Machine Learning - Google Developer

라벨(label) - 예측하게될 항목(스팸O, 스팸X)

특성(feature) - 라벨을 구분하기 위한 입력들 (이메일 주소, 이메일 텍스트 등)

라벨이 있는 예 - 학습에 사용(입력: 메일001 / 라벨: 스팸O)

라벨이 없는 예 - 예측 (입력: 메일 007 / 라벨: (없음))

\*\* 일부 라벨은 신뢰할 수 없을 수도 있음!!!!

선형관계

y’ = w1x1 + b

y’ : 라벨 / x1 : 특성

특성이 세가지라면

y’ = w1x1 + w2x2 + w3x3 + b

손실은 제곱손실 많이 사용함 avg((y - y’)^2)

손실 줄이기

손실 함수가 아래로 볼록한 함수 형태이다.

실제의 경우에는 계란판과 같은 모양일 수 있지만, 현재는 볼록함수라고 가정하자.

w1 을 정하고 그 시점에서의 경사를 계산하고, 그 방향으로 경사\*학습률만큼 움직인다.(경사하강법)

경사가 2.5, 학습률이 0.01이라면 0.0025만큼 움직이고 다시 경사계산 할거다.

학습률이 너무 작으면 극소값을 찾는데 굉장히 오랜 시간이 걸릴 것이고,

학습률이 너무 크면 극소값을 넘어가고 왔다갔다 할 것이다.

효과적으로 수렴할 정도로 학습률이 크지만, 발산할 정도로 크지 않은 적당한 학습률을 구해야 함.

골디락스 학습률 : 기울기가 작다면 더 큰 학습률을 시도해볼 수 있다.

배치

손실 줄이기에서 반복하여 극소를 찾아낼텐데, 반복 과정에서 데이터가 수억개 있다면 부담될 것.

몇개만 사용해서 이를 수행할 수 있음, 이 수행에서 사용하는 예의 총 개수를 배치(batch) 라고 함.

확률적 경사하강법(SGD)는 전체 예 중에서 무작위로 하나 골라서 수행하는 것 => 노이즈 심함

미니 배치 확률적 경사하강법(mini batch SGD)는 무작위 10개~1000개 사이의 예를 사용

python

pip install tensorflow

anaconda 설치함..

설치 후

export PATH="/Users/LWJ/opt/anaconda3/bin:$PATH"

conda create -n tf python=3.8

source /Users/LWJ/opt/anaconda3/etc/profile.d/conda.sh

conda activate tf

(conda 실행되어 (tf) 로 표시됨)

(tf) pip install tensorflow

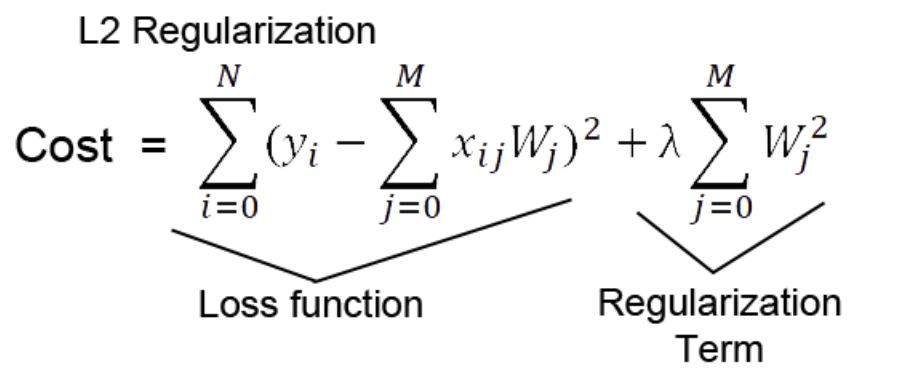
=> anaconda 굳이 필요없었음..

=> macOS 업그레이드 해야 되는거였음.

