1주차 강의 요약

1. 딥러닝 소개

1. 신경망은 무엇인가?

- 신경망이란 입력(x)와 출력(y)를 매칭하는 함수(f)를 찾는 과정
- 충분한 데이터가 주어지면 더 잘 알아낼 수 있음
- 해당 뉴런에 관계없는 x라도 입력으로 넣어줘야 함. 관계 여부는 신경망이 학습하면서 알아서 조절
- In the case of Housing Price Prediction...

  ✅ ReLU (Rectified Linear Unit) : 0과 결과 값 중 큰 값을 선택하는 함수

  ✅ 여러 개의 x를 사용하여 y를 추측 가능

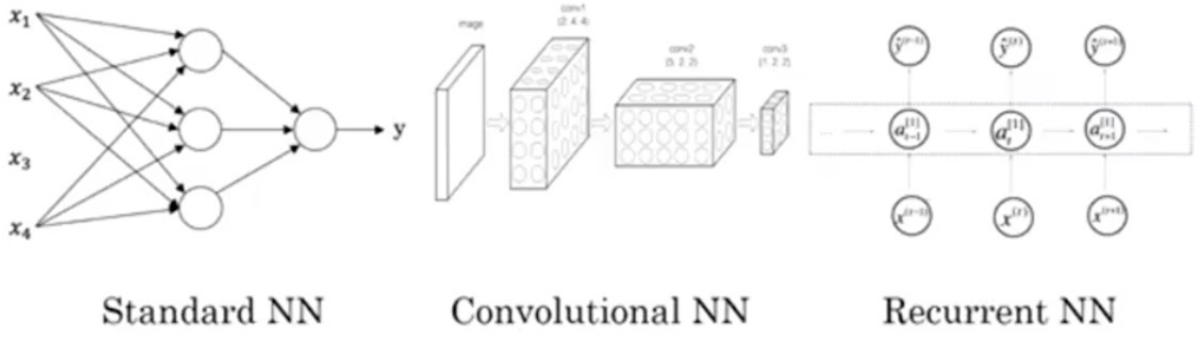
  ex. size, #bedrooms, zip code, wealth(x)→price(y)

2. 신경망을 이용한 지도학습

- 머신러닝의 방법에는 지도 학습, 비지도 학습 등 여러 가지 종류가 있음
- Supervised Learning : 정답이 주어져 있는 데이터를 사용해 컴퓨터를 학습시키는 방법

Input(x)	**Output**(y)	**Application**	**Type of NN**
Home features	Price	Real Estate	standard NN
Ad, user info	Click on ad? (0/1)	Online Advertising-어떤 광고를 내보내야 사람들이 잘 클릭하는지 예측함→굉장히 돈이 됨!	standard NN
Image	Object(1....n)	Photo tagging	CNN(image)
Audio	Text transcript	Speech recognition	RNN(sequence data, temporal factor)
English	Chinese	Machine translation	RNN
Image, Radar Info	Position of other cars	Autonomous driving	Custom/Hybrid

- NN examples



- Type of Data : 구조적 및 비구조적 데이터를 신경망에 사용하여 예측 가능

  - structured data : 정보의 특성이 잘 정의되어 있음

    ➡️ database

    ➡️ 광고, 사용자 맞춤 추천

  - unstructured data : 특징적인 값을 추출하기 어려운 형태의 데이터. 딥러닝 덕분에 컴퓨터가 비구조적 데이터를 잘 인식할 수 있게 됨

    ➡️ audio, image, text

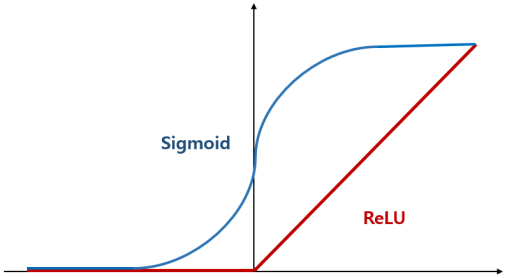
3. 왜 딥러닝이 뜨고 있을까요?

