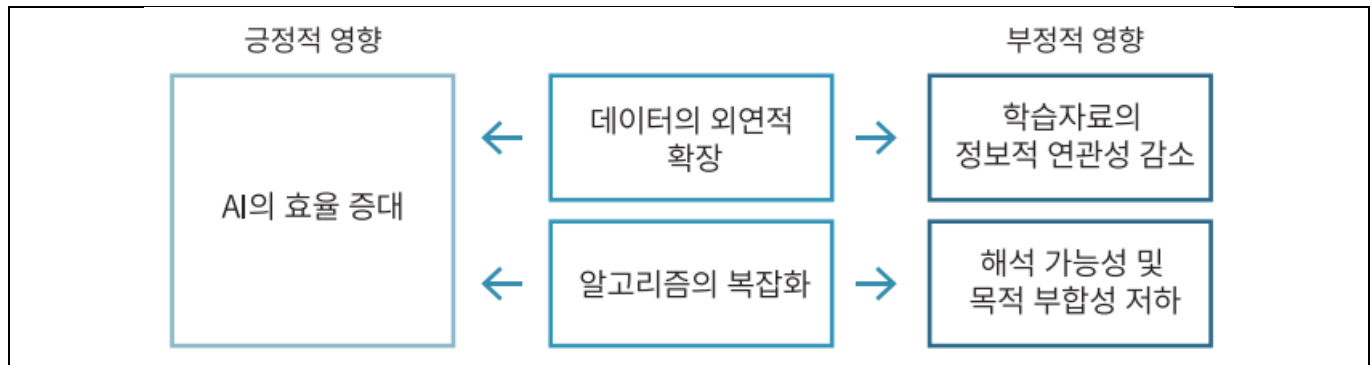


- 멀티 GPU 팜으로 구성된 AI 데이터 센터에서 멀티 GPU 기술로 대규모 AI 학습을 지원하는 플랫폼으로 활용됨.

“끝”

03	AI 시스템 윤리		
문제	AI 시스템에 대한 법적 이슈, 윤리적 문제, 기술적 문제에 대하여 설명하고, 해결방안을 제시하시오.		
도메인	인공지능	난이도	중(상/중/하)
키워드	데이터 편향, 저작권, 책임 소재, 적대적 공격, 정확성, 다양성		
출제배경	AI 생태계의 이해 확인 금융산업에서의 인공지능(AI) 활용 방안에 따른 리스크 요인 분석_자본시장연구원		
참고문헌	ITPE 기술사회 자료집		
해설자	정상반멘토 정상 기술사(제 124회 정보관리기술사 / itpe_peak@naver.com)		

I. AI 생태계 확장, AI 시스템에 대한 문제점 부각 배경



- AI 생태계가 확장 구성되고 사용이 증가하게 되면서 긍정적인 영향도 미치게 되지만 반대로 부정적인 영향까지 지속적으로 확대 발생하고 있는 상황

II. AI 시스템의 법적 이슈, 윤리적 문제, 기술적 문제

가. AI 시스템의 법적 이슈, 윤리적 문제

구분	이슈	설명
법적 이슈	데이터 프라이버시 및 보호	- 개인정보 보호법 및 유럽 기준인 GDPR 준수의 여부의 확인과 유출시의 문제 발생
	저작권 문제	- AI 생성 콘텐츠의 저작권 문제의 발생 - AI 소유자의 저작권 귀속 문제 발생
	책임 소재	- AI 시스템 사고 시 책임 주체 명확화 필요 - 자율주행차 사고 등 구체적 사례 필요
윤리적 문제	편향과 차별	- 데이터 편향에 따른 결과 편향 대응 필요 - 데이터 편향에 따라 특정 집단 차별 문제 발생 가능
	투명성과 설명 가능성	- AI 결정 과정의 데이터에 대한 투명성 부족 - 사용자의 이해 및 신뢰 확보 필요
	일자리 대체	- AI로 인한 기존의 일자리 감소 - 근로자 재교육 및 사회 안전망 구축 요구

	프라이버시 침해	<ul style="list-style-type: none"> - 개인 행동 감시 및 데이터화 과정에서의 개인 침해 발생 - 개인 프라이버시 보호관련 대응 문제 발생 가능
--	----------	---

