

# 딥러닝의 역사: 개념의 탄생부터 현대적 혁신까지

## I. 서론

### 딥러닝의 정의와 중요성

딥러닝(Deep Learning)은 여러 층(Layer)으로 구성된 인공신경망(Artificial Neural Networks, ANN)을 기반으로 하는 기계학습(Machine Learning, ML)의 한 분야이다.<sup>1</sup> 이는 데이터로부터 계층적인 표현(Hierarchical Representations)을 학습하는 것을 목표로 한다. 딥러닝은 컴퓨터 비전, 자연어 처리(Natural Language Processing, NLP), 음성 인식 등 다양한 분야에서 혁신적인 발전을 이끌며 인공지능(Artificial Intelligence, AI) 기술의 핵심 동력으로 자리 잡았다.<sup>4</sup> 인간의 뇌 구조와 기능에서 영감을 받았지만, 딥러닝 모델은 통계적 학습 알고리즘이며 인간 지능과 직접적인 유사성을 판단하기는 어렵다.<sup>9</sup>

### 보고서의 범위와 구조

본 보고서는 딥러닝의 역사적 발전 과정을 연대기적으로 추적하여, 초기 개념 정립부터 현대의 기술적 혁신에 이르기까지 주요 사건과 이론, 기술 발전을 종합적으로 고찰하는 것을 목적으로 한다. 이는 대학 수준의 보고서("학교 레포트") 형식에 맞춰 작성되었다. 보고서는 초기 인공신경망 개념의 태동, 연구 침체기였던 'AI 겨울', 핵심 알고리즘인 역전파의 개발, 주요 연구자들의 기여, 대표적인 딥러닝 아키텍처(CNN, RNN 등)의 진화, 대규모 데이터셋과 컴퓨팅 성능 향상의 영향, 그리고 다양한 응용 분야에서의 돌파구를 시대별로 나누어 살펴볼 것이다.<sup>10</sup>

## II. 인공신경망의 기원 (1940년대-1960년대): 아이디어의 씨앗

### 뇌의 초기 계산 모델

인공신경망 연구의 뿌리는 뇌의 작동 방식을 수학적으로 모델링하려는 시도에서 찾을 수 있다. 1943년, 워런 맥클락(Warren McCulloch)과 월터 피츠(Walter Pitts)는 신경세포(뉴런)의 기능을 모방한 최초의 수학적 모델을 제안했다.<sup>12</sup> 이 모델은 여러 입력을 받아 하나의 출력을 내는 단순한 구조로, 각 입력에는 가중치가 부여될 수 있음을 시사했다.<sup>13</sup> 맥클락-피츠 모델은 뉴런이 임계값을 넘어서면 활성화(fire)되는 방식을 모방하여 기본적인 논리 연산(AND, OR, NOT 등)을 수행할 수 있음을 보였다.<sup>14</sup> 이는 초기 인공지능 연구에서 중요한 진전이었으나, 스스로 학습하는 메커니즘은 결여되어 있었다.<sup>14</sup>

이후 1949년, 도널드 헵(Donald Hebb)은 생물학적 학습 원리, 즉 "함께 활성화되는 뉴런들은 서로 연결이 강화된다(neurons that fire together, wire together)"는 가설을 제시했다.<sup>12</sup> 헵의 학습 규칙은 신경망이 경험을 통해 스스로 연결 강도(가중치)를 조절하여 학습할 수 있다는 아이디어의 기초를 마련했으며, 이후 인공신경망 학습 알고리즘 개발에 큰 영감을 주었다.<sup>12</sup>

초기 인공지능 연구는 이처럼 뇌의 정보 처리 방식(맥클락-피츠 모델의 계산 기능)과 학습 메커니즘(헵의 학습 원리)을 모두 이해하고 모방하려는 이중적인 목표를 가지고 있었다. 이러한 계산과 적응이라는 두 가지 측면은 이후 딥러닝의 역사 전반에 걸쳐 지속적으로 나타나는 주제가 된다.

#### 퍼셉트론과 아달라인: 초기 학습 알고리즘

헵의 학습 원리에 영감을 받아, 1957년(또는 1958년) 코넬 항공 연구소의 프랭크 로젠블랫(Frank Rosenblatt)은 퍼셉트론(Perceptron)이라는 학습 가능한 신경망 모델을 개발했다.<sup>12</sup> 퍼셉트론은 맥클락-피츠 모델에 학습 능력을 부여한 것으로, 입력값에 가중치를 곱한 값들의 합이 특정 임계값(threshold)을 넘으면 1(활성화), 그렇지 않으면 -1 또는 0을 출력하는 구조였다.<sup>12</sup> 중요한 점은 퍼셉트론이 '학습 규칙'을 통해 입력 데이터로부터 가중치를 자동으로 조정하여 선형적으로 분리 가능한 패턴을 분류하는 방법을 학습할 수 있다는 것이었다.<sup>13</sup> 퍼셉트론의 등장은 큰 반향을 일으켰고, 언론에서는 이를 "스스로 학습하는 전자 두뇌"로 묘사하며 미래에 대한 기대를 높였다.<sup>17</sup>

퍼셉트론이 발표된 지 얼마 지나지 않은 1960년, 스탠포드 대학의 버나드 위드로(Bernard Widrow)와 그의 제자 테드 호프(Marcian "Ted" Hoff)는 아달라인(ADALINE: Adaptive Linear Neuron)이라는 개선된 모델을 개발했다.<sup>12</sup> 퍼셉트론이 최종 출력값(1 또는 -1)과 목표값의 차이를 이용해 가중치를 조정 한 반면, 아달라인은 활성화 함수(임계값 함수)를 통과하기 전의 순입력(net input)과 목표값 사이의 오차를 최소화하도록 가중치를 조정했다.<sup>12</sup> 이를 위해 아달라인은 비용 함수(Cost Function, 주로 오차 제곱합)를 정의하고, 경사 하강법(Gradient Descent)을 사용하여 이 비용 함수를 최소화하는 방향으로 가중치를 업데이트하는 델타 규칙(Delta Rule) 또는 LMS(Least Mean Squares) 알고리즘을 사용했다.<sup>12</sup>

이 초기 모델들에서 나타난 학습 방식의 차이는 주목할 만하다. 퍼셉트론은 최종 분류 결과의 정확성에 초점을 맞춘 반면, 아달라인은 연속적인 최적화 기법(경사 하강법)을 사용하여 활성화 이전 단계에서의 오차를 최소화하려 했다. 이는 이후 딥러닝에서 사용되는 다양한 손실 함수(loss function)와 최적화 알고리즘 논의의 초기 형태를 보여준다.

### III. 첫 번째 AI 겨울: 환멸의 시대 (1970년대-1980년대 초)

#### 이론적 한계의 노출

초기의 열광에도 불구하고, 단층 퍼셉트론(Single-Layer Perceptron)은 곧 명확한 한계에 부딪혔다. 가장 치명적인 약점은 선형적으로 분리할 수 없는 문제는 해결할 수 없다는 점이었다. 이는 간단한 논리 연산인 XOR(Exclusive OR) 문제조차 풀 수 없다는 사실로 극명하게 드러났다.<sup>11</sup>

이러한 한계는 1969년 마빈 민스키(Marvin Minsky)와 시모어 페퍼트(Seymour Papert)가

출간한 저서 "퍼셉트론(Perceptrons)"에서 수학적으로 엄밀하게 분석되면서 널리 알려졌다.<sup>15</sup> 이 책은 단층 퍼셉트론의 능력과 한계를 명확히 규명했으며, 특히 여러 층으로 구성된 다층 퍼셉트론(Multi-Layer Perceptron, MLP)이 XOR 문제 등을 해결할 잠재력은 인정하면서도, 당시에는 이러한 다층 구조를 효과적으로 학습시킬 알고리즘이 알려져 있지 않다는 점을 지적하며 회의적인 시각을 드러냈다.<sup>15</sup>

#### 외부적 압력과 연구 자금 삭감

민스키와 페퍼트의 비판은 학계와 연구 자금 지원 기관에 큰 영향을 미쳤다. 초기 연구자들이 제시했던 지나치게 낙관적인 예측들이 실현되지 못하면서<sup>23</sup>, 신경망 연구에 대한 회의론이 확산되었고, 이는 연구 자금 지원의 대폭적인 삭감으로 이어졌다.<sup>15</sup>

특히 영국에서는 1973년 제임스 라이트힐(James Lighthill) 경이 과학 연구 위원회(SRC)의 요청으로 작성한 보고서(라이트힐 보고서)가 결정적인 역할을 했다.<sup>23</sup> 이 보고서는 인공지능 연구 전반에 대해 비판적인 평가를 내렸는데, 특히 로봇 공학이나 언어 처리와 같은 기초 연구 분야가 현실 세계의 복잡성, 즉 '조합적 폭발(combinatorial explosion)' 문제를 해결하는 데 실패했다고 지적했다.<sup>26</sup> 라이트힐 보고서는 영국 내 대부분 대학의 AI 연구 지원 중단이라는 결과를 초래하며 AI 연구의 암흑기, 이른바 'AI 겨울(AI Winter)'을 심화시키는 데 크게 기여했다.<sup>26</sup>

#### 복합적인 원인들

첫 번째 AI 겨울은 단순히 퍼셉트론의 한계나 특정 보고서 때문만은 아니었다. 여러 요인이 복합적으로 작용한 결과였다. 당시 컴퓨터의 연산 능력 부족은 복잡한 신경망 모델의 구현과 학습을 현실적으로 어렵게 만들었다.<sup>23</sup> 또한, 초기 AI 연구가 직면했던 근본적인 문제들, 예를 들어 상식 추론의 어려움, 방대한 경우의 수를 처리해야 하는 조합적 폭발 문제 등은 신경망뿐만 아니라 기호주의(Symbolism) 기반의 AI 연구에도 큰 난관이었다.<sup>23</sup> 결국 이론적 한계, 비판적 평가, 연구 자금 삭감, 부족한 컴퓨팅 성능, 그리고 초기의 과장된 기대에 대한 실망감이 복합적으로 작용하여 AI 연구 전반, 특히 신경망 연구는 긴 침체기에 접어들게 되었다.<sup>1</sup>

이 시기의 침체는 신경망 연구에만 국한된 것이 아니었다는 점을 이해하는 것이 중요하다. 민스키와 페퍼트의 비판이 신경망 분야에 직접적인 타격을 주었지만, AI 겨울은 기호주의 AI가 직면한 어려움(예: 조합적 폭발<sup>23</sup>), 야심 찬 프로젝트들의 실패, 라이트힐 보고서와 같은 외부 요인<sup>26</sup> 등 AI 분야 전체에 영향을 미친 광범위한 현상이었다. 신경망 연구는 이러한 전반적인 AI 분야의 침체 분위기 속에서 함께 어려움을 겪었던 것이다.

