ai-mvp 핵심 이론

AI 역사

기계학습과 딥러닝의 역사를 획기적 전환기를 중심으로 설명해보겠습니다.
- 1950년대 - 기계학습의 기초가 마련됨
  앨런 튜링의 기계 지능 개념 제시, 퍼셉트론 등장

- 1980년대 - 신경망의 부흥
  역전파 알고리즘 등장, 심층 신경망 연구 활발

- 1990년대 - 겨울의 시대
  신경망의 한계 드러남, SVM, 결정 트리 등 대안 기법 등장

- 2000년대 초 - 딥러닝의 등장
  빅데이터와 GPU 보급으로 심층 신경망 성능 향상
- 2012년 - 알렉스넷의 이미지 분류 성과
  딥러닝이 이미지 인식에서 사람을 뛰어넘음

- 2015년 - 알파고의 바둑 승리
  딥러닝이 인간 수준의 지능 확보

- 현재 - 딥러닝 확산
  자율주행, 의료, 금융 등 다양한 분야로 확대 적용

주요 전환점은 신경망의 부흥, 딥러닝 등장, 알렉스넷과 알파고의 성과 등이 있습니다.

이를 기점으로 기계학습과 딥러닝이 비약적으로 발전했습니다.

앨런 튜링의 기계 지능

앨런 튜링은 1950년대에 기계가 지능을 갖출 수 있을 것이라는 개념을 제시했습니다.
튜링은 "기계는 어떤 문제에 대해 사람과 같은 방식으로 해결할 수 있다"고 주장했습니다. 이를 '튜링 테스트'라는 방식으로 설명했는데, 요약하면 다음과 같습니다.
- 사람과 기계가 서로 보이지 않는 방에서 대화를 한다.
- 대화의 내용만 보고 사람인지 기계인지 구분이 어려우면 기계는 지능이 있다고 볼 수 있다.
예를 들어, 체스 게임에서 컴퓨터 프로그램이 사람을 대항하여 동등하게 게임이 가능하다면 이 프로그램은 체스에서 사람 수준의 지능을 갖췄다고 볼 수 있습니다.
1997년 체스 프로그램 딥블루가 체스 그랜드마스터 카스파로프를 이긴 것이 튜링의 이 개념을 증명하는 사례입니다.
이렇듯 튜링은 기계도 사람과 유사한 지능을 내재할 수 있다는 획기적인 견해를 제시했습니다. 이후 AI 연구에 큰 영향을 미쳤습니다.

앨런 튜링의 기계 지능 개념을 증명하는 다른 대표적인 사례로는 다음과 같은 것들이 있습니다.
- 알파고: 2016년 바둑 프로그램 알파고가 세계 챔피언 이세돌을 물리친 사건은 기계의 바둑 능력이 사람을 뛰어넘음을 보여줬습니다.
- 자연어 처리: GPT, 번역기 등 자연어 처리 시스템의 발전은 기계와 자연스러운 대화 능력을 시사합니다.
- 이미지 인식: 이미지넷 도전에서 알렉스넷, VGGNet 등이 사람을 능가하는 정확도를 보여줬습니다.
- 음성 인식: Siri, 빅스비 등 음성 인식 시스템은 사람 수준의 음성 이해 능력을 갖췄다고 볼 수 있습니다.
- 자율주행: 테슬라 오토파일롯 등 자율주행 시스템은 주행 환경 판단 능력이 사람과 유사합니다.
이외에도 로보틱스, 의료 진단 등 다양한 분야에서 기계 지능의 발전 사례를 찾을 수 있습니다.

순전파와 역전파 알고리즘

아래는 순전파 알고리즘과 역전파 알고리즘의 주요 차이점과 장단점을 비교한 표입니다:

	순전파 알고리즘	**역전파 알고리즘**
설명	입력 데이터를 순서대로 각 층을 지나면서 전파하여 출력값 계산	순전파 과정을 거친 후 출력 오류를 역순으로 각 층으로 전파하며 가중치 업데이트
주요 단계	입력 데이터 전달 -> 각 층에서 활성화 함수 계산 -> 출력값 계산	순전파 과정 후 출력 오류 계산 -> 오류를 역순으로 각 층으로 전파하며 가중치 업데이트
목적	출력값을 계산하고 실제 값과 비교하여 오류를 확인	오류를 줄이는 방향으로 가중치를 수정하여 학습 성능 향상
주요 차이점	- 순전파는 오류 확인만 함 - 입력부터 출력까지의 단방향 전달	- 역전파는 오류를 역방향으로 전달하며 가중치 업데이트 수행
장단점	- 구현이 간단하지만 학습 능력이 낮음 - 병렬 처리에 유리함	- 학습 능력이 뛰어나지만 복잡한 연산과 메모리 요구량이 큼
결론	- 딥러닝 발전에 혁신적인 기여 - 단순한 모델에서도 사용 가능	- 딥러닝 학습의 초석이 되었으며, 다양한 신경망 구조에서 널리 사용됨

