Thesis review study

Learning representations by back-propagating errors(1986)

강한결, 조은비

CONTENT

1. 기존 퍼셉트론의 한계
2. Backpropagation 등장
3. Fig 1
4. Fig 2
5. Fig 3
6. Fig 4
7. Fig 5
8. 역전파 알고리즘

CONTENT

1. 기존 퍼셉트론의 한계
2. Backpropagation 등장
3. Fig 1
4. Fig 2
5. Fig 3
6. Fig 4
7. Fig 5
8. 역전파 알고리즘

1. 기존 퍼셉트론의 한계

단층 퍼셉트론

1) 단층 퍼셉트론이란? 단일층만 가진 모델. 1950~1960년대 고안된 신경망 구조.
2) 입력과 출력 사이에 가중치만 있음.
3) AND, OR 같은 선형 분리 문제만 해결
4) 입력값이 복잡하게 조합되는 XOR 같은 문제는 불가능

중간 유닛

1) 사람이 직접 설정
2) 학습되지 않고, 고정된 규칙만 가능

“기존 퍼셉트론은 중간 유닛(일종의 특징 추출기)이 있기는 했지만, 이 유닛들은 입력과의 연결이 ‘고정’돼 있었고 학습되지 않았다.”

- **기존의 단층 퍼셉트론에도 중간 유닛은 있었지만, 이 유닛들의 연결은 사람이 정해줬음. 그래서 표현은 할 수 있지만, 스스로 학습은 못하는 상태**

CONTENT

Backpropagation 이란?

1) 출력층에서 발생한 오차를 입력층으로 거슬러 올라가면서 가중치를 조정
2) 체인 룰을 사용해서 각 층 가중치의 영향을 계산

*출력층의 오차만 보고 은닉층의 가중치를 자동 조절

Backpropagation 흐름

1) 입력 -> 은닉 -> 출력 (forward)
2) 출력오차 -> 은닉 -> 입력 (backward)

순방향으로 계산하고, 역방향으로 오차를 전달해서 가중치를 조정

은닉 유닛

1) 단순히 입력을 출력으로 바꾸는 게 아니라, 의미 있는 내부 표현으로 바꾸는 역할
2) 사람이 정의하지 않은 특징을 스스로 학습

2. Backpropagation 등장

And, or는 간단해서 은닉이 필요 없다.
예를 들어 노란색에 And 하얀 줄이 있으면 참외다.
기침을 하거나 or 열이 나면 환자다. 이런 식으로 간단해서 선형으로 충분.

Xor은 조금 더 복잡해진다. 예를 들어 존댓말을 쓰거나 or 반말을 쓰면 사람인데, 둘다 쓰면 AI다.
이런식으로 구분한다. 단순히 선형으로 구분되지 않고 은닉 유닛의 학습이 필요해진다.

개발자가 '사람이면 1, AI면 0'이런식으로 정답만 준거고 내부 표현을 학습하기 위해 알아서 각 은닉 유닛들이 분석할 내용들을 담당한다.
은닉 유닛 1 : 반말만 감지
은닉 유닛 2 : 존댓말만 감지
은닉 유닛 3 : 둘 다 있을 때 활성화 -> 출력 유닛 억제해서 AI로 분류되게 함.

'반말' '존댓말' 등의 개념을 찾아내서 알아서 내부 표현을 학습한다.

