

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. 역전파 알고리즘

CONTENT

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

1. 기존 퍼셉트론의 한계

퍼셉트론

- 1) 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. ”

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

CONTENT

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

2. Backpropagation 등장

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

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

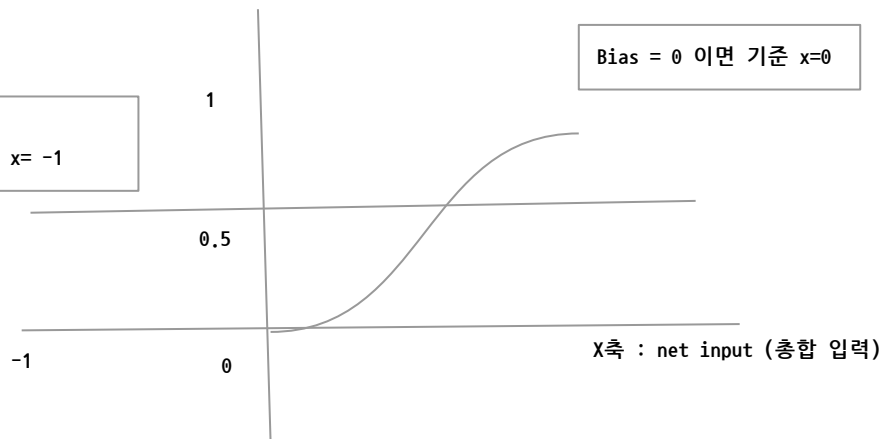
수식 1은 net input을 계산하는 식.

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_j 를 구하는 것.

Y축 : sigmoid 출력값(0~1)



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)

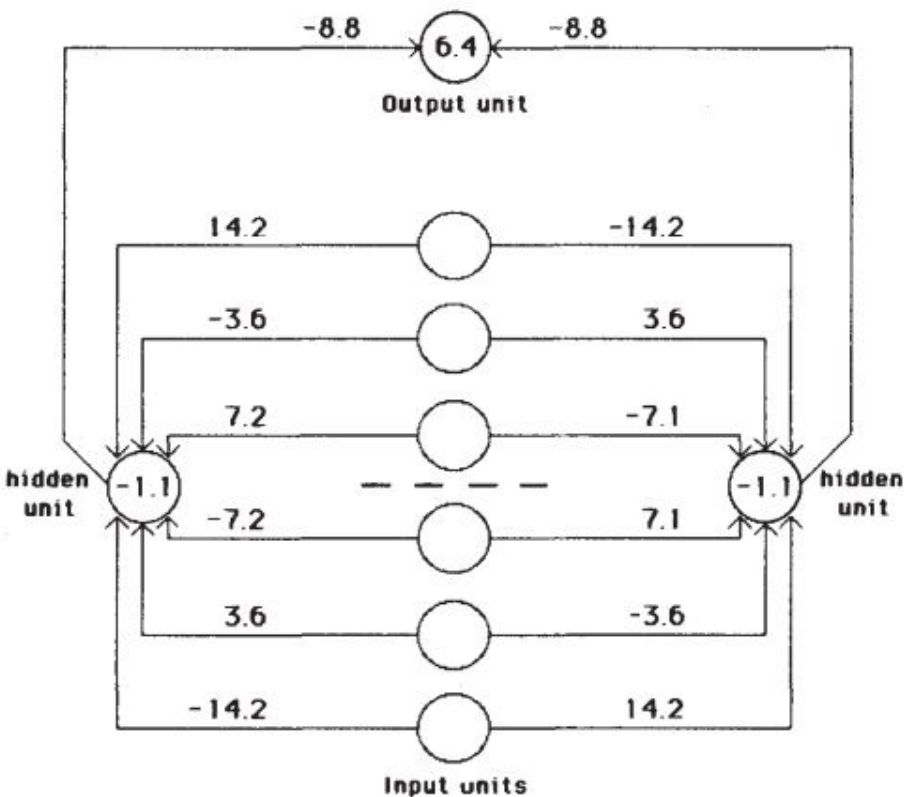
Bias는 기준점을 조정해서 이 유닛이 얼마나 쉽게 켜지질지 결정하는 역할.

CONTENT

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

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.

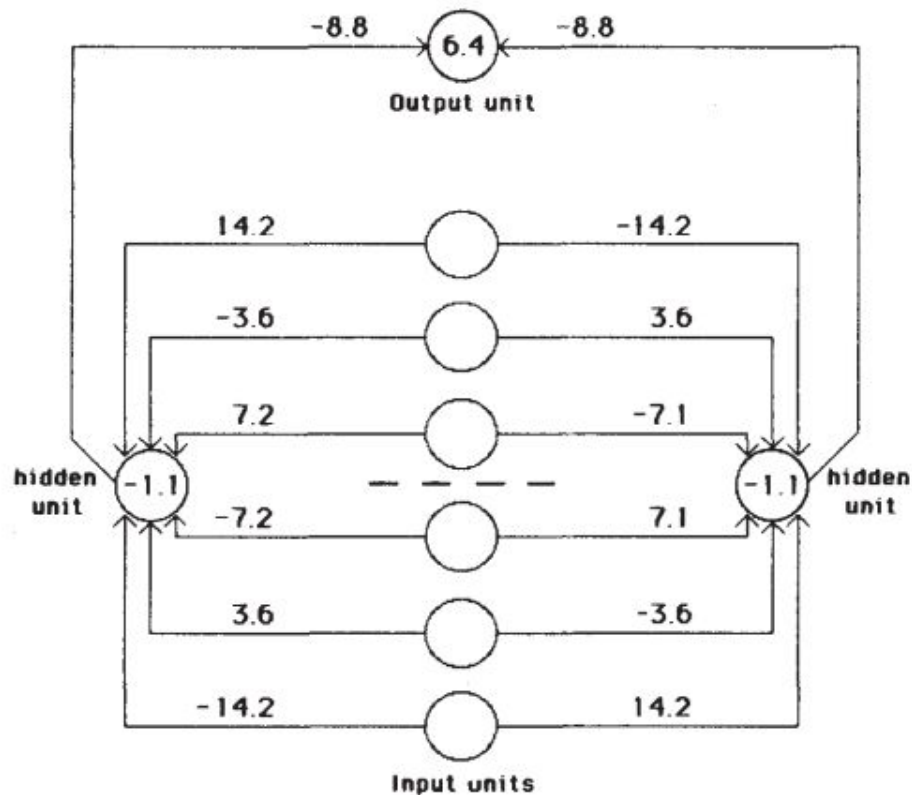
- 대칭일 때 양쪽 가중치 합 = 0
- Bias가 음수 → 은닉 유닛 꺼짐 → 출력 유닛 켜짐

