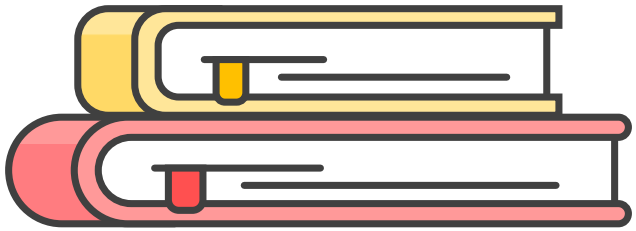
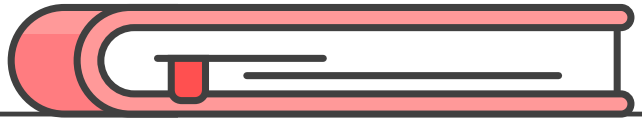


3.4~3.7 Softmax Regression

허지혜

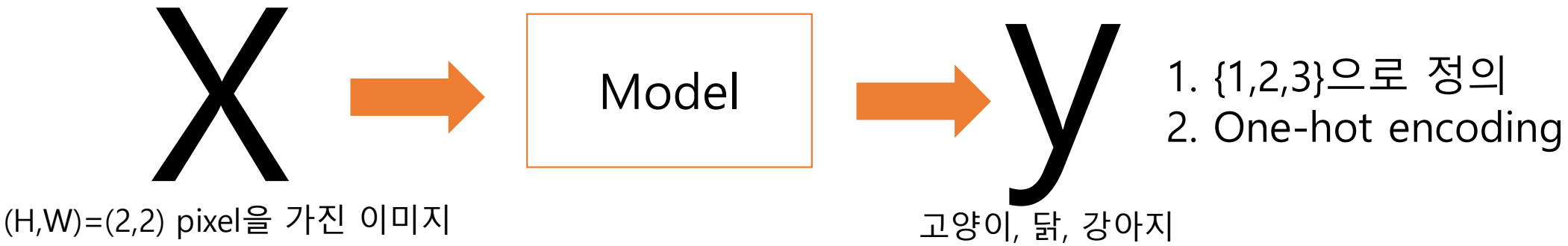


Softmax Regression

Regression	- 집 가격이 얼마인지 - 어떤 야구팀이 몇 번 승리를 할 것인지 - 환자가 몇 일만에 퇴원할 것인지
Classification	- 메일이 스팸인지 아닌지 - 고객이 구독 서비스에 가입할지 아닐지 - 이미지에 있는 객체가 무엇인지 - 어떤 물건을 구매할지

⇒카테고리별로 값을 할당하거나 어떤 카테고리에 속할 확률이 얼마나 되는지 예측하는 것

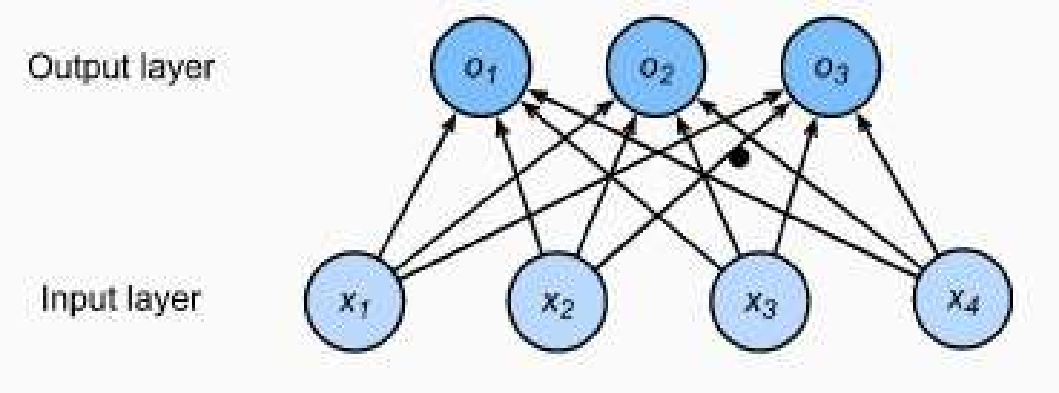
Classification Problem



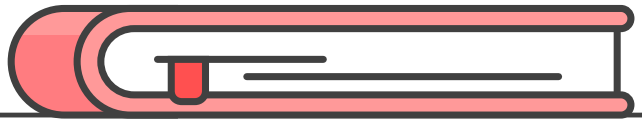
Network Architecture

여러 클래스를 분류할 때 예측할 때는 카테고리 개수와 같은 수의 출력들이 필요하다.
예) MNIST손글씨 : 0~9 분류 -> 출력층의 노드 수는 10개
예) 4개 특성과 3개의 동물 카테고리 출력들이 있으니 가중치는 12개의 scalar로 구성, 3개의 bias 구성

$$
\begin{aligned}
o_1 &= x_1 w_{11} + x_2 w_{21} + x_3 w_{31} + x_4 w_{41} + b_1, \\
o_2 &= x_1 w_{12} + x_2 w_{22} + x_3 w_{32} + x_4 w_{42} + b_2, \\
o_3 &= x_1 w_{13} + x_2 w_{23} + x_3 w_{33} + x_4 w_{43} + b_3.
\end{aligned}
$$