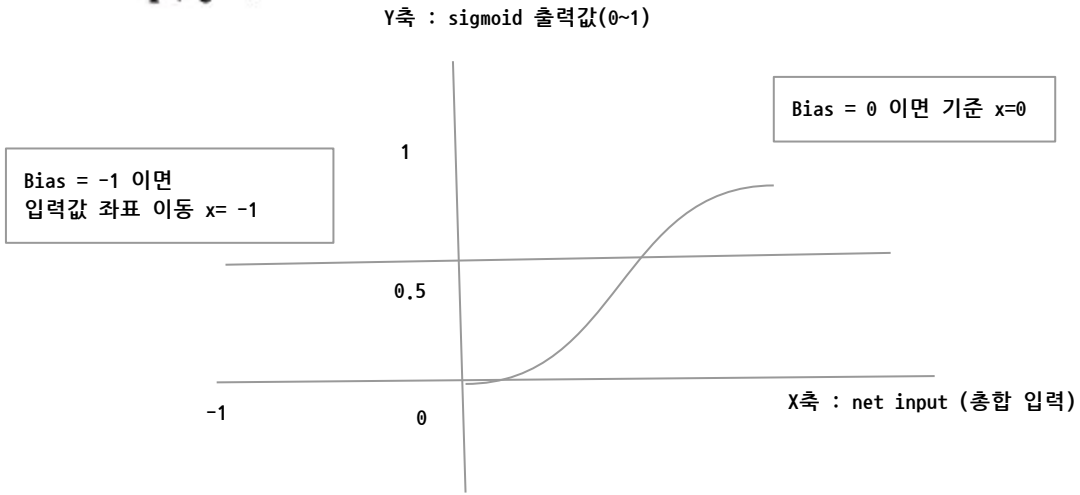
2. Backpropagation 등장

$$x_j = \sum_i y_i w_{ji}$$

수식 1은 net intput을 계산하는 식.
J번 유닛이 받는 총 입력 = 이전 유닛들의 출력 x 가중치
출력이 아니라, 그 유닛이 얼마나 자극받았는지 계산하는 것.

$$y_j = \frac{1}{1+e^{-x_j}}$$

수식 2는 수식1에서 구한 net input(x_j)를 sigmoid 함수에 통과시켜서 출력 y_j를 구하는 것.



Bias = 0 -> net input = 0 -> sigmoid(0) = 0.5
Bias = -1 -> net input = 0 -> sigmoid(-1) = 0.27

좌표를 작게 하면 출력이 작아지고, 뉴런이 덜 켜지게 됨

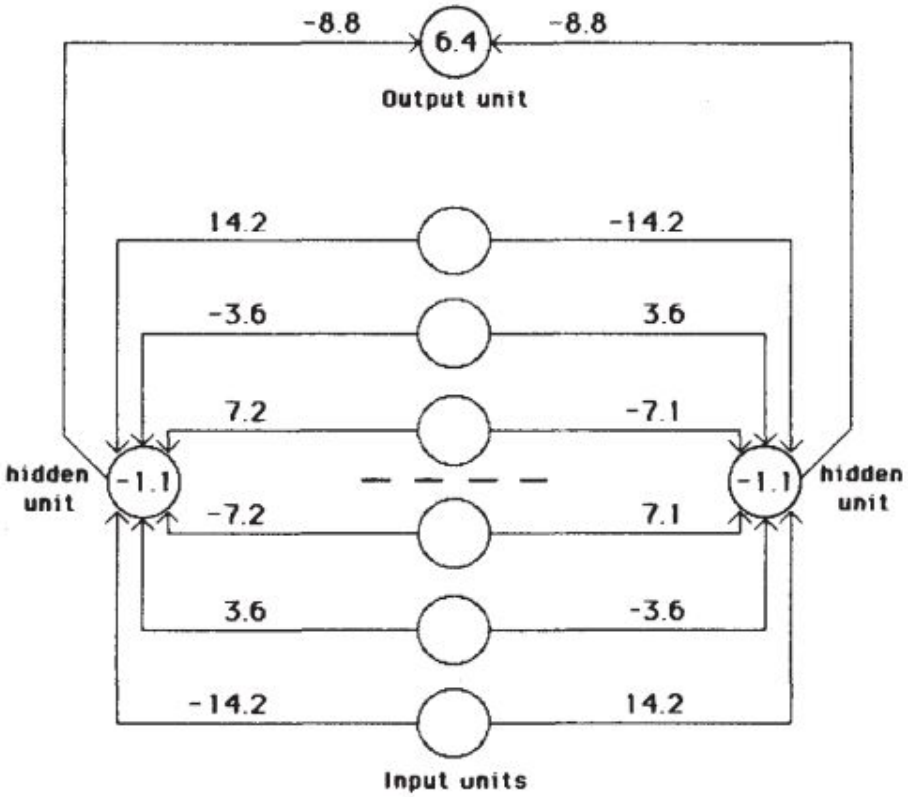
예를 들어 편의점 자동문이 열릴 때, net input이 사람 감지 센서에 걸리는 전체 신호이고 bias는 센서의 민감도를 뜻한다고 보면, bias가 음수가 되면 신호 기준이 미달돼 문이 안 열릴 수 있다.(출력 꺼짐 상태)

