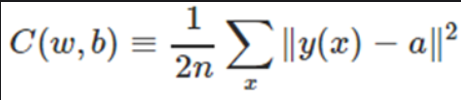
고급 소프트웨어 실습I 9주차 과제

1. Convolutional Neural Network의 Backpropagation에 대해 설명하시오. (1 page)

이전의 보고서에서, cnn은 필터가 입력데이터를 슬라이딩하면서 지역적 특징(feature)을 추출해내고, 이를 최대값(**Max Pooling**)이나 평균값(**Average Pooling**)으로 압축해 다음 레이어로 보내는 작업을 반복하여, 원하는 결과를 나타내는 것이라고 설명하였다.

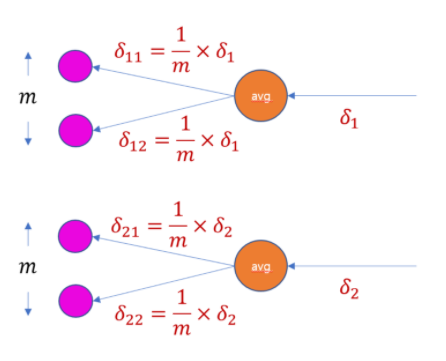
위의 과정을 통해 특징맵을 얻어내고, 활성화 함수인 relu(단, 실습에서는 sigmoid를 사용하였다)를 통하여 그 값을 활성화시킨다. 이 때, 신경망은 깊어질수록, 또는 input이 증가할 수록 가중치(w)나 바이어스(b)가 증가할 것이며, 이는 학습의 난이도를 증가시키는 결과를 가져온다. 따라서 이 두 개의 값을 잘 조절하여, cost function을 줄일 필요가 있을 것이다. 그러기 위해서 탄생한 것이 바로 **역전파(backpropagation)로,** 전체적으로 한 번 학습을 진행한 후 다시 한 번 그 계산된 값을 거꾸로 계산하는 과정이다.

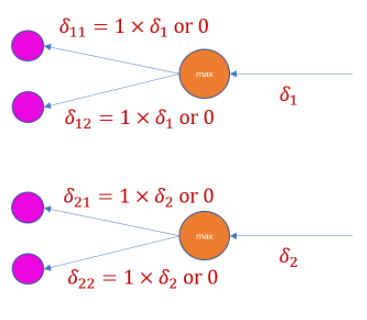


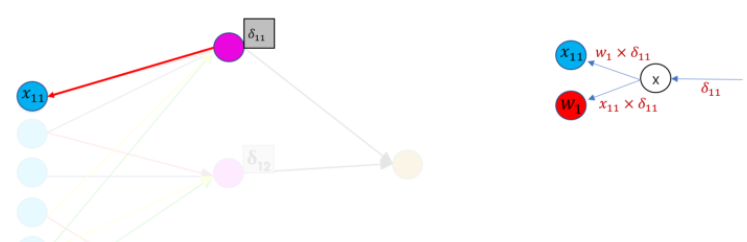
이 때 cost function은 위와 같이 표현되며, n은 입력의 수, y(x)는 입력 x를 통해 기대되는 출력, a는 x를 입력했을 때의 실제 출력이다.

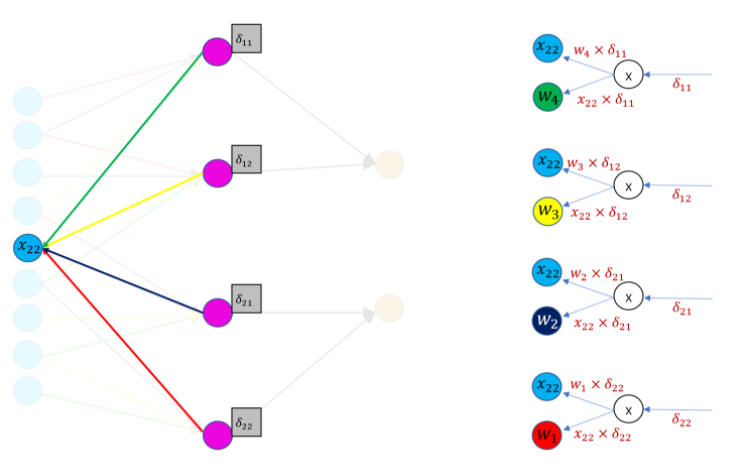
역전파를 구하는 과정은 말로 하면 간단하다. 가중치나 바이어스를 각각 편미분을 시키면, 출력 쪽에서는 아주 미세한 변화가 생길 것이다. 그리고 이러한 미세한 변화의 구간은 선형적이며, cnn의 cost function은 w와 b의 값과 직결되니, **출력에서 생긴 오차를 반대(입력)쪽으로 전파시키면서, cost function에 대한 편미분을 구하고 w와 b값을 갱신해나가면** cnn의 cost fucntion을 효과적으로 줄일 수 있을 것이다.

