CSCI E-89B Lecture 1
아무것도 몰라도 이해되는 신경망 기초

정리: InJoo 학습노트

Fall 2025

읽기 가이드

이 문서는 **전혀 모르는 사람**을 위해 작성되었다. 수학식은 꼭 필요한 만큼만, 대신 **비유 + 손계산 예시 + 체크리스트**로 이해를 도와준다.

- **핵심 비유:** 신경망은 "입력 재료 → 여러 단계의 조리(층) → 결과 요리"를 만드는 **레시피**이다.
- **핵심 목표:** 예측이 실제와 얼마나 다른지(오차)를 숫자로 재고(손실 함수), 그걸 줄이는 방향으로 레시피(가중치)를 조금씩 조정(경사 하강법)한다.
- **읽는 순서:** 1) 신경망이 뭔지 직관, 2) 층/뉴런/활성화함수, 3) 손실/비용 함수, 4) 순전파/역전파 계산, 5) 경사하강법(배치/미니배치/SGD), 6) 실전 팁 & FAQ.

1 신경망을 아주 직관적으로 이해하기

한 줄 요약

신경망(Neural Network)은 **입력** x를 받아 **여러 층(layer)**을 거치며 **출력** $\hat{y}$를 만드는 **함수들의 합성**이다:

$$\hat{y} = f^{(L)}\big(f^{(L-1)}(\cdots f^{(1)}(x)\cdots)\big).$$

여기서 각 $f^{(\ell)}$은 **선형변환(가중치/편향)**과 **비선형 활성화 함수**로 이루어진다.

왜 굳이 "깊게(Deep)" 써야 할까?

- 얕은(한두 단계) 모델은 **직선/완만한 곡선** 같은 단순한 경계만 만든다.
- 복잡한 문제(예: 이미지/언어)는 **비선형 변환을 여러 번** 적용해 **복잡한 모양의 결정 경계**가 필요하다.
- 층을 쌓으면, **간단한 조각 특징** → **중간 특징** → **고수준 의미** 식으로 표현이 점점 "추상화"된다.

2 뉴런, 층, 활성화 함수

2.1 두 입력, 은닉 2, 출력 1인 2-2-1 미니 네트워크

구성: 입력 $x = (x_1, x_2)$, 은닉층 뉴런 u_1, u_2, 출력 $\hat{y}$. 은닉층은 **ReLU**를, 출력층은 회귀면 **선형**, 이진분류면 **시그모이드**, 다중분류면 **소프트맥스**를 쓴다고 생각하면 된다.

$$u_1 = f\Big(w_{01}^{(1)} + w_{11}^{(1)} x_1 + w_{21}^{(1)} x_2\Big),$$
$$u_2 = f\Big(w_{02}^{(1)} + w_{12}^{(1)} x_1 + w_{22}^{(1)} x_2\Big),$$
$$\hat{y} = g\Big(w_0^{(2)} + w_1^{(2)} u_1 + w_2^{(2)} u_2\Big).$$

여기서 f는 은닉층 활성화, g는 출력층 활성화다.

2.2 활성화 함수(왜 비선형이 필요?)

- **ReLU** ($f(x) = \max(0, x)$): 빠르고 간단, 깊은 네트워크에서도 학습 잘 됨. 기본값으로 생각해도 좋다.
- **Sigmoid** ($\sigma(x) = 1/(1 + e^{-x})$): 출력이 (0,1)이라 **확률**처럼 해석 쉬움(이진분류 출력층에 주로 사용).
- **Softmax:** $\mathrm{softmax}_j(z) = \dfrac{e^{z_j}}{\sum_k e^{z_k}}$ (다중분류 출력층).
- **Tanh:** $(-1, 1)$ 범위. 옛날엔 자주 썼지만 ReLU에 밀림.

핵심: 활성화가 **비선형**이어야 층을 쌓을 의미가 생긴다. 선형만 쌓으면 전체가 결국 또 선형이 된다.

3 손실(LOSS)과 비용(COST) 정확히 구분하기

3.1 손실 함수 $L^{(i)}$ (한 샘플의 틀림 정도)

- **회귀(실수 예측):** $L^{(i)} = \big(\hat{y}^{(i)} - y^{(i)}\big)^2$ (MSE 단일항)
- **이진분류:** $L^{(i)} = -\big(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})\big)$
- **다중분류:** $L^{(i)} = -\sum_{c=1}^{M} y_c^{(i)} \log \hat{y}_c^{(i)}$ (원-핫 y 가정)

3.2 비용 함수 $J(\mathbf{w})$ (데이터 전체 평균 오차)

$$J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} L^{(i)}(\mathbf{w}).$$

요약: L = 개별샘플 오류, J = 전체 평균 오류(우리가 최소화 하려는 목표).