- 법적이슈, 윤리적 문제 이외 기술적 문제도 존재

나. AI 시스템의 기술적 문제

구분	이슈	설명
데이터 관점	데이터 품질 정확성	<ul style="list-style-type: none"> - 학습 데이터의 정확성과 신뢰성에 대한 검증 이슈 발생 - 노이즈나 오류가 포함된 데이터로 인한 이상 결과 노출
	데이터 편향성 위협	<ul style="list-style-type: none"> - 특정 집단에 유리하거나 불리한 데이터의 학습 가능성 - 편향된 결과 생성 가능성
	데이터 다양성 부족	<ul style="list-style-type: none"> - 다양한 상황을 반영하지 못하는 데이터 학습 문제 - 지식의 일반화 능력 저하로 인한 이상 결과 발생 가능성
	실시간 데이터 처리 문제	<ul style="list-style-type: none"> - 실시간 데이터 수집 및 처리의 어려움 발생 가능 - 지연 시간으로 시스템 전반적인 성능 문제 발생
보안 관점	모델의 해킹 문제	<ul style="list-style-type: none"> - 모델에 대한 직접적 공격 발생 가능 - 적대적 공격(Adversarial Attacks)의 위협 발생 가능
	데이터 유출	<ul style="list-style-type: none"> - 학습 데이터의 외부 유출 위험으로 인한 피해 가능성 - 민감한 정보 보호 필요성
	악의적인 데이터 삽입	<ul style="list-style-type: none"> - 악성 데이터로 인한 모델 성능 저하 가능성 - 데이터 무결성 손상으로 인한 가짜 결과물 생성 가능성
	시스템 취약점 발생	<ul style="list-style-type: none"> - AI 시스템의 보안 취약점 노출 가능성 발생 - 네트워크 공격 및 시스템 붕괴 위험 가능성 발생

- 다양한 관점에서의 기술적 문제에 대한 해결방안 필요

II. AI 시스템에 대한 문제점과 해결방안

가. AI 시스템의 법적, 윤리적 측면 해결방안

구분	해결방안	설명
법적 측면	데이터 보호법 준수	<ul style="list-style-type: none"> - 개인정보 보호법 및 유럽의 GDPR 가이드 준수 - 데이터 최소화 원칙 적용 및 데이터 익명화 및 가명화 기술 도입
	법률, 제도의 명확 규정	<ul style="list-style-type: none"> - AI 생성물에 대한 저작권 법률 명확화 - AI 창작물에 대한 새로운 저작권 규정 도입 - AI와 인간의 협업 결과물에 대한 권리 분배
	책임 주체의 명확화	<ul style="list-style-type: none"> - AI 시스템의 책임 주체 명확화 - 자율주행차 등 AI 시스템의 법적 프레임워크 구축 - 사고 시 책임 보험 제도 도입
윤리적 측면	데이터 검증 체계 강화	<ul style="list-style-type: none"> - 데이터 수집 및 처리 과정에서의 편향 제거 - 다양한 집단을 대표하는 데이터 사용 - 정기적인 모델 검토 및 업데이트

	설명가능한 AI 도입으로 투명성 확보	<ul style="list-style-type: none"> - AI 결정 과정의 투명성 강화 - 설명 가능한 AI(Explainable AI) 개발 - 사용자에게 AI 작동 원리 설명
	새로운 교육 프로그램 기획	<ul style="list-style-type: none"> - AI로 인해 대체된 근로자 재교육 프로그램 도입 - 사회적 안전망 강화 및 새로운 일자리 창출 - AI와 인간의 협력 모델 개발
	개인정보 절차 강화	<ul style="list-style-type: none"> - 데이터 수집 시 사용자의 동의 강화 - 프라이버시 보호 기술(예: 암호화, 익명화) 적용

- 법적이슈, 윤리적 문제 해결방안 이외 기술적 해결방안도 존재

나. AI 시스템 문제 기술적 측면 해결방안

구분	해결방안	설명
데이터 측면	데이터 품질 모니터링	<ul style="list-style-type: none"> - 데이터 정제 및 전처리 과정 개선 프로세스 도입 - 신뢰성 높은 데이터 소스 사용 필요
	편향 탐지 및 수정 도구 사용	<ul style="list-style-type: none"> - 편향 제거를 위한 알고리즘 개발 - 다양한 집단을 대표하는 균형 잡힌 데이터 수집
	도메인 전문가 협력	<ul style="list-style-type: none"> - 다양한 상황을 반영한 데이터 수집 - 다중 도메인 데이터 통합 - 도메인 전문가와 협력하여 데이터 다양성 확보
보안 측면	적대적 공격 방어기법	<ul style="list-style-type: none"> - 모델 취약성 점검 수시 점검 및 대응 방안 마련 - 안전한 모델 배포 및 업데이트
	데이터 암호화 기법 적용	<ul style="list-style-type: none"> - 접근 제어 및 데이터 보호 정책 강화 - 정기적인 데이터 보안 감사 실시
	지적 재산권 보호기술 적용	<ul style="list-style-type: none"> - 모델 워터마킹 및 지적 재산권 보호 기술 적용 - 모델 접근 권한 관리 강화 - 불법 복제 감지 시스템 도입
	시스템 취약점 개선 체계 마련	<ul style="list-style-type: none"> - AI 시스템 보안 취약점 분석 및 패치 - 네트워크 보안 강화 및 침입 탐지 시스템 적용 - 정기적인 보안 업데이트 및 모니터링

- 다양한 관점에서 기술적 해결방안으로 AI 시스템 문제점 방어 필요

“끝”