=> 에러메세지 보면 어떤거가 macOS 10.15 이상에서 빌드된거라고 하는데, 현재 macOS가 10.11(EL Capitan) 이었음, OS 업그레이드 후 anaconda 없이 정상 동작 확인, pycharm 내부 venv 사용해도 문제 없음.

L2 정규화

정규화는 과적합을 방지하는 도구



무조건 cost가 작아지는 방향으로 가야함.

w^2의 합을 추가하면서 w^2의 합이 작아지는게 왜 과적합 방지하는건지?

예를 들어 특성이 100가지가 있고, 이 중에서 10가지만 유효한 특성, 90가지는 쓰레기 특성이라고 하자.

Loss function 만 있을 때 : Loss function이 작아지는 방향으로 가기 때문에 과적합될 수 있다. (여기까지는 OK)

그런데 W^2의 합을 넣어서 이 값이 작아지도록 하는게 왜 과적합 방지냐? W값이 높은게 과적합인가? w값이 높아서 낮추려는게 아니라, 관련없는 특성의 w값을 0에 가깝게 만드는 것이 목적이다. 관련없는 특성의 w값도 높아서 이를 낮추는 것이 목적이고, 또한 관련있는 특성의 w 값들도 너무 과하게 높아졌으면 Loss function을 너무 과하게 낮추려고 너무 많이 움직인 것이기 때문에 과적합 되었을 가능성이 있다. (관련있는 특성의 w들이 과적합일수도 있고 아닐수도 있지만 어쨌든 과적합인 경우에 대해서 과적합이 아니게 만들어야 할 것, 그래서 너무 많이 진행하지 않고 적절히 끊는것과 같은 효과를 나타내도록 w값이 증가하면 낮추게 하는것이 Regularization term)

Loss function은 w 값을 조절하면서 작아지려고 노력할 것, 그 과정에서 w 값이 과적합될텐데, 10가지 특성만 제대로된 특성이고, 90가지는 제대로된 특성이 아니다. 90가지에 대해서는 w가 거의 0으로 가서 고려하지 않는 특성이 되어야한다.

하지만 Loss function만 있다면 학습 데이터에 의해 90가지 특성에 대해서도 어떻게든 분류 할거고, Loss function을 줄이는 방향으로(0.00…..001만 줄어든다고 할지라도) 적절한 w값이 선택될 것이다.

Loss function을 줄이는 것과, w^2을 줄이는 것이 균형을 이루는것이 목적인데,

90가지의 특성에 대해서는 Loss function 을 줄이는 것보다 w^2을 줄이는게 훨씬 효율적일 것이다. => 결국 w값은 거의 0으로 갈것이다.

10가지의 특성에 대해서는 Loss function 을 줄이는 것과 w^2을 줄이는 것이 균형을 이룰 것이다.

연관 없는 특성 : Loss function 낮추는 것보다 w 낮추는 것이 효율적 => w를 거의 0으로 함.

연관 있는 특성 : w 가 거의 0으로 가면 Loss function이 너무 높아질 것 => 균형을 맞춰야 함.

로지스틱 회귀

확률로 변환하여 판단하고자 할 경우, 선형 회귀의 형태라면 선형이므로 확률이 0보다 작아지는경우, 1보다 커지는 경우가 존재하게 될 수 있다. 이를 0~1 사이값으로 고정시키기 위해 로지스틱 회귀를 사용한다.

스팸을 예로 들면 선형회귀에서는 선형이므로 x가 10일때 y가 1로 100퍼센트라면 x가 11이 되면 y가 100퍼센트를 넘어가버린다. 이러한 경우 말이 안되는 상황이기 때문에 시그모이드 함수와 같은 방법을 사용하여 로지스틱 회귀를 사용한다.

이제 이 확률이 나왔을 때 스팸인지 아닌지 판단할 임계값이 필요하다.



