Deep Learning

## DeepLearning 의 역사

1980-1990년대에 이슈가 되었지만 컴퓨터 속도 및 데이터베이스 의 작은 크기 때문에 2000년도에 들어서 가라앉았다. 그러다가 2012년 싼 GPU, 많은 데이터로 다시 부상하기 시작했다.

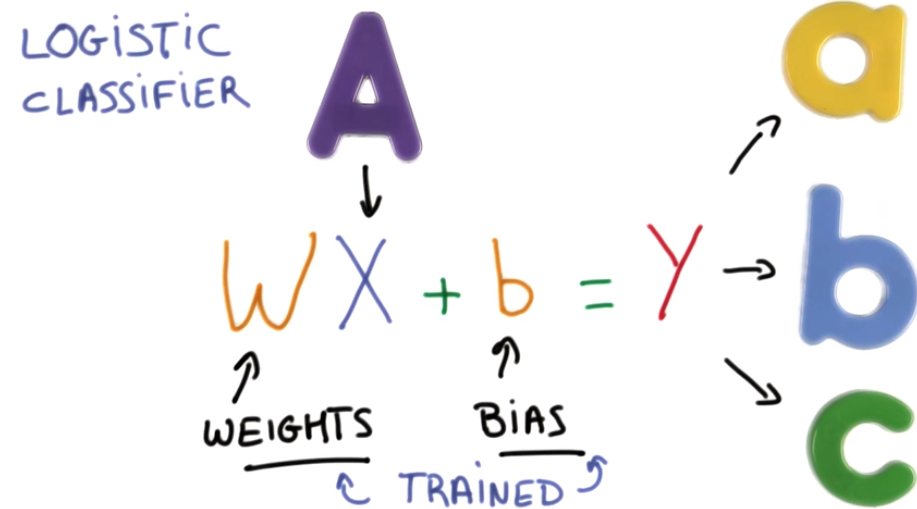
## Classification 방법

이미지의 경우 오브젝트로 잘라 원하는 이미지인지 아닌지 찾는다.

검색어 ranking의 경우 pair(query, web page)로 만든 후 관계가 있는지 없는지 확인한다.

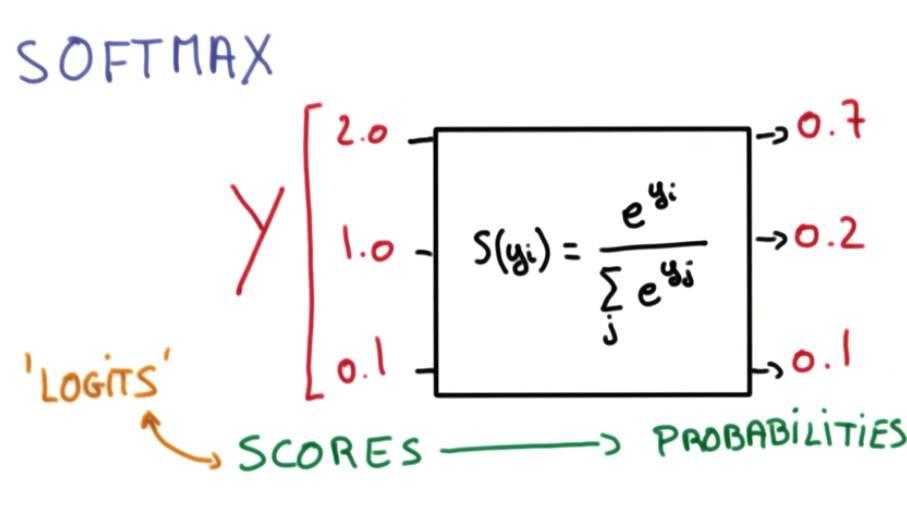
### Logistic classifier

Logistic classifier는 Linear classifier라고도 부른다. input을 받고 판단하기 위하여 Weight(Matrix)를 곱하고 조정(bias)하여 점수를 낸다. 이 점수로 확률을 만들어 원하는 클래스는 확률을 1로 다른 클래스는 0에 가깝게 만든다.



### Softmax function

Softmax function을 이용하여 점수를 확률로 변환시키는데 여러 종류의 점수를 받을 수 있고 그 점수를 적절한 확률로 변환 시킨다. SoftMax 방법은 각 요소(를 각 요소 ()로 각 합친 값으로 나누어 확률을 구한다. 만약 output(score)값이 크면 확률 차이가 크고 output값이 작으면 uniform해진다.

