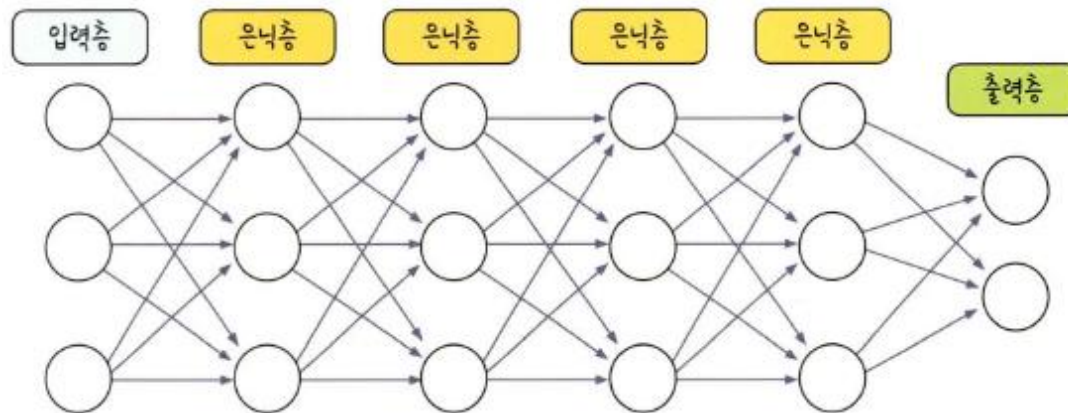


## 10. 신경망에서 딥러닝으로

# 서론

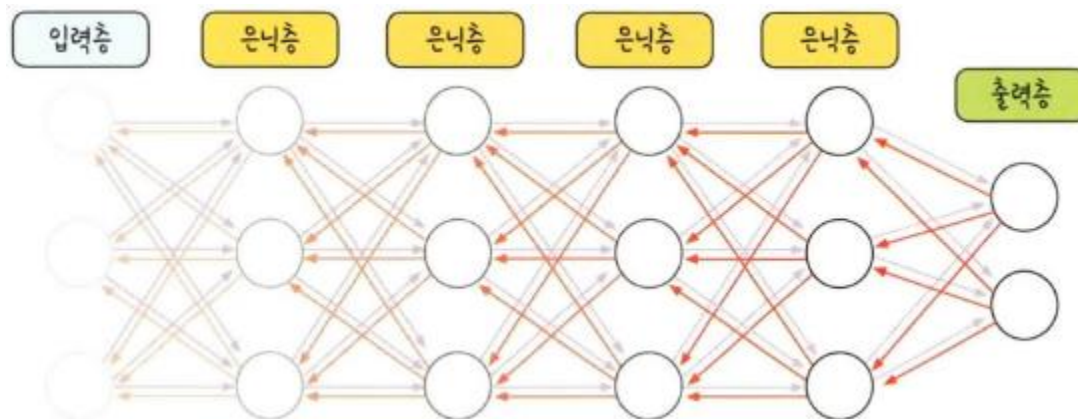
- ▶ 다층 퍼셉트론이 오차 역전파를 만나 신경망이 되었고, 신경망은 XOR 문제를 가볍게 해결했습니다.
- ▶ 따라서 이제 신경망을 차곡차곡 쌓아올리면 마치 사람처럼 생각하고 판단하는 인공지능이 금방이라도 완성될 것처럼 보였습니다.



- ▶ 하지만 기대만큼 결과가 좋아지지 않았습니다.
- ▶ 그 이유는 무엇이며, 이를 어떻게 해결했는지를 공부해 보겠습니다.