## IV. 불씨의 재점화: 역전파와 연결주의의 부상 (1980년대)

#### 역전파 알고리즘의 돌파구

단층 퍼셉트론의 한계를 극복하기 위해 다층 퍼셉트론(MLP) 구조가 제안되었지만<sup>17</sup>,

이를 효과적으로 학습시키는 방법, 즉 은닉층(hidden layer)의 가중치를 어떻게 조정해야 할지에 대한 문제가 해결되지 않고 있었다.<sup>17</sup> 아달라인의 델타 규칙은 단일 출력층에는 적용 가능했지만, 여러 층으로 구성된 네트워크의 내부 가중치를 학습시키는 데는 직접 적용하기 어려웠다.<sup>12</sup>

이 문제에 대한 해결책으로 등장한 것이 바로 역전파(Backpropagation) 알고리즘이다. 역전파의 핵심 아이디어는 출력층에서 계산된 오차(실제 값과 예측값의 차이)를 네트워크의 마지막 층부터 입력층 방향으로 역으로 전파시키면서, 각 가중치가 최종 오차에 얼마나 기여했는지를 미분 연쇄 법칙(chain rule)을 이용해 계산하고, 이를 바탕으로 경사 하강법을 적용하여 가중치를 업데이트하는 것이다.<sup>32</sup>

역전파 알고리즘 자체는 폴 베르보스(Paul Werbos)가 1974년 박사 학위 논문에서 이미 제안했지만<sup>33</sup>, 당시에는 큰 주목을 받지 못했다. 이 알고리즘이 널리 알려지고 신경망 연구의 부활을 이끈 것은 1986년 데이비드 루멜하트(David Rumelhart), 제프리 힌튼(Geoffrey Hinton), 로날드 윌리엄스(Ronald Williams)가 발표한 논문 "Learning Representations by Back-Propagating Errors"를 통해서였다.<sup>32</sup> 이 논문은 역전파 알고리즘을 이용해 다층 퍼셉트론이 XOR 문제와 같은 비선형 문제를 학습할 수 있음을 명확히 보여주었고<sup>33</sup>, 이는 신경망 연구에 새로운 활력을 불어넣었다. 역전파가 제대로 작동하기 위해서는 퍼셉트론의 계단 함수(step function)와 달리 미분 가능한 활성화 함수가 필요했고, 이 시기에 시그모이드(sigmoid) 함수 등이 널리 사용되기 시작했다.<sup>13</sup>

#### 다층 퍼셉트론(MLP)과 새로운 기대

역전파 알고리즘의 등장은 입력층과 출력층 사이에 하나 이상의 은닉층을 가진 다층 퍼셉트론(MLP)의 시대를 열었다. MLP는 은닉층을 통해 데이터의 내부 표현(internal representation)을 학습할 수 있었고, 이론적으로 충분한 노드와 적절한 가중치가 주어진다면 거의 모든 종류의 복잡한 비선형 함수를 근사할 수 있는 능력을 갖추게 되었다.

역전파를 통한 MLP의 성공적인 학습은 인공지능 연구에서 연결주의(Connectionism) 접근법의 부활을 알리는 신호탄이었다.<sup>10</sup> 이는 뇌의 신경망 구조에서 영감을 받아 분산된 표현과 병렬 처리 능력을 강조하는 접근 방식으로, 기호주의적 접근법과 경쟁하며 AI 연구의 또 다른 흐름을 형성했다. 이 시기는 신경망 연구의 두 번째 물결로 간주된다.<sup>10</sup>

한편, 1980년대에는 전문가 시스템(Expert Systems)이 상업적으로 성공하면서 첫 번째 AI 겨울을 끝내는 데 기여하기도 했다.<sup>1</sup> 전문가 시스템은 특정 분야의 전문가 지식을 규칙 기반으로 구현한 시스템이었으나, 지식 획득의 어려움과 새로운 상황에 대한 적응력 부족이라는 한계에 부딪히면서 결국 두 번째 AI 겨울의 원인 중 하나가 되었다.<sup>1</sup>

역전파 알고리즘의 재발견과 대중화는 민스키와 페퍼트가 제기했던 MLP 학습의 어려움이라는 핵심 문제를 해결한 기념비적인 사건이었다.<sup>15</sup> 하지만 이것이 곧바로

오늘날과 같은 딥러닝의 성공으로 이어진 것은 아니었다. 역전파 알고리즘 자체도 깊은 네트워크에서는 기울기 소실(vanishing gradient) 문제<sup>32</sup>라는 새로운 난관에 봉착했고, 여전히 부족한 컴퓨팅 파워<sup>23</sup>는 복잡하고 깊은 네트워크의 실용적인 학습을 제약했다. 이러한 문제들은 결국 신경망 연구가 다시 한번 침체기를 맞는, 두 번째 AI 겨울로 이어지는 배경이 되었다.

## V. 아키텍처 혁신과 핵심 연구자들 (1980년대 후반-2000년대)

분야의 선구자들: 힌튼, 르쿤, 벤지오

AI 겨울의 어려움 속에서도 신경망 연구의 명맥을 이어가며 현대 딥러닝의 토대를 마련한 핵심 연구자들이 있었다. 제프리 힌튼(Geoffrey Hinton), 얀 르쿤(Yann LeCun), 요슈아 벤지오(Yoshua Bengio)는 이 분야의 3대 거장으로 꼽히며, 이들의 끊임없는 연구와 협력은 딥러닝 발전에 결정적인 기여를 했다.<sup>38</sup> 이들은 딥러닝 분야에 대한 공로를 인정받아 2018년 ACM 튜링상(Turing Award)을 공동 수상했다.<sup>38</sup>

- 제프리 힌튼: 딥러닝의 "대부"로 불리는 힌튼은 루멜하트, 윌리엄스와 함께 역전파 알고리즘을 널리 알린 핵심 인물 중 한 명이다.<sup>33</sup> 그는 볼츠만 머신(Boltzmann Machine)<sup>39</sup>, 제한된 볼츠만 머신(Restricted Boltzmann Machine, RBM)<sup>42</sup>, 그리고 이를 쌓아 만든 심층 신뢰 신경망(Deep Belief Network, DBN)<sup>39</sup> 연구를 통해 깊은 신경망의 학습 문제를 해결하려는 중요한 시도를 했다. 특히, RBM과 DBN을 이용한 비지도 사전 학습(unsupervised pre-training) 방법은 초기 가중치를 효과적으로 설정하여 기울기 소실 문제를 완화하고 깊은 네트워크의 학습을 가능하게 하는 데 기여했다.<sup>44</sup> 또한 분산 표현(distributed representation) 개념을 발전시키고<sup>42</sup>, ReLU 활성화 함수의 효과를 강조하는 등<sup>9</sup> 딥러닝의 이론적, 실제적 발전에 지대한 영향을 미쳤다.
- 얀 르쿤: 르쿤은 합성곱 신경망(Convolutional Neural Network, CNN) 분야를 개척한 선구자이다.<sup>41</sup> 그는 1980년대 후반 AT&T 벨 연구소에서 CNN 구조의 초기 형태인 LeNet을 개발하여, 손글씨 숫자 인식(MNIST) 및 우편번호/수표 인식과 같은 실제 문제에 성공적으로 적용했다.<sup>2</sup> 이는 딥러닝 기술이 실용화된 최초의 사례 중 하나로 평가받는다.<sup>41</sup> 르쿤은 이후 페이스북 AI 연구소(FAIR)의 초대 소장을 맡아 연구 결과를 적극적으로 공개하고 학계와 교류하는 개방형 연구 모델을 추구하며 AI 연구 커뮤니티의 발전에 기여했다.<sup>5</sup>
- 요슈아 벤지오: 벤지오는 힌튼, 르쿤과 함께 현대 딥러닝 시스템을 완성하는 데 핵심적인 역할을 했다.<sup>38</sup> 그는 특히 순환 신경망(Recurrent Neural Network, RNN)과 장단기 메모리(Long Short-Term Memory, LSTM) 네트워크를 포함한 시퀀스 모델링<sup>38</sup> 및 신경망 언어 모델(Neural Language Models)<sup>38</sup> 분야에서 중요한 기여를 했다. 또한, 주의(Attention) 메커니즘<sup>38</sup>, 생성적 적대 신경망(Generative Adversarial Network, GAN)<sup>38</sup> 연구에도 참여했으며, ReLU 활성화 함수<sup>38</sup> 및 Xavier 초기화<sup>38</sup>와 같이 딥러닝 학습을 안정화하고 개선하는 기술 개발에도 기여했다. 그는 몬트리올 학습 알고리즘 연구소(MILA)를 이끌며 캐나다를 AI 연구의 허브로 만드는 데 중요한

역할을 했고, ElementAI라는 스타트업을 공동 창업하기도 했다.<sup>38</sup>

표 1: 딥러닝 선구자 3인의 주요 기여 요약

연구자	주요 기여	주요 시기/연도	중요성
제프리 힌튼	역전파 알고리즘 (대중화), 볼츠만 머신/RBM/DBN, 비지도 사전 학습, ReLU 활성화 함수 (옹호), 분산 표현	1980년대 - 2000년대	깊은 신경망 학습 난제 해결 시도, 비지도 학습 기반 초기화, 딥러닝 분야 창시자 역할
얀 르쿤	합성곱 신경망(CNN) 개척, LeNet 아키텍처 개발 (손글씨/문서 인식), 역전파 알고리즘 응용, FAIR 설립 (개방형 연구)	1980년대 후반 - 1990년대	컴퓨터 비전 분야 혁신, 초기 실용적 딥러닝 응용 성공, 개방형 연구 문화 확산 기여
요슈아 벤지오	순환 신경망(RNN)/LSTM, 신경망 언어 모델, 주의(Attention) 메커니즘, 생성적 적대 신경망(GAN) 공동 연구, ReLU/Xavier 초기화, ElementAI 설립	1990년대 - 2010년대	시퀀스 모델링 및 자연어 처리 발전, 생성 모델 연구 기여, 딥러닝 학습 안정화 및 성능 향상, AI 기술 상업화 및 생태계 조성 기여

이 세 연구자의 개별적이고 협력적인 노력은 딥러닝이 이론적 가능성을 넘어 실용적인 기술로 발전하는 데 결정적인 역할을 했다.

