2. 신경망과 로지스틱회귀

▼ 목차

이진 분류

1️⃣ Binary Classification

💠 이진 분류란 특성 벡터 X에 대한 레이블 y가 참(1)인지 거짓(0)인지를 예측(분류)함

- e.g. 고양이 사진 분류
  - 1(cat) / 0(non-cat)
  - 이미지 데이터는 입력된 사진의 픽셀 크기만큼의 행렬 3개가 RGB 픽셀 강도 값을 나타내는 형태로 표현되며, 모든 픽셀 강도 값을 벡터 X의 한 열로 나타내어 특성 벡터로 바꿀 수 있음

- 이미지가 64×64픽셀이라면, 64×64 크기의 RGB 강도 값 행렬 3개로 표현되며 이를 특성 벡터로 바꾼 것은 64×64×3=12,288차원상에 존재함

2️⃣ Notation

Notation

- one training sample : $(x, y) \longrightarrow x \in \mathbb{R}^{n_x}$, $y \in \{0, 1\}$

  n_x차원　　이진 분류

- training set (#m training samples) : $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

  ↓

  - $$X = \underbrace{\left.\begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix}\right\} n_x}_{m} \rightarrow X \in \mathbb{R}^{n_x \times m}$$

    (python)

    X.shape = (n_x, m)

  - $$Y = [y^{(1)}, y^{(2)}, \cdots, y^{(m)}] \rightarrow Y \in \mathbb{R}^{1 \times m}$$

    Y.shape = $(1, m)$

로지스틱 회귀

1️⃣ Logistic Regression

💠 로지스틱 회귀는 이진 분류 문제에서 주로 사용되며, 특성 벡터 X를 입력하면 시그모이드 함수를 적용하여 y의 예측값(ŷ)을 계산함

$$\hat{y} = P(y = 1|x),\ 0 \leq \hat{y} \leq 1$$

- y의 예측값(ŷ)은 입력될 특성 벡터 X에 대해 실제 값 y가 1일 확률
- 이진 분류의 경우 ŷ가 0과 1 사이의 값이어야 함

$$parameters : w \in \mathbb{R}^{n_x},\ b \in \mathbb{R}$$
$$output : \hat{y} = \sigma(w^T x + b),\ where\ \sigma(z) = \frac{1}{1+e^{-z}}$$

- 위 식(선형 회귀식에 시그모이드 함수 적용)을 이용해 입력 x에 대해 매개변수 w, b를 적절하게 조정하며 최적의 ŷ를 계산하게 됨

로지스틱 회귀의 비용함수

1️⃣ Loss Function

💠 손실함수는 ŷ와 y 간의 오차를 나타내며, training sample 하나에 관해 정의되어 그 하나가 얼마나 잘 예측되었는지 측정함

$$L(\hat{y},\ y) = \frac{1}{2}(\hat{y} - y)^2$$

- 일반적으로 사용하는 손실함수 식은 위와 같지만, 로지스틱 회귀에서는 이를 사용하면 지역 최솟값에 빠질 위험이 있어 아래 식을 사용함

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

- y = 0: L(ŷ, y) = -log(1 - ŷ)가 0에 가까워지도록 ŷ가 0에 수렴
- y = 1: L(ŷ, y) = -log ŷ가 0에 가까워지도록 ŷ가 1에 수렴

2️⃣ Cost Function (in Logistic Regression)