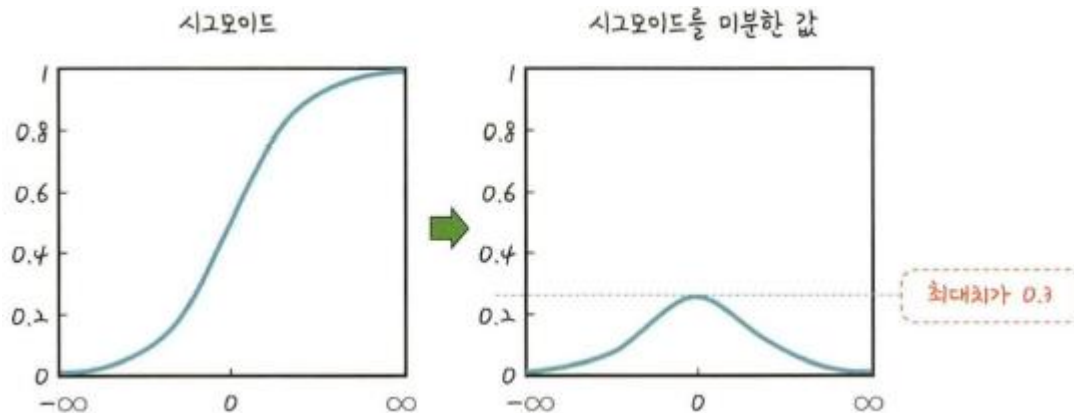
# 기울기 소실 문제와 활성화 함수

- ▶ 오자 역전파는 출력층으로부터 하나씩 앞으로 되돌아가며 각 층의 가중치를 수정하는 방법입니다.
- ▶ 가중치를 수정하려면 미분값 즉 기울기가 필요하다고 배웠습니다.
- ▶ 그런데 층이 늘어나면서 기울기가 중간에 0 이 되어버리는 기울기 소실(vanishing gradient) 문제가 발생하기 시작했습니다.