#### 합성곱 신경망(CNN): 공간적 계층 구조의 정복

이미지와 같은 고차원 데이터를 처리하기 위해 기존의 완전 연결(fully connected) MLP는 한계가 있었다. 이미지의 2차원 공간 구조 정보를 활용하지 못하고, 입력 픽셀 수가 늘어남에 따라 파라미터 수가 기하급수적으로 증가하여 학습이 비효율적이었다.<sup>51</sup> 이러한 문제를 해결하기 위해 얀 르쿤이 개척한 합성곱 신경망(CNN)이 등장했다.<sup>41</sup>

CNN의 핵심 구성 요소는 다음과 같다:

- **합성곱 층 (Convolutional Layer):** 학습 가능한 필터(커널)를 사용하여 입력 이미지 위를 이동하면서 합성곱 연산(필터와 입력 영역 간의 내적)을 수행한다.<sup>51</sup> 각 필터는



이미지의 지역적 특징(local feature), 예를 들어 특정 방향의 엣지, 코너, 색상, 질감 등을 감지하도록 학습된다.<sup>52</sup> 중요한 특징은 **\*\*파라미터 공유(parameter sharing)\*\***로, 하나의 필터가 이미지 전체 영역에 동일하게 적용되어 파라미터 수를 크게 줄이고 공간적 위치에 관계없이 동일한 특징을 감지할 수 있게 한다 (translation invariance).<sup>51</sup> 필터의 이동 간격인 **\*\*스트라이드(stride)\*\***와 입력 이미지 경계 처리 방식인 패딩(**padding**)(예: zero-padding)은 출력 특징 맵(feature map)의 크기를 조절하는 하이퍼파라미터이다.<sup>51</sup> 합성곱 층을 여러 개 쌓으면, 초기 층에서는 단순한 특징을 감지하고 후속 층에서는 이를 조합하여 더 복잡하고 추상적인 특징(예: 객체의 부분, 전체 객체)을 학습하는 **\*\*계층적 특징 추출(hierarchical feature extraction)\*\***이 가능해진다.<sup>52</sup>

- **풀링 층 (Pooling Layer) / 서브샘플링 층 (Subsampling Layer):** 합성곱 층 이후에 주로 위치하며, 특징 맵의 공간적 차원을 축소(downsampling)하는 역할을 한다.<sup>52</sup> 이는 모델의 파라미터 수와 계산량을 줄여 과적합(overfitting)을 방지하고, 입력의 작은 변화에 대해 모델이 좀 더 강인하게(robust) 반응하도록 돕는다 (약간의 translation invariance 제공).<sup>52</sup> 대표적인 풀링 방식으로는 특정 영역 내의 최댓값을 선택하는 **\*\*최대 풀링(Max Pooling)\*\***과 평균값을 사용하는 **\*\*평균 풀링(Average Pooling)\*\***이 있다.<sup>53</sup> 풀링 필터는 가중치 없이 고정된 연산을 수행한다.<sup>53</sup>

르쿤이 개발한 LeNet<sup>49</sup>은 이러한 합성곱 층과 풀링 층을 번갈아 사용하고 마지막에 완전 연결 층(Fully Connected Layer)을 두어 분류를 수행하는 CNN의 기본 구조를 확립했다.<sup>49</sup> LeNet은 MNIST 손글씨 숫자 데이터셋과 미국 우편 서비스의 우편번호 인식 등에서 뛰어난 성능을 보여주며 CNN의 가능성을 입증했다.<sup>2</sup> 이는 신경망이 실제 산업 문제에 적용될 수 있음을 보여준 중요한 초기 성공 사례였다.<sup>2</sup>

순환 신경망(RNN)과 LSTM: 순서와 기억의 모델링

텍스트나 시계열 데이터와 같이 순서가 중요한 데이터를 처리하기 위해 순환 신경망(Recurrent Neural Network, RNN)이 고안되었다. RNN은 네트워크 내부에 순환(loop) 구조를 가지고 있어 이전 단계의 정보를 현재 단계의 계산에 활용할 수 있다. 이를 통해 시퀀스 내의 의존성(dependency)을 모델링하고 일종의 "기억"을 유지할 수 있다.<sup>38</sup>

하지만 기본적인 RNN 구조는 기울기 소실(vanishing gradient) 문제에 취약했다.<sup>60</sup> 시퀀스가 길어질 경우, 역전파 과정(Backpropagation Through Time, BPTT)에서 오차 신호가 시간을 거슬러 전파되면서 점차 작아져서 거의 사라지게 된다.<sup>9</sup> 이는 RNN이 시퀀스 내에서 멀리 떨어진 요소들 간의 장기 의존성(long-range dependency)을 학습하기 어렵게 만드는 주요 원인이었다.<sup>62</sup> 기울기가 너무 커지는 기울기 폭발(exploding gradient) 문제도 존재했다.<sup>60</sup> 기울기 소실 문제는 1991년 제프 호크라이터(Sepp Hochreiter)의 석사 논문에서 이미 분석된 바 있다.<sup>60</sup>

이러한 RNN의 한계를 극복하기 위해 1997년, 제프 호크라이터와 워르겐

슈미트후버(Jürgen Schmidhuber)는 장단기 메모리(Long Short-Term Memory, LSTM) 네트워크를 제안했다.<sup>61</sup> LSTM은 기울기 소실 문제를 해결하도록 특별히 설계된 RNN의 변형 구조이다.<sup>61</sup>

LSTM의 핵심 아이디어는 **\*\*메모리 셀(memory cell)\*\***과 게이트(**gate**) 메커니즘이다.<sup>65</sup> 메모리 셀은 정보를 장기간 저장할 수 있는 내부 상태(internal state)를 가지며, 이 상태는 거의 선형적인 방식으로 업데이트되어 정보가 오래 유지될 수 있다 (이를 "constant error carousel"이라고도 함 <sup>66</sup>). 세 종류의 게이트 - 입력 게이트(**input gate**), 망각 게이트(**forget gate**), 출력 게이트(**output gate**) - 는 학습 가능한 신경망 유닛으로, 각각 셀 상태에 새로운 정보를 얼마나 추가할지, 기존 정보를 얼마나 잊을지(망각 게이트는 1999년 Gers 등에 의해 추가됨 <sup>65</sup>), 그리고 셀 상태의 정보를 얼마나 출력으로 내보낼지를 제어한다.<sup>65</sup> 이러한 게이트 구조 덕분에 LSTM은 필요한 정보를 선택적으로 기억하고 불필요한 정보는 잊으면서 장기 의존성을 효과적으로 학습할 수 있게 되었다.

계속되는 도전: 두 번째 **AI** 겨울과 잔존 문제들

역전파, CNN, LSTM과 같은 중요한 발전에도 불구하고, 1990년대와 2000년대 초반까지 신경망 연구는 여전히 여러 난관에 직면해 있었다.<sup>17</sup>

기울기 소실 문제는 특히 네트워크가 깊어지거나 RNN에서 긴 시퀀스를 처리할 때 여전히 심각한 장애물이었다.<sup>9</sup> 또한, 당시의 컴퓨터 성능과 가용 데이터셋의 규모는 복잡하고 큰 모델을 효과적으로 훈련시키는 데 제약이 되었다.<sup>23</sup>

이러한 기술적 한계와 더불어, 1980년대 후반 전문가 시스템 시장의 붕괴<sup>1</sup>는 AI 분야 전반에 대한 투자를 위축시켰다. 신경망 역시 제한적인 성능과 높은 계산 비용 등으로 인해 기대에 미치지 못한다는 인식이 퍼지면서<sup>25</sup>, 연구는 다시 한번 침체기를 맞게 되는데, 이를 두 번째 **AI** 겨울(대략 1980년대 후반부터 1990년대 중반/2000년대 초반까지)이라고 부른다.<sup>24</sup>

그러나 AI 겨울이 완전한 연구 중단을 의미하는 것은 아니었다. 오히려 이 시기는 과장된 기대를 걷어내고 근본적인 문제들을 해결하기 위한 노력이 지속된 기간으로 볼 수 있다. 첫 번째 AI 겨울이 퍼셉트론의 한계(XOR 문제)를 드러내고 MLP와 역전파 연구를 촉발했듯이<sup>25</sup>, 두 번째 AI 겨울 동안 제기된 문제들(기울기 소실, 계산 비용 등<sup>25</sup>)은 LSTM<sup>66</sup>, ReLU 활성화 함수<sup>9</sup>, 더 나은 초기화 기법, 비지도 사전 학습<sup>42</sup> 등 후속 혁신을 위한 자양분이 되었다. 결국 이러한 꾸준한 연구 노력과 외부 환경의 변화(특히 컴퓨팅 파워의 발전<sup>68</sup>)가 결합되면서 딥러닝은 다시 부흥기를 맞이하게 된다.

## VI. 딥러닝 시대 (2006년-현재): 융합과 가속

완벽한 폭풍: 빅 데이터와 병렬 컴퓨팅

2000년대 중반 이후, 딥러닝이 폭발적으로 성장하게 된 배경에는 크게 두 가지 핵심적인



외부 요인이 있었다: 바로 대규모 데이터셋의 등장과 병렬 컴퓨팅 기술의 발전이다.