💠 비용함수는 모든 입력에 대해 계산한 손실함수의 평균값으로, 매개변수 w, b가 training set 전체를 얼마나 잘 예측하는지 측정함

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

경사하강법

1️⃣ Gradient Descent

> 경사하강법은 비용함수를 최소로 만드는, 즉 전체 데이터셋에 대해 예측이 잘 이루어지도록 하는 파라미터 w와 b를 찾아내는 방법 중 하나임

- 임의의 점에서 시작하여 비용함수가 최소가 되는 지점을 찾게 되며, 따라서 비용함수는 볼록한 형태여야 함

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

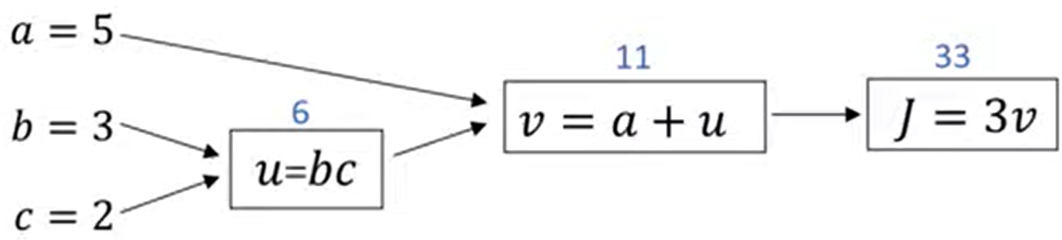
- α는 학습률로 나아갈 스텝의 크기를 정함
- 비용함수의 도함수값 dw가…
  - dw > 0: 파라미터 w는 기존의 w값보다 작은 방향으로 업데이트됨
  - dw < 0: 파라미터 w는 기존의 w값보다 큰 방향으로 업데이트됨
- 두 개 이상의 변수에 대한 도함수는 함수의 기울기가 분모에 있는 변수의 방향으로 얼마나 변했는지 나타냄

계산 그래프와 미분

1️⃣ Computation Graph

$$J(a, b, c) = 3(a + bc) = 3(5 + 3 \times 2) = 33$$
$$u = bc$$
$$v = a + u$$
$$J = 3v$$

- 위 식을 계산 그래프로 나타낸 것은 다음과 같음



2️⃣ Derivatives With Computation Graphs

같은 식에서 다양한 미분 예제를 살펴보자.

$$\frac{dJ}{dv} = 3$$
$$\frac{dv}{da} = 1$$
$$\frac{dJ}{da} = \frac{dJ}{dv} \times \frac{dv}{da} = 3$$

위와 같이 합성함수의 도함수를 합성함수를 구성하는 함수의 미분을 곱함으로써 구하는 것을 미분의 연쇄법칙이라고 한다. 이어서 J를 b에 대해 미분해 보자.

$$\frac{dJ}{db} = \frac{dJ}{du} \times \frac{du}{db} = 3 \times c = 6$$

로지스틱 회귀의 경사하강법

1️⃣ Logistic Regression Derivatives

로지스틱 회귀에서의 손실함수를 각각의 변수로 미분한 것은 다음과 같다.

$$da = -\frac{y}{a} + \frac{1-y}{1-a}$$
$$dz = a - y$$
$$dw_1 = \frac{dL}{dw_1} = x_1 dz$$
$$db = \frac{dL}{db} = dz$$

---

m개 샘플의 경사하강법

1️⃣ Logistic Regression on m Examples

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{i=m}(L(a^{(i)},\ y^{(i)}))$$

비용함수 식을 손실함수를 L로 하여 다시 쓰면 위와 같으며, 이는 의사 코드로 다음과 같이 나타낼 수 있다.

$J=0$; $dw_1=0$; $dw_2=0$; $db=0$

For $i=1$ to m

$\quad z^{(i)} = w^T x^{(i)} + b$

$\quad a^{(i)} = \sigma(z^{(i)})$

$\quad J \mathrel{+}= -[\,y^{(i)} \log a^{(i)} + (1-y^{(i)}) \log(1-a^{(i)})\,]$

$\quad dz^{(i)} = a^{(i)} - y^{(i)}$

$\quad dw_1 \mathrel{+}= x_1^{(i)} dz^{(i)}$

$\quad dw_2 \mathrel{+}= x_2^{(i)} dz^{(i)}$ $\quad n=2$

$\quad db \mathrel{+}= dz^{(i)}$

$J \mathrel{/}= m$

$dw_1 \mathrel{/}= m$; $dw_2 \mathrel{/}= m$; $db \mathrel{/}= m$

- 위와 같이 for문을 사용한 코드의 경우 특성의 개수가 많아진다면 다중 for문을 사용하게 되어 complexity가 높아지는 문제가 있음