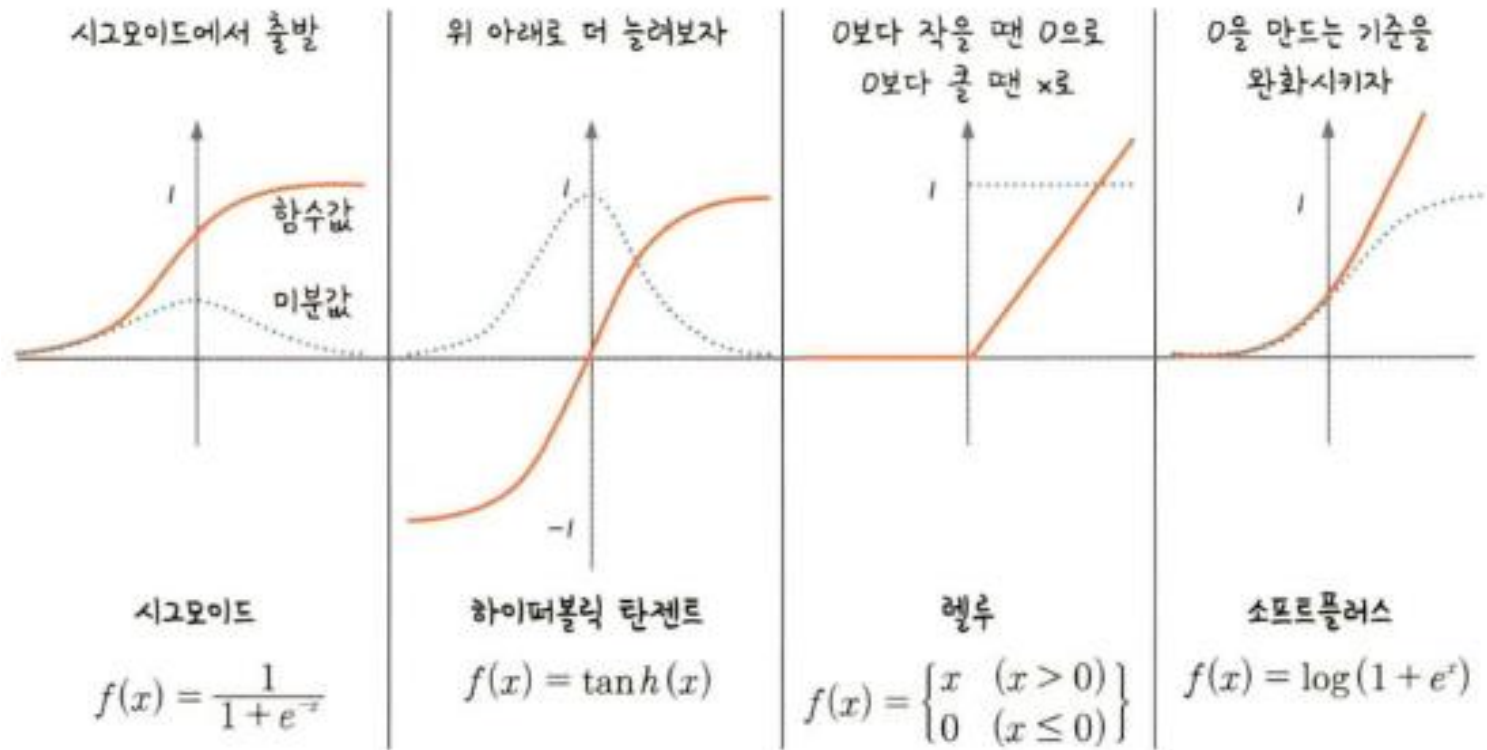
# 기울기 소실 문제와 활성화 함수

- ▶ 이는 활성화 함수로 사용된 시그모이드 함수의 특성 때문입니다.
- ▶ 앞의 그림에서 처럼 시그모이드를 미분하면 최대치가 0.3 입니다.
- ▶ 보다 작으므로 계속 곱하다 보면 0에 가까워집니다.
- ▶ 따라서 층을 거쳐 갈수록 기울기가 사라져 가중치를 수정하기가 어려워지는 것입니다.



# 기울기 소실 문제와 활성화 함수

- ▶ 이를 해결하고자 활성화 함수를 시그모이드가 아닌 여러 함수로 대체하기 시작했습니다.



## 기울기 소실 문제와 활성화 함수

---

- ▶ 먼저, 시그모이드 함수의 범위를  $-1$ 에서  $1$ 로 확장한 개념인 하이퍼볼릭 탄젠트( $\tanh$ )는 미분한 값의 범위가 함께 확장되는 효과를 가져왔습니다.
- ▶ 하지만 여전히  $1$ 보다 작은 값이 존재하므로 기울기 소실 문제는 사라지지 않습니다.



# 기울기 소실 문제와 활성화 함수

---

- ▶ 토론토대학교의 제프리 힌튼 교수가 제안한 렐루 (ReLU)는 시그모이드의 대안으로 떠오르며 현재 가장 많이 사용되는 활성화 함수입니다.
- ▶ 렐루는  $x$ 가 0보다 작을 때는 모든 값을 0으로 처리하고, 0보다 큰 값은  $x$ 를 그대로 사용하는 방법입니다.
- ▶ 단순해 보이지만, 이 방법을 쓰면  $x$ 가 0보다 크기만 하면 미분 값이 1이 됩니다.
- ▶ 따라서 여러 은닉층을 거치며 곱해지더라도 맨 처음 층까지 사라지지 않고 남아있을 수 있습니다.
- ▶ 이 간단한 방법이 여러 층을 쌓을 수 있게 했고, 이로써 딥러닝의 발전에 속도가 붙게 됩니다.



## 기울기 소실 문제와 활성화 함수

---

- ▶ 이후 렐루의 0이 되는 순간을 완화한 소프트플러스 (softplus) 등 렐루를 변형한 함수도 개발중이며, 좀더 나은 활성화 함수를 만들기 위한 노력이 이어지고 있습니다.





# 속도와 정확도 문제를 해결하는 고급 경사하강법

---

- ▶ 가중치를 업데이트하는 방법으로 우리는 경사 하강법을 배웠습니다.
- ▶ 그런데 경사 하강법은 정확하게 가중치를 찾아가지만, 한 번 업데이트할 때마다 전체 데이터를 미분해야 하므로 계산량이 매우 많다는 단점이 있습니다.
- ▶ 이러한 점을 보완한 고급 경사 하강법이 등장하면서 딥러닝의 발전속도는 더 빨라졌습니다.



