**1. Overfitting의 정의**

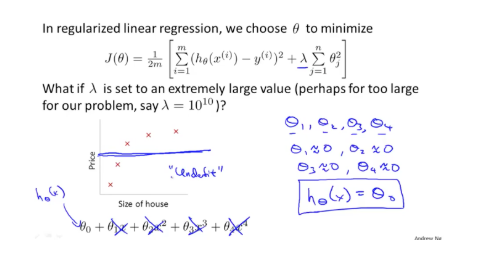
🡺 If we have too many features, the learned hypothesis may fit the training set very well, but fail to generalize to new examples (predict new examples)

* 너무 과하게 최적화가 되는 바람에 일반화가 어려워진 경우를 의미함. 여기서 일반화란, 학습에 사용된 데이터 외에 다른 새로운 데이터를 넣었을 때도 모델이 좋은 결과를 내는 것을 의미함
* 일반적으로 Machine Learning의 결과로 나타내지는 그래프는 right case에 “최근사” 그래프이기 때문

**2-1. Regularization**

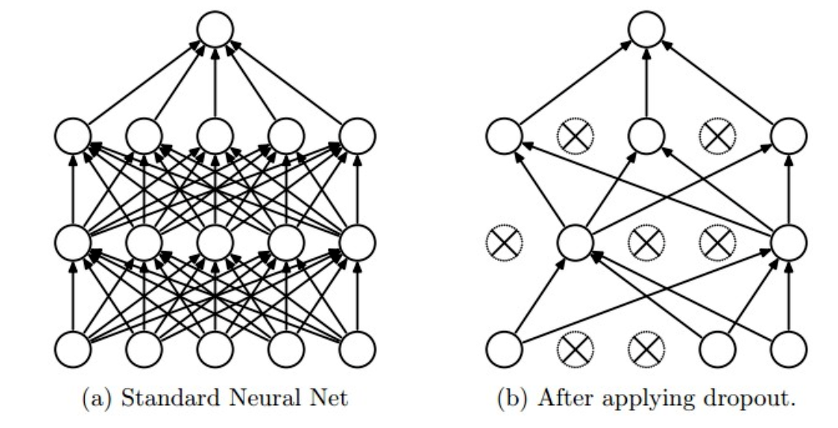
1) Add term to loss

정의한 Loss function에 “직접” 파라미터를 추가하여 overfitting을 막는 방법. 다른 방법들 과 구별되는 특징으로 유일하게 function에 직접적인 조작을 가하는 방법임. 추가하는 term의 개수에는 제약이 없으나 보통 1~2개 정도를 추가하며, Multi-model의 경우 3개 이상의 term을 추가하기도 함.

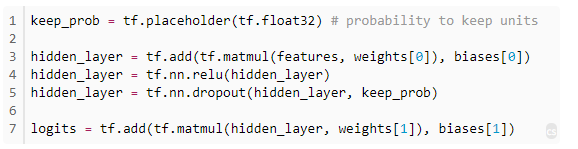


2) Dropout

In each forward pass에서 뉴런의 집합을 random하게 0값을 주어서 활성화를 시키지 않는 방법으로 이는 overfitting의 원인이 “too many features”이기 때문에 이를 해결하기 위한 방법이기도 함. 대부분 0.5를 사용하며, 경우에 따라서 그 숫자를 조절함.



여기서 주의할 점은 Tensorflow(아마도 2.0 버전 이전의 버전)와 다른 언어에서 Dropout의 패러미터가 가지는 의미가 다르다는 점이다.



예를 들자면 위에서 tf.nn.dropout은 dropout을 적용할 텐서와 keep\_prob(drop을 하지 않을 확률) 두 가지를 인자로 받는다. 이때 drop된 유닛을 보상하기 위해 tf.nn.dropout() 함수는 drop되지 않은 모든 유닛에 1/keep\_prob을 곱하게 된다.   
다시 말해 Tensorflow에서는 만약 Keep\_prob의 값이 0.1이라면, 10%의 입력 요소들이 drop될 것이라는 것을 의미한다. (2.0 버전 기준)

하지만 2.0버전 이전에 Tensorflow에서 저 비율이 의미하는 것은 10%의 입력 요소들만 사용될 것이라는 의미로서, 90%의 입력 요소들이 drop될 것이라는 것을 의미하는 것이다.

