[Week 5] 1. Deep Layer Neural-network

Deep Model?

더 많은 은닉층이 있는 신경망을 말한다. 얕은 모델은 할 수 없는 학습들은 깊은 신경망을 사용하면 학습이 가능하다.

Deep Neural-Network notation

- L : num of layers
- n^[l] : num of units in layer l(l번째 레이어에 있는 유닛 수)

  **인덱스 값 주의 : 처음 input은 0번째 레이어로 취급한다.
- a^[l] : activations in layer l

  ** l번째 레이어에서의 활성값. g(z^[l])을 계산한 값이다. a^[0] = X, a^[L] = ^y(예측값).
- w^[l] : 가중치
- b^[l] : bias

Why Deep Representations?

왜 깊은 신경망이 더 잘 작동할까?

- 직관 1: 네트워크가 더 깊어 질 수록, 더 많은 특징을 잡아낼 수가 있습니다. 낮은 층에서는 간단한 특징을 찾아내고, 깊은 층에서는 탐지된 간단한 것들을 함께 모아 복잡한 특징을 찾아낼 수 있습니다.
- 아래의 예시에서 약 20개의 은닉층이 이 이미지를 어떻게 계산하는지 살펴보자.

  얼굴을 인식하는 신경망을 만든다고 했을 때, 심층신경망의 첫번째 층은 feature나 edge detector의 역할을 할 수 있을 것이다. 우리는 이러한 사진들을 보고 모서리가 어디에 있는지 파악한다. 모서리를 형성하기 위해 픽셀을 그룹화하여 사진에서 모서리가 어디에 있는지 알아본다. 그 후 감지된 모서리와 그룹화된 모서리를 받아서 얼굴의 일부를 형성한다.

  예를 들어, 어떤 뉴런에서는 눈의 일부를 찾고, 어떤 뉴런에서는 코의 일부를 찾아내게 되는 것이다. 이러한 많은 모서리를 모아 얼굴의 일부를 감지하는 것이다. 그리고 최종적으로 서로 다른 얼굴의 일부(눈,코,입,)를 한데 모아서 서로 다른 얼굴을 감지할 수 있게 된다.

  초기의 레이어에서는 모서리와 같은 **간단한 함수**를 감지하게 되고, 그 이후의 신경망의 층에서 이것들을 조합하여 더 **복잡한 함수**를 학습한다. 또한 초기의 레이어에서는 이미지에서 상대적으로 작은(단순한) 영역을 보게 되고, 점차 레이어가 깊어질수록 얼굴과 같은 큰(복잡한) 영역을 모으고 찾게 된다.

  이러한 원리는 사진인식에만 적용되는 것이 아니다. 음성인식에서도 적용될 수 있다. 음성을 인식할 때, 초기의 레이어에서는 낮은 단계의 음성 파형의 특징을 탐지하고(톤이 높아지고/낮아지는 것, 백색소음 등), 음소를 학습하고, 그 다음 단어를 인식하고, 최종적으로 단어들을 구성해서 구나 문장을 인식한다.

  따라서 심층신경망에서는 초기의 레이어에서는 간단한 특징을 학습하게 되고, 간단하게 탐지된 영역들을 모아 깊은 레이어에서는 더 복잡한 것들을 탐지하게 된다. (사람의 뇌도 같은 방식으로 작동한다고 한다)

- 직관 2: 순환 이론에서 따르면, 상대적으로 은닉층의 개수가 작지만 깊은 심층 신경망에서 계산할 수 있는 함수가 있습니다. 그러나 얕은 네트워크로 같은 함수를 계산하려고 하면, 즉 충분한 은닉층이 없다면 기하급수적으로 많은 은닉 유닛이 계산에 필요하게 될 것입니다.
  - Circuit Theory

    얕은 네트워크 = 은닉층이 적음 ⇒ 하나의 레이어에 더 많은 hidden unit의 개수를 필요로 함.

What does Deep Learning have to do with the brain?