- **대규모 데이터셋의 역할:** 딥러닝 모델, 특히 깊은 신경망은 성능을 제대로 발휘하기 위해 방대한 양의 학습 데이터를 필요로 한다.<sup>24</sup> 1990년대 중반 이후 인터넷의 폭발적인 확산은 엄청난 양의 디지털 데이터를 생성했고<sup>9</sup>, 이는 딥러닝 모델 학습을 위한 귀중한 자원이 되었다. 특히, **ImageNet** 프로젝트는 딥러닝 발전에 결정적인 기폭제가 되었다.<sup>70</sup> 2006년경 페이페이 리(Fei-Fei Li) 교수가 시작한 이 프로젝트는 WordNet의 계층 구조를 기반으로 수백만 개 이상의 이미지를 수집하고 레이블을 부착하여, 당시까지 존재하지 않았던 규모의 고품질 이미지 데이터셋을 구축했다 (최종적으로 1400만 개 이상의 이미지, 2만 개 이상의 카테고리 포함).<sup>70</sup> ImageNet은 딥러닝 모델, 특히 컴퓨터 비전 분야의 모델을 훈련하고 평가하는 데 필수적인 자원이 되었다.<sup>1</sup>
- **ImageNet Large Scale Visual Recognition Challenge (ILSVRC):** 2010년부터 매년 개최된 ILSVRC는 ImageNet 데이터셋의 일부(일반적으로 1000개 클래스, 약 120만 개의 훈련 이미지<sup>73</sup>)를 사용하여 이미지 분류(classification), 객체 탐지(detection), 객체 위치 추정(localization) 등의 과제에 대한 알고리즘 성능을 겨루는 대회였다.<sup>70</sup> 이 대회는 연구자들 간의 건전한 경쟁을 유도하고 혁신적인 모델 개발을 촉진하는 표준화된 벤치마크 역할을 수행하며 컴퓨터 비전 분야의 발전을 급격히 가속화했다.<sup>70</sup>
- **GPU 기반 병렬 컴퓨팅의 혁신:** 딥러닝 모델의 학습에는 엄청난 양의 행렬 및 벡터 연산이 필요하다. 전통적인 CPU(Central Processing Unit)는 복잡한 순차적 작업에 최적화되어 있어 이러한 대규모 병렬 연산을 효율적으로 처리하기 어려웠다.<sup>68</sup> 반면, **\*\*GPU(Graphics Processing Unit)\*\***는 원래 3D 그래픽 처리를 위해 설계되었지만, 수백 또는 수천 개의 단순한 코어를 가지고 있어 대규모 병렬 연산을 동시에 처리하는 데 매우 효과적이라는 점이 발견되었다.<sup>1</sup> 또한 GPU는 CPU보다 훨씬 높은 메모리 대역폭을 제공하여 대량의 데이터를 빠르게 처리할 수 있었다.<sup>68</sup>
- **CUDA와 가속 라이브러리:** GPU를 범용 병렬 컴퓨팅에 활용할 수 있게 만든 결정적인 기술은 2007년 NVIDIA가 출시한 **CUDA(Compute Unified Device Architecture)** 플랫폼이었다.<sup>69</sup> CUDA는 개발자들이 C/C++과 같은 언어를 사용하여 GPU에서 실행되는 병렬 프로그램을 작성할 수 있게 해주었다.<sup>69</sup> 여기에 더해, NVIDIA는 딥러닝 연산(합성곱, 풀링, 활성화 함수 등)에 최적화된 라이브러리인 **cuDNN(CUDA Deep Neural Network library)**(2014년 출시<sup>69</sup>)을 제공하여 딥러닝 프레임워크(TensorFlow, PyTorch, MXNet 등)들이 GPU 가속 기능을 쉽고 효율적으로 활용할 수 있도록 지원했다.<sup>68</sup> GPU와 CUDA/cuDNN의 조합은 딥러닝 모델의 훈련 시간을 극적으로 단축시켰고(수 주 걸리던 작업을 수 일 또는 수 시간으로 단축<sup>78</sup>), 이전에는 불가능했던 훨씬 더 크고 깊은 모델의 훈련을 현실화했다.<sup>9</sup>

이처럼 대규모 데이터셋의 가용성 증대와 GPU 기반 병렬 컴퓨팅 기술의 발전이라는 두 가지 요소가 시기적절하게 맞물리면서, 딥러닝 알고리즘과 아키텍처의 잠재력이 마침내

폭발적으로 발현될 수 있는 환경이 조성되었다.

## 알고리즘 개선과 훈련 기법

빅데이터와 GPU라는 외부 환경 변화와 더불어, 딥러닝 모델 자체의 학습을 개선하기 위한 다양한 알고리즘 및 기법들도 개발되었다.

- **ReLU 활성화 함수:** 이전의 시그모이드(sigmoid)나 하이퍼볼릭 탄젠트(tanh) 함수는 입력값이 특정 범위를 벗어나면 기울기가 0에 가까워져 깊은 네트워크에서 기울기 소실 문제를 유발하는 경향이 있었다.<sup>32</sup> **ReLU(Rectified Linear Unit)** 함수( $f(x) = \max(0, x)$ )는 양수 입력에 대해서는 기울기가 1로 일정하고 계산이 매우 간단하여<sup>32</sup>, 기울기 소실 문제를 크게 완화하고 학습 속도를 높이는 데 기여했다.<sup>9</sup> 힌튼 등이 ReLU의 효과를 강조하면서<sup>9</sup> 딥러닝 모델의 표준적인 활성화 함수 중 하나로 자리 잡았다. Leaky ReLU, ELU 등 ReLU의 변형들도 제안되었다.<sup>62</sup>
- **가중치 초기화:** 신경망 학습은 초기 가중치 값에 민감한데, 잘못된 초기화는 학습을 불안정하게 만들거나 기울기 소실/폭발 문제를 악화시킬 수 있다. Xavier/Glorot 초기화<sup>38</sup>나 He 초기화와 같은 더 정교한 가중치 초기화 기법들이 개발되어 깊은 네트워크의 안정적인 학습을 도왔다.
- **드롭아웃 (Dropout):** 깊고 파라미터가 많은 신경망은 훈련 데이터에 과적합(overfitting)되기 쉽다. 드롭아웃은 훈련 과정에서 각 뉴런을 일정 확률(예: 0.5)로 무작위로 비활성화(drop out)시키는 기법이다.<sup>11</sup> 이는 마치 매번 다른 구조의 네트워크를 훈련시키는 것과 유사한 효과를 내어, 특정 뉴런에 과도하게 의존하는 것을 방지하고 모델의 일반화 성능을 향상시키는 강력한 정규화(regularization) 기법으로 널리 사용된다.<sup>58</sup>
- **배치 정규화 (Batch Normalization):** 각 층의 입력 분포가 훈련 과정에서 계속 변하는 현상(Internal Covariate Shift)은 학습을 불안정하게 만들 수 있다. 배치 정규화는 미니배치(mini-batch) 단위로 각 층의 입력값을 평균 0, 분산 1로 정규화하고, 추가적인 학습 가능한 파라미터(scale, shift)를 통해 표현력을 유지하는 기법이다.<sup>61</sup> 이는 학습 속도를 높이고, 초기화에 덜 민감하게 만들며, 정규화 효과도 일부 제공하여 딥러닝 모델 훈련에 필수적인 요소 중 하나가 되었다.<sup>83</sup>
- **비지도 사전 학습 (Unsupervised Pre-training):** 충분한 레이블 데이터가 부족하거나 깊은 네트워크의 초기화가 어려웠던 시기에는, 제프리 힌튼 등이 제안한 RBM/DBN 기반의 비지도 사전 학습이 중요한 역할을 했다.<sup>42</sup> 레이블 없는 대량의 데이터를 사용하여 네트워크의 각 층을 탐욕적(greedy) 방식으로 사전 훈련시킨 후, 레이블 데이터를 사용하여 전체 네트워크를 미세 조정(fine-tuning)하는 방식이다.<sup>44</sup> 이는 좋은 초기 가중치를 제공하여 이후 지도 학습의 성능을 높이는 데 도움을 주었으나, 이후 ReLU, 더 나은 초기화 기법, 배치 정규화 등의 발전과 대규모 레이블 데이터셋(ImageNet 등)의 등장으로 그 중요성은 다소 감소했다.

이러한 알고리즘 및 훈련 기법들의 발전은 딥러닝 모델이 더 깊고 복잡해지면서도

안정적으로 학습되고 좋은 성능을 낼 수 있도록 하는 데 필수적이었다.

기념비적 성공 사례와 그 영향

강력해진 컴퓨팅 파워, 풍부해진 데이터, 그리고 개선된 알고리즘이 결합되면서 딥러닝은 다양한 분야에서 획기적인 성공 사례들을 만들어내기 시작했다.