Softmax Regression은 단일층의 NN으로 구성된다.
출력은 모든 입력들과 연관 되서 계산되기 때문에 위의 출력층은 Fully Connected이다.



Softmax Operation

$$o_1 = x_1 w_{11} + x_2 w_{21} + x_3 w_{31} + x_4 w_{41} + b_1,$$
$$o_2 = x_1 w_{12} + x_2 w_{22} + x_3 w_{32} + x_4 w_{42} + b_2,$$
$$o_3 = x_1 w_{13} + x_2 w_{23} + x_3 w_{33} + x_4 w_{43} + b_3.$$

I번째 카테고리에 대한 Confidence level을 표현하기 위해서 출력을 o_1 으로 표현한다.
이렇게 구성하면 어떤 카테고리에 속하는지를 결과 값들 중에서 가장 큰 값의 클래스로 선택하면 되고 $argmax\ o_1$ 으로 계산할 수 있다.

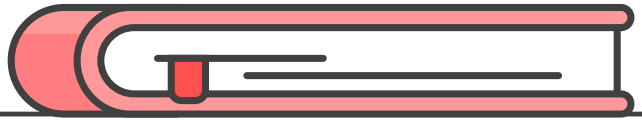
그렇지만 출력층의 값을 직접 사용하면 두가지 문제가 있다.

1. 출력값의 범위가 불확실해서 시각적으로 이 값들의 의미를 판단하기 어렵다는 것이다.
2. 실제 label은 이산값을 갖기 때문에 불특성 범위를 갖는 출력값과 레이블 값의 오류를 측정하는 것은 어렵다. 위를 확률값으로 나오도록 해볼 순 있지만 새로운 데이터가 주어졌을 때 확률값이 0또는 확률값이고 전체 합이 1이 된다는 것을 보장할 수 없다.

⇒ 위를 다루기 위해 softmax regression 분류 모델을 만들었다.
⇒ 선형 회귀와 다르게 모든 결과값들의 합이 1이 되도록 하는 비선형성에 영향을 받는다.
⇒ 위 연산은 예측하는 카테고리의 결과를 안바꾸면서 적절한 의미부여를 해준다.
⇒ 각 결과값이 0 또는 양수값을 가진다.

벡터 표현법 : $\mathbf{o}^{(i)} = \mathbf{W}\mathbf{x}^{(i)} + \mathbf{b}, \hat{\mathbf{y}}^{(i)} = \mathrm{softmax}(\mathbf{o}^{(i)})$

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o}) \text{ where } \hat{y}_i = \frac{\exp(o_i)}{\sum_j \exp(o_j)}$$



Vectorization for Minibatches

연산 효율을 더 높이기 위해 데이터의 Minibatch에 대한 연산을 벡터화 해보자.

차원 : d
배치 크기 : n 인 데이터들의 미니 배치 X
결과 카테고리 : q
미니 배치 feature X는 $R^{n\times d}$ 에 속한다.
가중치 W는 $R^{d\times q}$ 에 속한다.
편향 b는 R^{q} 에 속한다.

$$\mathbf{O} = \mathbf{X}\mathbf{W} + \mathbf{b}$$
$$\hat{\mathbf{Y}} = \mathrm{softmax}(\mathbf{O})$$