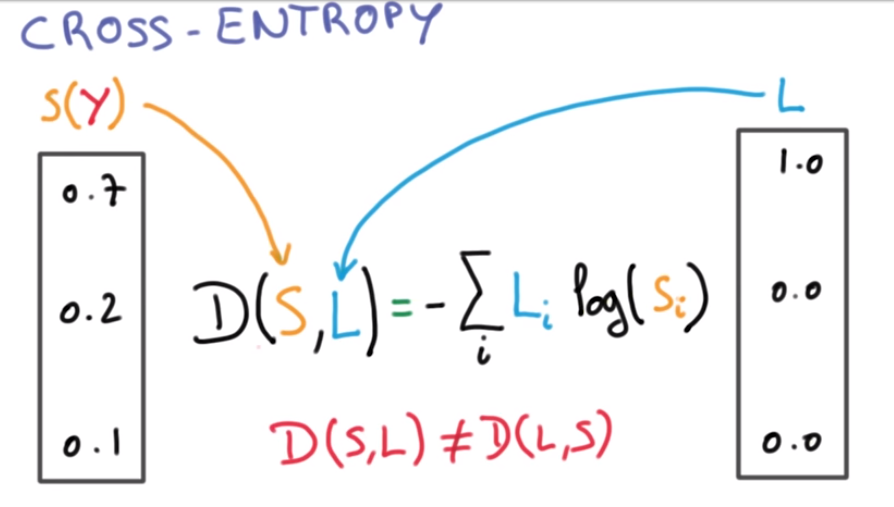


### One-hot encoding

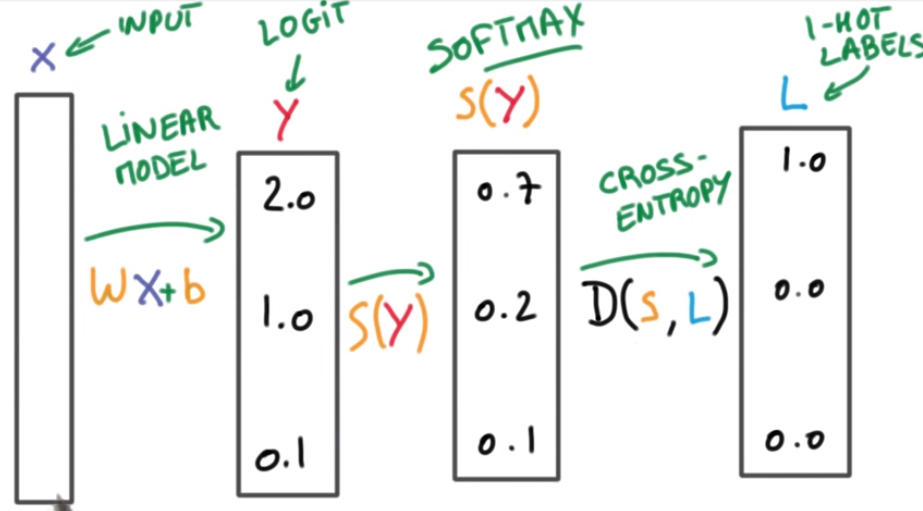
One-hot encoding은 카테고리(클래스)마다 하나만 1 나머진 0 인 vector로 만들어 클래스를 찾을 때 사용한다. 클래스가 많으면 사용하기 힘들다.

### Cross-Entropy

Cross-Entropy는 One-hot Label이랑 SoftMax를 통하여 나온 확률과 얼마나 관계있는지를 구하는 식이다. D는 distance로 correct class면 distance가 가깝고 wrong class면 distance가 멀다



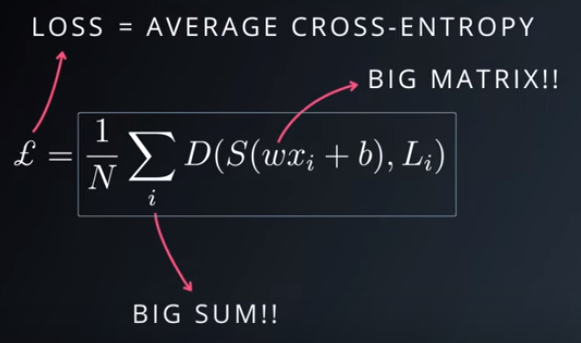
모든 과정을 나타내면 다음과 같다.



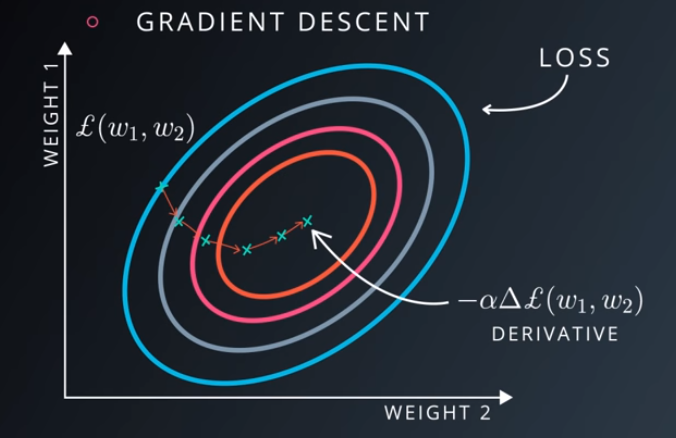
D(S(wx + b), L)

### LOSS

LOSS는 Average Cross-Entropy를 의미한다. 모든 input의 training set, label에 대하여 수행하여 구하는데. 이 LOSS를 줄여야 한다.



가장 간단하게는 Gradient Descent방법으로 줄일 수 있다.