*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(\pm)여서, 오직 완벽한 좌우 대칭만이 net input을 0으로 만들어 은닉 유닛을 꺼지게 만들 수 있다. 라는걸 학습해냄.
- 대칭 아니면 반드시 값이 남음 \rightarrow 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

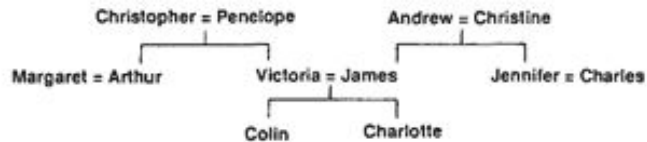
- 은닉 유닛이 직접 대칭성 같은 추상 개념을 학습함
- Backprop는 은닉층이 학습을 할 수 있게 함
- 딥러닝이 '표현학습을 할 수 있다'는 것을 최초로 보여준 실험

CONTENT

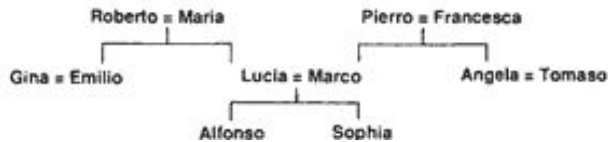
1. 기존 퍼셉트론의 한계
2. Backpropagation 등장
3. Fig 1
4. Fig 2
5. Fig 3
6. 역전파 알고리즘

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

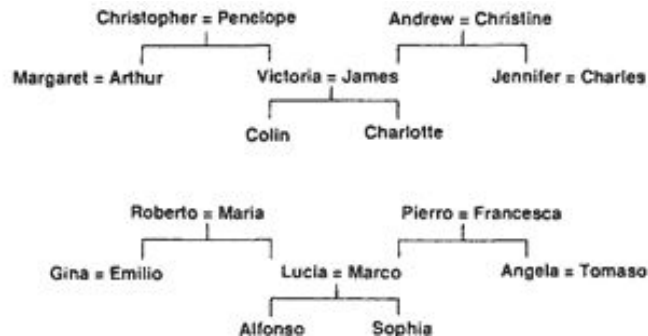
- 사람 이름과 관계를 나타낸 트리 구조
- 24명의 이름과 12개의 관계명(엄마, 아빠, 아들, 이모...) 사용
- (이름, 관계, 이름)을 묶어 triple 구조를 만들고 이를 통해 학습

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

4. Fig 2

단층 퍼셉트론과의 비교



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

다층 퍼셉트론은

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

즉, 단층 퍼셉트론에서 다층 퍼셉트론으로 진화하면서 두 가지가 가능해졌다.

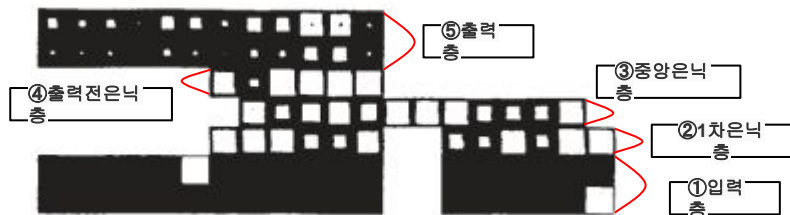
1. 비선형 구조의 학습
2. 일반화

CONTENT

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

5. Fig 3

모델 내부 구조



총 5개의 층으로 구성

- 입력층과 출력층만 있던 단층 퍼셉트론에서 중간에 은닉층이 있는 다층 구조로 진화
- 부분연결의 개념이 본격적으로 등장하기 이전 시기이므로 모든 층이 완전연결층으로 구성됨
- 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 6개의 뉴런으로 구성되어 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

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

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 만들
- 다수의 함수가 연쇄적으로 작동하여 출력값이 생성됨

6. 역전파 알고리즘

Chain rule(합성함수의 편미분)

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

- 하나의 변수가 합성함수의 출력값에 어느 정도 영향을 미치는지를 계산하는 방법
 - 불의 세기가 라면 익는 속도에 어떤 영향을 미칠까?
 - Chain rule(미분의 연쇄법칙)을 이용하여 계산
-

다층 퍼셉트론에서 Chain rule의 활용

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

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

- 출력의 오차값(E)를 가중치(x,y...)에 대해 편미분
- 가중치가 오차값에 어떤 영향을 미칠까?
- 출력층에서 은닉층으로, 은닉층에서 입력층으로 입/출력값이 전달되며 연쇄적으로 계산됨