Bias는 기준점을 조정해서 이 유닛이 얼마나 쉽게 켜지질지 결정하는 역할.

CONTENT

3. Fig 1

실험 목표 : 사람이 규칙을 알려주지 않아도, 네트워크가 스스로 입력 벡터의 대칭을 알아챌 수 있는지 확인하기



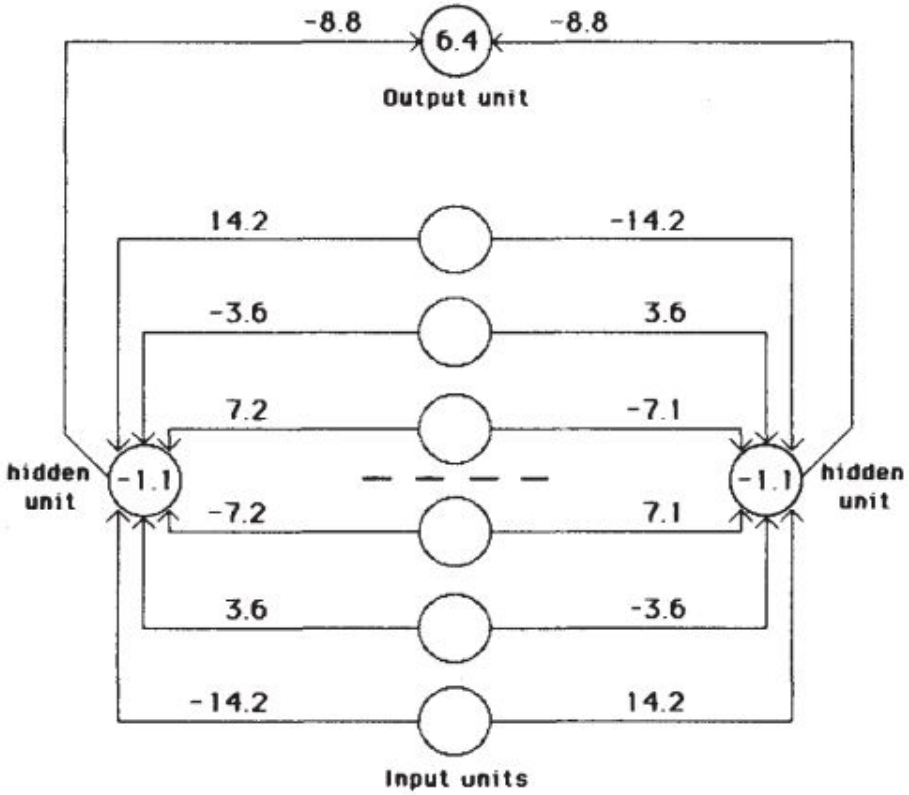
- 입력 유닛 6개 -> 은닉 유닛 2개 -> 출력 1개
- 각 가중치와 편향(bias)가 조정돼 학습됨
- 총 64개의 입력 벡터를 1,425회 반복 학습함.