여기에서 정확성은 전체 4칸 중 초록색 2칸 (T\* / \*)

정밀도는 양성이라고 판단했을 때 이게 얼마나 정밀한지 판단하는 것이므로 (TP / \*P)

재현율은 실제 양성이었던 것들 중에서 내가 양성이라고 집어낸 것이 얼마나 되는지 이므로 실제 양성은 TP + FN 이고, 따라서 (TP / (TP+FN))

0.01% 만 가지고 있는 희귀병 판단 시 모두 음성이라고 하면 정확도는 99.99% 이다. 따라서 정확도만으로 모델을 판단할 수 없다. 하지만 정밀도와 재현율에서는, 양성이라고 판단한 경우가 없어 TP = 0 이므로 정밀도, 재현율 모두 0이다.

임계값이 높아진다면 더 확실한 경우에만 양성이라고 집어낼 것이므로 정밀도는 올라갈것이고, 재현율은 떨어질 수 있다.

임계값이 낮아진다면 좀 더 불확실하더라도 일단 양성이라고 집어낼 것이므로 정밀도는 내려갈것이고, 재현율은 좀 더 올라갈 수 있다.

L1 정규화

L2 정규화는 관련없는 특성의 가중치를 0에 가깝게 만들지만 완전히 0으로 만들지는 못한다. (제곱이므로 미분하였을 때 항이 남아있어 비율로 감소시키기 때문에 0이되지는 않는다)

(반면 L1 정규화는 절대값이므로 미분하였을 때 상수로 되어 일정값을 감소시켜 0이 가능)

L1 정규화는 관련없는 특성의 가중치를 0으로 만들어 특성을 삭제할 수 있다.

특성곱을 사용해야 하는 경우 특성이 너무 많으면 너무 과하게 커지므로 필요없는 특성을 삭제할 필요가 있다. (x1이 10000, x2가 10000 이면 x1x2 는 1억이된다.)

주의점 - 대신 약하게나마 유용한 특성도 0으로 만들 수 있다.

신경망(Neural Network)

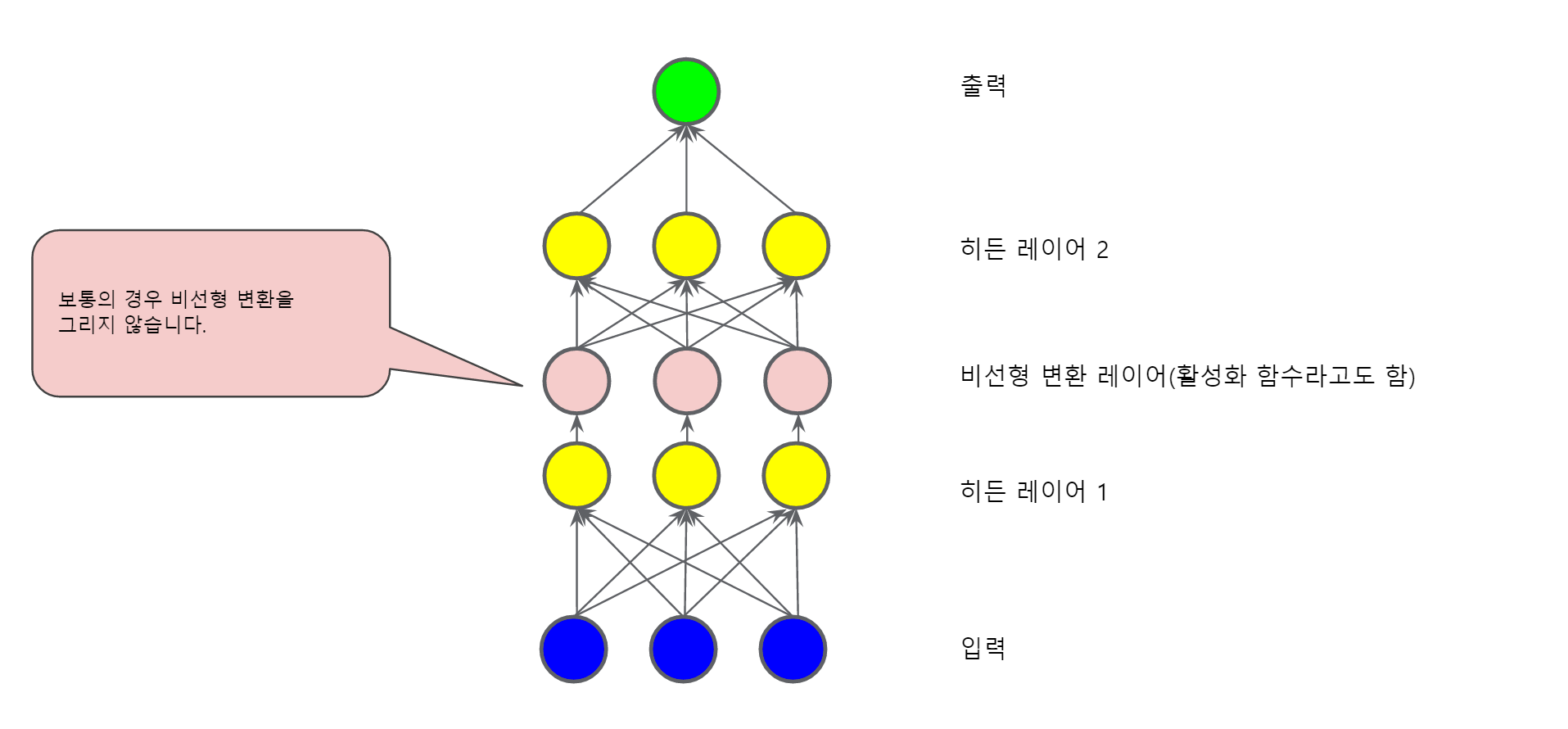
x1, x2, x1x2, x1^2, x2^2, sin(x1), sin(x2), … 여러 특성교차가 가능하다.