역전파 알고리즘이 신경망 학습에 혁신을 가져왔습니다. 그것은 최적화하기 위해 오류를 반대로 전파하는 능력을 통해 딥러닝의 발전에 기여했습니다

이 알고리즘은 복잡한 연산과 메모리 요구량 때문에 추가적인 비용이 들지만, 그것은 딥러닝 모델의 학습 능력을 크게 향상시켰습니다.

신경망 한계(90년대)

- 당시 신경망은 수렴이 잘 안되는 문제가 있었다. 특히 심층 신경망은 각 층의 가중치 초기화와 전파되는 gradient 발산 문제로 정확한 학습이 어려웠다.
- 컴퓨터 처리 능력의 제약으로 인해 심층 신경망을 효과적으로 학습시키기 어려웠다. 대규모 데이터셋과 많은 연산량이 필요했는데 이를 만족시킬 하드웨어가 없었다.
- 당시 주류를 이루던 전통적인 기계학습 기법들(SVM, 결정 트리 등)에 비해 신경망의 정확도가 나을 것이라는 확신이 없었다. 오히려 SVM 등의 성능이 좋다고 여겨졌다.
- 신경망 학습 알고리즘의 이론적 토대가 약했다. 기존 알고리즘의 한계를 극복할 새로운 아이디어가 제시되지 않았다.
- 딥러닝 관련 연구가 주목받지 못했고, 연구 자금과 인력 투입도 부족했다.

기계학습 주목

1980년대에 신경망은 역전파 알고리즘의 등장 등으로 관심을 끌며 연구가 활발히 진행되었습니다.
하지만, 1990년대 들어 신경망은 복잡한 문제 해결능력의 한계, 과적합 문제, 계산 자원 제약 등의 이유로 침체기를 맞이하였습니다.
이에 따라 1990년대에는 신경망의 대안으로 SVM, 결정 트리 등의 기계학습 기법들이 주목을 받으며 관련 연구가 활발해졌습니다.
SVM과 결정 트리는 계산 효율성, 해석 가능성, 이론적 토대의 견고함 등의 장점으로 신경망을 대체하였습니다.
그러나 2000년대 들어서면서 딥러닝의 등장으로 인해 신경망 구조가 부활하였습니다.

대안 기법 SVM, 결정트리

SVM 사례]

구체적 사례로, 스팸 메일 분류, 음성 인식, 암 진단, 주식 가격 예측 등에 SVM이 유용하게 사용되고 있습니다.

대표적으로 SVM은 다음과 같은 분야에서 유용하게 쓰입니다.
- 이미지 분류
  필기체 인식, 얼굴 인식 등 이미지 분류 문제에 SVM 활용
- 문서 분류
  스팸 메일 분류, 문서 주제 분류 등 텍스트 문서 분류에 강점
- 생물정보분류
  유전자 발현 데이터, 단백질 구조 데이터 등 생명정보 분류에 용이
- 시계열 분석
  주식 가격 변동, 공정 데이터 등 시계열 패턴 인식에 사용
- 작은 데이터셋
  데이터가 적은 문제에서 과적합 방지에 도움이 됨

기계 학습의 분류와 주요 알고리즘

아래 표는 기계학습의 주요 분류와 각 분류에 속하는 주요 알고리즘을 설명합니다.

분류	개념	주요 알고리즘
지도 학습 (Supervised Learning)	입력 데이터와 해당하는 정답(label) 데이터를 이용하여 모델을 학습하는 방법입니다. 모델은 입력과 출력 사이의 관계를 학습하고, 새로운 입력에 대한 출력을 예측할 수 있습니다.	선형 회귀 (Linear Regression), 로지스틱 회귀 (Logistic Regression), 결정 트리 (Decision Trees), 랜덤 포레스트 (Random Forests), 서포트 벡터 머신 (Support Vector Machines), 나이브 베이즈 (Naive Bayes), K-최근접 이웃(K-Nearest Neighbors) 등
비지도 학습 (Unsupervised Learning)	정답(label) 데이터 없이 입력 데이터만으로 모델을 학습하는 방법입니다. 주로 데이터에서 숨겨진 패턴, 구조, 군집 등을 발견하기 위해 사용됩니다.	군집화(Clustering): k-평균 군집화(k-Means Clustering), DBSCAN(Density-Based Spatial Clustering of Applications with Noise) 등 차원 축소(Dimensionality Reduction): 주성분 분석(Principal Component Analysis, PCA), t-SNE(t-Distributed Stochastic Neighbor Embedding) 등
강화 학습 (Reinforcement Learning)	에이전트가 환경과 상호작용하며 보상(reward) 신호를 받아 행동(action)을 선택하고 그 결과로부터 학습합니다. 목표는 최대의 누적 보상을 얻는 것입니다.	Q-Learning, 딥 Q 네트워크(Deep Q-Networks, DQN), 정책 그래디언트(Policy Gradient) 등