대칭적인 입력 벡터가 들어왔을 때, 은닉 유닛은 꺼지고, 출력 유닛은 bias 덕분에 저절로 켜지게 된다.
- 대칭일 때 양쪽 가중치 합 = 0
- Bias가 음수 -> 은닉 유닛 꺼짐 -> 출력 유닛 켜짐

*음수면 왜 꺼질까? Net input이 0이면, sigmoid 함수 출력은 0.5다. Bias가 음수면 net input이 줄어들어서 sigmoid 출력이 0.5보다 작아지고, 출력이 꺼진 상태에 가까워진다.

비대칭 입력 벡터가 들어왔을 때, 은닉 유닛 하나가 켜지고, 출력이 억제된다.(출력값이 0에 가깝다. 거의 꺼졌다는 의미)
- 좌우 대칭이 깨지면 은닉 유닛이 하나 켜짐
- 이 유닛이 출력 유닛을 억제함 -> 출력 꺼짐

3. Fig 1



1:2:4 가중치 비율 *입력패턴을 유일하게 구분하는 구조*

- 입력 가중치 비율이 1:2:4로 설정됨
- 은닉 유닛에 들어오는 좌우 입력의 가중치가 정확히 같고 부호가 반대(±)여서, 오직 완벽한 좌우 대칭만이 net input을 0으로 만들어 은닉 유닛을 꺼지게 만들 수 있다. 라는걸 학습해냄.
- 대칭 아니면 반드시 값이 남음 -> 은닉 유닛이 반응함
- 사람이 한 건 '대칭이면 1, 아니면 0'이라는 문제를 제시한 게 다고, 이후에 bias와 가중치를 스스로 학습해서 조정하고 '대칭성' 개념과 판단 기준을 찾아낸 것.

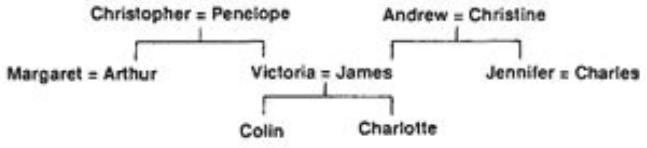
결과

- 은닉 유닛이 직접 대칭성 같은 추상 개념을 학습함
- Backprop는 은닉층이 학습을 할 수 있게 함
- 딥러닝이 '표현학습을 할 수 있다'는 것을 최초로 보여준 실험

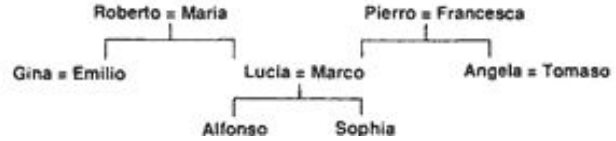
CONTENT

4. Fig 2

미국인 가족 12명



이탈리아인 가족 12명



관계명 12개

father	아버지	uncle	삼촌
mother	어머니	aunt	이모 / 고모
husband	남편	brother	형제
wife	아내	sister	자매
son	아들	nephew	조카 (남자)
daughter	딸	niece	조카 (여자)

학습 과정

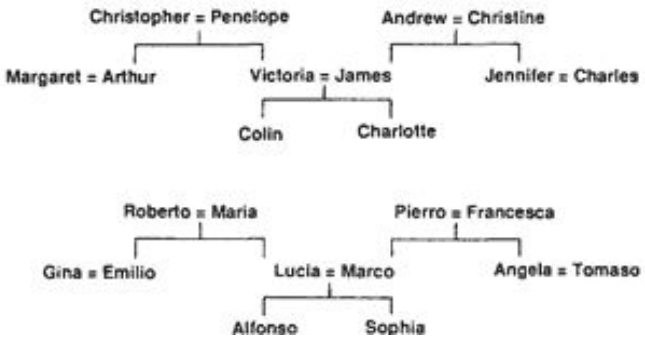
- 사람 이름과 관계를 나타낸 트리 구조
- 24명의 이름과 12개의 관계명(엄마, 아빠, 아들, 이모…) 사용
- (이름, 관계, 이름)을 묶어 triple 구조를 만들고 이를 통해 학습

단층 퍼셉트론이라면?