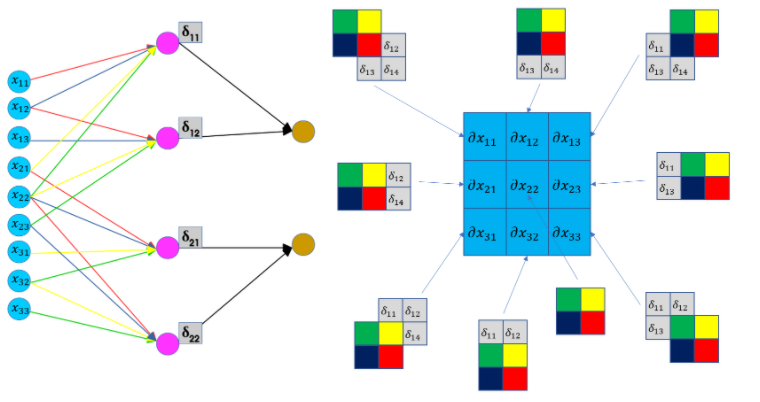
W와 b는 현재의 국소적인 정보(local information)을 토대로, 가장 기울기가 큰 방향을 향해 내려가 결과적으로 최소값에 다다르는 gradient descent라는 방법을 통해 cost function을 최소로 다다르게 한다. 이러한 역전파의 과정은, 아래와 같이 쉽게 설명이 가능하다.

현재 지점의 그래디언트는 chain rule에 의해 오게 된 gradient에 local-gradient(=상대방의 변화량)를 곱한 값과 같다. 평균은 모든 값을 더한 값에 개체의 수를 나눈 값이니, m개의 요소로 구성되어 있는 상황에서 average pooling을 한다면 그 지점에서의 local-gradient는 1/m이 될 것이다. 여기에 chain rule에 의해 넘겨받은 gradient인 d1과 곱하면, d11의 계산이 가능하다. 이와 같은 방식으로 d12, d21, d22를 구하는 것이 가능하다.

max pooling에서의 역전파 과정은, 우선 최대값이 속한 요소의 그래디언트는 1, 나머지는 0이므로, 위와 같은 계산식으로 구할 수 있다.

이를 토대로 conv layer의 역전파를 구해보자. 우선 x11은, 오른쪽에서 보다시피 2\*2필터의 붉은 가중치(w1)하고만 합성곱을 수행하였다. 따라서, 역전파도 단 한 번만 일어나게 된다. 이 때, x11의 gradient는 chain rule에 의해 들어온 gradient인 d11에 local-gradient인 w1을 곱하는 것으로 구할 수 있다. 만일 특정 노드가 여러 개의 가중치하고 합성곱을 수행했을 경우엔, 해당하는 gradient와 전부 계산을 수행하면 된다.

여기에서 x22는 3\*3의 input feature map중 (2,2)이며, 따라서 2\*2필터를 총 4번 거친다. 따라서 총 4번 연결되어 있음을 알 수 있다.

이를 간단히 도식화하면 다음과 같다. 2\*2크기의 gradienat행렬을, conv layer을 만들 때 사용한 필터가 반대방향으로 슬라이딩을 하면서 값을 구하는 구조이다. 가령, 빨-파-노-초 필터는 초-노-파-빨의 순으로 바꿔 그래디언트 행렬에 합성곡을 수행하면, 입력벡터에 대한 gradient를 구하는 것이 가능하다. 예를 들어, x11의 gradient는 w1\*d11로, 위에서 계산한 것과 일치한다.

이 때, 필터의 gradient는 chain rule에 의해 생성된 gradient(d11, d12, d21, d22)에 local-gradient를 곱해서 구할 수 있는데, 이 각각의 local-gradient는 합성곱의 필터의 가중치로 연결된 값이므로, dw11은 x11d11+x12d12+x21d21+x22d22로 나오게 된다.

이러한 작업을 함으로써 역전파를 구하는 것이 가능하다.