# 속도와 정확도 문제를 해결하는 고급 경사하강법

---

## ▶ 확률적 경사 하강법(SGD)

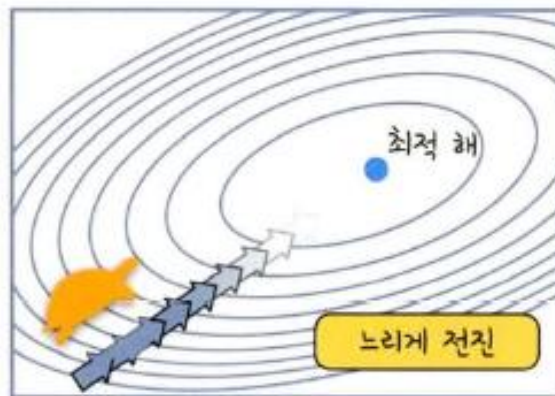
- ▶ 경사 하강법의 불필요하게 많은 계산량은 속도를 느리게 할 뿐 아니라, 최적 해를 찾기 전에 최적화 과정이 멈출 수도 있습니다.
- ▶ 확률적 경사 하강법 (Stochastic Gradient Descent, SGD)은 이러한 속도의 단점을 보완한 방법입니다.
- ▶ 전체 데이터를 사용하는 것이 아니라, 랜덤하게 추출한 일부 데이터를 사용합니다.
- ▶ 일부 데이터를 사용하므로 더 빨리 그리고 자주 업데이트를 하는 것이 가능해졌습니다.



# 속도와 정확도 문제를 해결하는 고급 경사하강법

## ▶ 확률적 경사 하강법(SGD)

- ▶ 다음 그림은 경사하강법과 확률적 경사하강법의 차이를 보여 줍니다.
- ▶ 랜덤한 일부 데이터를 사용하는 만큼 확률적 경사 하강법은 중간 결과의 진폭이 크고 불안정해 보일 수도 있습니다.
- ▶ 하지만 속도가 확연히 빠르면서도 최적 해에 근사한 값을 찾아낸다는 장점 덕분에 경사 하강법의 대안으로 사용되고 있습니다.



경사 하강법



확률적 경사 하강법

# 속도와 정확도 문제를 해결하는 고급 경사하강법

---

## ▶ 모멘텀

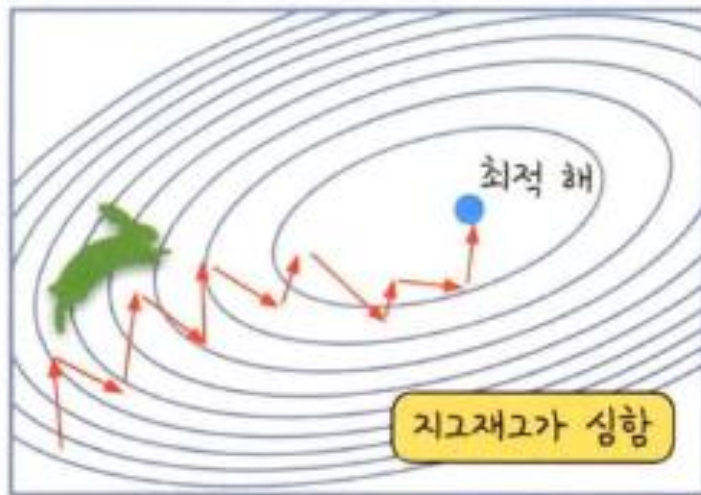
- ▶ 모멘텀 (momentum)이란 단어는 관성, 탄력, 가속도라는 뜻입니다.
- ▶ 모멘텀 SGD란 말 그대로 경사 하강법에 탄력을 더해 주는 것입니다.
- ▶ 다시 말해서, 경사 하강법과 마찬가지로 매번 기울기를 구하지만, 이를 통해 오차를 수정하기 전 바로 앞 수정 값과 방향(+, -)을 참고하여 같은 방향으로 일정한 비율만 수정되게 하는 방법입니다.