위 표는 기계학습의 일부만 다룬 것이며, 다른 유형의 알고리즘과 변형들도 존재합니다. 또한 최신 연구 및 개발 동향에 따라 새로운 알고리즘이 개발될 수 있습니다.

딥러닝 등장(2000년대)

신경망의 한계 극복 : 빅데이터(대규모 데이터셋), 컴퓨팅 파워, 알고리즘(모델), 오픈소스, 알고리즘 경쟁

1990년대 'AI 겨울'의 원인이었던 신경망의 한계를 극복하여 2000년대 초 딥러닝이 등장할 수 있었던 배경은 다음과 같습니다.
- 하드웨어 발전:
  GPU 등 병렬처리 칩의 발전으로 신경망 학습 속도가 빨라졌다.
- 빅데이터 축적:
  인터넷의 확산으로 대규모 데이터 생성 및 저장이 용이해졌다.
- 알고리즘 개선:
  옵티마이저, 정규화 등을 통해 과적합 문제 개선하였다.
- 신경과학 지식 적용:
  뇌 신경회로 연구로 신경망 관련 인사이트를 얻었다.
- 오픈소스 확산:
  TensorFlow, PyTorch 등 오픈소스로 알고리즘 공유가 원활해졌다.
- 대회 유행:
  ImageNet, Kaggle 등 AI 알고리즘 경쟁이 활발해졌다.

이러한 다양한 요인들이 복합적으로 작용하여 1990년대 신경망의 문제점을 극복하고 2000년대 딥러닝을 가능하게 했다고 볼 수 있습니다.

신경망의 리브랜딩 '딥러닝'

딥러닝은 기존 신경망 기술의 진화된 버전이지만, 과거의 한계극복/혁신을 강조하기 위해 리브랜딩했다고 볼 수 있습니다.

1. 딥러닝은 신경망 기반 기술
- 딥러닝은 신경망 모델을 바탕으로 한 기술로, 과거 신경망 연구의 연장선상에 있다.
2. '딥러닝'이라는 새로운 이름 사용
- 2000년대 초 기존 '신경망' 대신 '딥러닝'이라는 이름을 내세웠다.
3. 과거의 낙인에서 벗어나기 위해
- 1990년대 신경망의 한계로 인한 부정적 이미지 극복을 위해 새로운 브랜딩 시도
4. 진화된 기술 강조
- '딥'러닝이라는 이름으로 과거와의 차별성과 진보를 강조
5. 붐 조성 목적
- '딥러닝'이라는 용어로 신기술 트렌드성 부각, 연구/투자 붐 조성

다양한 AI 모델

2023년 현재, 인기 있는 모델을 선정하는 것은 다소 주관적일 수 있습니다. 그 이유는 사용되는 분야, 목적, 필요한 자원 등에 따라 가장 적합한 모델이 달라지기 때문입니다.

그러나, GPT-3와 같은 대형 언어 모델들이 많은 주목을 받고 있습니다. GPT-3는 1750억 개의 파라미터를 가진 대형 모델로서 텍스트 생성 능력이 매우 뛰어나며 다양한 어플리케이션에서 활용되고 있습니다.

또한 이미지 처리 분야에서는 EfficientNet과 Vision Transformer(ViT)가 주목받고 있습니다. EfficientNet은 네트워크 깊이(depth), 너비(width), 해상도(resolution) 등 여러 요소들을 동시에 고려하여 최적화된 성능을 제공하며 Vision Transformer(ViT)는 전체 이미지 맥락을 파악하는데 유용하게 사용됩니다.

DALL·E 역시 흥미로운 모델 중 하나로, 자연어 설명만으로 다양한 이미지를 생성할 수 있는 능력으로 인해 많은 관심을 받았습니다.

따라서 각 분야별로 보면 GPT-3(자연어 처리), EfficientNet 및 ViT(이미지 처리), DALL·E(이미지 생성) 등이 인기 있는 AI 모델로 볼 수 있겠습니다.

