



# 9장 신경망에서 딥러닝으로



# 목 차



- 
- 1 기울기 소실 문제와 활성화 함수
  - 2 속도와 정확도 문제를 해결하는 고급 경사 하강법



# 신경망에서 딥러닝으로

- 다층 퍼셉트론이 오차 역전파를 만나 신경망이 되었고, 신경망은 XOR문제를 가볍게 해결함
- 이제 신경망을 차곡차곡 쌓아올리면 마치 사람처럼 생각하고 판단하는 인공지능이 금방이라도 완성될 것처럼 보임

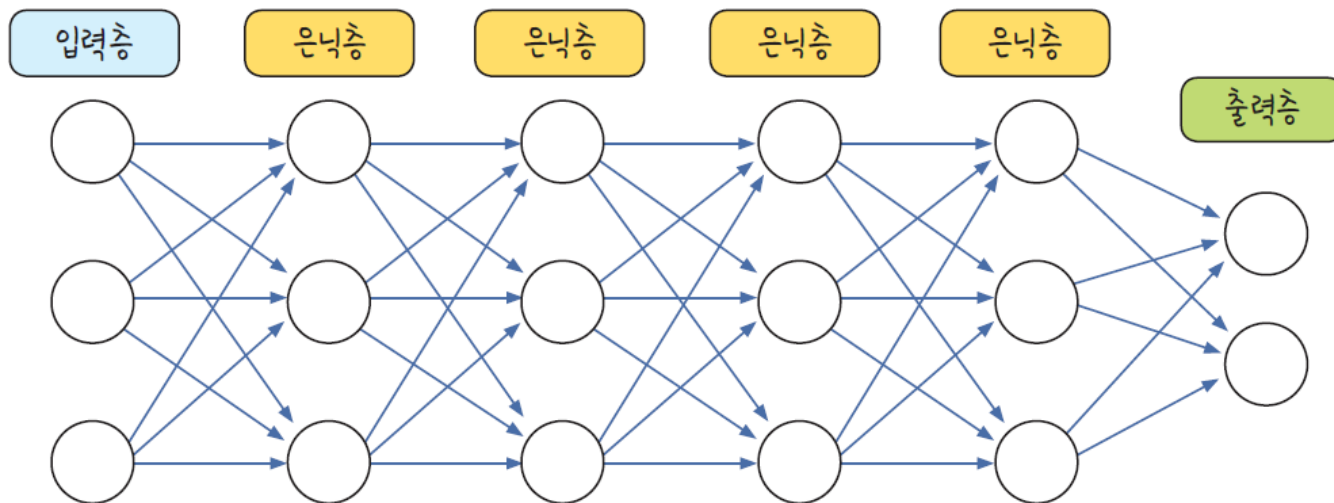


그림 9-1 다층 확장

# ○○○ 1 기울기 소실 문제와 활성화 함수 ○○○

- 오차 역전파 :