- 깊은 모델일수록 더 많은 데이터가 필요하며, 더 뛰어난 성능을 나타냄
- 최근 딥러닝이 급부상하는 이유

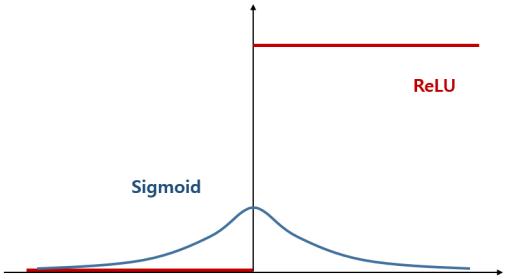
  ✅ 데이터 양 증가

  ✅ 컴퓨터 성능 향상 : CPU→GPU (*학습 과정이 매우 iterative하기 때문에 빠른 계산이 중요함: Idea→Code→Experiment)

  ✅ 알고리즘의 개선



(1) sigmoid와 ReLU 함수



(2) sigmoid와 ReLU 함수의 미분

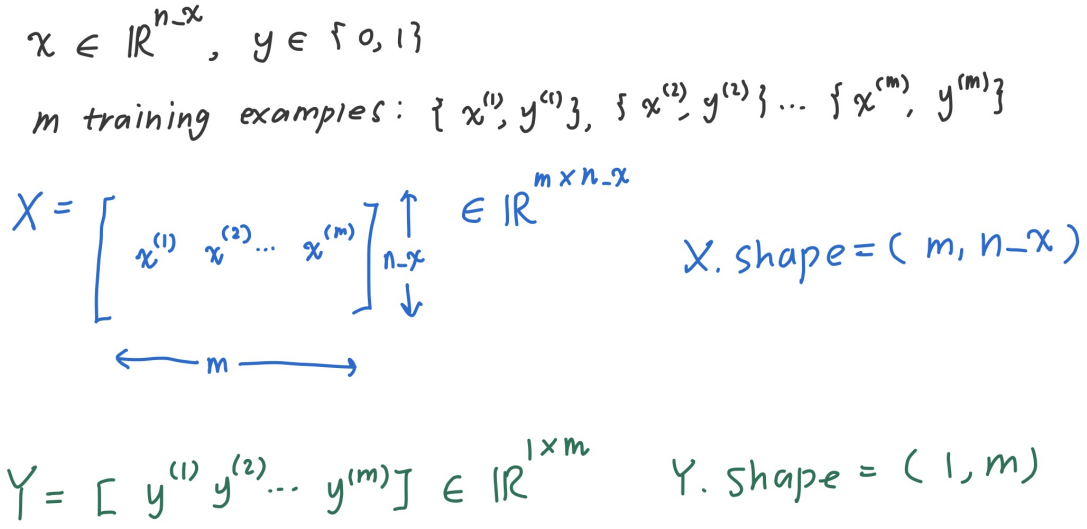
> 💡 sigmoid는 오른쪽, 왼쪽 끝으로 가면 미분값이 0이 되기 때문에 gradient가 vanishing하는 문제가 발생하는데, ReLU 함수를 사용해서 문제를 해결할 수 있음→경사 하강법 알고리즘이 훨씬 빨라짐!

2. 신경망과 로지스틱회귀

1. 이진 분류

- 신경망에서 학습하는 방법에는 **정방향 전파(forward pass)**와 **역전파(backpropagation)가** 있음
- 이진 분류란 그렇다(1)/아니다(0) 2개의 클래스로 분류하는 것
- 로지스틱 회귀(logistic regression)이란 이진 분류에 사용되는 알고리즘
- **Binary Classification**

