

## 4.2 딥러닝 구조

### 4.2.1 딥러닝 용어

- **뉴런 (Neuron)**: 하나의 입력을 받아 계산 후 출력하는 기본 단위. 각각은 가중치(weight)를 가지고 입력을 곱한 뒤, 비선형 함수를 거쳐 출력값을 낸다.
- **레이어 (Layer)**: 뉴런들이 모여 있는 층. 입력층(Input layer), 은닉층(Hidden layer), 출력층(Output layer)으로 구성됨.
- **가중치 (Weight)**: 입력값에 곱해지는 값으로, 학습을 통해 조정되는 핵심 요소.
- **바이어스 (Bias)**: 출력값을 조정해주는 상수항. 가중치와 함께 모델의 유연성을 향상시킴.
- **활성화 함수 (Activation Function)**: 뉴런의 출력을 비선형으로 변환하는 함수. 대표적으로 ReLU, Sigmoid, Tanh 등이 있음.



## 1. Sigmoid

- 수식:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- 출력 범위: (0, 1)
  - 장점: 확률 해석이 가능
  - 단점: 기울기 소실(vanishing gradient), 학습 느림
- 

## 2. Tanh (Hyperbolic Tangent)

- 수식:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- 출력 범위: (-1, 1)
  - 장점: 중심이 0이라 수렴 빠름
  - 단점: 여전히 기울기 소실 발생
- 

## 3. ReLU (Rectified Linear Unit)

- 수식:

$$f(x) = \max(0, x)$$

- 출력 범위: [0, ∞)
  - 장점: 연산 간단, 기울기 소실 방지
  - 단점: 음수 입력 시 죽은 뉴런(dead neuron) 문제
- 

## 4. Leaky ReLU / PReLU

- Leaky ReLU:

$$f(x) = \max(0.01x, x)$$

- PReLU: 음수 기울기를 학습 가능하게 만들
  - 장점: 죽은 뉴런 문제 완화
- 

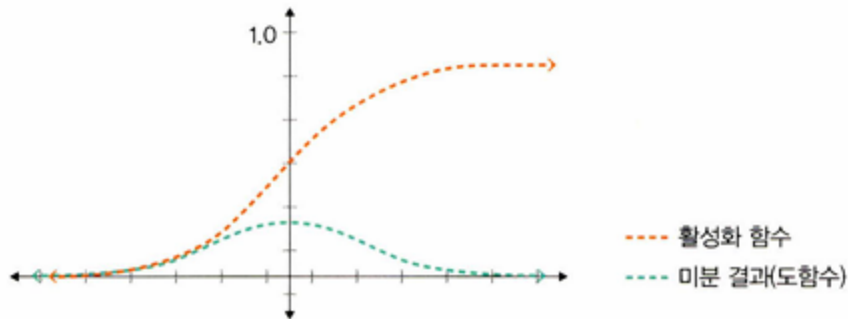
## 5. Softmax

- 용도: 출력층 (다중 클래스 부호)