출력층으로부터 하나씩 앞으로 되돌아가며 각 층의 가중치를 수정하는 방법

- 가중치를 수정하려면 미분 값, 즉 기울기가 필요하다고 배움

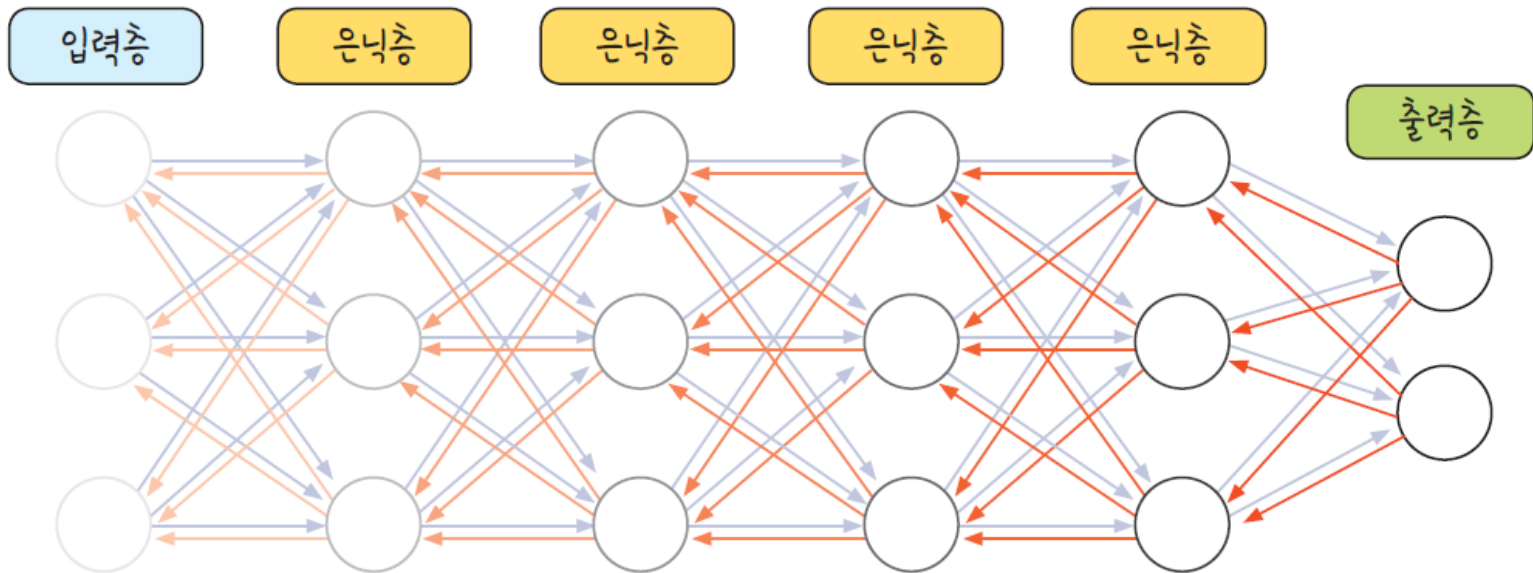


그림 9-2 기울기 소실 문제 발생!

# 1 기울기 소실 문제와 활성화 함수

- 기울기 소실(vanishing gradient) 문제가 발생하기 시작한 것은 활성화 함수로 사용된 시그모이드 함수의 특성 때문임
- 여러 층을 거칠수록 기울기가 사라져 가중치를 수정하기가 어려워지는 것임