분류 모델 평가 도구 Confusion matrix

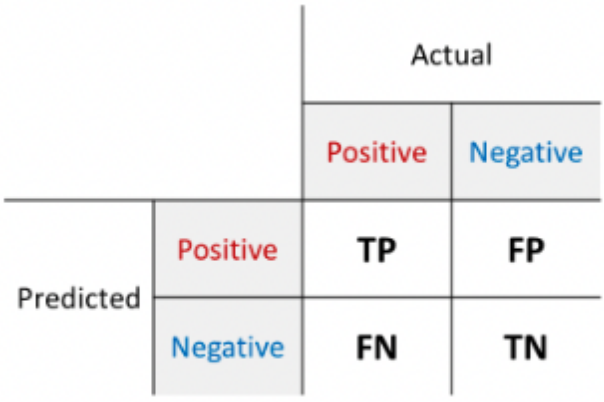


Figure 1. Confusion matrix

분류 모델 성능 지표
[!]Mermaid UML

Accuracy(정확도) : 분류 모델 평가하는 가장 단순한 지표, 불균형한 클래스를 가진 데이터셋 평가 어려운 단점 존재

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

불균형한 클래스 예시)

positive와 negative 비율이 2:8로 불균형한 클래스의 경우에 모든 예측을 negative로 하더라도 80%의 정확도

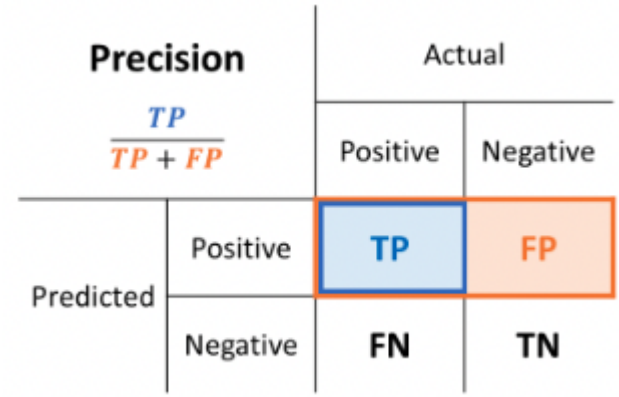
즉, postive 분량에 대해서는 모두 오류인 상황이지만 전체 정확도는 80%가 됨

그럼, 모델이 postive로 판정한 것 중에 실제 postive인 것과

실제 positive인 것 중에 모델이 positive로 판정한 것을 확인해 보면 좋겠다.

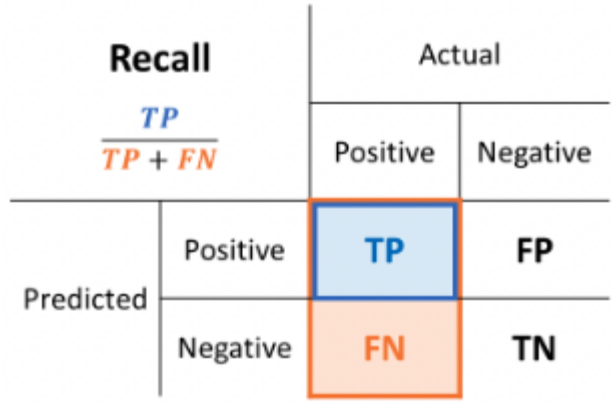
Precision(정밀도)

- 모델이 postive로 판정한 것 중에 실제 postive인 것



Recall(재현율)

- 실제 positive인 것 중에 모델이 positive로 판정한 것



분류모델의 성능을 동시에 고려하기 'F1 score'

- precision과 recall 의 조화평균

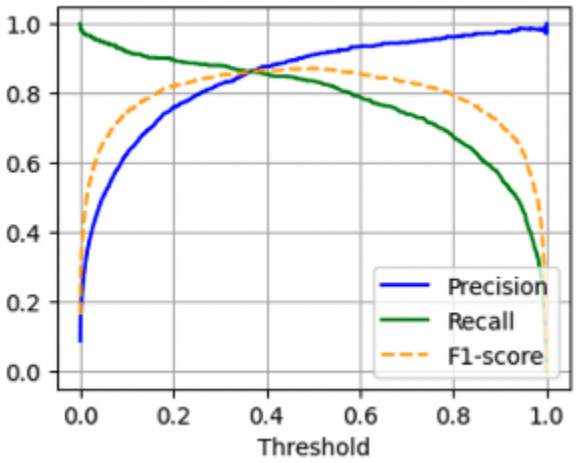
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