하지만 다른 언어(ex. Keras)에서는 만약 그 비율이 20%라면 5개의 입력 데이터 중 1의 입력 데이터가 each update cycle에서 제외된다는 것이다.

3) A common pattern

데이터 셋에서 비슷한 특징들이 발견될 시에 그것들을 이용하는 방법으로 그 예로는

Dropout, Batch Normalization, Data Augmentation, DropConnect, Fractional Max Pooling이 있다

그 중에서

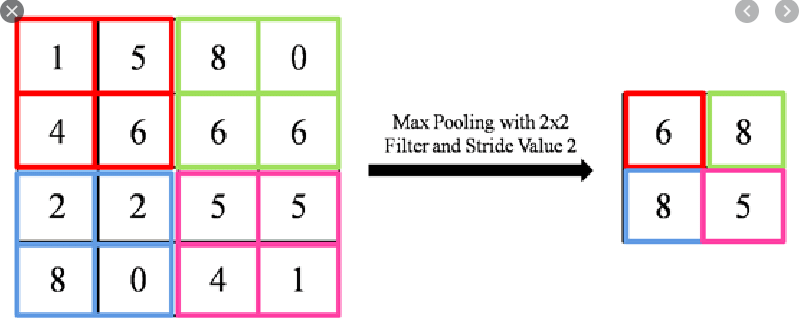
-1) Data Augmentation

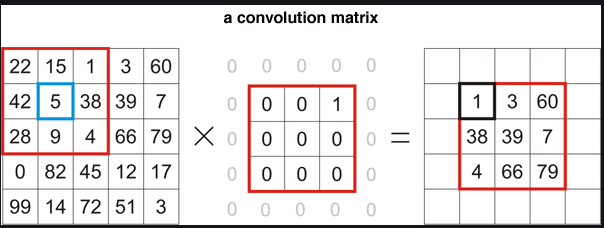
Data Augmentation은 말 그대로 데이터를 회전, 반전, 색깔 변화 등을 주어서 데이터 양을 늘리는 방법이다. 그런데, Overfitting의 정의를 보면 너무 많은 feature로 인하여 Model이 과최적화 된 것이라고 하였다. 그렇기 때문에 Data Augmentation이 Overfitting을 방지한다는 것이 언뜻 이해가 가지 않을 수가 있다. 예를 들어 말 사진을 구분하는 Model을 만든다고 가정했을 때, 만약 모든 말의 머리가 오른쪽으로 향한 사진을 가지고 학습을 진행하면 말머리가 왼쪽을 향하는 사진의 경우 제대로 classification을 못하는 경우가 발생하게 된다. 이 경우 Data Augmentation을 통해 말 머리가 왼쪽을 향하는 사진을 만들게 되면 이러한 문제를 해결할 수가 있게 된다. 이러한 의미에서 Data Augmentation이 Overfitting을 해결하는 방법이 되는 것이다. **Overfitting은 “특정 Data에 대하여 과 최적화되어 있는 상태”**이기 때문이다.

-2) Fractional Max Pooling

Fractional Max Pooling은 Max Pooling을 위한 pooling size를 전부 다르게 주는 방법을 의미한다.