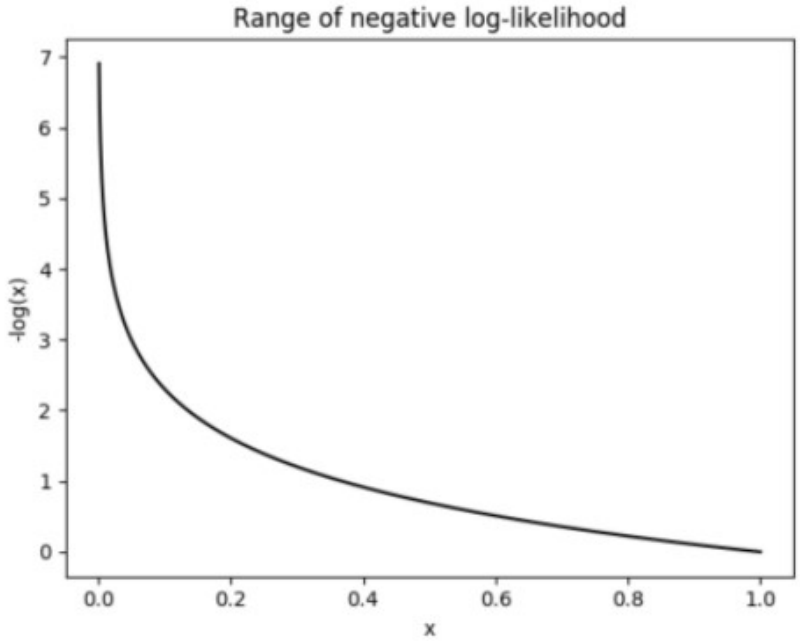
다음과 같이 정의하면 가장 많이 차지하는 연산을 가속화할 수 있다.
즉, WX가 행렬-벡터 곱에서 행렬-행렬 곱으로 변환된다.
Softmax는 결과 O에 지수 함수를 적용하고, 지수 함수들의 값의 합으로 정규화 하는것으로 계산.

Loss Function : N log_likelihood

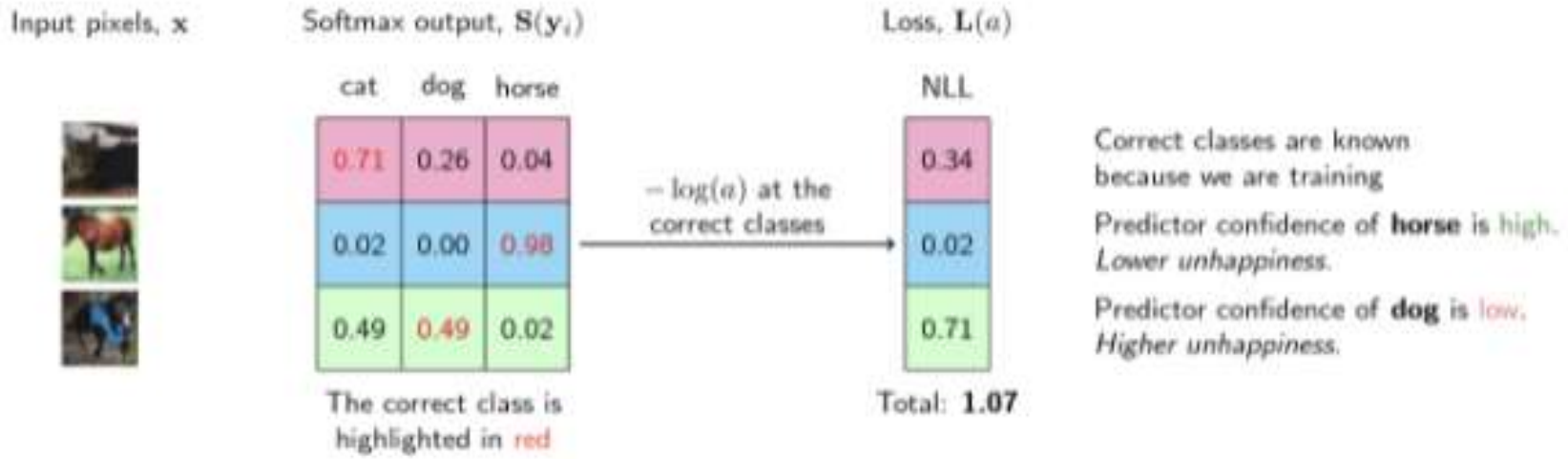
실제 softmax를 쓰게 되면 loss function은 Negative log-likelihood와 함께 사용한다.

$$-\log P(\mathbf{Y} \mid \mathbf{X}) = \sum_{i=1}^{n} -\log P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}) = \sum_{i=1}^{n} l(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)}),$$

Parameter가 주어졌을 때 loss의 최소값을 찾아야 한다.