- **AlexNet (2012):** 2012년 ILSVRC 대회에서 알렉스 크리제프스키(Alex Krizhevsky), 일리야 수츠케버(Ilya Sutskever), 제프리 힌튼이 개발한 AlexNet은 딥러닝 역사의 전환점으로 평가받는다.<sup>58</sup> AlexNet은 8개의 층(5개의 합성곱 층, 3개의 완전 연결 층)으로 구성된 깊은 CNN 구조를 가졌으며<sup>81</sup>, ReLU 활성화 함수<sup>58</sup>, 드롭아웃 정규화<sup>58</sup>, 데이터 증강(data augmentation)<sup>80</sup> 등의 기법을 효과적으로 활용했다. 특히, 두 개의 NVIDIA GTX 580 GPU를 사용하여 병렬로 훈련함으로써 당시로서는 매우 큰 모델(약 6천만 개의 파라미터)의 학습을 가능하게 했다.<sup>79</sup> AlexNet은 ILSVRC 2012에서 Top-5 오류율 15.3%를 기록하며, 2위(26.2%)를 큰 차이로 따돌리고 압도적인 우승을 차지했다.<sup>58</sup> 이 결과는 전통적인 컴퓨터 비전 기법의 성능을 훨씬 뛰어넘는 것으로, 딥러닝의 엄청난 잠재력을 전 세계 연구자들에게 각인시키고 현대 AI 붐을 촉발하는 결정적인 계기가 되었다.<sup>58</sup>
- **ResNet (2015):** 네트워크 깊이가 깊어질수록 오히려 성능이 저하되는 문제(degradation problem)를 해결하기 위해 마이크로소프트 리서치 아시아(MSRA)의 허카이밍(Kaiming He) 등이 제안한 잔차 신경망(Residual Network, ResNet)은 또 다른 중요한 혁신이었다.<sup>57</sup> ResNet의 핵심 아이디어는 \*\*잔차 블록(residual block)\*\*과 \*\*스킵 연결(skip connection)\*\*이다.<sup>57</sup> 스킵 연결은 몇 개의 층을 건너뛰어 입력을 직접 출력에 더해주는 구조로, 네트워크가 학습해야 할 목표를 원래의 출력( $H(x)$ ) 대신 입력과의 차이, 즉 잔차(residual,  $F(x) = H(x) - x$ )를 학습하도록 유도한다.<sup>83</sup> 이는 네트워크가 항등 함수(identity mapping, 즉 입력을 그대로 출력)를 배우기 쉽게 만들어, 층을 깊게 쌓더라도 최소한 성능이 저하되지는 않도록 보장하며 기울기 소실 문제 완화에도 도움을 준다.<sup>57</sup> ResNet은 이 구조를 통해 이전보다 훨씬 깊은 네트워크(예: 152개 층)의 학습을 성공적으로 수행했으며<sup>84</sup>, ILSVRC 2015 대회에서 이미지 분류, 객체 탐지, 객체 위치 추정 등 여러 부문에서 1위를 차지하며 딥러닝 모델의 깊이 한계를 크게 확장시켰다.<sup>57</sup> ResNet 아키텍처는 이후 다양한 컴퓨터 비전 모델과 심지어 트랜스포머와 같은 다른 분야 모델에도 널리 영향을 미쳤다.<sup>76</sup>
- **음성 인식의 발전 (Deep Speech 등):** 딥러닝은 자동 음성 인식(Automatic Speech Recognition, ASR) 분야에서도 혁명을 일으켰다.<sup>6</sup> 특히 RNN, LSTM과 같은 시퀀스 모델링 기법이 음성 신호의 시간적 특성을 효과적으로 포착하는 데 활용되었다.<sup>89</sup> 2014년 바이두(Baidu) 연구팀이 발표한 **Deep Speech** 시스템은 대표적인 예시이다.<sup>88</sup> Deep Speech는 RNN(및 이후 CNN) 기반의 종단간(end-to-end) 딥러닝 모델을 사용하여, 기존의 복잡한 음향 모델(acoustic model), 발음 모델(pronunciation model), 언어 모델(language model) 파이프라인 없이도 높은

인식 성능을 달성했다.<sup>90</sup> 특히 잡음이 많은 환경에서도 강인한 성능을 보였으며, 대규모 데이터 합성과 GPU 기반 훈련 시스템이 핵심적인 역할을 했다.<sup>92</sup> 구글 역시 2015년경 CTC(Connectionist Temporal Classification) 손실 함수와 LSTM을 결합하여 음성 인식 성능을 극적으로 향상시켰다고 보고했으며<sup>89</sup>, 이는 구글 어시스턴트 등 상용 서비스에 적용되었다.<sup>6</sup> 이러한 딥러닝 기반 ASR 시스템들은 기존의 은닉 마르코프 모델(Hidden Markov Model, HMM) 기반 시스템들을 빠르게 대체하며<sup>88</sup> Siri, Alexa, Cortana 등 오늘날 우리가 사용하는 음성 비서 기술의 기반이 되었다.<sup>6</sup>

- **AlphaGo (2016):** 구글 딥마인드(DeepMind)가 개발한 AlphaGo는 인공지능 역사상 가장 상징적인 사건 중 하나를 만들었다.<sup>96</sup> 2016년, AlphaGo는 세계 최정상급 프로 바둑 기사 이세돌 9단을 5번의 대국에서 4승 1패로 꺾었다.<sup>96</sup> 바둑은 경우의 수가 우주 전체의 원자 수보다 많다고 할 정도로 복잡성이 높고<sup>96</sup>, 직관과 창의성이 중요하게 여겨져 오랫동안 인공지능에게 난공불락의 영역으로 간주되어왔다.<sup>96</sup> AlphaGo의 성공 비결은 딥러닝과 강화학습(Reinforcement Learning), 그리고 몬테 카를로 트리 탐색(Monte Carlo Tree Search, MCTS)의 결합에 있었다.<sup>96</sup> \*\*정책망(policy network)\*\*과 \*\*가치망(value network)\*\*이라는 두 개의 깊은 신경망을 사용하여, 정책망은 다음 수의 후보를 제안하고 가치망은 현재 국면의 승률을 예측했다.<sup>102</sup> 이 신경망들은 방대한 양의 인간 기보 데이터를 이용한 지도 학습(supervised learning)과 AlphaGo 스스로 수많은 대국을 두며 학습하는 강화 학습(self-play reinforcement learning)을 통해 훈련되었다.<sup>96</sup> MCTS는 이 신경망들의 예측을 바탕으로 효율적으로 가능한 수들을 탐색하여 최종 결정을 내렸다.<sup>100</sup> 특히 2국에서 보여준 '37수'는 인간의 정석을 벗어난 창의적인 수로 평가받으며 AI가 인간의 지능을 모방하는 것을 넘어 새로운 전략을 창조할 수 있음을 시사했다.<sup>96</sup> AlphaGo의 승리는 딥러닝과 강화학습의 강력한 시너지를 입증했으며, 이후 인간의 데이터 없이 오직 스스로 학습하여 이전 버전들을 압도한 AlphaGo Zero<sup>99</sup>와 다른 게임(체스, 쇼기)까지 정복한 AlphaZero<sup>100</sup>로 이어지며 범용 인공지능 연구에 큰 영감을 주었다.
- **트랜스포머와 BERT (2017-2018):** 자연어 처리(NLP) 분야에서는 2017년 구글 브레인 팀이 발표한 논문 "Attention Is All You Need"에서 제안된 트랜스포머(Transformer) 아키텍처가 혁신을 가져왔다.<sup>106</sup> 기존의 RNN이나 LSTM은 시퀀스 데이터를 순차적으로 처리해야 했기 때문에 병렬 처리가 어렵고 장기 의존성 학습에 여전히 한계가 있었다.<sup>106</sup> 트랜스포머는 이러한 순환 구조를 완전히 배제하고, 셀프 어텐션(self-attention) 메커니즘만을 사용하여 입력 시퀀스 내의 단어들 간의 관계(의존성)를 거리에 상관없이 직접적으로 파악하고 중요도를 계산한다.<sup>106</sup> 이를 통해 병렬 처리가 가능해져 학습 속도가 크게 향상되었고, 장기 의존성 문제도 효과적으로 해결했다.<sup>106</sup> 2018년, 구글 연구팀은 트랜스포머의 인코더(encoder) 구조만을 활용하여 BERT(Bidirectional Encoder Representations from Transformers)라는 언어 모델을 개발했다.<sup>8</sup> BERT의 핵심적인 특징은 이름 그대로 \*\*양방향성(bidirectionality)\*\*이다. 이전 모델들이 텍스트를 왼쪽에서 오른쪽 또는

오른쪽에서 왼쪽으로 단방향으로만 처리했던 것과 달리, BERT는 **\*\*마스크 언어 모델(Masked Language Model, MLM)\*\***이라는 독창적인 사전 훈련(pre-training) 방식을 통해 문장 내 모든 단어의 왼쪽과 오른쪽 문맥을 동시에 고려하여 깊은 문맥적 의미를 학습한다.<sup>107</sup> (MLM은 입력 문장의 일부 단어를 무작위로 마스킹하고, 주변 문맥을 이용하여 원래 단어를 예측하도록 학습하는 방식이다.) 또한 다음 문장 예측(**Next Sentence Prediction, NSP**) 태스크를 통해 문장 간의 관계도 학습한다.<sup>112</sup> BERT는 대규모 텍스트 코퍼스(위키피디아, BooksCorpus 등<sup>8</sup>)로 사전 훈련된 후, 특정 NLP 태스크(질의응답, 감성 분석, 개체명 인식 등)에 맞게 소량의 레이블 데이터로 미세 조정(fine-tuning)하는 방식을 통해<sup>8</sup>, 당시 존재하던 11개 이상의 주요 NLP 벤치마크(GLUE, SQuAD 등)에서 최고 성능(State-Of-The-Art, SOTA)을 달성하며 NLP 분야에 혁명을 일으켰다.<sup>8</sup> BERT의 성공은 사전 훈련-미세 조정 패러다임을 NLP의 표준으로 만들었으며, 구글 검색<sup>107</sup> 등 다양한 실제 서비스에 적용되었고, RoBERTa<sup>109</sup>, ALBERT<sup>113</sup>, DistilBERT<sup>8</sup> 등 수많은 후속 모델 개발의 기반이 되었다.

#### 광범위한 적용 가능성과 현재 동향

딥러닝 기술은 앞서 언급된 분야 외에도 광범위한 영역에서 성공적으로 적용되고 있다. 컴퓨터 비전 분야에서는 이미지 분류<sup>71</sup>, 객체 탐지<sup>71</sup>, 이미지 분할(segmentation)<sup>83</sup> 등에서 핵심 기술로 사용된다. 자연어 처리 분야에서는 기계 번역<sup>106</sup>, 텍스트 요약<sup>8</sup>, 챗봇 및 대화형 AI<sup>8</sup> 등 다양한 응용을 가능하게 했다. 음성 인식<sup>6</sup> 기술은 스마트폰, 스마트 스피커 등 일상 기기와의 상호작용 방식을 바꾸고 있다. 강화학습과 결합된 딥러닝(Deep Reinforcement Learning)은 게임 플레이(Atari 게임<sup>98</sup>, Go<sup>96</sup>, Starcraft 등)에서 초인적인 성능을 보여주었을 뿐만 아니라 로봇 제어 등 현실 문제 해결에도 활용되고 있다. 이 외에도 신약 개발<sup>3</sup>, 단백질 구조 예측(AlphaFold<sup>96</sup>), 의료 영상 분석<sup>24</sup>, 금융 시장 예측<sup>1</sup>, 자율 주행 자동차<sup>24</sup> 등 과학, 산업, 사회 전반에 걸쳐 딥러닝의 영향력이 확대되고 있다.<sup>3</sup>

현재 딥러닝 분야는 더욱 강력한 성능을 위해 모델의 크기를 키우는 대규모 모델(**Large-scale Models**), 특히 GPT 시리즈와 같은 대규모 언어 모델(**Large Language Models, LLM**)<sup>3</sup> 개발 경쟁이 치열하게 벌어지고 있다. 또한, 이미지, 텍스트, 음성 등 여러 종류의 데이터를 함께 처리하는 멀티모달(**Multimodal**) 학습, 모델의 훈련 및 추론 효율성을 높이는 연구, 그리고 모델의 편향성(bias), 투명성(transparency), 공정성(fairness) 등 윤리적 문제에 대한 고민과 책임감 있는 AI 개발<sup>4</sup>에 대한 관심도 높아지고 있다.