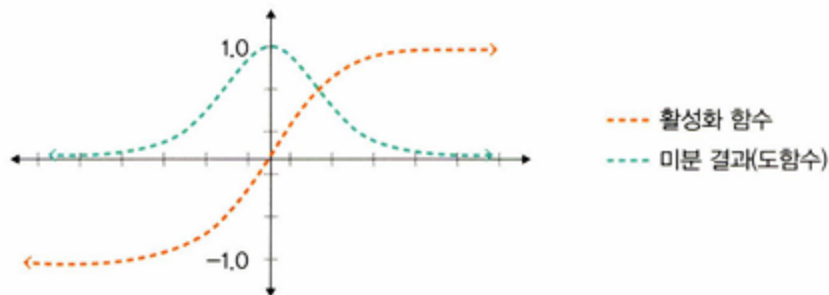
- 역할: 입력값을 확률 분포로 변환

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

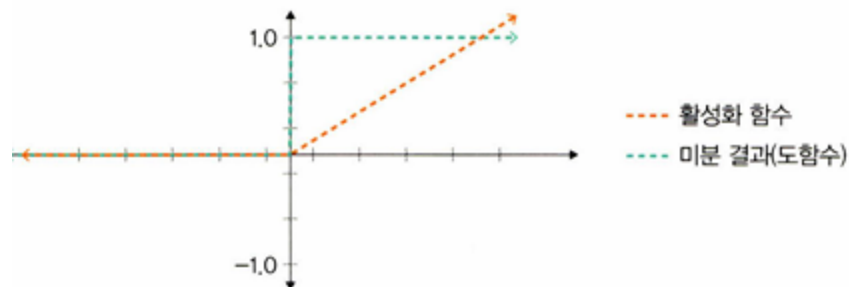
- 
- Sigmoid



Tanh



ReLU



## 4.2.2 딥러닝 학습

- **순전파(Forward Propagation):** 입력 데이터를 받아 각 레이어를 거쳐 출력을 계산함. 각 레이어는 입력에 가중치를 곱하고 활성화 함수를 적용.
- **오차 계산:** 출력값과 실제 값 사이의 차이를 손실 함수로 계산.
- **역전파(Backpropagation):** 오차를 출력층부터 입력층 방향으로 전달하며 각 가중치에 대한 기울기를 계산.
- **파라미터 업데이트:** 기울기를 기반으로 최적화 알고리즘(예: Adam, SGD 등)을 통해 가중치와 바이어스를 조정함.

이 과정을 수많은 반복(epoch) 동안 수행하며 모델이 점점 정답을 잘 맞추게 됨.

### 4.2.3 딥러닝의 문제점과 해결 방안

#### 1. 과적합(Overfitting)

- 문제: 학습 데이터에 너무 잘 맞아서 새로운 데이터에 성능이 떨어짐
- 해결: Dropout, 정규화(L2/L1), 조기 종료(EarlyStopping), 데이터 증강

#### 2. 기울기 소실/폭주 문제

- 문제: 역전파 시 기울기가 0에 가까워지거나 너무 커져서 학습이 잘 되지 않음
- 해결: ReLU 계열 함수 사용, Batch Normalization, 적절한 초기화

#### 3. 학습 시간 오래 걸림

- 문제: 깊은 네트워크일수록 학습 시간이 오래 걸림
- 해결: GPU 사용, 미니배치 학습, 최적화 기법 개선(Adam 등)

#### 4. 데이터 부족

- 문제: 대량의 데이터가 필요함
- 해결: 전이학습(Transfer Learning), 데이터 증강(Data Augmentation)

### 4.2.4 딥러닝을 사용할 때의 이점

- 비선형 관계 학습 가능: 복잡한 데이터 구조를 자동으로 학습 가능
- 특징 추출 자동화: 사람이 설계하지 않아도 유용한 특징을 스스로 학습
- 범용성: 이미지, 음성, 텍스트 등 다양한 도메인에서 사용 가능
- 대규모 데이터에 최적화: 많은 양의 데이터일수록 성능 향상이 큼
- 최신 성능 확보: 최신 연구와 프레임워크 덕분에 빠르게 발전

## 4.3 딥러닝 알고리즘

딥러닝에서 사용되는 다양한 모델들은 특정한 데이터 특성과 문제 유형에 맞게 발전되어 왔습니다. 이 절에서는 딥러닝을 구성하는 주요 알고리즘들과 학습 방법론을 다루며, 각각의 구조적 특성과 학습 절차에 대해 구체적으로 살펴봅니다.

---

## 학습 방식

딥러닝 모델은 크게 아래와 같은 네 단계로 학습을 수행합니다:

#### 1. 순전파 (Forward Propagation)

- 입력 데이터를 받아 각 층을 통과시키며 출력값을 계산합니다.
- 각 층에서는 입력값에 가중치를 곱하고 바이어스를 더한 후, 활성화 함수를 통해 출력값을 생성합니다.

- 마지막 출력층에서는 예측값(예: 확률, 회귀값 등)을 산출합니다.

## 2. 오차 계산 (Loss Calculation)

- 모델의 출력값과 실제 정답값(Label) 간의 차이를 손실 함수(Loss Function)를 통해 계산합니다.
- 이 손실 값은 모델의 예측 정확도를 수치화한 것으로, 작을수록 좋은 예측임을 의미합니다.