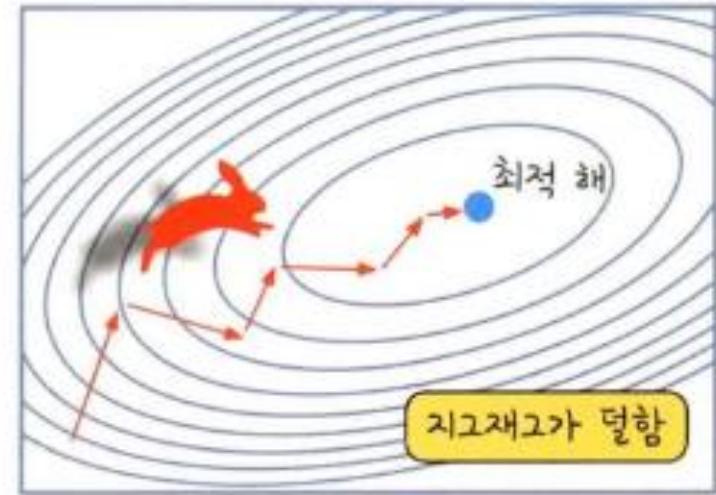
# 속도와 정확도 문제를 해결하는 고급 경사하강법

## ▶ 모멘텀

- ▶ 따라서 수정 방향이 양수(+) 방향으로 한 번, 음수(-) 방향으로 한 번 지그재그로 일어나는 현상이 줄어들고, 이전 이동 값을 고려하여 일정 비율만큼만 다음 값을 결정 하므로 관성의 효과를 낼 수 있습니다.



확률적 경사 하강법



모멘텀을 적용한 확률적 경사 하강법

# 속도와 정확도 문제를 해결하는 고급 경사하강법

---

- ▶ 이밖에 딥러닝 구동에 필요한 고급 경사 하강법과 케라스 내부에서의 활용법을 표로 정리하면 다음과 같습니다.
- ▶ 이제 다음 장부터는 케라스가 본격적으로 사용됩니다.
- ▶ 각 고급 경사하강법의 케라스 사용법도 함께 눈여겨 보기 바랍니다.



# 속도와 정확도 문제를 해결하는 고급 경사하강법

고급 경사 하강법	개요	효과	케라스 사용법
1. 확률적 경사 하강법 (SGD)	랜덤하게 추출한 일부 데이터를 사용해 더 빨리, 자주 업데이트를 하게 하는 것	속도 개선	<code>keras.optimizers.SGD(lr = 0.1)</code> 케라스 최적화 함수를 이용합니다.
2. 모멘텀 (Momentum)	관성의 방향을 고려해 진동과 폭을 줄이는 효과	정확도 개선	<code>keras.optimizers.SGD(lr = 0.1, momentum = 0.9)</code> 모멘텀 계수를 추가합니다.
3. 네스테로프 모멘텀 (NAG)	모멘텀이 이동시킬 방향으로 미리 이동해서 그레디언트를 계산. 불필요한 이동을 줄이는 효과	정확도 개선	<code>keras.optimizers.SGD(lr = 0.1, momentum = 0.9, nesterov = True)</code> 네스테로프 옵션을 추가합니다.
			<code>keras.optimizers.Adagrad(lr = 0.01, epsilon = 1e - 6)</code> 아다그라드 함수를 사용합니다.



# 속도와 정확도 문제를 해결하는 고급 경사하강법

고급 경사 하강법	개요	효과	케라스 사용법
4. 아다그라드 (Adagrad)	변수의 업데이트가 잦으면 학습률을 적게 하여 이동 보폭을 조절하는 방법	보폭 크기 개선	<code>keras.optimizers.Adagrad(lr = 0.01, epsilon = 1e - 6)</code> 아다그라드 함수를 사용합니다.  ※ 참고: 여기서 <code>epsilon</code> , <code>rho</code> , <code>decay</code> 같은 파라미터는 바꾸지 않고 그대로 사용하기를 권장하고 있습니다. 따라서 <code>lr</code> , 즉 <code>learning rate</code> (학습률) 값만 적절히 조절하면 됩니다.
5. 알엠에스프롭 (RMSProp)	아다그라드의 보폭 민감도를 보완한 방법	보폭 크기 개선	<code>keras.optimizers.RMSprop(lr = 0.001, rho = 0.9, epsilon = 1e - 08, decay = 0.0)</code> 알엠에스프롭 함수를 사용합니다.
6. 아담(Adam)	모멘텀과 알엠에스프롭 방법을 합친 방법	정확도와 보폭 크기 개선	<code>keras.optimizers.Adam(lr = 0.001, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e - 08, decay = 0.0)</code> 아담 함수를 사용합니다.



# 속도와 정확도 문제를 해결하는 고급 경사하강법

---

- ▶ 각 방법이 개발된 순서 대로 정리하였습니다.
- ▶ 먼저 나온 방법의 단점을 보완하여 다음 방법이 나온 만큼 나중에 나온 고급 경사 하강법이 좋은 성과를 보입니다.
- ▶ 맨 마지막에 기재된 아담(Adam)은 현재 가장 많이 사용되는 고급 경사 하강법입니다.



---

# Q&A