precision과 recall 의 trade-off 관계

- precision이 올라가면 recall이 떨어지고,

  recall이 올라가면 precision이 떨어지기 때문에

  적정의 decision threshold 설정 필요



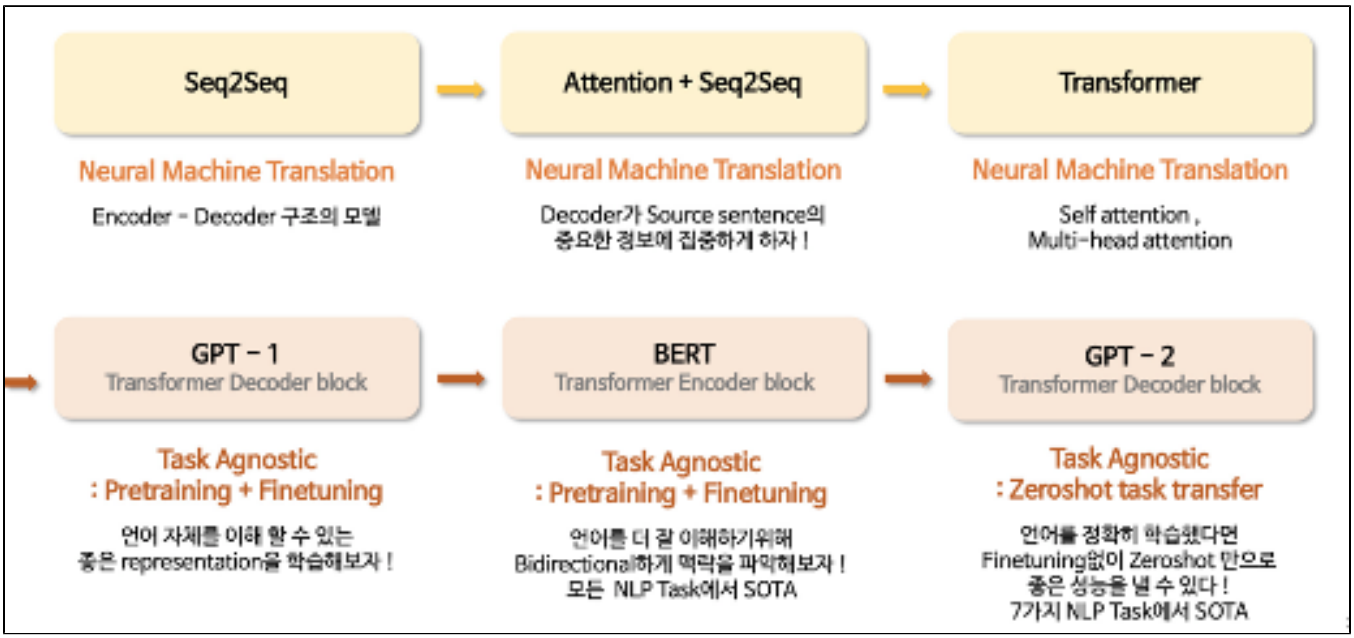
AI 모델 개발 시 중요 요소

분류	요소	설명
데이터 관련	데이터의 질과 양	AI 모델의 성능은 대부분 사용하는 데이터의 질과 양에 의해 결정됩니다. 더 많고 다양한 데이터를 사용하면 일반적으로 더 좋은 성능을 얻을 수 있습니다.
목표 설정	문제 정의 및 목표 설정	모델이 해결하려는 문제나 달성하려는 목표를 명확하게 설정하는 것이 중요합니다. 이것은 모델 선택, 학습 방법, 평가 메트릭 등에 영향을 줍니다.
리소스 관련	계산 리소스 및 저장 공간	AI 모델을 훈련시키기 위해서는 계산 리소스(예: GPU), 저장 공간 등이 필요합니다. 이런 리소스의 제약사항은 선택할 수 있는 모델의 종류와 크기에 영향을 줄 수 있습니다.
알고리즘 및 구조	네트워크 구조 및 최적화 알고리즘 선택	적절한 네트워크 구조와 최적화 알고리즘이 선택되어야 합니다. 예를 들어 이미지 분류 작업에는 CNN(Convolutional Neural Network)이 적합할 수 있으며, 시계열 데이터 처리에는 RNN(Recurrent Neural Network)이 적합할 수 있습니다.
성능 평가	성능 평가 메트릭 설정	어떻게 성능을 평가할 것인지 결정하는 것도 중요합니다. 일반적으로 훈련 데이터셋 외에 검증 및 테스트 데이터셋에서의 성능도 체크해야 합니다.
일반화 능력	오버피팅(과적합) 방지	모델이 새로운, 본 적 없는 데이터에 대해서도 잘 작동하도록 만드는 것입니다.
사회 윤리	윤리 및 사회적 고려사항	Ai 모델 개발과 사용 시 윤리적인 사용과 사회적 파급력 등을 고려해야 합니다.