## 3. 역전파 (Backpropagation)

- 출력층에서 계산된 오차를 기반으로 각 층의 가중치에 대해 미분(기울기)을 계산합니다.
- 체인 룰(Chain Rule)을 사용하여 오차가 입력층까지 거꾸로 전파됩니다.
- 이 과정을 통해 어떤 가중치가 얼마나 잘못되었는지 파악할 수 있습니다.

## 4. 최적화 (Optimizer)

- 계산된 기울기를 바탕으로 가중치와 바이어스를 조정합니다.
- 최적화 알고리즘은 학습률(learning rate)을 포함한 다양한 전략을 사용하여 손실 값을 최소화하는 방향으로 파라미터를 업데이트합니다.

---

## 손실 함수 (Loss Function)

손실 함수는 문제 유형에 따라 선택됩니다:

- **회귀 문제**
  - **MSE (Mean Squared Error):** 평균 제곱 오차. 예측값과 실제값 차이를 제곱해 평균. 큰 오차에 더 민감.
  - **MAE (Mean Absolute Error):** 평균 절대 오차. 제곱 대신 절댓값을 사용해 이상치 영향이 적음.
- **분류 문제**
  - **Binary Cross Entropy:** 이진 분류에서 사용. 0과 1 사이의 확률 출력을 학습할 때 사용됨.
  - **Categorical Cross Entropy:** 다중 클래스 분류에서 사용. softmax 출력과 함께 사용되어 확률 분포 간 차이를 줄임.

---

## 최적화 알고리즘 (Optimizer)

- **SGD (Stochastic Gradient Descent):** 가장 기본적인 방식으로, 한 샘플씩 또는 미니배치 단위로 기울기를 계산해 업데이트.
- **Momentum:** 이전 기울기의 이동 방향을 고려해 관성을 부여함. 진동을 줄이고 수렴 속도 향상.

- **RMSProp**: 학습률을 각 파라미터마다 개별적으로 조정해주는 방식. 자주 변화하는 파라미터는 학습률을 낮추고, 덜 변화하는 파라미터는 높입니다.
  - **Adam**: Momentum과 RMSProp을 결합한 방식으로, 현재 가장 널리 사용되는 최적화 알고리즘입니다.
    - 학습률 자동 조정, 빠른 수렴, 대부분의 딥러닝 문제에 안정적으로 적용 가능.
- 

## 정규화 기법 (Regularization Techniques)

모델이 학습 데이터에만 과하게 적합하지 않도록 다양한 정규화 방법이 사용됩니다:

- **Dropout**
  - 학습 중 무작위로 일부 뉴런을 비활성화(0으로 설정)하여 신경망의 특정 뉴런에 의존하는 것을 방지.
  - 과적합을 줄이고 일반화 성능을 높입니다.
- **Batch Normalization**
  - 각 층의 출력값을 평균 0, 분산 1로 정규화하여 학습 안정화.
  - 기울기 소실 방지, 학습 속도 향상, 초기 가중치 설정에 덜 민감.
- **EarlyStopping**
  - 검증 데이터의 성능이 일정 에폭 이상 향상되지 않으면 학습을 조기 종료.
  - 과적합 방지 및 불필요한 학습 시간 단축에 효과적.
- **Weight Decay (L2 정규화)**
  - 가중치 값이 너무 커지는 것을 방지하기 위해 손실 함수에 가중치의 제곱합을 추가하여 모델 복잡도 억제.

딥러닝에서 사용되는 다양한 모델들은 특정한 데이터 특성과 문제 유형에 맞게 발전되어 왔습니다. 이 절에서는 딥러닝을 구성하는 주요 알고리즘들을 다루며, 각각의 구조적 특성과 용도에 대해 구체적으로 살펴봅니다.

---

### 4.3.1 심층 신경망 (Deep Neural Network, DNN)

#### 개념 및 구조

- 심층 신경망은 다층 퍼셉트론(MLP, Multi-Layer Perceptron)을 기반으로 한 인공 신경망 구조로, 은닉층이 2개 이상 존재합니다.
- 입력층 → 여러 개의 은닉층 → 출력층 순으로 구성되며, 각 층은 fully connected 구조로 구성됩니다.