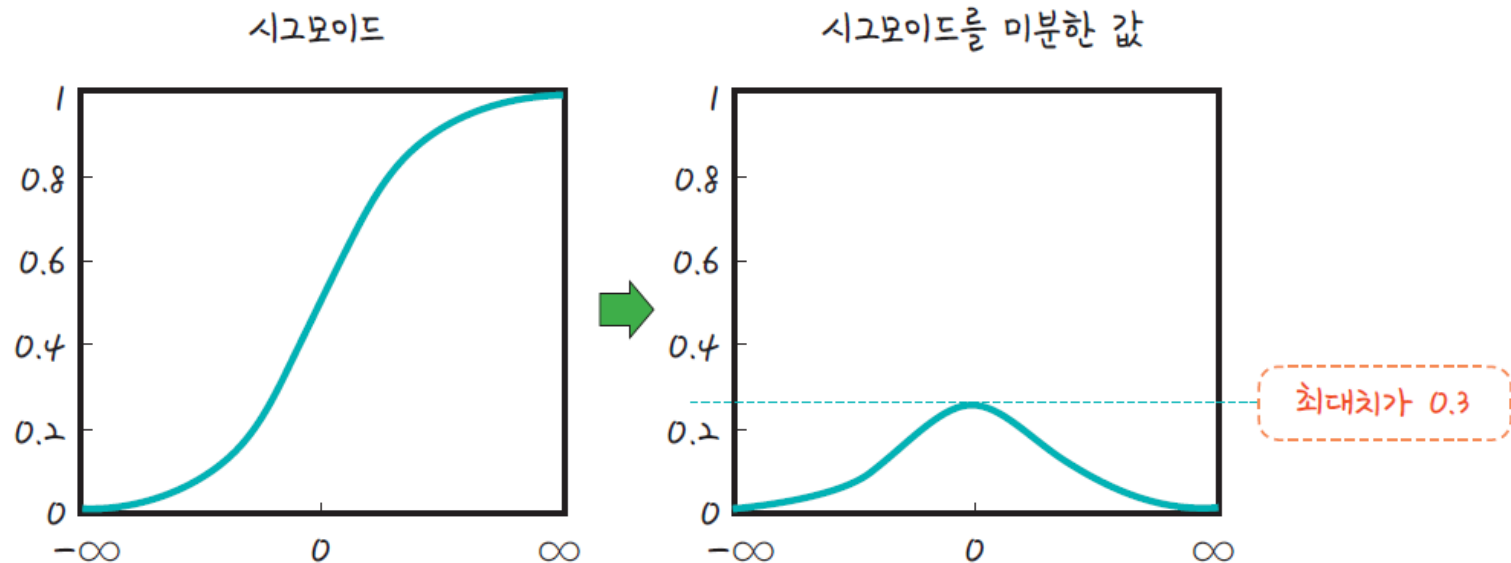


그림 9-3 시그모이드의 미분

# 1 기울기 소실 문제와 활성화 함수

- 이를 해결하고자 활성화 함수를 시그모이드가 아닌 여러 함수로 대체하기 시작함

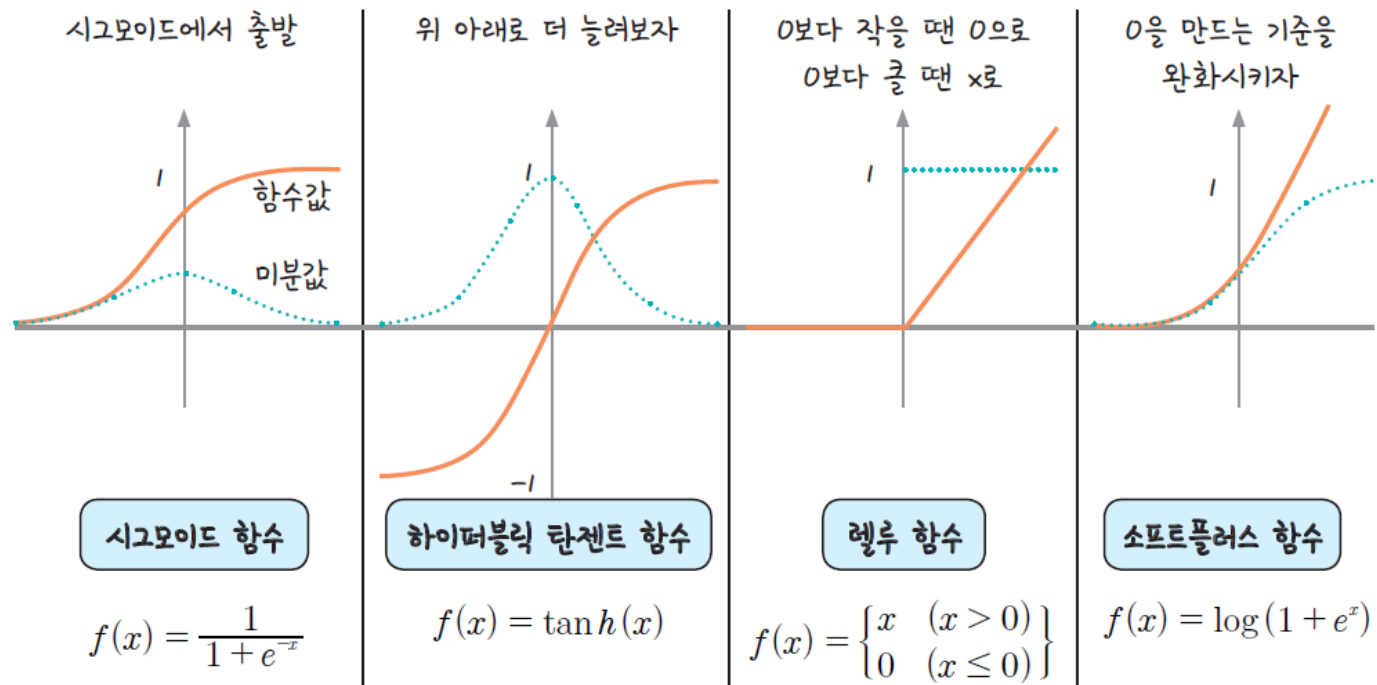


그림 9-4 여러 활성화 함수의 도입



# 1 기울기 소실 문제와 활성화 함수



- 하이퍼볼릭 탄젠트(tanh) 함수
  - 미분한 값의 범위가 함께 확장되는 효과를 가져옴
  - 다양한 여전히 1보다 작은 값이 존재하므로 기울기 소실 문제는 사라지지 않음
- 렐루(ReLU) 함수
  - 시그모이드 함수의 대안으로 떠오르며 현재 가장 많이 사용되는 활성화 함수임
  - 여러 은닉층을 거치며 곱해지더라도 맨 처음 층까지 사라지지 않고 남아있을 수 있음
  - 이 간단한 방법이 여러 층을 쌓을 수 있게 했고, 이로써 딥러닝의 발전에 속도가 붙게 됨



# 1 기울기 소실 문제와 활성화 함수



- 소프트플러스 (softplus) 함수
  - 이후 렐루의 0이 되는 순간을 완화





## 2 속도와 정확도 문제를 해결하는 고급 경사 하강법

- 가중치를 업데이트하는 방법으로 우리는 경사 하강법을 배웠음
- 경사 하강법은 정확하게 가중치를 찾아가지만, 한 번 업데이트할 때마다 전체 데이터를 미분해야 하므로 계산량이 매우 많다는 단점이 있음
- 이러한 점을 보완한 고급 경사 하강법이 등장하면서 딥러닝의 발전 속도는 더 빨라짐