04	개방형 API(Open Application Programming Interface)		
문제	3. 개방형 API(Open Application Programming Interface)와 관련하여 다음을 설명하시오. 가. 정의 및 특징 나. SOAP 및 REST 구성요소 다. 취약점 및 대응방안		
도메인	디지털서비스, 보안	난이도	중(상/중/하)
키워드	API Gateway, API Portal, UDDI, WSDL, XML, HTTP, SOAP Envelope/Header/Body/Fault		
출제배경	Open API의 활용 증가로 기본 이해 확인		
참고문헌	ITPE 기술사회 자료집		
해설자	정상반멘토 정상 기술사(제 124회 정보관리기술사 / itpe_peak@naver.com)		

I. 누구나 이용할 수 있도록 공개 인터페이스, 개방형 API의 정의 및 필요성

가. 개방형 API(Open Application Programming Interface)의 정의

- 누구나 사용할 수 있도록 공개된 API를 말하며, 개발자에게 사유 응용 소프트웨어나 웹 서비스에 프로그래밍적인 권한을 제공

나. 개방형 API의 특징

특징	설명
공개성	- 누구나 접근할 수 있도록 공개되어 있으며, 개발자들이 자유롭게 사용
표준화	- 일반적으로 REST, SOAP, GraphQL 등의 표준 프로토콜을 사용
호환성	- 다양한 플랫폼과 호환되도록 설계되어 있어, 여러 시스템 간의 통합을 용이
확장성	- 새로운 기능을 추가하거나 기존 기능을 확장할 수 있도록 유연하게 설계
버전 관리	- 버전 관리가 이루어지며, 이를 통해 기존 사용자가 영향을 받지 않고 새로운 기능을 도입

II. SOAP 및 REST 구성요소

가. SOAP(Simple Object Access Protocol) 아키텍처 설명

구분	설명	
정의	- Open API 환경에서 사용되는 확장성 있는 메시지 프로토콜로, HTTP 통신 프로토콜을 기반으로 XML을 사용하여 데이터를 교환한다	
특징	- HTTP 프로토콜 사용으로 API 사용자의 접근성이 뛰어남 - XML 형식으로 데이터를 표현하기 때문에 태그, 속성 등을 통해 부가정보의 표현이 자유로움	
개념도	<pre> graph LR Provider([Open API 제공자]) -- "API 등록 (WSDL)" --> UDDI[UDDI] UDDI -- "API 검색 (WSDL)" --> User([Open API 사용자]) User -- "API 사용 (SOAP MESSAGE)" --> Provider </pre>	
구성요소	UDDI	Universal Description, Discovery and Integration Open API 공개 및 검색을 위한 표준으로 제공자가 저장한 API를 사용자가 이용
	WSDL	Web Services Description Language Open API를 설명하기 위한 XML 표준 언어로 사용절차 등을 문서 지향 형태로 제공
	SOAP Message	Open API 요청 및 응답 메시지 규격으로 사용자는 규격에 맞게 데이터를 구성하여 제공자에게 요청, 제공자는 동 규격으로 처리결과 반환

나. REST (Simple Object Access Protocol) 아키텍처 설명

구분	설명	
정의	인터넷 상에 존재하는 API 대상 자원(데이터 또는 서비스)에 주소를 지정하고, HTTP 통신 프로토콜을 사용하여 해당 자원을 관리하는 아키텍처	
특징	<ul style="list-style-type: none"> - 가장 일반적으로 사용되는 Open API 시스템의 아키텍처 - 사용자 환경에 대해 제약이 적어 범용성, 편의성, 접근성 등이 뛰어남 	
개념도	<pre> API 사용 (HTTP or HTTPS) Open API 제공자 <--> Open API 사용자 HTTP POST https://www.apisample.com/members { "id" : "itpe", "name" : "Bob" } </pre>	
구성요소	자원 (Resource)	URI를 통해 식별 가능한 데이터 또는 서비스, Open API로 제공하는 대상을 의미
	행위 (Behavior)	자원에 대한 생성·조회·수정·삭제 행위를 HTTP Method(POST, GET, PUT, DELETE)로 표현
	메시지 (Message)	자원에 대한 행위를 위한 요청정보와 행위에 대한 결과인 응답정보를 출력하는 방식
Method	GET(조회)	기존 자원에 대한 정보 조회 (예시) 사용자 정보 조회: GET https://www.apisample.com/users
	POST (생성)	신규 자원의 생성 (예시) 'itpe'의 사용자 정보 생성: POST https://www.apisample.com/users/itpe
	PUT (수정)	기존 자원에 대한 정보 수정 (예시) 'itpe'의 사용자 정보 변경: PUT https://www.apisample.com/users/itpe
	DELETE (삭제)	기존 자원의 삭제 (예시) 'itpe'의 사용자 정보 삭제: DELETE https://www.apisample.com/users/itpe