Transformer

영어적 의미로는 작은 변화가 아니라 완전히 변경한다는 뜻

ㅇ 트랜스포머 기반 모델의 발전 과정



아래 표는 트랜스포머(Transformer)의 부상 배경과 진화 과정을 설명합니다.

연도	배경 및 진화 과정	주요 모델
2017	기존 RNN의 순차적 계산 한계를 극복하기 위해 동시 계산 가능한 non-RNN 모델이 필요하다는 인식으로 시작됨.	'Attention is All You Need' 논문에서 트랜스포머가 처음 등장. Encoder-Decoder 구조에 Attention mechanism 적용.
2018	RNN 없이 순차 입력의 위치 정보를 필요로 하지 않고 병렬 계산이 가능한 트랜스포머 모델 개선이 이루어짐.	BERT(Bidirectional Encoder Representations from Transformers) 제안, 사전 학습(pre-training) 방식 개선.
2019	GPT-2(Generative Pre-trained Transformer 2)와 XLNet 등으로 트랜스포머 모델이 발전함.	GPT-2: 대규모 양방향 사전 학습 언어 모델 XLNet: 자기 주변 문맥을 고려하는 사전 학습 언어 모델
2020	GPT-3(Generative Pre-trained Transformer 3)와 같은 거대한 트랜스포머 모델이 등장하며 NLP 성능 대폭 향상됨.	GPT-3: 거대한 다목적 자연어 처리 AI 모델 1,750억 개 파라미터를 가진 거대한 언어 모델
2021	GPT-3보다 더 큰 모델 연구 진행	GPT-Neo, GPT-J 등
2022	PaLM 발표. 2000억 개 파라미터 규모	PaLM (Pathways Language Model)
2022	ChatGPT 발표. 대화형 AI에 Transformer 적용	GPT-3.5(2022.11월)
2023	GPT-4(2023.3월)	한 번에 처리할 수 있는 단어량(token)[4]을 3,000개에서 25,000개로 8배 이상 확대 AI가 사실과 다른 것을 마치 진실인 것처럼 강한 확신을 담아 답변하는 문제인 할루시네이션을 상당 부분 줄이는 데 성공

트랜스포머 계열의 모델들은 RNN의 한계를 극복하고 병렬 처리와 대규모 학습을 통해 NLP 분야에서 주류로 자리잡게 되었습니다. 이러한 진화 과정을 거치면서 성능은 지속적으로 발전하고 있습니다.

BERT와 GPT과의 비교

[!]Mermaid UML

특징	BERT (Bidirectional Encoder Representations from Transformers)	GPT (Generative Pre-trained Transformer)
개념	문장 내 단어의 문맥을 이해하고 단어들 간의 관계를 파악하는 데 중점을 두며, 다음 단어를 생성하는 것이 아니라 문장 내에서 단어를 이해 - 양방향 언어 모델로, 주변 문맥을 고려하여 텍스트의 빈 칸(마스크된 단어)을 예측	자연어 생성에 중점 - 주어진 문맥에서 다음 토큰(단어)을 예측하고 텍스트를 자동으로 생성
접근 방식	양방향 Transformer 모델 - Autoencoder 기반의 bidirectional 모델	단방향 Transformer 모델 - Autoregressive 기반의 unidirectional 모델
사전 훈련 목적	빈칸을 예측 및 언어 모델 - Masked language modeling과 next sentence prediction에 초점	언어 모델 - 단일한 language modeling 태스크에 최적화
입력 표현	WordPiece embeddings 사용	Byte-pair encoding 사용
사전 훈련 데이터	대규모 텍스트 코퍼스 (e.g., 위키피디아)	대규모 텍스트 코퍼스 (e.g., 인터넷 글)
훈련 방법	양방향 언어 모델 훈련	단방향 언어 모델 훈련
특징 추출	각 단어의 표현을 추출하는 용도	문맥에서 다음 단어를 생성하는 용도
언어 이해 및 생성	언어 이해에 주로 사용	자연어 생성 및 이해에 사용
Fine-tuning	다양한 NLP 작업에 쉽게 적용 가능	주로 생성 작업에 적합
자연어 이해	문장의 의미와 관련된 작업에 강점	텍스트 생성 및 완성에 강점
언어 생성	텍스트 생성 작업에서 제한적인 사용	텍스트 생성 및 자유로운 작업에 적합
예시 응용 분야	텍스트 분류, 개체명 인식, 문장 유사도 등 - 주로 downstream task fine-tuning에 사용	자동 번역, 질의 응답, 텍스트 생성 등 - 주로 생성 모델로 사용