Loss Function : N log_likelihood



Softmax and Derivatives

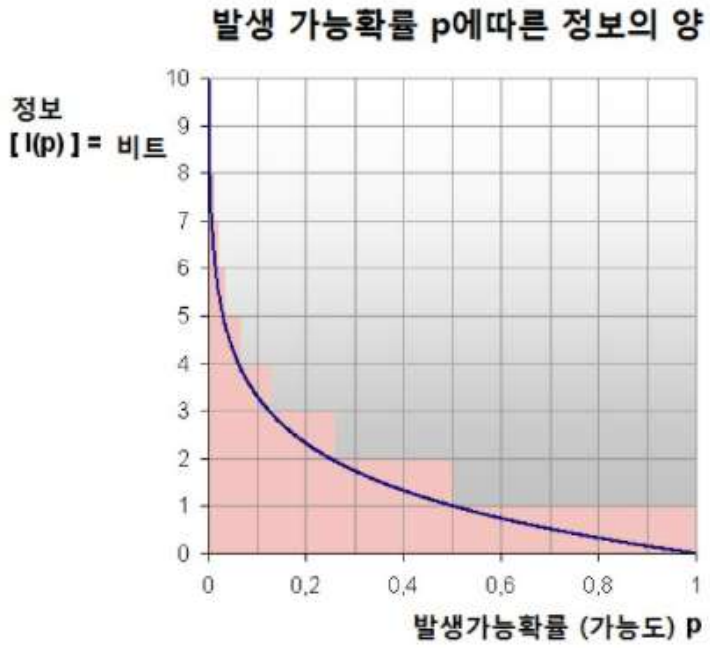
범주형 데이터를 대상으로 하는 손실함수의 기반, cross-entropy error

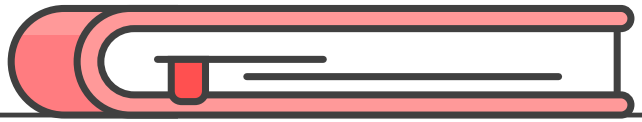
1. 정보 이론에서 entropy란?

A. 정보 이론

어떤 사람이 정보를 많이 알수록, 새롭게 알 수 있는 정보의 양이 감소한다.

흔히 볼 수 있는 사건 -> 정보량 낮음
희귀한 사전 -> 정보량 매우 큼





Softmax and Derivatives

범주형 데이터를 대상으로 하는 손실함수의 기반, cross-entropy error

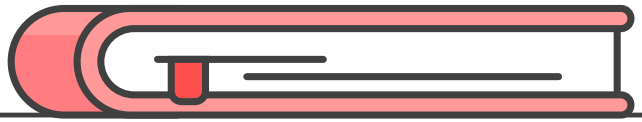
1. 정보 이론에서 entropy란?

$$H[P] = \sum_j -P(j) \log P(j).$$

B. Entropy

사건 A를 반복 실행하였을 때
얻을 수 있는 평균 정보량
또는
어떤 사건에 대한 정보량의 기댓값

엔트로피 큼 -> 사건 A의 확률이 낮음
엔트로피 = 어떤 상태에서의 불확실성
예측하기 어려움 -> 정보량 증가 -> 엔트로피 큼



Softmax and Derivatives

범주형 데이터를 대상으로 하는 손실함수의 기반, cross-entropy error

2. 교차 엔트로피 오차(Cross Entropy Error(CEE))

$$H(P,Q) = -\sum_{x} P(x) ln Q(x)$$

Q(x) : 신경망의 출력값
P(x) : 정답 레이블 -> One-hot 벡터를 사용한다.

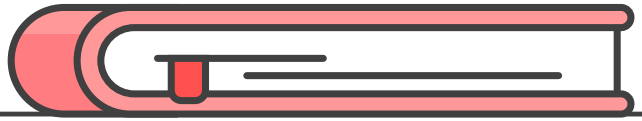
예) 분류가 4개인 데이터

$$label = [0,0,1,0]$$

$$pred = [0.1,0.2,0.6,0.1]$$

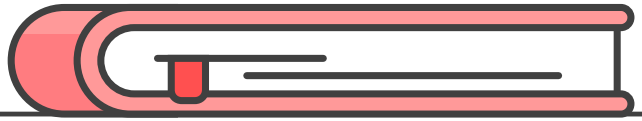
$$E = -(0 * ln(0.1) + 0 * ln(0.2) + 1 * ln(0.6) + 0 * ln(0.1)) = -ln(0.6) = 0.51$$

교차 엔트로피 오차는 특정 클래스에 속할 정보량을 이용한다는 것을 알 수 있다.
교차 엔트로피 오차 역시 정보량이 0에 가까워져 발생 확률이 1에 가깝게 만드는 것이 목적이다.

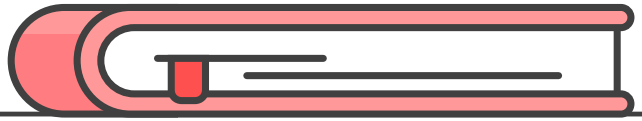


Model Prediction and Evaluation

Softmax regression model을 훈련한 뒤면 각 출력 클래스의 확률을 예측할 수 있다.
일반적으로 예측 확률이 가장 높은 클래스를 출력 클래스로 사용한다.
실제 클래스와 레이블이 일치하면 예측이 정확하다.
Model의 성능은 정확도를 사용하여 평가한다.
이것은 정확한 예측 수와 총 예측 수 사이의 비율과 같다.



구현 및 실습



끝