III. 개방형 API의 취약점

가 종류 측면의 취약점

구분	취약점	설명
기술	- 객체 식별자 제어값 API 중단 노출	- API 특정 값 변조 통해 접근 권한 없는 데이터 조회 - 자동차 원격 제어값 노출, 온라인 문서ID 노출 사례
	- 인증 토큰 조작	- 잘못 구현된 인증 메커니즘 취약점 이용 인증 토큰 조작 - API의 쿼리 일괄 처리 기능 악용 요청 속도 우회 공격 사례
	- 접근 권한 검증의 부적절한 구현	- 자산의 모든 속성을 외부에 노출하거나 변조 가능 - 다른 사용자의 민감 정보 노출, 청구 금액 변조 사례
	- 과도한 자동화 접근	- 민감한 비즈니스 흐름 찾아 과도한 접근하여 손상 발생 - 구매 코드 자동화하여 대량 매수 후 재판매 악용 사례
	- 보안 구성 오류	- 패치되지 않은 결함, 디폴트 구성, 잘못된 디렉토리 구성 - 네트워크 아웃바운드 구성 오류로 인한 악성코드 실행 사례
관리	- 과도한 API 요청 허용	- 자원 사용량 과다 발생 및 비용 증가 유발 - 비밀번호 찾기 SMS 과다 발생 사례
	- API 이용 수준 검사 미흡	- 불분명한 관리자 및 사용자 기능 구분 취약점 - 관리자 및 사용자 리소스에 대한 복잡한 접근 제어 정책
	- URI 유효성 미검증	- Server Side Request Forgery 공격 취약점 - 원격 리소스 사용자의 URI 유효성 미검증 취약
	- 가시성 확보 및 유지보수 미흡	- API 이용 환경 및 권한이 구분 불가한 취약점 - API 버전 관리 및 문서화 부재 취약점
	- 맹목적인 API 신뢰	- 암호화되지 않은 채널의 타사 API 이용 취약점 - 타사 API로 수집된 정보에 대한 검증 없이 처리

나. 경로 측면의 취약점

구분	취약점	설명
API 외부	- 클라우드 서비스 API Key 저장 위협	- 클라우드 기반 Storage에 보안 미적용 API Key 저장 취약점 - Amazon S3와 같은 클라우드 및 Github와 같은 Repository
	- API 호출 스니핑	- MITM(Man-In-The-Middle) 공격 통해 API 트래픽 스니핑 - MITM 프록시 도구 사용하여 비공개 API 역공학 및 해킹
API 내부	- API 로직 결함	- API에 악용 가능한 버그 또는 논리적인 결함 - Steam 버그 이용 모든 게임의 CD Key 접근 사례
	- 하드코딩된 API Key	- 디컴파일을 통한 API 키 또는 자격증명 노출 - 스마트 잠금/허브에서 발견된 임베디드 API Key 취약 사례

- API의 보안 위협 대응을 위해 관리적, 기술적 측면의 대응 필요

IV. API의 보안 대응 방안

가. 관리적 측면의 보안 대응 방안

구분	보안 대응 방안	설명
API 설계	- API 표준화 수립	- 조직의 API 표준 및 스타일 지침 관리
	- API 프로세스 공동 참여	- 개발자 및 비개발자 모두가 접근하여 문서화/프로세스 기여
API 구현	- API 안정성 강화	- 의미 있는 API 테스트 사례 생성하여 테스트 범위 확장
	- API 보안성 강화	- API 취약점 식별 및 개발 프로세스 적용(DevSecOps)
API 배포	- API 카탈로그 제공	- API 명세 조회, 게이트웨이 사용량 분석 표시
	- API 제어 중앙 집중화	- 확장성과 일관성 위해 API 구성 관리 중앙 집중화
API 관리	- API 버전 관리	- 동일한 API의 여러 버전을 쉽게 제공 및 동시 작동 필요
	- 미사용 API 비활성화	- 사용이 종료된 API는 서비스 비활성화 및 폐기 처리

나. 기술적 측면의 보안 대응 방안

구분	보안 대응 방안	설명
기존 보안 솔루션과 종합적 운영	- 기존 웹 어플리케이션 방화벽 활용 API 보호	- API 관리 및 ID 접근관리와 함께 API 보호 수행
	- API 관리 수행	- API 게이트웨이 및 사용자 지정 가능한 API 개발자 포털 구성
	- 콘텐츠 위협탐지 및 OAS 메시지 유효성 검사 수행	- OAS(OpenAPI Specification)에 대한 검사 강화 기능 수행
	- IDaaS 활용	- ID 접근 관리는 온프레미스와 클라우드 기반 관리 병행
	- 인앱 보호 활용	- API 스크래핑 및 악의적 API 사용에 대한 보호
	- 전문 API 보안 도구 활용	- API 게이트웨이 및 프록시 역할 포함하여 API 검색 및 보안 테스트 종합적 수행 가능한 전문 도구 활용
서비스 메시 환경의 운영	- 웹방화벽 API 공격 탐지 및 차단	- 웹 어플리케이션 방화벽은 API 통해 들어오는 공격 탐지/차단
	- API 클라이언트 인증	- API 게이트웨이는 클라이언트 인증 및 권한 할당 토큰 주입
	- Micro Service 트래픽 라우팅 및 로드밸런싱	- Ingress Controller 트래픽 분산 처리에 따른 부하 감소
	- Side-Car TLS 적용	- 세분화된 권한 부여 및 암호화로 API 트래픽 보호
로깅 보안 제어 활용	- API 보안 가시성 확보	- API 사용 비정상 패턴 탐지 및 잠재적 보안 위협 대응 - 집계 스캔, 모니터링, 경고, 로그 분석 등 내결함성 정책 설정
	- SIEM, SOAR 활용	- 기존 보안 로깅 기술과 연계하여 종합적인 이벤트 상관 분석 - API 보안 업무 자동화 및 유지관리 효율성 증대 효과