특성교차가 아니라 다른 방법으로 일반적인 데이터를 분류할 수 있으면 좋겠다.

x1, x2 만 가지고 할 수 없나? 비선형 문제는 선형으로 해결할 수 없다.

그치만 선형으로 여러 레이어 추가해봤자 선형인데 어떻게 할 수 있을까?

비선형성을 도입해야 한다. (ReLU, Sigmoid, tanh 등)



비선형 함수를 추가한 후에는 Layer를 추가하여 비선형성이 누적될수록 더 복잡한 모델링이 가능해진다.

|  |  |
| --- | --- |
|  |  |

소프트맥스

스팸/스팸아님 의 문제 말고

사진이라면 (개, 고양이, 사과, 바나나, 사탕) 5가지라고 하면

개OX, 고양이 OX, .. 의 노드를 하나 더 추가하면 된다. 이 확률들의 합은 1일 것이고, 이게 소프트맥스이다. 모든 가능한 클래스의 확률을 계산함

=> 사진 하나에 개와 고양이가 같이있다면 이를 사용하지 못함. (동시에 여러 클래스)

TensorFlow - 초보자 가이드

<https://www.tensorflow.org/tutorials/keras/classification>

데이터 전처리

데이터는 28\*28 pixel이고, 하나의 픽셀은 0~255 값을 가지고 있다.

이를 0~1 값으로 변환하기 위해 /255를 수행하고, 28\*28 pixel을 1차원으로 늘려 784pixel의 1차원배열로 변환한다.

그러면 x1, x2, ... , x784 로 생각할 수 있고, 각 값은 0~1 사이의 값이다.

model = keras.Sequential([

keras.layers.Flatten(input\_shape=(28, 28)), => 784pixel의 1차원배열

keras.layers.Dense(128, activation='relu'),

keras.layers.Dense(10, activation='softmax')

])

Layer를 넣을텐데, 첫 Layer는 128개의 노드이고, relu 활성화 함수를 사용한다.

마지막 Layer는 10개의 노드이고, softmax 를 사용한다. 이는 사진의 카테고리가 총 10가지이고, 그 중 하나에 속하는 확률을 구하기 때문에 확률의 합은 1이다.