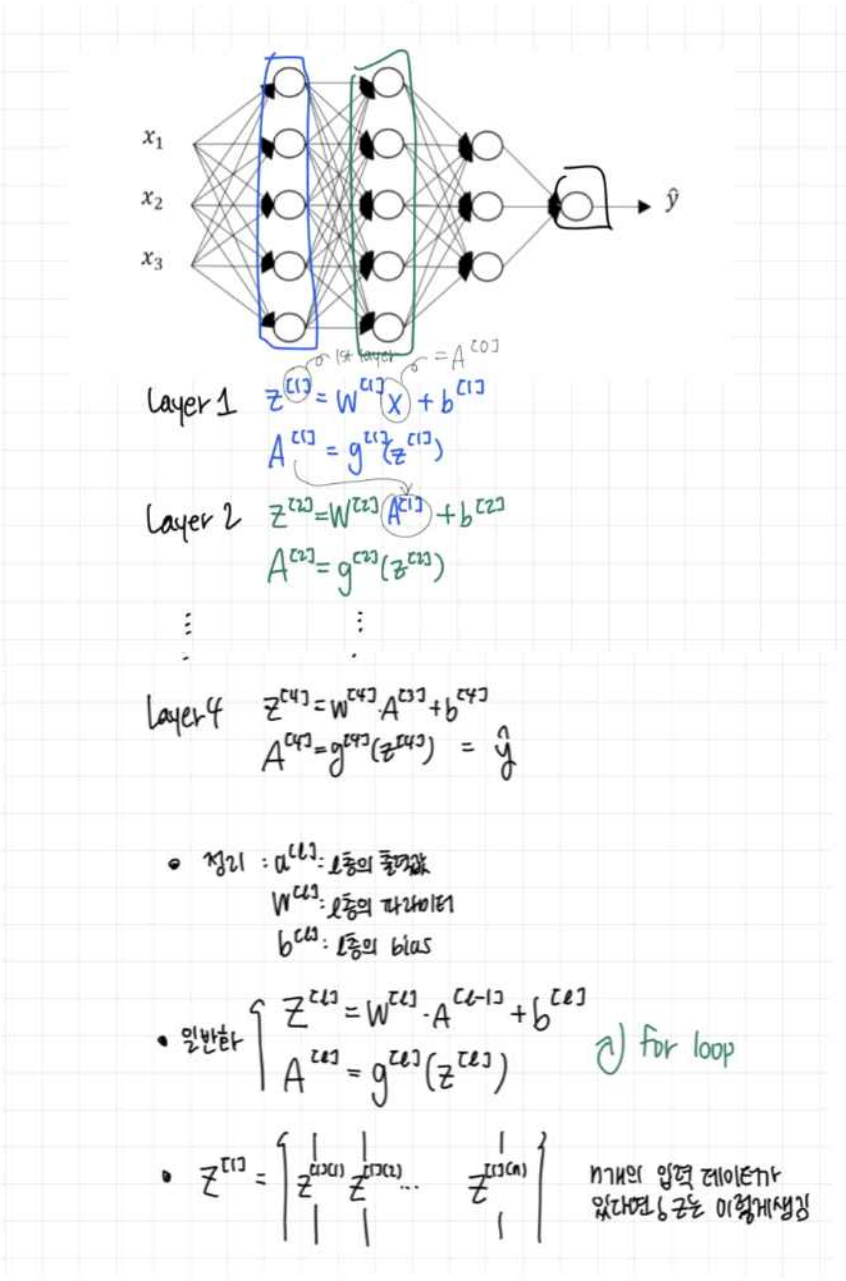
$$
\begin{aligned}
Z^{[1]} &= W^{[1]}X + b^{[1]} \\
A^{[1]} &= g^{[1]}(Z^{[1]}) \\
Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\
A^{[2]} &= g^{[2]}(Z^{[2]}) \\
&\vdots \\
A^{[L]} &= g^{[L]}(Z^{[L]}) = \hat{Y}
\end{aligned}
$$

$$
\begin{aligned}
dZ^{[L]} &= A^{[L]} - Y \\
dW^{[L]} &= \frac{1}{m} dZ^{[L]} A^{[L]^T} \\
db^{[L]} &= \frac{1}{m} np.\text{sum}(\text{d}Z^{[L]}, axis = 1, keepdims = True) \\
dZ^{[L-1]} &= dW^{[L]^T} dZ^{[L]} g'^{[L]}(Z^{[L-1]}) \\
&\vdots \\
dZ^{[1]} &= dW^{[L]^T} dZ^{[2]} g'^{[1]}(Z^{[1]}) \\
dW^{[1]} &= \frac{1}{m} dZ^{[1]} A^{[1]^T} \\
db^{[1]} &= \frac{1}{m} np.\text{sum}(\text{d}Z^{[1]}, axis = 1, keepdims = True)
\end{aligned}
$$

- 신경망과 인간의 뇌 간의 관계는 크지 않다.
- 오늘날 신경 과학자들조차도 하나의 뉴런이 무엇을 하는지 거의 모릅니다. 우리가 신경과학에서 특징짓는 것보다 하나의 뉴런은 훨씬 더 복잡하고 알기 어렵다. 뉴런이 신경망 처럼 역전파를 통해서 학습 하는지도 의문이다. 따라서 이런 비유가 최근에는 점점 무너져 가고 있다.