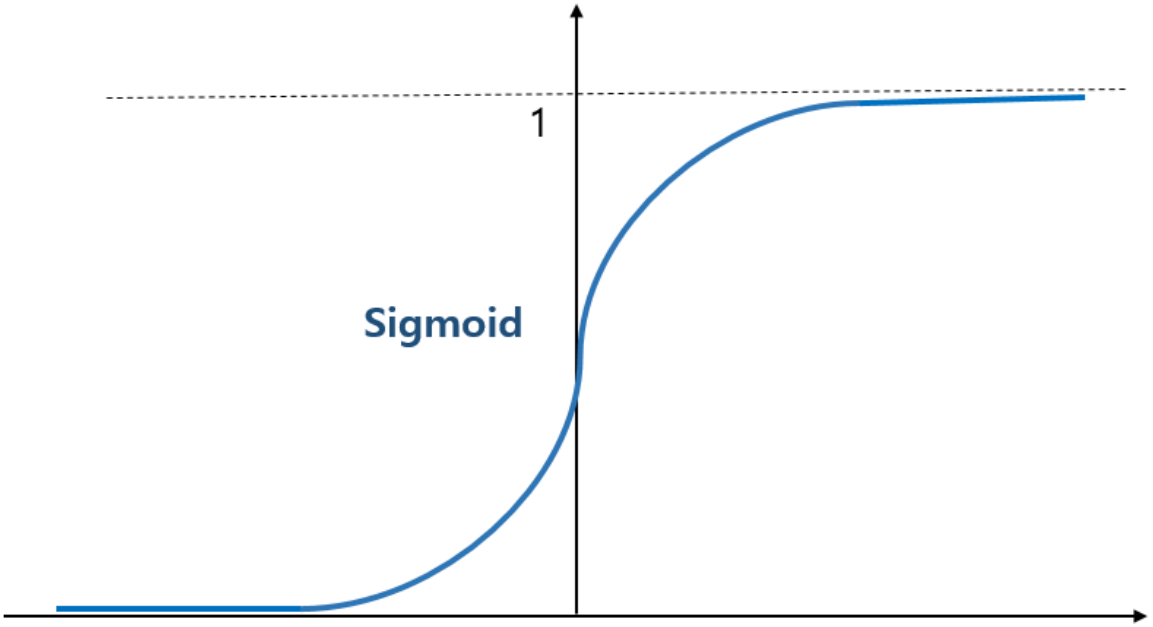
  ➡️ 64×64 짜리 map이 3개(RGB)→일차원으로 flatten→input(x)의 크기(n_x): 64x64×3=12288→신경망을 통과해 고양이 사진인지(1) 아닌지(0)를 분류
- Notation



2. 로지스틱 회귀

- **로지스틱 회귀**란 답이 0 또는 1로 정해져 있는 이진 분류 문제에 사용되는 알고리즘
- X(입력 특성), y(주어진 입력 특성 X에 해당하는 실제 값), y^(y의 예측값)
- $0 \le \hat{y} \le 1$, $\hat{y} = P(y=1 \mid x)$ ➡️ x가 주어졌을 때 y가 1일 확률

- Parameters: w, b
- 선형 회귀 시 y^=w^T*X+b를 통해 계산하지만, 해당 값은 0과 1 범위를 벗어나기 때문에 sigmoid 함수를 통해 해당 값을 0과 1 사이의 값으로 변환시켜줌
- 따라서 로지스틱 회귀를 이용해 **y^=*σ*(w^*T**X+b)** 로 구하게 됨 (*σ(z)=1/1+e^-z, z=w^T*X+b)



(1) Sigmoid 함수

3. 로지스틱 회귀의 비용함수

- 학습의 목표는 실제값(y)에 가까운 예측값(y^)를 구하는 것
- **손실 함수**: 하나의 입력특성(x)에 대한 실제값(y)과 예측값(y^) 사이의 오차를 계산하는 함수

💡 **손실 함수**

$$L(\hat{y}, y) = -(y log \hat{y} + (1-y) log(1-\hat{y}))$$

(1) y=0 인 경우 손실 함수($L(y\hat{},y)=-\log(1-y\hat{})$)는 0에 가까워지도록 y^는 0에 수렴하게 됨→log(1-y^)이 최대한 커야 하며 y^는 최소여야 함(부호가 음수이기 때문)

(2) y=1인 경우 손실 함수($L(y\hat{},y)=-\log y\hat{}$)는 0에 가까워지도록 y^는 1에 수렴하게 됨→logy^가 최대한 커야 하며 y^는 최대여야 함(부호가 음수이기 때문)

➡️ 두 경우 모두 손실 함수를 최소화하는 w, b를 찾는 것이 목표!