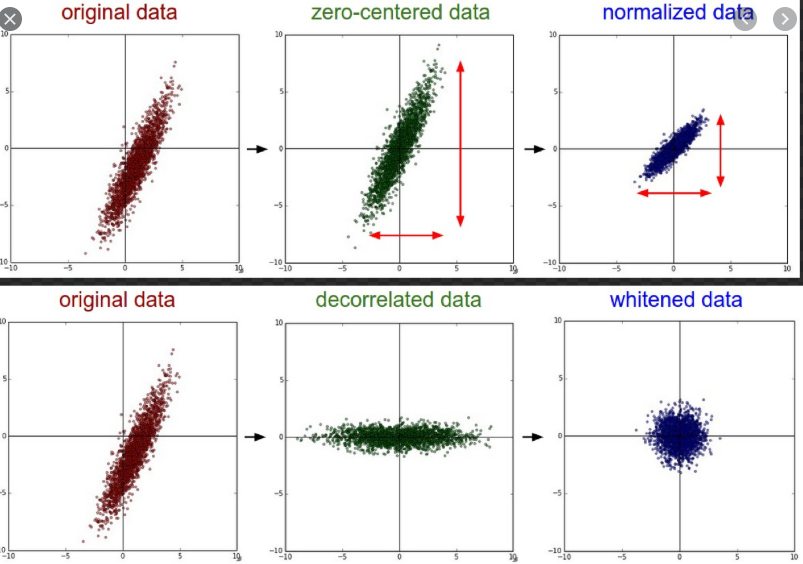
예를 들자면 First Max Pooling layer에서 size를 2x2로 하였다면 다음 Max Pooling layer에서 그 size를 예를 들어 2x4 혹은 4x4와 같은 방식으로 변화를 주는 방법이다. 이는 아래와 같이 (첫 번째 그림은 2x2 Pooling size, 두 번째 그림은 3x3 pooling size의 예시이다) Pooling size에 따라서 실제로는 하나의 데이터지만, 마치 여러 개의 데이터를 가지고 훈련을 하는 것과 같은 효과를 누릴 수 있다. 이렇게 할 시 **“특정 Data에 대하여 과 최적화되어 있는 상태”**를 완화시켜 줄 수 있기 때문에 이 또한 Overfitting을 방지하는 방법이 된다.





**2-2. Normalization**

Normalization은 data preparation에 주로 사용되는 방법이며, overfitting을 막아주는 방법 중에 하나이기도 하다. Normalization의 효과는 아래의 그림의 위에서 3번째 그림을 통해서도 알 수가 있다. Normalization은 Regularization 과는 다르게 Hyperparameter 및 add function term을 이용한 것이 아닌 Data를 정제해서 Overfitting을 막는 다는 점에서 차이점을 가지고 있다.



정규화의 목표는 값 범위의 차이를 왜곡하지 않고 데이터 집합의 숫자 열 값을 공통 척도로 변경하는 것이다. (The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values).

하지만 모든 데이터 셋이 Normalization을 필요로 하는 것은 아니다. Machine learning에서 만약 data set의 feature들이 다른 범위를 가질 때만 Normalization이 필요하다.

예를 들자면 어떠한 데이터 셋이 2가지 feature (age, income)를 가지고 있다고 가정해보자.

이때 age의 경우 0~100의 값을 가질 것이며, income의 경우 0~20,000 혹은 20,000 이상의 값을 가지게 될 것이다. 이와 같이 그 범위의 차이가 유의미하게 클 경우에만 Normalization이 필요로 하게 된다. 하지만 이것은 이론적이 이야기일 뿐, 실제로는 Normalization을 언제 사용해야 하는 지에 대한 논의는 아직도 계속되고 있는 상태이다. 하나의 예로 batch Normalization이 그 예라고 볼 수 있다. Batch Normalization은 mini-batch size 별로 normalization을 하는 방법인데, 이러한 방법은 range가 크게 차이가 안 나더라도 사용하는 경우가 있기 때문이다.

<https://mlexplained.com/2018/11/30/an-overview-of-normalization-methods-in-deep-learning/>

위의 링크에서 weigh normalization, group normalization, instance normalization, layer normalization에 대한 설명이 매우 잘 되어 있다. 간단하게 설명하자면 normalization을 무엇을 기준으로 하느냐에 따라 차이를 보인다고 보면 된다.

**2-3. Standardization (Z-score Normalization)**

Normalization과 Standardization의 차이점은Normalization은 입력 데이터들을 [0,1]의 범위로 rescale 시키는 것이고, Standardization의 경우 평균 =0, 표준편차가 1로 rescale을 시키는 것이라고 보면 된다. 그 외에는 거의 비슷하다고 생각하면 된다.