[Week 5] 2. Forward Propagation in a Deep Network

심층 신경망에서 정방향 전파가 어떻게 이루어지는지 알아보자. 전체 데이터셋에 대해 학습을 한번에 수행하도록 벡터화하여 수식으로 나타내보면,

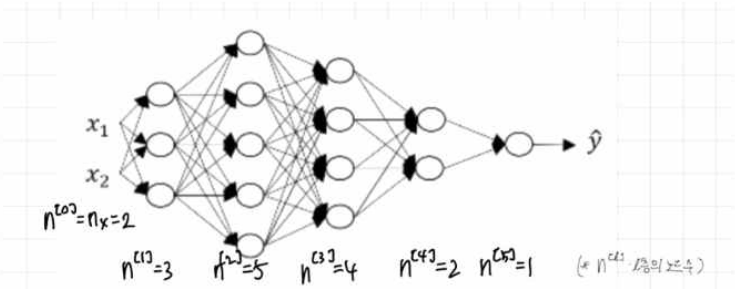


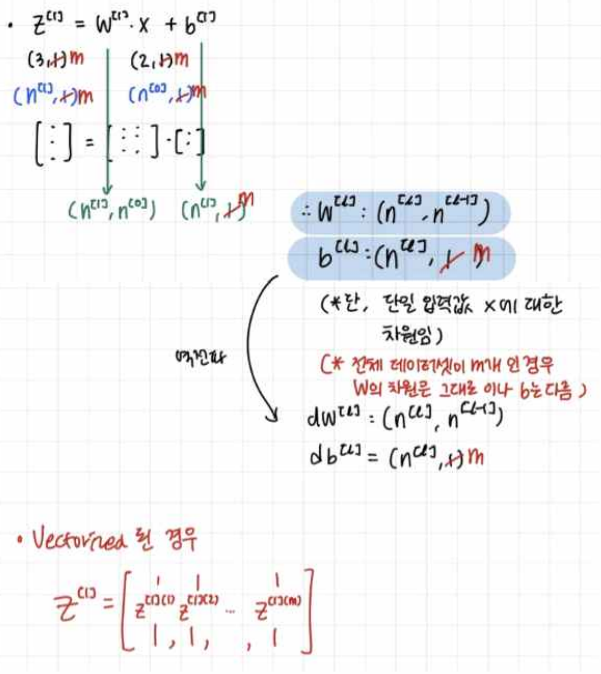
- X처럼 모든 학습 데이터는 왼쪽에서 오른쪽으로 저장된 열벡터의 형태이다.
- 각 레이어에 대해 활성값을 계산하는 과정에서는 반복문이 불가피하다.
- 심층 신경망을 구현할 때 에러를 최소화하려면 작업하는 행렬 차원에 대해 완벽히 이해하고 구현해야 한다.

[Week 5] 3. Matrix Demensions

L: 5 (hidden layers : 4, output layer : 1)

bias 는 무시하고 매개변수 W에 집중해 보자. Z,W,X의 차원을 생각해보자. 여기서 빨간색으로 표시한 부분은 전체 데이터셋에 대해 벡터화를 하였을 때의 차원이다.





• Vectorized 된 경우

$$Z^{[1]} = \begin{bmatrix} | & | & & | \\ z^{[1](1)} & z^{[1](2)} & \cdots & z^{[1](m)} \\ | & | & & | \end{bmatrix}$$

*행렬 미분을 수행할 때 차원이 바뀌나?(행렬곱이 없다면 바뀌지 않음)

하나의 행렬에 대한 미분을 수행할 때, 행렬 내 각 요소에 대한 미분이 이루어집니다. 이 경우에도 일반적으로 미분의 결과로서 동일한 크기의 행렬이 생성됩니다. 다만 각 요소별로 미분이 이루어지므로, 각 요소의 변화율을 표현하는 행렬이 생성됩니다.

예를 들어, 다음과 같은 행렬 함수가 있다고 가정해봅시다:

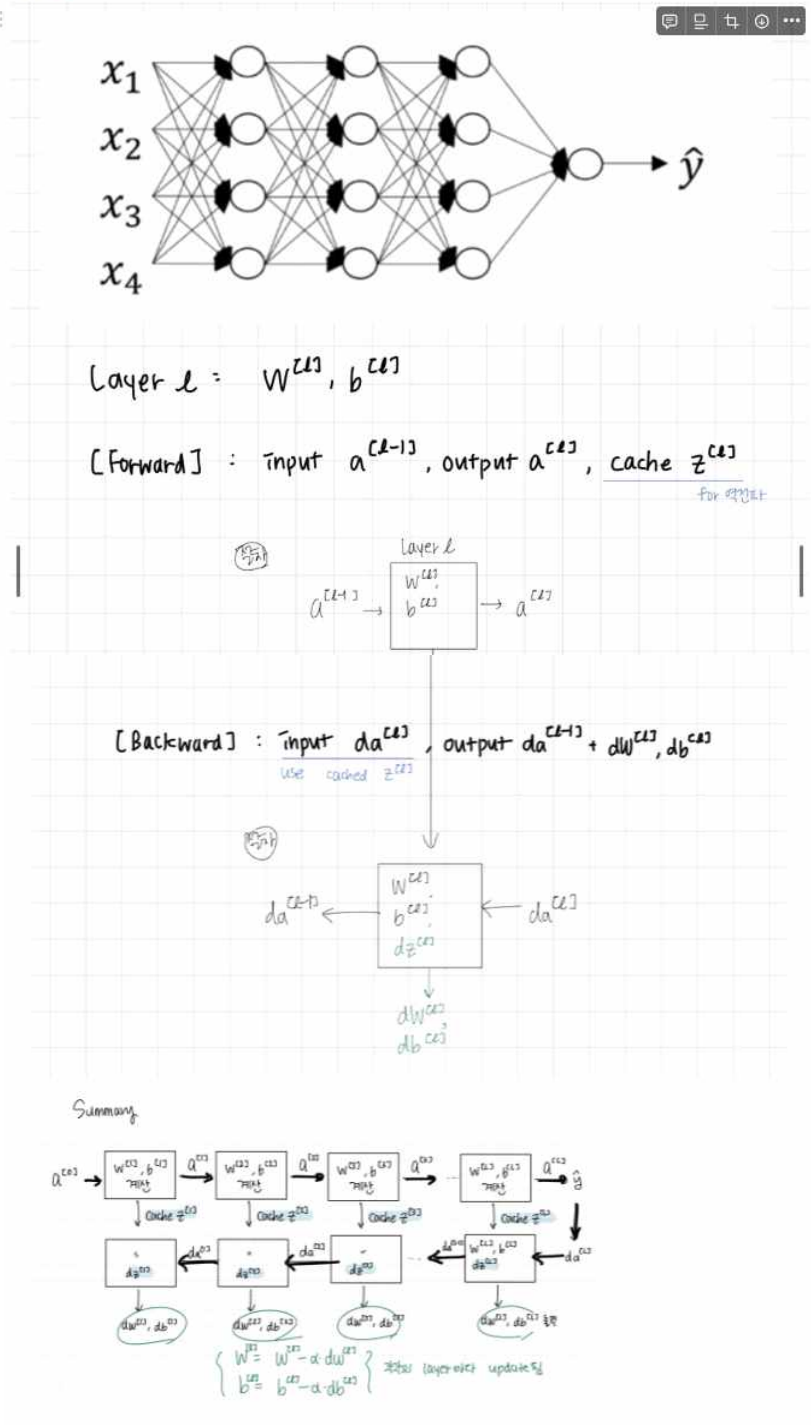
$$f(X) = \begin{bmatrix} x_{11}^2 & x_{12}^3 \\ x_{21}^2 & x_{22}^3 \end{bmatrix}$$

이러한 행렬 함수를 미분하면, 각 요소에 대한 미분이 이루어지며, 결과는 원래 행렬과 동일한 크기의 미분된 행렬이 생성됩니다. 즉, 위의 행렬 $f(X)$를 X에 대해 미분하면, 각 요소별로 미분이 이루어지므로 동일한 크기의 행렬이 생성됩니다. 이 결과 행렬은 각 요소별로 $f(X)$에 대한 미분값을 포함하게 됩니다.

따라서 행렬 내 각 요소에 대한 미분을 수행할 때에도, 일반적으로 차원이 변하지 않고, 결과로서 동일한 크기의 행렬이 생성됩니다.

[Week 5] 4. Building Blocks of a Deep Neural-Network

심층신경망에 영향을 주는 기본 구성요소들을 알아보자.



- z^[l]의 값을 저장하면 역전파 단계에서 유용하게 쓰일 수 있다.
- 캐시를 역방향함수에 대한 z값을 저장하는 곳이라고 이해하자. 캐시가 각 레이어마다 w와 b값을 얻어 역방향 함수에 넣기 때문에, 실제로는 캐시에 z뿐만아니라 W와 b도 저장된다.

[Week 5] 5. Parameters VS Hyperparameters

Hyperparameter?

- 우리가 지금까지 배운 모델의 파라미터는 W와 b이다.
- Hyperparameter는 parameter W와 b를 통제하는 매개변수이다. 매개변수의 값을 결정하는 매개변수를 하이퍼 파라미터라고 한다.

  ex) learning rate(α) , *iterations of gradient descent, #of Hidden layers(L), # of hidden units, activation function,* (+ momentum term, mini-batch size ...)

Deep learning is a very empirical process.

많은 하이퍼 파라미터의 값을 변경해가면서 시도하고 작동되는지 확인한다는 것을 의미한다.

여러 가지 경우를 시도해보면서 문제에 가장 적합한 하이퍼파라미터를 찾아가야 한다.