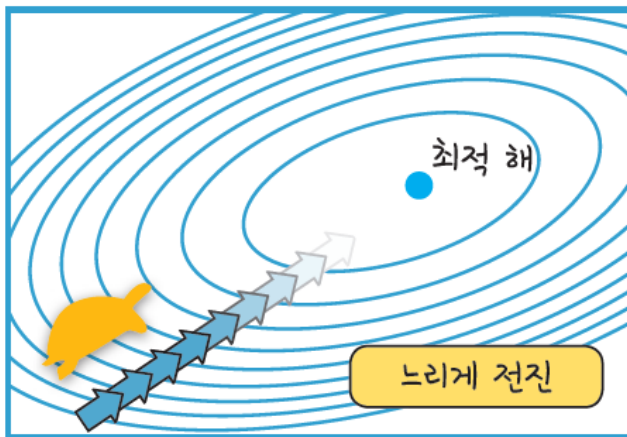
## 2 속도와 정확도 문제를 해결하는 고급 경사 하강법

### 확률적 경사 하강법

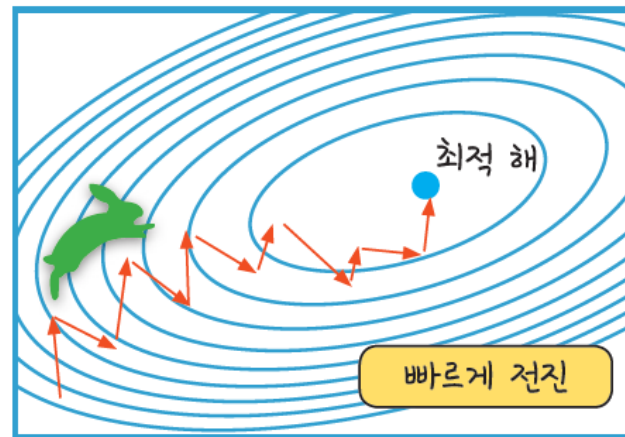
- 경사 하강법은 불필요하게 많은 계산량은 속도를 느리게 할 뿐 아니라, 최적 해를 찾기 전에 최적화 과정이 멈출 수도 있음
- 경사 하강법의 이러한 단점을 보완한 방법
- 전체 데이터를 사용하는 것이 아니라, 랜덤하게 추출한 일부 데이터를 사용함
- 일부 데이터를 사용하므로 더 빨리 그리고 자주 업데이트를 하는 것이 가능해짐

## 2 속도와 정확도 문제를 해결하는 고급 경사 하강법

- 랜덤한 일부 데이터를 사용하는 만큼 확률적 경사 하강법은 중간 결과의 진폭이 크고 불안정해 보일 수도 있음
- 속도가 확연히 빠르면서도 최적 해에 근사한 값을 찾아낸다는 장점 덕분에 경사 하강법의 대안으로 사용되고 있음