- 클라우드 서비스의 경우 AWS CloudWatch/CloudTrail, GCP Apigee, Azure API Management 등 솔루션 제공

“끝”

05	클라우드 전환		
문제	클라우드 전환 사업의 단계별 감리 방법과 검토 항목에 대하여 설명하시오.		
도메인	디지털서비스	난이도	중(상/중/하)
키워드	클라우드 서비스 기회식별, 클라우드 서비스 요건정의, 클라우드서비스 전환		
출제배경	클라우드, 인공지능 등 지능정보 기술의 감리를 위해 한국지능정보사회진흥원(NIA)에서 「지능정보기술 감리 실무 가이드(2023.2)」 발간		
참고문헌	ITPE 기술사회 자료		
해설자	BP반 김찬일 기술사(제 130회 정보관리기술사 / s2carey@naver.com)		

I. 지능정보 기술의 감리, 클라우드 서비스 활용 사업 감리의 개요

가. 클라우드 서비스 활용 사업의 개념

- 감정보자원의 구축, 고도화 등을 목적으로 하는 공공정보화사업 중 정보자원의 일부 또는 전부를 직접 구축하는 대신 기 구축된 G-Cloud 또는 Private Cloud, 민간 클라우드를 활용하는 사업

나. 클라우드 서비스 활용 사업 감리의 필요성

감리 가이드 부재	- 기존에 클라우드 활용 사업에 대한 감리 가이드가 없어 사업적 특성 반영한 감리 가이드 필요
클라우드 전환	- 공공,행정분야 클라우드 전환 사업을 통해 공공정보화사업에 클라우드 서비스 활용 높아짐

- 공공분야에 클라우드 도입이 활발해지면서 이에 대한 감리가이드 제정 필요

II. 클라우드 전환 사업의 단계별 감리 방법

[표 3-16] 클라우드 전환 감리 점검가이드(안)

단계	활동	검토항목	적용모델	
			정부	민간
클라우드전환 계획수립 및 준비	전환 타당성 검토	01. 클라우드 전환 가능성에 대한 검토가 적절하게 이루어졌는가?	●	●
	요구사항 및 영향도 분석	02. 클라우드 전환에 따른 이해관계자(발주자, 정부 등)의 요구사항 및 운영체제 및 자원변경 등의 영향도가 체계적으로 파악되었는가?	●	●

단계	활동	검토항목	적용모델	
			정부	민간
클라우드 전환 계획수립 및 준비	시스템 구성 및 사용량 확정	03. 최적의 성능 및 안정성을 확보를 위한 시스템 구성 및 사용조건을 확정하였는가?	●	●
	전환방식결정 및 비용산정	04. 클라우드 서비스 대상(IaaS, PaaS, SaaS, IaaS/PaaS/ SaaS, IaaS/PaaS), 민간클라우드 서비스 사업자 유형 (CSP, MSP) 등 전반적인 사업방식을 결정이 적절하게 이루어졌는가?		●
		05. 클라우드 전환비용과 클라우드서비스 이용료 산정이 적절하게 이루어졌는가?		●
	보안성 검토 및 서비스수준 정의	06. 국가 정보보안 기본지침에 의거하여, 도입하는 정보자원에 대한 보안성 검토를 수행하였는가?		●
		07. 요구되는 클라우드 서비스에 대한 서비스수준(SLA)을 적절하게 작성하였는가?		●
	구축 및 테스트 계획의 수립	08. 시스템 전환에 따른 업무 중단을 최소화하고 안정성을 확보할 수 있는 방안을 고려한 체계적인 구축 및 테스트 계획인 마련되었는가?	●	●
		09. 시스템 이전/전환에 따른 상용S/W의 업그레이드, 마이그레이션, 환경설정 등 시스템 SW 영향도 분석 및 IP 변경에 따른 기 운영 프로그램에 대한 영향도를 분석하여 최적의 구축방안이 제시되었는가?	●	●
	nTOPS계정 신청	10. nTOPS를 통한 계정 발급 신청, 권한 신청은 적정하게 이루어졌는가?	●	
	원격접근 및 방화벽 포트 허용신청	11. 시스템의 환경구성에 따른 검증작업과 방화벽 포트 오픈 및 시스템접근 작업에 대한 허용신청은 적정하게 이루어졌는가?	●	
	발주방식결정 및 서비스제공자 선정	12. 클라우드 서비스에 대한 발주방식(일반 조달계약, 디지털서비스 전문계약제도 등)을 결정하고 결정된 방식에 따라 클라우드 서비스 제공자를 적절하게 선정하였는가?		●
클라우드	계약체결 및 SLA	13. 서비스제공자(MSP, CSP)와의 계약체결과 서비스수준합의(SLA)는 적정하게 이루어졌는가?		●
	어플리케이션 수정/개발,	14. IP변경, 플랫폼 변경/개선 등으로 인한 어플리케이션 수정 요구에 대해서 적정하게 어플리케이션 수정과 테스트가 수행되었는가?	●	●