ILSVRC나 GLUE와 같은 벤치마크 대회와 표준 데이터셋은 딥러닝 기술 발전을 측정하고 가속화하는 데 중요한 역할을 수행했다.<sup>71</sup> AlexNet, ResNet, BERT와 같은 모델들은 이러한 경쟁 환경 속에서 탄생하고 검증받으며 그 성능을 입증했다. 그러나 특정 벤치마크에서의 높은 점수가 반드시 실제 세계에서의 강인함이나 범용적인 지능을 의미하지는 않는다는 비판도 제기된다.<sup>70</sup> ImageNet 데이터셋 자체의 편향성 문제나 특정



벤치마크에 과도하게 최적화되는 경향 등은 연구자들이 해결해야 할 과제로 남아있다.<sup>70</sup> 이러한 문제 인식은 딥러닝 연구가 단순히 벤치마크 점수를 높이는 것을 넘어, 보다 신뢰할 수 있고 일반화 가능한, 그리고 사회적으로 책임감 있는 방향으로 나아가야 함을 시사한다.

딥러닝의 역사를 관통하는 또 다른 중요한 축은 하드웨어와 소프트웨어(알고리즘) 간의 상호작용이다. 다층 퍼셉트론이나 깊은 신경망과 같은 아이디어는 종종 이를 효과적으로 구현할 수 있는 하드웨어보다 먼저 등장했다.<sup>23</sup> 역으로, GPU와 CUDA와 같은 강력하고 프로그래밍 가능한 하드웨어의 등장은<sup>68</sup> 기존 알고리즘의 잠재력을 폭발시키고(예: AlexNet의 성공<sup>79</sup>), 새로운 규모의 모델(예: 대규모 트랜스포머<sup>8</sup>) 개발을 가능하게 했다. 하드웨어의 발전 없이는 알고리즘의 혁신이 실현되기 어려웠고, 알고리즘의 요구는 다시 하드웨어 발전을 견인하는, 이 둘의 공진화(co-evolution)가 딥러닝 발전의 핵심 동력이었음을 알 수 있다.

## VII. 결론

딥러닝의 역사는 뇌를 모방하려는 초기 이론적 시도에서 시작하여, 여러 차례의 기대와 실망(AI 겨울)의 순환을 거쳐 오늘날 인공지능의 핵심 기술로 자리 잡기까지 길고 역동적인 여정을 거쳐왔다. 맥클락-피츠의 뉴런 모델, 로젠블랫의 퍼셉트론, 위드로의 아달라인 등 초기 연구는 가능성의 씨앗을 뿌렸지만, 이론적 한계와 기술적 제약으로 인해 첫 번째 AI 겨울을 맞았다.

1980년대 역전파 알고리즘의 등장은 다층 퍼셉트론 학습의 길을 열어 연결주의의 부활을 이끌었으나, 기울기 소실 문제와 컴퓨팅 성능의 한계는 여전히 깊은 네트워크의 실현을 가로막으며 두 번째 AI 겨울로 이어졌다. 이 침체기 동안에도 힌튼, 르쿤, 벤지오와 같은 선구자들의 꾸준한 연구는 CNN, LSTM 등 혁신적인 아키텍처와 알고리즘 개선의 토대를 마련했다.

2000년대 중반 이후, ImageNet과 같은 대규모 데이터셋의 등장과 GPU 기반 병렬 컴퓨팅 기술의 발전이라는 결정적인 외부 환경 변화는 딥러닝의 잠재력을 폭발시키는 기폭제가 되었다. ReLU, 드롭아웃, 배치 정규화 등 개선된 학습 기법과 결합된 딥러닝 모델들은 AlexNet의 ILSVRC 제패를 시작으로 ResNet, Deep Speech, AlphaGo, BERT 등 다양한 분야에서 인간의 능력을 뛰어넘거나 이전에는 불가능했던 문제들을 해결하며 그 위력을 입증했다.

딥러닝의 역사는 생물학적 영감에서 출발했지만<sup>9</sup>, 결국 수학적 원리, 알고리즘 혁신, 데이터의 힘, 그리고 컴퓨팅 기술 발전의 복합적인 상호작용을 통해 발전해왔음을 보여준다. AI 겨울이라는 침체기는 단순히 연구가 중단된 시기가 아니라, 이전 시대의 한계를 극복하기 위한 근본적인 문제 해결 노력이 이루어진 성찰과 재정비의 시간이었다. 또한, ILSVRC와 같은 벤치마크는 경쟁을 통해 발전을 가속화했지만, 동시에 연구 방향에 영향을 미치는 양면성을 지녔다. 하드웨어와 소프트웨어의 긴밀한 공진화 역시 딥러닝



발전의 중요한 축이었다.

현재 딥러닝은 대규모 모델, 멀티모달 학습, 효율성 증대, 그리고 윤리적 책임감이라는 새로운 과제와 기회에 직면해 있다.<sup>3</sup> 지난 수십 년간의 역동적인 발전을 고려할 때, 딥러닝은 앞으로도 인공지능 기술의 핵심 동력으로서 과학, 산업, 그리고 우리 사회 전반에 걸쳐 지속적으로 중요한 영향을 미칠 것으로 전망된다.

## VIII. 참고문헌

1

### 참고 자료

1. [칼럼] 인공지능 연구의 거울에서 얻는 지능적 수행에 대한 통찰 - THE AI, 4월 13, 2025에 액세스, <https://www.newstheai.com/news/articleView.html?idxno=3829>
2. 1. 딥러닝 소개 — Deep Learning with Python(2판) - 코딩알지, 4월 13, 2025에 액세스, [https://codingalzi.github.io/dlp2/what\\_is\\_deep\\_learning.html](https://codingalzi.github.io/dlp2/what_is_deep_learning.html)
3. 딥러닝의 역사 알아보기, 4월 13, 2025에 액세스, <https://basicdl.tistory.com/entry/%EB%94%A5%EB%9F%AC%EB%8B%9D%EC%9D%98-%EA%B0%84%EB%8B%A8%ED%95%9C-%EC%97%AD%EC%82%AC>
4. AI 거울: 과대 기대와 실망이 만들어낸 혁신의 주기, 4월 13, 2025에 액세스, <https://contentstailor.com/entry/AI-%EA%B2%A8%EC%9A%B8-%EA%B3%BC%EB%8C%80-%EA%B8%B0%EB%8C%80%EC%99%80-%EC%8B%A4%EB%A7%9D%EC%9D%B4-%EB%A7%8C%EB%93%A4%EC%96%B4%EB%82%B8-%ED%98%81%EC%8B%A0%EC%9D%98-%EC%A3%BC%EA%B8%B0>
5. 인공지능(AI)분야의 4대천왕, 4월 13, 2025에 액세스, <https://neo-platform.tistory.com/59>
6. 딥 러닝 - 위키백과, 우리 모두의 백과사전, 4월 13, 2025에 액세스, [https://ko.wikipedia.org/wiki/%EB%94%A5\\_%EB%9F%AC%EB%8B%9D](https://ko.wikipedia.org/wiki/%EB%94%A5_%EB%9F%AC%EB%8B%9D)
7. Chapter.1 Deep Learning 개요, 4월 13, 2025에 액세스, <https://velog.io/@fagaram112/Deep-Learning>
8. BERT 101 - State Of The Art NLP Model Explained - Hugging Face, 4월 13, 2025에 액세스, <https://huggingface.co/blog/bert-101>
9. [연구리딩] 인공지능의 역사 및 기술 동향(2018) - velog, 4월 13, 2025에 액세스, <https://velog.io/@sjinu/%EC%97%B0%EA%B5%AC%EB%A6%AC%EB%94%A9-%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5%EC%9D%98-%EC%97%AD%EC%82%AC-%EB%B0%8F-%EA%B8%B0%EC%88%A0-%EB%8F%99%ED%96%A52018>
10. [2] 딥러닝의 역사적 동향 <1> : 네이버 블로그, 4월 13, 2025에 액세스, <https://blog.naver.com/lcj1725/221816928959?viewType=pc>
11. 딥러닝의 역사, 4월 13, 2025에 액세스, [https://jinseob2kim.github.io/deep\\_learning.html](https://jinseob2kim.github.io/deep_learning.html)
12. [인공지능 만화] 8. 신경세포의 초기모델 ADALINE - 위데이터랩, 4월 13, 2025에 액세스, <https://wedatalab.tistory.com/64>
13. 신경망 의 퍼셉트론 (1) : 네이버 블로그, 4월 13, 2025에 액세스,