BERT와 GPT 인코더/디코더 아키텍처 측면 비교

특징	BERT (인코더 기반)	GPT (디코더 기반)
모델 유형	인코더-인코더 구조	디코더-디코더 구조
사전 훈련 목적	양방향 언어 모델 훈련	단방향 언어 모델 훈련
입력 표현	문장의 양쪽에서 문맥을 고려하여 표현합니다	이전 토큰들만을 고려하여 표현합니다
언어 이해 및 추출	텍스트 이해와 관련된 작업에 주로 사용됩니다	텍스트 생성 및 완성에 사용됩니다
특수 토큰	[CLS] 토큰을 추가하여 문장의 전체 의미를 표현	[PAD] 토큰을 통해 입력 길이를 일정하게 맞춥니다
자연어 생성	제한적인 텍스트 생성 능력	풍부한 자연어 생성 능력
Fine-tuning 가능 여부	다양한 NLP 작업에 적용 가능	주로 자연어 생성 작업에 적합
예시 응용 분야	언어 이해 분야 - 텍스트 분류, 개체명 인식, 문장 유사도 등	생성 분야 - 자동 번역, 질의 응답, 텍스트 생성 등

GPT 비교

[!]Mermaid UML	flowchart LR A[GPT-1 사전학습과 파인튜닝] -->|지도학습인 파인튜닝 제거| B[GPT-2 제로샷 학습 시도] B -->|대규모 데이터와 파라미터 인 컨텍스트 러닝| C[GPT-3 문맥을 추론하고 이해] C -->|RLHF추가 도입| D[GPT-3.5] D -->|멀티모달| E[GPT-4]

모델	기술 내용	파라미터 수
GPT-1	두 단계의 학습 과정 : 비지도학습의 pre-training + 지도학습의 fine-tuning 1) 비지도 학습을 통한 언어 모델 사전 훈련 2) 미세 조정을 통한 특정 작업 최적화. Transformer 네트워크 구조 사용.	1.17억 개

GPT-2	"제로샷" 학습 방법 시도 : 새로운 태스크에 대해서도 fine-tuning 없이 바로 사용이 가능함 - 지도학습인 fine-tuning을 제거 - 지도학습 없이도 다양한 작업 수행 가능. - 웹텍스트라는 대규모 데이터셋에 기반하여 학습 - Layer normalization 위치 변경 등 네트워크 구조 개선.	15억 개
GPT-3	많은 데이터와 파라미터 + 인 컨텍스트 러닝 엄청난 크기의 파라미터를 가짐. 인 컨텍스트 러닝 본격 적용 : 주어진 문맥 속에서 추론하고 이해하는 능력 - fine-tuning과 다르게 LLM 자체는 건드리지 않고, inference 시에(질문할 때) 질문을 잘 해보자는 접근 (LLM을 훈련시키는 fine tuning : LLM 이 점점 커지면서 fine tuning(재훈련)이 너무 오래 걸리고 비싼 작업으로 효용성 낮아짐) "퓨샷(적은 입력)" 학습 방법 도입 : 몇 가지 예시만으로도 작업 수행 방법 습득 가능	1750억개 토큰수 : 2048개
GPT-3.5	GPT-3의 학습방식을 그대로 차용했으나 RLHF라는 기법을 추가로 도입 - Reinforcement Learning from Human Feedback (사람이 직접 피드백을 주는 방식으로 언어모델을 최적화하는 기법) - in-context learning으로 input을 이해하고, few-shot learning을 통해 prompt를 수행	1750억개
GPT-4	많은 글자수 추론, 그림을 분석하는 것까지 확장 - 32,768개의 토큰을 받을 수 있으며 50페이지가 넘는 텍스트를 읽고 추론이 가능 멀티 모달(여러 가지 종류의 데이터를 함께 사용하여 인공지능 모델의 성능을 높이는 기술 멀티 모달은 이미지, 음성, 비디오 등 다양한 유형의 데이터를 함께 사용한다. 이를 통해 모델이 보다 현실적인 문제를 해결할 수 있게 되며, 사용자 경험을 높일 수 있다.	조단위 예상(미공개)

GPT3.5와 GPT4.0

모델	GPT 4.0	GPT 3.5
개요	보다 정교한 자연어 또는 코드의 이해와 생성	정교한 자연어 또는 코드의 이해와 생성
모델 크기	1조 7600억 개의 파라미터	1750억 개의 파라미터
입력 크기	최대 32,768개의 토큰	최대 4096개의 토큰
멀티 모달 (Multimodal)	도입	미도입
계산 요구 비용	상대적으로 높음	상대적으로 낮음

GPT 4.0의 장점

- 의료 검사: 약 92%의 정확도(기존보다 약 5%상승)
- 법률 애플리케이션: 약 88%의 정확도(기존보다 약7%상승)
- 감성 분석: GPT3.5보다 향상된 성능
- 이미지를 이해할 수 있음