4 순전파(Forward) & 역전파(Backprop) — 손계산으로 감 잡기

4.1 설정(회귀)

은닉 ReLU, 출력 **선형.** 입력/가중치/정답을 **일부러** 간단히 잡아 계산이 한 번에 눈에 보이도록 한다.

$$
\begin{aligned}
&x_1 = 1,\ x_2 = 2, \qquad y = 2.0\\
&(\text{은닉}1)\ w_{01}^{(1)} = 0.1,\ w_{11}^{(1)} = 0.5,\ w_{21}^{(1)} = 0.3\\
&(\text{은닉}2)\ w_{02}^{(1)} = -0.1,\ w_{12}^{(1)} = 0.4,\ w_{22}^{(1)} = 0.1\\
&(\text{출력})\ w_{0}^{(2)} = 0.2,\ w_{1}^{(2)} = 1.0,\ w_{2}^{(2)} = 0.5
\end{aligned}
$$

1) 순전파

$$
\begin{aligned}
z_1 &= 0.1 + 0.5(1) + 0.3(2) = 1.2 \Rightarrow u_1 = \max(0, 1.2) = 1.2\\
z_2 &= -0.1 + 0.4(1) + 0.1(2) = 0.4 \Rightarrow u_2 = 0.4\\
\hat{y} &= 0.2 + 1.0 \cdot 1.2 + 0.5 \cdot 0.4 = 1.6\\
L &= (\hat{y} - y)^2 = (1.6 - 2)^2 = 0.16
\end{aligned}
$$

2) 역전파(미분)

핵심은 **연쇄법칙(Chain Rule).** $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w}$.

출력층:

$$
\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) = 2(-0.4) = -0.8
$$

$$
\frac{\partial L}{\partial w_0^{(2)}} = -0.8, \quad \frac{\partial L}{\partial w_1^{(2)}} = -0.8 \cdot u_1 = -0.96, \quad \frac{\partial L}{\partial w_2^{(2)}} = -0.8 \cdot u_2 = -0.32.
$$

은닉층: 먼저 은닉 출력에 대한 민감도:

$$
\frac{\partial L}{\partial u_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial u_1} = -0.8 \cdot w_1^{(2)} = -0.8, \qquad \frac{\partial L}{\partial u_2} = -0.8 \cdot w_2^{(2)} = -0.4.
$$

ReLU의 도함수 $f'(z) = \mathbf{1}\{z > 0\}$. 여기선 $z_1, z_2 > 0 \Rightarrow f'(z_1) = f'(z_2) = 1$.

$$
\frac{\partial L}{\partial z_1} = -0.8, \quad \frac{\partial L}{\partial z_2} = -0.4.
$$

이제 은닉 가중치:

$$
\begin{aligned}
&\frac{\partial L}{\partial w_{01}^{(1)}} = -0.8 \cdot 1 = -0.8, \quad \frac{\partial L}{\partial w_{11}^{(1)}} = -0.8 \cdot x_1 = -0.8, \quad \frac{\partial L}{\partial w_{21}^{(1)}} = -0.8 \cdot x_2 = -1.6,\\
&\frac{\partial L}{\partial w_{02}^{(1)}} = -0.4 \cdot 1 = -0.4, \quad \frac{\partial L}{\partial w_{12}^{(1)}} = -0.4 \cdot x_1 = -0.4, \quad \frac{\partial L}{\partial w_{22}^{(1)}} = -0.4 \cdot x_2 = -0.8.
\end{aligned}
$$

3) 가중치 업데이트(경사 하강법)

$$w \leftarrow w - \alpha \frac{\partial L}{\partial w}.$$

학습률을 $\alpha = 0.02$로 작게 잡아 **손실이 실제로 줄어드는지** 확인한다.