단계	활동	검토항목	적용모델	
			정부	민간
클라우드 전환	클라우드 서비스 이용환경 구축	17. 클라우드 정보자원을 효율적으로 이용할 수 있도록 환경을 설정하였는가?	●	●
	보안 취약점 점검	18. HW, 어플리케이션 등 자원에 대해 영향을 미칠 수 있는 다양한 위협요인을 파악하고, 이들 위협요인에 대한 취약성 분석과 대응방안이 마련되었는가?	●	●
	데이터 이관	19. 데이터 이관계획에 따라 데이터를 이관하고 검증이 이루어졌는가?	●	●
	통합테스트 수행	20. 사용자 환경에서 이용시나리오를 기반으로 수립된 통합시험계획에 따라 시험이 적정하게 수행되었는가?	●	●
		21. 시스템 성능 및 연계시스템과의 호환성이 확보되었는가?	●	●
서비스 안정화	서비스 전환 및 안정화	22. 안정화를 위한 어플리케이션 및 상용SW에 대한 기술지원은 적정하게 이루어지고 있는가?	●	●

클라우드 전환사업의 감리 점검가이드는 전환계획 수립 및 준비, 전환, 서비스 안정화 등 3단계의 16개 활동으로 구분되며, 16개 활동은 다시 22개 검토항목으로 구성되었고, 각 검토항목은 '2. 검토항목'에 별도로 기술하고 있다.

III. 클라우드 전환 사업의 검토 항목

번호	검토 항목	설명
1	클라우드 전환 가능성	정보시스템 특성과 클라우드 환경 간의 적합성 검토
2	장비 사용연한 만료 여부	장비 교체 주기를 고려하여 전환 필요성 검토
3	SW의 기술지원 만료 여부	소프트웨어의 기술지원 만료 상태 확인
4	상용 SW를 공개 SW로 전환 가능 여부	현재 상용 소프트웨어를 오픈 소스로 전환 가능성 평가
5	고립망 또는 DR 필요 여부	재해복구(Disaster Recovery)나 고립망 구성 필요성 검토
6	업무 단위 이관 가능 여부	업무 시스템을 클라우드로 전환 가능한지 여부 검토
7	시스템 용량 산정	CPU, 메모리, 디스크 용량 등 시스템 자원 산정
8	시스템 구성	클라우드 전환을 위한 시스템 구성 요소 결정
9	인터넷 서비스와 행정망 서비스 간 데이터 연동 필요 여부	데이터 연동 필요성 검토
10	보안 취약점 점검	보안 취약점 분석 및 대응 방안 마련
11	데이터 이관 계획 및 검증	데이터 이관 계획 수립 및 검증 절차 수행
12	통합 테스트 수행	사용자 환경에서의 통합 테스트 수행
13	시스템 성능 및 연계 시스템과의 호환성	성능 평가 및 연계 시스템과의 호환성 확인
14	서비스 전환 및 안정화	서비스 전환 과정 및 안정화 작업 수행
15	클라우드 서비스 이용 환경 구축	클라우드 자원 효율적 이용을 위한 환경 설정
16	보안 취약점 분석 및 대응 방안 마련	클라우드 서비스 보안 취약점 및 대응 방안 수립
17	클라우드 전환 비용 및 이용료 산정	전환 비용 및 클라우드 서비스 이용료 산정
18	서비스 수준(SLA) 정의 및 계약 조건 결정	SLA 정의 및 계약 조건 결정
19	클라우드 모델 결정 및 최적화	클라우드 모델 선정 및 최적화
20	클라우드 서비스 제공자 선정 및 검토	CSP/MSP 선정 및 제안 내용 검토
21	클라우드 서비스 운영 관리	운영 관리 체계 수립 및 운영 절차 확인
22	서비스 수준 및 사용량 측정	서비스 수준 유지 및 사용량 모니터링

“끝”