- Victoria는 Colin의 엄마다.. Victoria는 Charlotte의 엄마다.. Charlotte은 Colin의 자매다.. 단순히 매핑하여 암기할 뿐, 구조 및 관계에 대한 이해는 없음
- 이미 학습된 데이터에 대한 정답률도 떨어지고, 정답을 직접 알려주지 않은 데이터에 대해서는 예측이 불가능

4. Fig 2

단층 퍼셉트론과의 비교



다층 퍼셉트론은

- 부모가 같으면 형제자매다.. 엄마의 자매가 이모다.. 등 구조를 파악
- Jennifer가 Colin의 이모라는 걸 알려주면 Angela나 Gina가 Sophia의 이모라는 것도 예측함

결과

- 학습에 직접 사용되지 않은 4개의 triple을 포함해 104개의 triple에 대해 전부 정답을 맞힘
- 모델이 정답값만 외우는게 아니라, 구조 자체를 학습한다는 증거
- 단층 퍼셉트론은 학습에 쓰지 않은 4개는 아예 맞힐 수 없는 구조였고, 나머지 100개도 꽤 틀리곤 했던 것에 비해 비약적인 발전을 이룸

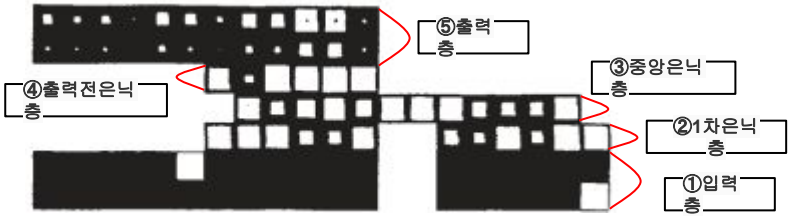
즉, 단층 퍼셉트론에서 다층 퍼셉트론으로 진화하면서 두 가지가 가능해졌다.

1. 비선형 구조(복잡한 패턴)의 학습
2. 일반화

CONTENT

5. Fig 3

모델 내부 구조



총 5개의 층으로 구성

- 입력층과 출력층만 있던 단층 퍼셉트론에서 중간에 은닉층이 있는 다층 구조로 진화
- 그림의 하얀 부분은 활성화 정도를 의미

---

① 입력층
- 사람과 관계정보가 one-hot 방식으로 입력됨
- 총 36개(사람24, 관계12)의 뉴런 중 2개(사람 이름 1개, 관계명 1개)가 활성화

② 1차 은닉층
- 사람과 관계정보를 분산표현(벡터값)으로 변환
- 각 뉴런이 부분적으로 활성화되어 6차원 벡터값을 표현
- 사람 벡터(6차원)와 관계 벡터(6차원)는 분리되어 병렬적으로 존재

③ 중앙 은닉층
- 사람과 관계의 의미를 통합하여 연결
- 의미 학습과 관계 추론의 중심이 됨

④ 출력 전 은닉층
- 뉴런 수를 6개로 줄여 노이즈를 삭제하고 정보 정리
- 핵심적인 신호만 남겨 예측을 강화

⑤ 출력층
- 24개의 뉴런(사람)이 부분적으로 활성화
- Sigmoid 활성화함수를 사용하여 값이 0.2 이하인 경우는 오답, 0.8 이상인 경우는 정답으로 처리