- <http://blog.naver.com/msnayana/80186743513>
14. Andrey Kurenkov의 신경망과 딥러닝의 간략한 역사 : 네이버 블로그, 4월 13, 2025에 액세스, <https://blog.naver.com/dlcksgod1/220992607547?viewType=pc>
  15. 인공지능 역사 연대표 상세 정리 - 브런치스토리, 4월 13, 2025에 액세스, <https://brunch.co.kr/@mentats1/672>
  16. [지능형 시스템] Chapter 3. 초기 퍼셉트론, 아달린(Adaline), 최소평균제곱법(LMS) - velog, 4월 13, 2025에 액세스, <https://velog.io/@diduya/%EC%A7%80%EB%8A%A5%ED%98%95-%EC%8B%9C%EC%8A%A4%ED%85%9C-Chapter-3.-%EC%B4%88%EA%B8%B0-%ED%8D%BC%EC%85%89%ED%8A%B8%EB%A1%A0%EA%B3%BC-%EC%B5%9C%EC%86%8C%ED%8F%89%EA%B7%A0%EC%A0%9C%EA%B3%B1%EB%B2%95LMS>
  17. 퍼셉트론부터 CNN까지, 딥러닝의 역사 - 카이스트신문, 4월 13, 2025에 액세스, <http://times.kaist.ac.kr/news/articleView.html?idxno=4675>
  18. 인공신경망 초기 모델, ADALINE(아달라인) 이해, 손실함수(Loss Function) MSE (Mean Squared Error) - ITDA - 티스토리, 4월 13, 2025에 액세스, <https://ju-blog.tistory.com/59>
  19. 신경세포의 초기 모델, ADALINE - 브런치스토리, 4월 13, 2025에 액세스, <https://brunch.co.kr/@1wbE/64>
  20. [거의 모든 AI의 역사] ④ AI 부족 간 전쟁의 시작, 4월 13, 2025에 액세스, <http://www.firenzedt.com/news/articleView.html?idxno=30785>
  21. 인공지능 역사 연대표 상세 정리 - 브런치스토리, 4월 13, 2025에 액세스, <https://brunch.co.kr/@2fbJ/672>
  22. 왜 지금인가? - 딥러닝의 역사, 4월 13, 2025에 액세스, <https://skyl.tistory.com/6>
  23. 인공지능의 역사 : 네이버 블로그 - Naver Blog, 4월 13, 2025에 액세스, [https://blog.naver.com/keit\\_newtech/220978388608?viewType=pc](https://blog.naver.com/keit_newtech/220978388608?viewType=pc)
  24. AI의 과거, 현재, 미래: 혁신과 윤리를 아우르는 인공지능 기술의 여정 - Goover AI, 4월 13, 2025에 액세스, <https://seo.goover.ai/report/202504/go-public-report-ko-7f2d920e-e922-487e-90ff-9973aa876875-0-0.html>
  25. [디지털 치료] 인공지능의 겨울 vs. 디지털치료제의 겨울 : 네이버 블로그, 4월 13, 2025에 액세스, <https://blog.naver.com/digitaltherapeutics1/223078286664>
  26. Dicklesworthstone/the\_lighthill\_debate\_on\_ai: A Full Transcript of the Lighthill Debate on AI from 1973, with Introductory Remarks - GitHub, 4월 13, 2025에 액세스, [https://github.com/Dicklesworthstone/the\\_lighthill\\_debate\\_on\\_ai](https://github.com/Dicklesworthstone/the_lighthill_debate_on_ai)
  27. GitHub - Dicklesworthstone/the\_lighthill\_debate\_on\_ai: A Full Transcript of the Lighthill Debate on AI from 1973, with Introductory Remarks - BestofAI, 4월 13, 2025에 액세스, [https://bestofai.com/article/github-dicklesworthstonethe\\_lighthill\\_debate\\_on\\_ai-a-full-transcript-of-the-lighthill-debate-on-ai-from-1973-with-introductory-remarks](https://bestofai.com/article/github-dicklesworthstonethe_lighthill_debate_on_ai-a-full-transcript-of-the-lighthill-debate-on-ai-from-1973-with-introductory-remarks)
  28. Lighthill Report - Artificial Intelligence & Complex Event Processing, 4월 13, 2025에 액세스, <https://complexevents.com/lighthill-report-artificial-intelligence/>
  29. Lighthill report - Wikipedia, 4월 13, 2025에 액세스, [https://en.wikipedia.org/wiki/Lighthill\\_report](https://en.wikipedia.org/wiki/Lighthill_report)
  30. [AI 이야기] 인공지능의 결정적 순간들, 두 번째 순서 - 레터웍스, 4월 13, 2025에

- 엑세스, <https://www.lettr.ai/ko/blog/story-20211105-1>
31. 02화 AI 겨울을 이겨낸 노장 과학자들 - 브런치스토리, 4월 13, 2025에 액세스, <https://brunch.co.kr/@10ca/209>
  32. 인공 신경망의 역사 - 개발일지 - 티스토리, 4월 13, 2025에 액세스, <https://devjournal.tistory.com/131>
  33. CNN과 딥러닝을 통한 인공 신경망 발전 역사 : 네이버 블로그, 4월 13, 2025에 액세스, <https://blog.naver.com/kckoh2309/223739750271>
  34. <7> 딥러닝 역사의 전환점들 - 한국경제, 4월 13, 2025에 액세스, <https://www.hankyung.com/article/202202179087i>
  35. Backpropagation - Wikipedia, 4월 13, 2025에 액세스, <https://en.wikipedia.org/wiki/Backpropagation>
  36. Paul Werbos - Wikipedia, 4월 13, 2025에 액세스, [https://en.wikipedia.org/wiki/Paul\\_Werbos](https://en.wikipedia.org/wiki/Paul_Werbos)
  37. 2022 International Conference on Dr. Paul Werbos\_ICCSI2021 - AGIST, 4월 13, 2025에 액세스, <https://agist.org/iccsi2022/Keynote3.html>
  38. 요슈아 벤지오 - 나무위키, 4월 13, 2025에 액세스, <https://namu.wiki/w/%EC%9A%94%EC%8A%88%EC%95%84%20%EB%B2%A4%EC%A7%80%EC%98%A4>
  39. 제프리 힌튼, 요슈아 벤지오, 얀 르쿤과 챗GPT 원리 - 데이터 과학, 4월 13, 2025에 액세스, <https://tsyoon.tistory.com/148>
  40. [거의 모든 AI의 역사] ⑥ '캐나다 학파'와 AI 3대 천왕, 4월 13, 2025에 액세스, <http://www.firenzedt.com/news/articleView.html?idxno=30856>
  41. [거의 모든 AI의 역사] #9. 합성곱 신경망의 아버지, 얀 ... - 피렌체의 식탁, 4월 13, 2025에 액세스, <http://www.firenzedt.com/news/articleView.html?idxno=30903>
  42. 2024 노벨 물리학상 수상자 제프리 힌튼과 1세대 연구자들 - 메일리, 4월 13, 2025에 액세스, <https://maily.so/seanlee/posts/w6ovjl3zk5>
  43. #1. 3대 천왕 이야기 - 어쩌다 인공지능, 4월 13, 2025에 액세스, <https://audreyprincess.tistory.com/52>
  44. [All Around AI 1편] AI의 시작과 발전 과정, 미래 전망 - SK하이닉스 뉴스룸, 4월 13, 2025에 액세스, <https://news.skhyunix.co.kr/post/all-around-ai-1>
  45. 인공지능은 어떻게 발달해왔는가 인공지능의 역사 - Samsung SDS, 4월 13, 2025에 액세스, [https://www.samsungsds.com/global/ko/support/insights/091517\\_CX\\_CVP3.html](https://www.samsungsds.com/global/ko/support/insights/091517_CX_CVP3.html)
  46. [Deep Learning] 딥러닝의 역사 - Data Science - 티스토리, 4월 13, 2025에 액세스, <https://lebi.tistory.com/18>
  47. 3-2. 딥러닝의 역사 - TCP School, 4월 13, 2025에 액세스, [https://www.tcpschool.com/deeplearning/deep\\_history](https://www.tcpschool.com/deeplearning/deep_history)
  48. 얀 르쿤 - 나무위키, 4월 13, 2025에 액세스, <https://namu.wiki/w/%EC%96%80%20%EB%A5%B4%EC%BF%A4>
  49. LeNet Architecture: A Complete Guide - Kaggle, 4월 13, 2025에 액세스, <https://www.kaggle.com/code/blurredmachine/lenet-architecture-a-complete-guide>
  50. LeNet-5: A Simple Yet Powerful CNN for Image Classification - Paravision Lab, 4월 13, 2025에 액세스, <https://paravisionlab.co.in/lenet-5-architecture/>
  51. Convolutional neural network - Wikipedia, 4월 13, 2025에 액세스,

- [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)
52. Convolutional Neural Networks, Explained - Towards Data Science, 4월 13, 2025에 액세스,  
<https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939/>
  53. What are Convolutional Neural Networks? - IBM, 4월 13, 2025에 액세스,  
<https://www.ibm.com/think/topics/convolutional-neural-networks>
  54. Convolutional Neural Networks (CNN) Overview - Encord, 4월 13, 2025에 액세스,  
<https://encord.com/blog/convolutional-neural-networks-explained/>
  55. A Comprehensive Exploration of Pooling in Neural Networks: A Python demo to Pooling in CNN - Paperspace Blog, 4월 13, 2025에 액세스,  
<https://blog.paperspace.com/a-comprehensive-exploration-of-pooling-in-neural-networks/>
  56. Basic CNN Architecture: A Detailed Explanation of the 5 Layers in Convolutional Neural Networks - upGrad, 4월 13, 2025에 액세스,  
<https://www.upgrad.com/blog/basic-cnn-architecture/>
  57. Introduction to Resnet or Residual Network - Great Learning, 4월 13, 2025에 액세스,  
<https://www.mygreatlearning.com/blog/resnet/>
  58. AlexNet: A Revolutionary Deep Learning Architecture - viso.ai, 4월 13, 2025에 액세스,  
<https://viso.ai/deep-learning/alexnet/>
  59. LeNet - Wikipedia, 4월 13, 2025에 액세스,  
<https://en.wikipedia.org/wiki/LeNet>
  60. Recurrent Neural Networks (RNN) - The Vanishing Gradient Problem - Blogs - SuperDataScience | Machine Learning | AI | Data Science Career | Analytics | Success, 4월 13, 2025에 액세스,  
<https://www.superdatascience.com/blogs/recurrent-neural-networks-rnn-the-vanishing-gradient-problem>
  61. Vanishing gradient problem - Wikipedia, 4월 13, 2025에 액세스,  
[https://en.wikipedia.org/wiki/Vanishing\\_gradient\\_problem](https://en.wikipedia.org/wiki/Vanishing_gradient_problem)
  62. Vanishing Gradient Problem : Everything you need to know - Engati, 4월 13, 2025에 액세스,  
<https://www.engati.com/glossary/vanishing-gradient-problem>
  63. The Vanishing Gradient Problem in Recurrent Neural Networks | Nick McCullum, 4월 13, 2025에 액세스,  
<https://www.nickmccullum.com/python-deep-learning/vanishing-gradient-problem/>
  64. Long Short-Term Memory - Deep Learning, CMU, 4월 13, 2025에 액세스,  
<https://deeplearning.cs.cmu.edu/S23/document/readings/LSTM.pdf>
  65. Long short-term memory - Wikipedia, 4월 13, 2025에 액세스,  
[https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory)
  66. LONG SHORT-TERM MEMORY 1 INTRODUCTION, 4월 13, 2025에 액세스,  
<https://www.bioinf.jku.at/publications/older/2604.pdf>
  67. 10.1. Long Short-Term Memory (LSTM) - Dive into Deep Learning, 4월 13, 2025에 액세스,  
[https://d2l.ai/chapter\\_recurrent-modern/lstm.html](https://d2l.ai/chapter_recurrent-modern/lstm.html)
  68. GPU Acceleration in AI: How Graphics Processing Units Drive Deep Learning - Gcore, 4월 13, 2025에 액세스,  
<https://gcore.com/blog/deep-learning-gpu>
  69. Accelerating Deep Learning using GPUs - IRJET, 4월 13, 2025에 액세스,