06	SOM(Self Organizing Map)		
문제	6. 군집분석 기법인 SOM(Self Organizing Map)에 대하여 설명하시오. 가. SOM 정의 및 특징 나. SOM 구성요소 다. SOM과 신경망 분석기법의 차이점		
도메인	인공지능	난이도	중(상/중/하)
키워드	클러스터링, 비지도학습, 입력층, 경쟁층, 가중치, 노드		
출제배경	k-means 알고리즘의 단점을 보완한 군집화 알고리즘 이해		
참고문헌	ITPE 기술사회 자료, 모의고사 자료		
해설자	정주행 조종흥 기술사(제127회 정보관리기술사 / choheung@naver.com)		

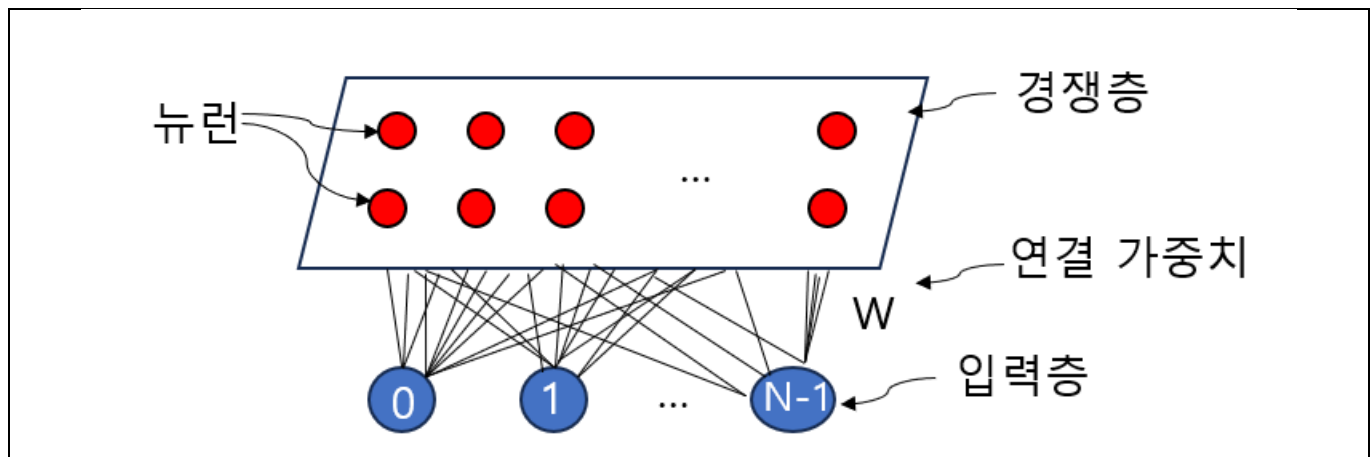
I. K-Means 알고리즘 단점 보완, SOM(Self Organizing Map)의 개요

정의	비지도 학습으로, 고차원 데이터의 토폴로지를 보존하여 저 차원 그리드에 매핑하는 데 사용되며, 주로 데이터 시각화와 군집화에 활용하는 알고리즘	
특징	군집화	- 유사한 데이터를 그룹화하여 군집화를 수행하는 알고리즘
	차원축소	- 고차원 데이터를 저차원 그리드에 매핑하여 효과적으로 표현
	시각화	- SOM은 데이터의 시각적인 표현을 제공하는 알고리즘
	비지도 학습	- SOM은 레이블이 없는 데이터를 학습하며, 데이터의 내재된 구조를 발견하는 데 사용
	경쟁학습	- 뉴런들이 서로 경쟁하여 활성화되며, 가장 가까운 뉴런(우승 뉴런)이 데이터 포인트에 적응
	토폴로지 보존	- 고차원 데이터의 거리 관계를 저차원 맵에서 보존하여 시각적으로 유사한 데이터 포인트가 인접하게 배치

- 다양한 유형의 데이터에 적용할 수 있는 유연한 구조와 데이터의 형태나 분포에 대한 가정이 필요하지 않음

II. SOM의 개념도 및 구성요소

가. SOM의 개념도



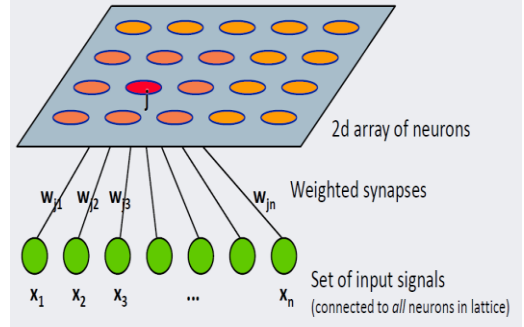
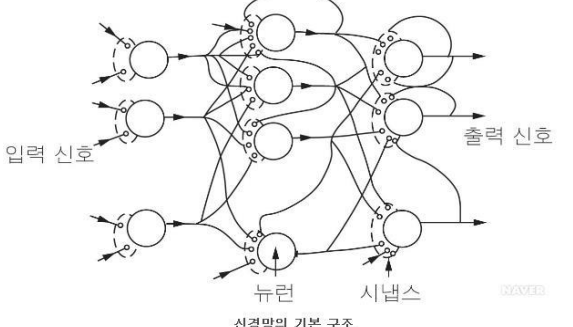
- 입력층과 경쟁층 등 2개 층 구조이며 각 층간은 가중치(W)로 Fully Connected로 연결되어 있음