GPT 3.5의 장점

- 속도가 GPT 4.0에 비해 빠름
- 언어번역 측면에서 GPT 4.0와 비슷한 성능을 보임

주의사항

- GPT 모델 사용 시 "프롬프트"와 생성된 출력 값 길이의 합이 모델이 허용하는 최대 길이를 넘어서는 안됨

인 컨텍스트 러닝(GPT-3)

문맥 이해 → 사전 지식 없이 추론	여기서 인 컨텍스트 러닝 모델은 먼저 문맥(1번)을 이해합니다. 그리고 그 문맥 아래에서 질문(2번)을 받으면 문맥을 바탕으로 답변(3번)을 생성합니다. 즉, 사전 지식 없이 주어진 문맥 내에서 질문을 이해하고 답변을 추론하는 능력을 보여줍니다. 이는 기존 모델을 새로 훈련하지 않고도 가능한 인 컨텍스트 러닝의 가장 큰 장점입니다

```
1.
"    ."
2.
"    ?"
3.
"    ."
```

퓨샷 러닝 예시 (GPT-3)

[!]Mermaid UML

task, example, prompt	수행할 업무(명령), 조금의 예시, 프롬프트를 입력해주면 업무 내용과 예시를 통해 결과를 예측

```
#task
  :

       #example
       few =>
    shot =>
    learning =>

    #prompt
    important => _____
```

RLHF(GPT-3.5)

- 이슈 : LLM은 훈련 데이터를 기반으로 다음 단어를 예측할 뿐이기 때문에 어떤 문장들을 만들어낼지 알기 어렵다.

  엉뚱하게도 가끔 정치적, 종교적, 인종적으로 문제가 되는 말을 하기도 했습니다.

- 해소 : 가이드라인을 따르도록 좀 더 훈련(좋은 피드백을 받을 수 있는 쪽으로 훈련 → 훈련이 잘 되면 LLM은 좋은 피드백인 보상(사람들의 피드백)을 획득)

  - 참고 사항 : RLHF를 거쳤다고 해서 LLM의 성능이 향상되는 것은 아니며, RLHF의 목적은 LLM의 응답을 특정 방향으로 틀어 주는 역할(LLM의 기본 능력은 사전 학습 모델 자체로 결정)

RLAIF (Anthropic)

- 이슈 : RLHF 수행 시 요구되는 사람을 투입하는 비용과 확장성의 제약
- 해소 : AI에게 LLM의 응답을 평가하게 하여 이 평가를 피드백으로 사용하자는 아이디어 → Reinforcement Learning with AI Feedback
- 솔루션 : 피드백용LL을 만들어서 대응

  - 몇가지 규칙을 정해 설계 : " 네가 한 응답이 윤리적, 사회적으로 문제가 되는 점이 있는지 말해봐."

```
:           .
LLM:      ......
 LLM:    ,      .
LLM:       .           .
 LLM:       .
LLM:  AI AI        .
```

LLaMA : 작지만 강력한 모델

- 이슈 : LLM은 좋은 기술이지만 모델이 너무 큼
- 해소 : 작은 모델로 유사한 성능 발휘 시도 → LLaMA(Large Language Model Meta AI)
- 솔루션 : 더 많은 데이터를 써서 훈련시키고, 더 오래 훈련시키기를 통해 성능 향상
  - 결과 : Meta의 발표에 따르면 LLaMA 13B 모델은 GPT-3 175B 보다 좋고, LLaMA 65B 모델은 PaLM 540B와 비슷

GPT의 토큰

GPT의 token : 텍스트 처리 단위로 한국어의 형태소와 유사한 개념 예) tokenization은 2개의 토큰으로 분리됨(token + ization)

- 한국어가 상대적으로 토큰의 양이 많아짐

NER

NER(Named Entity Recognition)은 자연어 처리에서 중요한 태스크 중 하나로, 주어진 텍스트에서 개체명(Entity)을 인식하고 태깅하는 기술입니다.
NER의 주요한 목적은 텍스트에 언급된 개체명들을 정의된 유형별로 분류하는 것입니다.
대표적으로 인명(Person), 지명(Location), 기관명(Organization) 등의 고유명사(Proper Noun)를 인식합니다.
예를 들어 "김철수는 서울에서 삼성전자에 다닌다"는 문장이 주어졌을 때, NER은 '김철수'를 인명, '서울'을 지명, '삼성전자'를 기관명으로 올바르게 태깅할 수 있어야 합니다.
NER은 의미 있는 정보를 추출하기 위한 전처리 과정으로, 기계 번역, 질의응답, 정보검색 등 자연어 처리의 다양한 응용 분야에서 널리 사용되고 있습니다