업데이트:

$$\Delta w = -\alpha \nabla L \Rightarrow \begin{cases} w_0^{(2)} \leftarrow 0.2 + 0.016 = 0.216 \\ w_1^{(2)} \leftarrow 1.0 + 0.0192 = 1.0192 \\ w_2^{(2)} \leftarrow 0.5 + 0.0064 = 0.5064 \\ w_{01}^{(1)} \leftarrow 0.1 + 0.016 = 0.116 \\ w_{11}^{(1)} \leftarrow 0.5 + 0.016 = 0.516 \\ w_{21}^{(1)} \leftarrow 0.3 + 0.032 = 0.332 \\ w_{02}^{(1)} \leftarrow -0.1 + 0.008 = -0.092 \\ w_{12}^{(1)} \leftarrow 0.4 + 0.008 = 0.408 \\ w_{22}^{(1)} \leftarrow 0.1 + 0.016 = 0.116 \end{cases}$$

업데이트 후 순전파(손실 감소 확인):

$$\begin{aligned} z_1 &= 0.116 + 0.516(1) + 0.332(2) = 1.296 \Rightarrow u_1 = 1.296 \\ z_2 &= -0.092 + 0.408 + 0.232 = 0.548 \Rightarrow u_2 = 0.548 \\ \hat{y} &= 0.216 + 1.0192 \cdot 1.296 + 0.5064 \cdot 0.548 \approx 1.8144 \\ L &= (1.8144 - 2)^2 \approx 0.0345 \quad (\textbf{처음}\ 0.16 \rightarrow \textbf{감소}) \end{aligned}$$

교훈: α가 너무 크면 오히려 손실이 커질 수 있다(발산/오버슈팅). 작게부터 시도하고 모니터링하자.

5 경사 하강법(최적화) — 세 가지 모드

- **배치 GD**: 전체 데이터(m개) 평균 그라디언트로 한 번 업데이트. 정확하지만 느림.
- **미니배치 GD**: s개씩 묶어 평균 그라디언트로 업데이트($1 < s < m$). 현업 표준.
- **SGD**: $s = 1$. 가볍고 빠르지만 요동이 큼.

모두 공통 업데이트식은

$$w \leftarrow w - \alpha \cdot \frac{1}{s} \sum_{i \in \text{batch}} \nabla L^{(i)}(w).$$

6 실전 체크리스트(왕초보용)

1. **데이터 분할:** train/validation/test를 시간 순서 존중 또는 무작위로 적절히 나눈다.
2. **스케일링:** 입력을 표준화/정규화하면 학습이 더 안정적.

3. **초기화:** He/Xavier 등 합리적인 초기화(프레임워크 기본값 활용).

4. **기준선:** 평균 예측/최빈값/지속성 같은 **베이스라인**과 꼭 비교.

5. **얼리 스토핑:** 검증 손실이 나빠지면 학습 중단.

6. **학습률 스케줄:** 처음엔 조금 크게, 점점 줄이는 전략도 유효.

7 (선택) 분류 출력층 한눈 요약

이진분류(라벨 0/1)

출력층 활성화 $g = \sigma$ (시그모이드), 손실은 **Binary Cross-Entropy**.

$$\hat{y} = \sigma(z), \quad L = -\big(y \log \hat{y} + (1-y) \log(1-\hat{y})\big).$$

다중분류(라벨 1-of-K)

출력층 활성화 $g = \text{softmax}$, 손실은 **Categorical Cross-Entropy**.

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{z}), \quad L = -\sum_{c=1}^{K} y_c \log \hat{y}_c.$$

8 자주 헷갈리는 것들(FAQ)

Q1. *Loss*와 *Cost* 차이? 손실(Loss)은 **개별 샘플**의 오차, 비용(Cost)은 **전체 평균** 오차(최소화 대상).

Q2. 기울기(그라디언트)가 0이면 항상 최적? 아니다. **안장점/최대점**일 수도 있다. 실전에서는 검증성능/학습곡선으로 판단.

Q3. 왜 활성화가 꼭 비선형이어야 하나? 선형만 겹치면 전체가 결국 **한 번의 선형변환**과 다를 바 없다. 복잡한 패턴을 못 배운다.

Q4. 학습률은 어떻게 정하지? 작게 시작(예: $10^{-3} \sim 10^{-2}$)해서 **학습곡선**을 보고 조정. 발산하면 더 작게, 너무 느리면 조금 키운다.

Q5. 미니배치 크기 s는? 하드웨어/데이터에 따라 다르지만 32, 64, 128이 무난. 너무 크면 **평탄부**에 갇히기도.

마무리 요약(핵심만 쏙)

- 신경망은 **선형 + 비선형** 변환을 층층이 쌓아 **복잡한 함수**를 근사한다.
- **손실**은 한 샘플, **비용**은 전체 평균. 우리는 비용을 **내리도록** 가중치를 업데이트한다.

- 순전파로 예측, 역전파로 기울기, 경사하강법으로 업데이트. **학습률/미니배치**가 실전 핵심 노브.