- **손실 함수**는 **하나의 입력**에 대한 오차를 계산하는 함수
- **비용 함수**는 **모든 입력**에 대한 오차를 계산하는 함수
- 따라서 비용 함수는 모든 입력에 대한 손실 함수의 평균 값으로 구할 수 있음

$$J(w,b) = -\frac{1}{m}\sum_{i=1}^{i=m}(y^{(i)} log \hat{y}^{(i)} + (1-y^{(i)}) log(1-\hat{y}^{(i)}))$$

➡️ **최종 목표는 비용 함수를 최소화하는 w, b를 찾는 것! (참값에 가까운 예측값을 찾는 것)**

4. 경사 하강법(Gradient Descent)

- **경사 하강법**은 비용함수를 최소화하는 파라미터 w, b를 찾아내는 방법 중 하나
- 비용 함수는 **볼록한(curve)** 형태여야 함. 만약 볼록하지 않은 형태의 함수를 쓰게 되면 경사 하강법을 이용해 최적의 파라미터를 찾을 수 없음
- 함수의 최솟값을 모르기 때문에 일단 임의의 점을 골라서 시작
- 경사 하강법은 가장 가파른 방향, 즉 함수의 기울기를 따라서 최적의 값으로 한 스텝씩 업데이트하게 됨

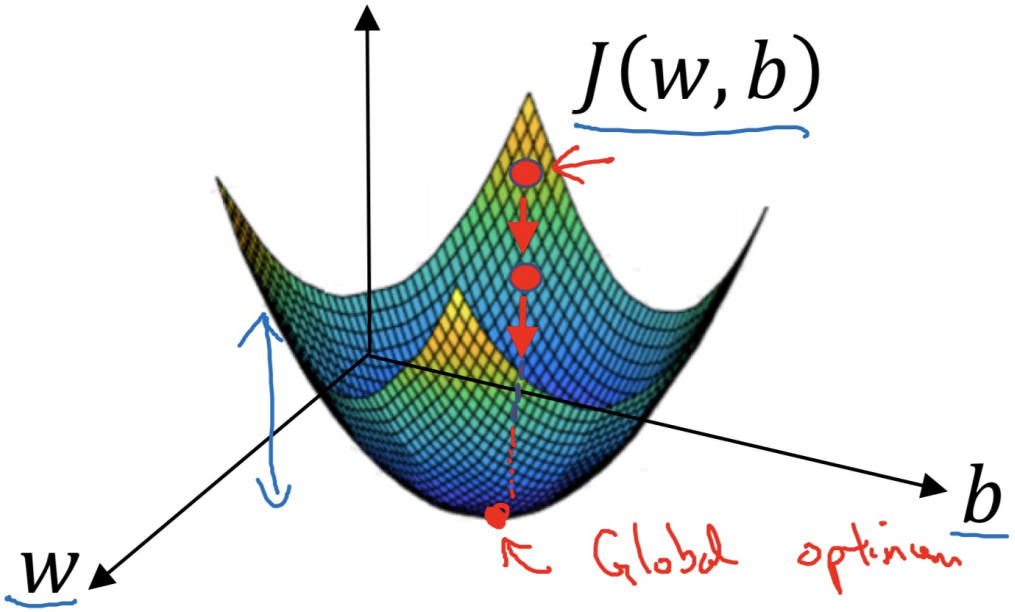
💡 **알고리즘: w, b의 값을 반복적으로 업데이트**

$$w : w - \alpha \frac{dJ(w,b)}{dw}$$

$$b : b - \alpha \frac{dJ(w,b)}{db}$$

(* α: 학습률이라고 하며, 얼마만큼의 스텝으로 나아갈 것인지 정함,
dJ(w,b)/dw: 도함수라고 하며, 미분을 통해 구한 값이자 기존 w/b 지점에서의 기울기. dw라고 표기하기도 함. w와 b에서 각각의 도함수는 함수의 기울기가 w/b 방향으로 얼만큼 변하는지 나타냄)

- 만약 **dw>0**이면, 파라미터 w는 기존의 w보다 **작은** 방향으로 업데이트 될 것이고 만약 **dw<0**이면 파라미터 w는 기존의 w보다 **큰** 방향으로 업데이트 될 것 ➡️ global optima에 도달해야 함



5. 미분

- 도함수(=어떤 함수의 기울기)란 변수 a를 조금만 변화했을 때, 함수 f(a)가 얼만큼 변하는지를 측정하는 것

$$\frac{d}{da} f(a) = \frac{df(a)}{da}$$

6. 더 많은 미분 예제

- 기울기 혹은 도함수는 함수에 따라 달라질 수 있음

함수 : $f(a)$	a^2	a^3	$\ln(a)$
도함수 : $\frac{d}{da}f(a)$	$2a$	$3a^2$	$\frac{1}{a}$