CONTENT

6. Fig4

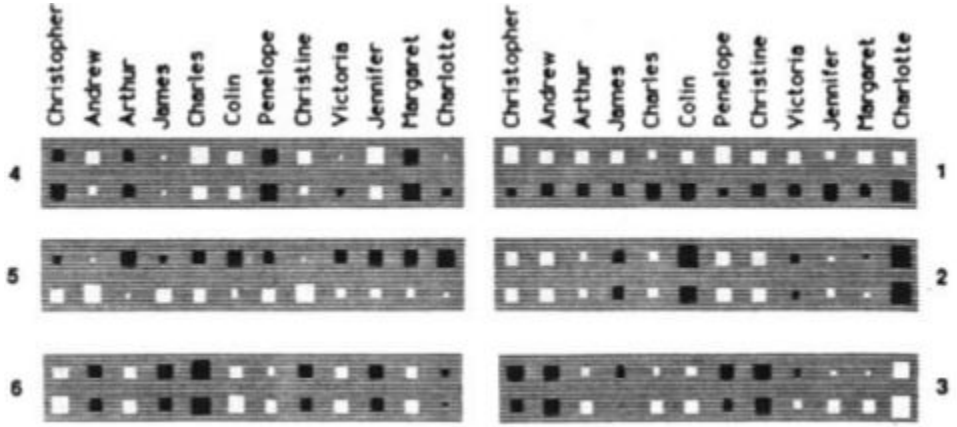


Fig4는 입력층 -> 은닉층 가중치를 시각화한 그림이다.

하얀색 박스가 클 수록 높은 가중치를 뜻하고, 검은색 박스가 클수록 가중치가 적다는 것을 의미한다.

입력층에 12명의 영국인, 12명의 이탈리아인으로 24개의 유닛

연구자들이 은닉층을 6개로 두고, 역전파 알고리즘으로 의미있는 뭔가를 만들어내봐! 라고 한 상황.

그 결과, 네트워크가 '분산 표현'으로 유닛1, 2, 6에서 '국적', '세대', '가족 분가'를 알아냄

Unit1: 주로 영국인과 이탈리아인을 구분하는 역할을 담당. 다른 대부분의 유닛들은 이 구분을 무시하는데, 영국인의 표현이 이탈리아인의 표현과 매우 유사하다는 의미함 다시 말해, 영국 가족이랑 이탈리아 가족이 같은 구조를 가졌다는 걸 알아챈 뒤, 한쪽에서 배운 걸 다른 쪽에도 적용했다는 것. '아 가족관계가 서로 비슷하구나' 라고 생각한 것처럼.

Unit2: 한 사람이 속한 세대를 인코딩

Unit6: 그들이 나온 가족의 분가를 인코딩

분산 표현 : 각 사람을 하나의 숫자로 표현하는 게 아니라, 여러 차원의 특징들의 조합으로 표현해서 일반화가 가능해짐

- 기존 : Christopher - 1
- 분산 : christopher - [영국인, 3세대, A분가]

입력 : Christopher -> 은닉층: 6개 유닛이 각각 다른 활성도 -> 출력: Andrew

Christopher의 아버지는 Andrew다. 이런식으로 관계를 학습

6. Fig4

1500번 훈련

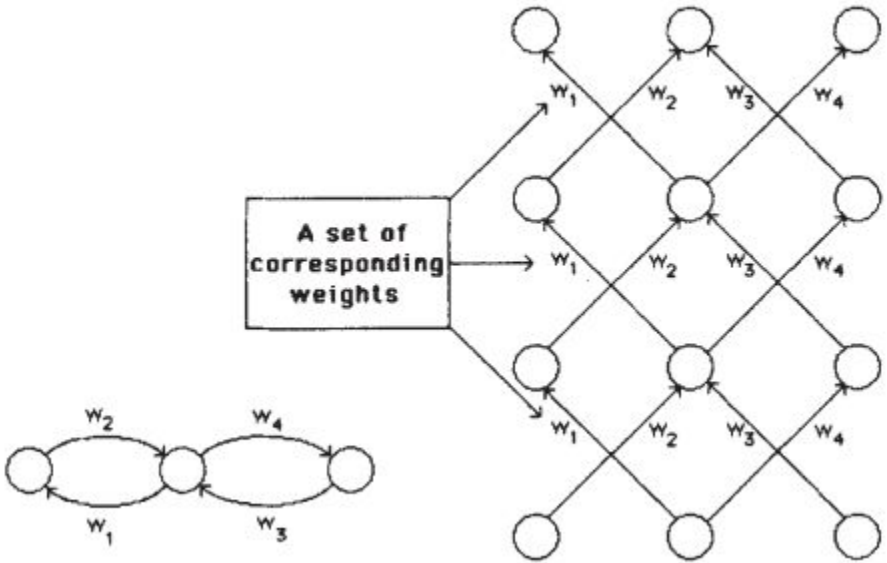
학습률(ε)
- 처음에는 작은 학습률로 방향을 잡고
- 나중에 큰 학습률로 속도를 높임