경사 하강법



확률적 경사 하강법

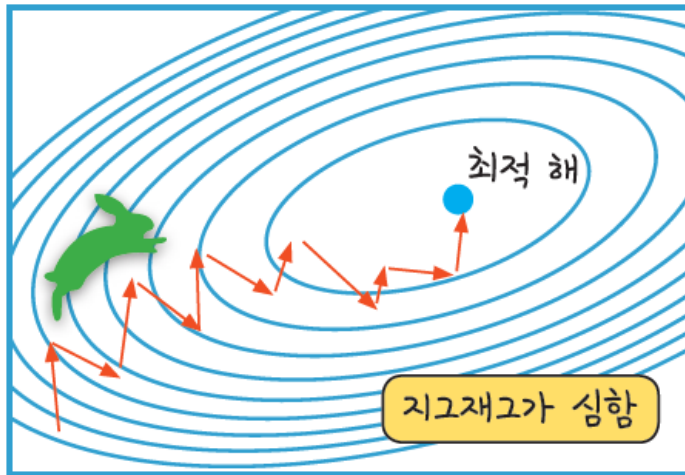
그림 9-5 경사 하강법과 확률적 경사 하강법의 비교

## 2 속도와 정확도 문제를 해결하는 고급 경사 하강법

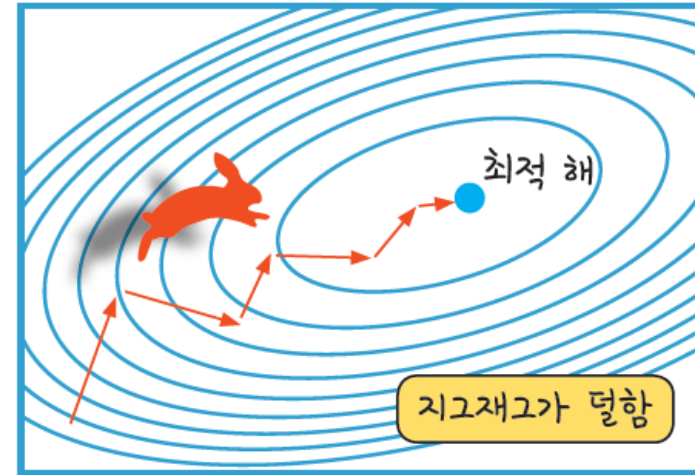
### 모멘텀

- 모멘텀(momentum)이란 단어는 '관성, 탄력, 가속도'라는 뜻
- 모멘텀 SGD란 말 그대로 경사 하강법에 탄력을 더해 주는 것
- 다시 말해서, 경사 하강법과 마찬가지로 매번 기울기를 구하지만, 이를 통해 오차를 수정하기 전 바로 앞 수정 값과 방향(+, -)을 참고하여 같은 방향으로 일정한 비율만 수정되게 하는 방법
- 수정 방향이 양수(+) 방향으로 한 번, 음수(-) 방향으로한 번 지그재그로 일어나는 현상이 줄어들고, 이전 이동 값을 고려하여 일정 비율만큼만 효과를 낼 수 있음

## 2 속도와 정확도 문제를 해결하는 고급 경사 하강법



확률적 경사 하강법



모멘텀을 적용한 확률적 경사 하강법

그림 9-6 모멘텀을 적용했을 때

## 2 속도와 정확도 문제를 해결하는 고급 경사 하강법

| 고급 경사 하강법           | 개요  | 효과          | 케라스 사용법   |
|---------------------|---|-------------|---|
| 확률적 경사 하강법<br>(SGD) | 랜덤하게 추출한 일부 데이터를 사용해<br>더 빨리, 자주 업데이트를 하게 하는 것            | 속도 개선       | <code>keras.optimizers.SGD(lr = 0.1)</code><br>케라스 최적화 함수를 이용합니다.   |
| 모멘텀<br>(Momentum)   | 관성의 방향을 고려해 진동과 폭을 줄이<br>는 효과                             | 정확도<br>개선   | <code>keras.optimizers.SGD(lr = 0.1,<br/>momentum = 0.9)</code><br>모멘텀 계수를 추가합니다.   |
| 네스테로프 모멘텀<br>(NAG)  | 모멘텀이 이동시킬 방향으로 미리 이동해<br>서 그래디언트를 계산. 불필요한 이동<br>을 줄이는 효과 | 정확도<br>개선   | <code>keras.optimizers.SGD(lr = 0.1,<br/>momentum = 0.9, nesterov = True)</code><br>네스테로프 옵션을 추가합니다.  |
| 아다그라드<br>(Adagrad)  | 변수의 업데이트가 잦으면 학습률을 적게<br>하여 이동 보폭을 조절하는 방법                | 보폭 크기<br>개선 | <code>keras.optimizers.Adagrad(lr = 0.01,<br/>epsilon = 1e - 6)</code><br>아다그라드 함수를 사용합니다.<br><br>※ 참고: 여기서 epsilon, rho, decay 같은<br>파라미터는 바꾸지 않고 그대로 사용하기를<br>권장하고 있습니다. 따라서 lr, 즉 learning<br>rate(학습률) 값만 적절히 조절하면 됩니다. |

## 2 속도와 정확도 문제를 해결하는 고급 경사 하강법

| 고급 경사 하강법           | 개요                    | 효과                  | 케라스 사용법   |
|---------------------|-----------------------|---------------------|---|
| 알엠에스프롭<br>(RMSProp) | 아다그라드의 보폭 민감도를 보완한 방법 | 보폭 크기<br>개선         | <code>keras.optimizers.RMSprop(lr = 0.001, rho = 0.9, epsilon = 1e - 08, decay = 0.0)</code><br>알엠에스프롭 함수를 사용합니다.             |
| 아담(Adam)            | 모멘텀과 알엠에스프롭 방법을 합친 방법 | 정확도와<br>보폭 크기<br>개선 | <code>keras.optimizers.Adam(lr = 0.001, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e - 08, decay = 0.0)</code><br>아담 함수를 사용합니다. |

표 9-1 딥러닝 구동에 사용되는 고급 경사 하강법 개요 및 활용법