- 은닉층이 많아질수록 더 복잡한 데이터 표현이 가능하며, 이를 통해 높은 수준의 추상적 개념을 학습할 수 있습니다.

### 활용 분야

- 전통적인 구조화된 데이터 분류 및 회귀
- 간단한 이미지/텍스트/신호 분석 문제

### 장점

- 다양한 문제 유형에 폭넓게 적용 가능
- 구조가 단순하여 구현과 실험이 용이

### 한계

- 이미지나 시계열 데이터와 같은 특수 구조에 최적화되지 않음
- 깊어질수록 학습이 어려워지며, 기울기 소실 문제 발생 가능

### 보완 기법

- ReLU 계열 활성화 함수 도입으로 기울기 소실 문제 완화
  - Dropout, Batch Normalization 등으로 과적합 및 학습 안정성 보완
- 

## 4.3.2 합성곱 신경망 (Convolutional Neural Network, CNN)

### 개념

- CNN은 주로 이미지 처리에 사용되며, 입력 데이터의 공간적 구조(위치, 패턴)를 고려하여 특징을 추출하는 신경망입니다.

### 핵심 구성 요소

1. **합성곱 계층(Convolutional Layer):** 필터(커널)를 통해 입력 이미지에서 지역적 특징(엣지, 텍스처 등)을 추출합니다.
2. **활성화 함수(ReLU 등):** 비선형성 부여
3. **풀링 계층(Pooling Layer):** 공간 크기를 축소하여 연산량과 과적합을 줄입니다 (예: MaxPooling).
4. **완전연결층(Fully Connected Layer):** 추출된 특징을 기반으로 최종 분류 수행

### 활용 분야

- 이미지 분류 (ImageNet, MNIST 등)



- 객체 탐지, 얼굴 인식, 자율주행, 의료 영상 분석 등

## 장점

- 이미지에 특화된 구조로 학습 효율성 뛰어남
- 파라미터 수가 적어 과적합에 강함
- 다양한 구조 개선 모델 존재: LeNet, AlexNet, VGG, ResNet 등

## 한계

- 시계열 데이터나 자연어 처리에는 적합하지 않음 (해결: RNN, Transformer)
- 

### 4.3.4 제한된 볼츠만 머신 (Restricted Boltzmann Machine, RBM)

#### 개념

- RBM은 확률적 비지도 학습 모델로, 입력 데이터의 분포를 모델링하기 위해 사용됩니다.
- 2계층 구조: 가시층(입력층) + 은닉층
- 같은 층 내에서는 뉴런 간 연결이 없고, 서로 다른 층 간에만 연결이 존재함 (Restricted 구조)

#### 학습 방법

- 에너지 기반 모델로서 상태의 에너지 값을 최소화하는 방향으로 학습
- Contrastive Divergence (CD)라는 근사 방법으로 파라미터 업데이트

#### 용도

- 차원 축소, 특징 추출, 데이터 생성

#### 한계 및 현재 위상

- 학습 속도가 느리고 불안정함
  - 현재는 AutoEncoder, GAN 등으로 대체됨
  - 그러나 딥러닝 역사상 중요한 모델로, 이후 DBN, Deep Autoencoder 등에 영향을 줌
- 

### 4.3.5 심층 신뢰 신경망 (Deep Belief Network, DBN)

#### 개념

- DBN은 여러 개의 RBM을 층층이 쌓은 구조로 구성되며, 각 층은 비지도 학습을 통해 특징을 학습합니다.

## 학습 절차

1. 각 RBM을 순차적으로 학습 (layer-wise pretraining)
2. 전체 네트워크를 소규모 지도 학습 데이터로 fine-tuning

## 장점

- 초기 학습 파라미터를 안정적으로 설정할 수 있어 학습 효율 증가
- 복잡한 구조에서도 학습이 가능했던 딥러닝 초창기 주요 모델

## 활용 분야

- 이미지/문자 인식 초기 모델로 사용됨 (예: MNIST)

## 현황

- 최근에는 더 나은 성능과 효율성을 가진 CNN, RNN, Transformer 등이 등장하며 실무 활용은 줄었지만, 딥러닝 기초 모델로 학습에 중요