- <https://www.irjet.net/archives/V8/i3/IRJET-V8I3318.pdf>
70. ImageNet Definition | DeepAI, 4월 13, 2025에 액세스, <https://deepai.org/machine-learning-glossary-and-terms/imagenet>
  71. ImageNet Dataset: Evolution & Applications (2025) - viso.ai, 4월 13, 2025에 액세스, <https://viso.ai/deep-learning/imagenet/>
  72. Imagenet - (History of Science) - Vocab, Definition, Explanations | Fiveable, 4월 13, 2025에 액세스, <https://library.fiveable.me/key-terms/history-science/imagenet>
  73. ImageNet - Wikipedia, 4월 13, 2025에 액세스, <https://en.wikipedia.org/wiki/ImageNet>
  74. ImageNet: A Pioneering Vision for Computers - History of Data Science, 4월 13, 2025에 액세스, <https://www.historyofdatascience.com/imagenet-a-pioneering-vision-for-computers/>
  75. ImageNet Dataset: Key Features, Limitations, and How to Get Started - Kolena, 4월 13, 2025에 액세스, <https://www.kolena.com/guides/imagenet-dataset-key-features-limitations-and-how-to-get-started/>
  76. ImageNet: VGGNet, ResNet, Inception, and Xception with Keras - PyImageSearch, 4월 13, 2025에 액세스, <https://pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>
  77. A Comprehensive Guide to NVIDIA's AI Stack for Deep Learning Projects - OpenCV, 4월 13, 2025에 액세스, <https://opencv.org/blog/nvidia-ai-deep-learning-projects/>
  78. Accelerating AI with GPUs: A New Computing Model - NVIDIA Blog, 4월 13, 2025에 액세스, <https://blogs.nvidia.com/blog/accelerating-ai-artificial-intelligence-gpus/>
  79. AlexNet - Wikipedia, 4월 13, 2025에 액세스, <https://en.wikipedia.org/wiki/AlexNet>
  80. [NIPS 2012] AlexNet: Review and Implementation - Deep Learning Viet Nam, 4월 13, 2025에 액세스, <https://deeplearning.vn/post/alexnet/>
  81. What were the major innovations introduced by AlexNet in 2012 that significantly advanced the field of convolutional neural networks and image recognition? - EITCA Academy, 4월 13, 2025에 액세스, <https://eitca.org/artificial-intelligence/eitc-ai-adl-advanced-deep-learning/advanced-computer-vision/convolutional-neural-networks-for-image-recognition/examination-review-convolutional-neural-networks-for-image-recognition/what-were-the-major-innovations-introduced-by-alexnet-in-2012-that-significantly-advanced-the-field-of-convolutional-neural-networks-and-image-recognition/>
  82. A Review of Popular Deep Learning Architectures: AlexNet, VGG16, and GoogleNet, 4월 13, 2025에 액세스, <https://www.digitalocean.com/community/tutorials/popular-deep-learning-architectures-alexnet-vgg-googlenet>
  83. Detailed Guide to Understand and Implement ResNets - CV-Tricks.com, 4월 13, 2025에 액세스, <https://cv-tricks.com/keras/understand-implement-resnets/>
  84. Residual neural network - Wikipedia, 4월 13, 2025에 액세스,



- [https://en.wikipedia.org/wiki/Residual\\_neural\\_network](https://en.wikipedia.org/wiki/Residual_neural_network)
85. Deep Residual Networks (ResNet, ResNet50) 2024 Guide - viso.ai, 4월 13, 2025에 액세스, <https://viso.ai/deep-learning/resnet-residual-neural-network/>
  86. Deep Residual Learning for Image Recognition - The Computer Vision Foundation, 4월 13, 2025에 액세스, [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf)
  87. Evolution and Challenges in Speech Recognition Technology: From Early Systems to Deep Learning Innovations | Applied and Computational Engineering, 4월 13, 2025에 액세스, <https://www.ewadirect.com/proceedings/ace/article/view/19493>
  88. The Evolution of Voice Recognition Technologies | - iPlum, 4월 13, 2025에 액세스, <https://www.iplum.com/blog/the-evolution-of-voice-recognition-technologies>
  89. Speech recognition - Wikipedia, 4월 13, 2025에 액세스, [https://en.wikipedia.org/wiki/Speech\\_recognition](https://en.wikipedia.org/wiki/Speech_recognition)
  90. What is Automatic Speech Recognition? A Comprehensive Overview of ASR Technology, 4월 13, 2025에 액세스, <https://www.assemblyai.com/blog/what-is-asr>
  91. The History of Speech Recognition to the Year 2030 - Awni Hannun, 4월 13, 2025에 액세스, <https://awni.github.io/future-speech/>
  92. [1412.5567] Deep Speech: Scaling up end-to-end speech recognition - arXiv, 4월 13, 2025에 액세스, <https://arxiv.org/abs/1412.5567>
  93. A Historical Perspective of Speech Recognition - Communications of the ACM, 4월 13, 2025에 액세스, <https://cacm.acm.org/research/a-historical-perspective-of-speech-recognition/>
  94. (PDF) A Historical Perspective of Speech Recognition - ResearchGate, 4월 13, 2025에 액세스, [https://www.researchgate.net/publication/262157198\\_A\\_Historical\\_Perspective\\_of\\_Speech\\_Recognition](https://www.researchgate.net/publication/262157198_A_Historical_Perspective_of_Speech_Recognition)
  95. The Past, Present, and Future of Speech-to-Text and AI Transcription | iMerit, 4월 13, 2025에 액세스, <https://imerit.net/blog/the-past-present-and-future-of-speech-to-text-and-ai-transcription-all-una/>
  96. AlphaGo - Google DeepMind, 4월 13, 2025에 액세스, <https://deepmind.google/research/breakthroughs/alphago/>
  97. AlphaGo: AI Conquers Go, Redefining Human Intelligence - Open MedScience, 4월 13, 2025에 액세스, <https://openmedscience.com/alphago-ai-conquers-go-redefining-human-intelligence/>
  98. Google DeepMind - Wikipedia, 4월 13, 2025에 액세스, [https://en.wikipedia.org/wiki/Google\\_DeepMind](https://en.wikipedia.org/wiki/Google_DeepMind)
  99. AlphaGo by DeepMind: The AI that Mastered Go - Perplexity, 4월 13, 2025에 액세스, <https://www.perplexity.ai/page/alphago-by-deepmind-the-ai-tha-zpNCBbdIQ4eAcjGHwvq4dA>
  100. AlphaGo - Wikipedia, 4월 13, 2025에 액세스, <https://en.wikipedia.org/wiki/AlphaGo>



101. The story of AlphaGo - Google Arts & Culture, 4월 13, 2025에 액세스,  
<https://artsandculture.google.com/story/the-story-of-alphago-barbican-centre/kQXBk0X1qEe5KA?hl=en>
102. AI Behind AlphaGo: Machine Learning and Neural Network, 4월 13, 2025에 액세스,  
<https://illumin.usc.edu/ai-behind-alphago-machine-learning-and-neural-network/>
103. Move 37: Artificial Intelligence, Randomness, and Creativity - John Menick, 4월 13, 2025에 액세스,  
<https://www.johnmenick.com/writing/move-37-alpha-go-deep-mind.html>
104. AlphaGo and AI Progress - Future of Life Institute, 4월 13, 2025에 액세스,  
<https://futureoflife.org/recent-news/alphago-and-ai-progress/>
105. AlphaGo Zero: Starting from scratch - Google DeepMind, 4월 13, 2025에 액세스,  
<https://deepmind.google/discover/blog/alphago-zero-starting-from-scratch/>
106. Transformer Models in Natural Language Processing - Netguru, 4월 13, 2025에 액세스,  
<https://www.netguru.com/blog/transformer-models-in-nlp>
107. Understanding BERT: The Model That Revolutionized NLP | by Zahra Waleed - Medium, 4월 13, 2025에 액세스,  
<https://medium.com/@ms.scales1480/understanding-bert-the-model-that-revolutionized-nlp-ec6ba6bde43f>
108. BERT Vs GPT-4: Evolution Of Transformer Model - Xonique, 4월 13, 2025에 액세스,  
<https://xonique.dev/blog/the-evolution-of-transformer-models-in-nlp/>
109. The Definitive Guide to BERT Models | deepset Blog, 4월 13, 2025에 액세스,  
<https://www.deepset.ai/blog/the-definitive-guide-to-bertmodels>
110. 10 Things You Need to Know About BERT and the Transformer Architecture That Are Reshaping the AI Landscape - Neptune.ai, 4월 13, 2025에 액세스,  
<https://neptune.ai/blog/bert-and-the-transformer-architecture>
111. BERT: The Start of Modern NLP - Multimodal.dev, 4월 13, 2025에 액세스,  
<https://www.multimodal.dev/post/bert-the-start-of-modern-nlp>
112. RoBERTa vs. BERT: Exploring the Evolution of Transformer Models - DS Stream, 4월 13, 2025에 액세스,  
<https://www.dsstream.com/post/roberta-vs-bert-exploring-the-evolution-of-transformer-models>
113. BERT (language model) - Wikipedia, 4월 13, 2025에 액세스,  
[https://en.wikipedia.org/wiki/BERT\\_\(language\\_model\)](https://en.wikipedia.org/wiki/BERT_(language_model))
114. Feedback From Automatic Speech Recognition to Elicit Clear Speech in Healthy Speakers, 4월 13, 2025에 액세스,  
<https://pmc.ncbi.nlm.nih.gov/articles/PMC10721250/>
115. CUDA Libraries Expand Accelerated Computing Into New Science, Industrial Applications, 4월 13, 2025에 액세스,  
<https://blogs.nvidia.com/blog/cuda-accelerated-computing-energy-efficiency/>