가중치 해석을 쉽게 하기 위해 각 가중치 변화 후 모든 가중치를 0.2%씩 감소시킴

가중치 감쇠
- 정말 중요한 가중치는 계속 강화돼서 감쇠를 이겨내고 살아남지만, 별로 안 중요한 가중치는 점점 0에 가까워지면서 ‘중요도’를 측정함
- 최종적으로 큰 가중치만 남으니까 어떤 게 중요한 연결인 지 알아냄

CONTENT

5. Fig 5 RNN의 이론이 됐던 실험



순환 구조를 시간 순으로 펼쳐서 학습하는 아이디어

우리가 아는 형태 : 입력층 - 은닉층 - 출력층 일직선
실제 뇌 : 복잡한 뉴런 연결, 앞뒤로 순환하는 정보

앞서 역전파를 적용했던 실험들
A -> B -> C 일방향 구조에만 적용해왔음 (역전파 오차만 C -> B -> A 방향으로 학습했음)

순환 네트워크는 : A <-> B <-> C 이렇게 생겼음.

A와 C 뉴런이 직접 연결된 건 아니지만, A의 신호가 B를 거쳐 C에 도달하고, C의 신호가 B를 거쳐 A에 도달하는 식으로 상호작용. 인접한 뉴런들 간의 양방향 연결과 시간에 따른 반복처리가 핵심.

순환 구조 신경망을 어떻게 역전파로 학습시킬 수 있을까?

첫 번째 반복의 결과를 다시 입력으로 넣어서 두 번째 반복…
-> 이 과정을 옆으로 펼친 그림. -> 순환 네트워크를 펼쳐서 일반 다층 네트워크로 만들었고, 역전파를 적용할 수 있게 함.

A : 밥 먹었니? B : 응 너는? A 나는 이따 먹으려고 B : 같이 먹을래? 이렇게 대화가 있다고 하면,
순환 네트워크 A <-> B는 어디서부터 시작인지 모르기에 역전파를 적용할 수 없지만,
이걸 시간 순으로 일방향으로 펼쳐서 A1 -> B1 -> A2 -> B2 역전파를 적용할 수 있다.

CONTENT

6. 역전파 알고리즘

합성함수란?

- 한 함수의 출력값을 다른 함수의 입력값으로 사용하는 새로운 함수
- 다층구조에서 역전파 알고리즘의 핵심 개념

합성함수의 예시

Function 1

불이 세지면
온도가 높아진다.

Function 2

온도가 높아지면
라면이 빨리
익는다.

합성함수의 특징

- 두 개 이상의 함수가 순차적으로 결합되어 하나의 함수를 만듦
- 다수의 함수가 연쇄적으로 작동하여 출력값이 생성됨

편미분이란?