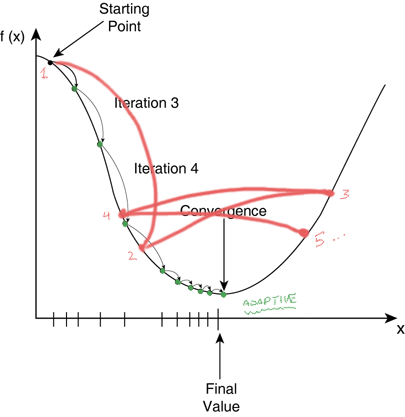
Normalized Inputs는 input을 계산하기 쉽고 범위가 크지 않게 바꾸는 것이다. 예를 들어 이미지의 경우 픽셀당 (256,256,256)의 값을 가지고 있을 것인데 (pixels - 128) / 128 해서 Normalize 시켜주는 것이다. initial weights는 Gaussian distribution에서 랜덤하게 고르는데 값에 따라 SoftMax값이 차이가 나므로 적절한 값을 선택하기 위해 작은 값 부터 고르자.

### dataset

데이터셋(dataset)을 Training set, Validation set, Test set으로 데이터를 나누어 학습한다. Training set만 사용한다면 학습을 적게 했는지(Under Fitting) 아니면 많이 했는지(Over Fitting)알 방법이 없다. Under Fitting의 경우 학습이 덜 되서 제대로 찾지 못하는 경우이고 반대로 Over Fitting의 경우 training set은 잘 분류하나 새로운 데이터가 오면 제 기능을 못하는 경우이다. 그래서 Validation을 하면서 적절히 학습한 정도를 찾아야 한다. Test Set은 본 적 없는 새로운 데이터로 수행한다.

### Learning rate

Learning rate는 기계학습을 할 때 LOSS를 줄이기 위해서 Gradient Descent하는데 값이 수렴되는 정도를 나타낸 것이다. 만약 Learning rate가 높다면 수렴되지 않고 발산될것이다. 반면에 너무 낮으면 학습을 많이 반복해도 줄어들다가 말 것이다. 그래서 적절한 Learning rate가 필요하다.

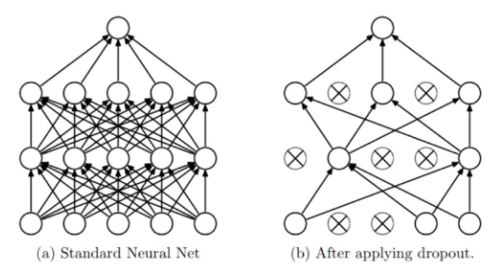
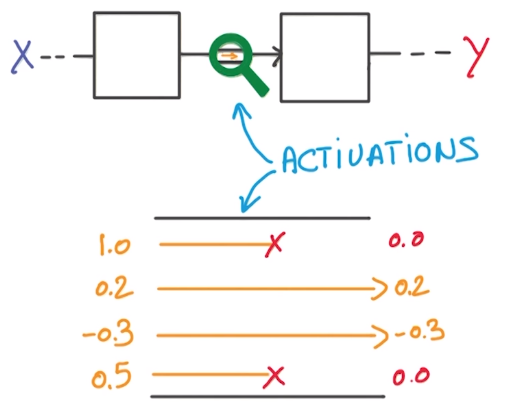


### Stochastic Gradient Descent

Stochastic Gradient Descent는 Gradient Descent를 이용하여 LOSS를 구하기에는 비용(시간)이 너무 커 임의로(Stochastic) Mini-batch라는 데이터의 일부분만을 사용하여 LOSS를 구한 뒤 전체 데이터에 적용시키는 방법이다

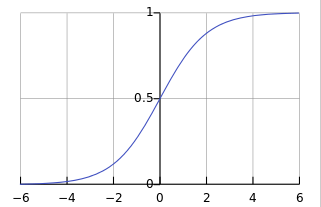
### Dropout

Dropout은over fitting을 막는 regularization 방법으로 한 레이어에서 다음 레이어로 넘어가는 것을 activations(결과에 효과를 주기 때문에)라고 부르는데 이 때 랜덤하게 0으로 만들어버린다.



### Sigmoid

Sigmoid는 activations에서 값이 엄청 크거나 작을 때 미치는 영향을 줄이기 위해서 0~1사이의 값으로 바꾸어 주는 함수이다 생긴 모습은 다음과 같다

.

### ReLU

ReLU는 Sigmoid를 사용할 경우 Back Propagation을 할때 미분을 하게되면 0에 아주 가까운수 0.01 같은 수가 나올 수 있는데 deep learning이니까 이 0.01이 여러번 나오게 된다. 결국 0.01 \* 0.01 \* 0.01 --- 를 하게 되어 기울기가 매우작아지는 현상이 발생한다. 그래서 ReLU를 사용한다. 0보다 작으면 0, 0보다 크면 그 수를 취하는 방식이다. 생긴건 다음과 같다. 마지막은 0~1사이의 수가 나와야 하므로 sigmoid를 취한다.