나. SOM의 구성과 절차

구분	구성 요소	설명
구성	입력층 (Input Layer)	- 입력층은 SOM에 제공되는 고차원의 입력 데이터를 받는 층
	경쟁층 (Competitive Layer)	- 경쟁층은 SOM의 핵심이며, 그리드 형태로 구성되어 있습니다. 각 유닛은 입력 데이터와의 거리를 측정하여 입력 데이터와 가장 가까운 유닛을 선정하는 경쟁을 진행
	뉴런(Neurons)	- 경쟁층 내의 각 유닛을 의미하며, 경쟁 통해 유사 뉴런 선택됨.
	가중치(Weights) 벡터	- 초기에는 무작위로 설정되며, 학습 과정에서 조정되는 가중치
	BMU (Best Matching Unit)	- 주어진 입력 벡터와 가장 유사한 가중치를 가지는 노드
	이웃 함수 (Neighborhood Function)	- BMU 주변 노드들이 얼마나 영향을 받는지 결정하는 함수
	학습률 (Learning Rate)	- 학습 과정에서 가중치 업데이트의 크기를 결정하는 파라미터
절차	1. 초기화 (Initialization)	- 가중치는 일반적으로 입력 데이터의 범위에 맞추어 무작위로 초기화
	2. 경쟁(Competition)	- 입력 데이터와 각 뉴런의 가중치 벡터 사이의 거리를 계산하여 입력 데이터와 가장 가까운 뉴런을 선택
	3. 이웃성 함수 적용 (Neighborhood Function)	- 선택된 뉴런을 중심으로 주변의 뉴런들에게 가중치 조정을 위한 영향을 주는 이웃성 함수를 적용
	4. 가중치 업데이트 (Weight Update)	- 선택된 뉴런과 그 주변의 뉴런들의 가중치를 조정 이웃성 함수를 사용하여 뉴런 간의 거리에 따라 가중치를 업데이트하면서 입력 데이터와의 거리를 줄이는 방향으로 이동
	5. 반복(Iteration)	- 학습이 진행될수록 유사한 입력 데이터는 서로 가까이 매핑

- 군집이 잘 수행되었는지 주요 평가 방법인 실루엣 계수를 통해 검증이 필요함.

III. SOM과 신경망 분석기법의 차이점

구분	SOM	신경망 분석
개념		

목적	데이터 시각화 및 군집화	분류, 회귀, 예측
분석요소	입력데이터의 차원, 크기, 노드의 격자 형태, 노드 수, 초기화 방법, BMU(Best Matching Unit)	뉴런, 레이어, 가중치, 손실함수, 활성화 함수, 편향, 순전파, 역전파
학습 방식	비지도 학습	지도 학습
학습 방법	경쟁 학습	역전파 학습
알고리즘		
구조	2차원 격자형태의 맵	다층 구조(입력층, 은닉층, 출력층)
입력 데이터	고차원 벡터	고차원 벡터
출력 형식	2차원 시각화 맵	출력 벡터, 예측 값 또는 분류 결과
학습률 조정	점진적 감소	고정 또는 적응형
데이터 표현	유사성 기반	입출력 관계
주요 응용 분야	데이터 시각화, 패턴 인식, 클러스터링, 특징 추출, 이상탐지 등	이미지 및 영상 처리, 음성 및 오디오 처리, 자연어 처리, 시계열 분석, 추천시스템, 의료 및 생명과학, 게임 인공지능, 로봇틱스 등

- SOM은 경쟁 학습 방식을 사용하여 우승 뉴런과 이웃 뉴런의 가중치를 조정하는 반면, 신경망은 역전파(backpropagation)를 통해 오류를 최소화하는 방향으로 가중치를 업데이트를 함.

“끝”



ITPE 기술사회

제134회 정보처리기술사 기출문제 해설집

대 상	정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험
발행일	2024년 07월 27일
집 필	강정배PE, 전일PE, 백현PE, 조종흥PE, 정상PE, 김찬일PE
출 판	ITPE(Information Technology Professional Engineer)
주 소	ITPE 대치점 서울시 강남구 선릉로 86길 17 선릉엠티빌딩 7층 ITPE 선릉점 서울시 강남구 선릉로 86길 15, 3층 IT교육센터 아이티피이 ITPE 강남점 서울시 강남구 테헤란로 52길 21 파라다이스벤처타워 3층 303호 ITPE 영등포점 서울시 영등포구 당산동2가 하나비즈타워 7층 ITPE
연락처	070-4077-1267 / itpe@itpe.co.kr

본 저작물은 [ITPE\(아이티피이\)](#)에 저작권이 있습니다.

저작권자의 허락없이 **본 저작물을 불법적인 복제 및 유통, 배포**하는 경우
법적인 처벌을 받을 수 있습니다.