- 다변수함수에서 특정 변수에 대한 변화율(그 변수의 영향력)만을 계산하는 방법

6. 역전파 알고리즘

합성함수의 편미분(Chain rule)

$$\frac{\partial z}{\partial x_i} = \frac{df}{dy} \cdot \frac{\partial y}{\partial x_i}$$

- 하나의 변수가 합성함수의 출력값에 어느 정도 영향을 미치는지를 계산하는 방법
- ex) 불의 세기가 라면 익는 속도에 어떤 영향을 미칠까?
- 미분의 연쇄법칙을 이용하여 계산

---

(4)(5) 출력층에서 오차를 입력값에 대해 미분

$$\partial E/\partial x_j = \partial E/\partial y_j \cdot dy_j/dx_j$$

$$\partial E/\partial x_j = \partial E/\partial y_j \cdot y_j(1-y_j)$$

- 출력의 오차값(E)를 입력값(xj)에 대해 편미분
- 출력층에서 은닉층으로, 은닉층에서 입력층으로 연쇄적으로 연산이 이루어짐
- 각 층에서 오차에 얼만큼 기여했는지를 알 수 있음
- 각 층의 오차는 그 위에 있는 모든 층에 영향을 미치기 때문에, 이 층이 얼만큼 틀렸는지를 함께 다른 층에 전달하기 위함

E : 오차
x : 입력값
y : 출력값
j : 출력층(sigmoid 사용)

6. 역전파 알고리즘

(6) 오차를 가중치로 미분

$$\partial E/\partial w_{ji} = \partial E/\partial x_j \cdot \partial x_j/\partial w_{ji}$$
$$= \partial E/\partial x_j \cdot y_i$$

- 모델이 가중치를 업데이트하기 위해 필요한 건 오차에 대한 가중치의 기여도
- 가중치의 기울기값(각 층의 가중치가 오차에 어떤 영향을 미치는지)을 알 수 있음

---

(7) 은닉층에서의 오차 기여도 계산

$$\partial E/\partial y_i = \sum_j \partial E/\partial x_j \cdot w_{ji}$$

- 출력층(5)에서는 라벨값(정답)이 있으니까 직접 오차 계산이 가능한데, 은닉층엔 라벨값이 없어서 오차 자체를 구할 수 없음
- 그래서 출력층에서 계산한 오차에서 출발해서 층을 지나면서 오차를 가중합하고, 그걸 기반으로 오차 기여도를 계산함

6. 역전파 알고리즘

(8) 가중치 업데이트

$$\Delta w = -\varepsilon \partial E / \partial w$$

- (학습률 * 가중치의 기울기값)이 실제 가중치 업데이트량이 됨
- 층을 순차적으로 되돌아가며 반복적으로 수행됨

---

(9) 수식(8)에 모멘텀(관성)을 추가한 버전

$$\Delta w(t) = -\varepsilon \partial E / \partial w(t) + \alpha \Delta w(t-1)$$

- 가중치에 모멘텀계수(0~1)을 곱한 값을 더해주는 방식
- 이전 업데이트가 옳은 방향일 때는 약간의 가속도를 부여하고 틀린 방향일 때는 브레이크 역할을 해줌
- Adam 등 현대의 최적화 기법들에서도 널리 사용되는 방식

※참고 : 국소최적해(local optimum) 이슈

- 이론적으로, 경사하강법은 국소최적에 빠질 가능성이 있는 알고리즘임
- 그러나 실제로는 모델 자체 성능이 크게 떨어지지만 않는다면 이 이슈가 발생하는 경우는 거의 없음

그래서 이 논문은..!

- 입력층과 출력층만 있던 단층 퍼셉트론에 은닉층을 추가하고
- 합성함수의 미분(chain rule)과 편미분 기법을 사용하여
- 각 층을 거꾸로 거쳐가며 가중치를 업데이트하는 역전파 알고리즘을 제안
- 다양한 실험 모델을 만들어 실용성 테스트

그 결과

- 단층 퍼셉트론이 해결할 수 없던 XOR 문제 등 비선형 구조 학습에 성공
- 학습시키지 않은 케이스에 대한 일반화된 관계 추론에 성공
- 현대 딥러닝 모델 연구의 초석이 됨