

BOAZ Spring Study : 6th & 7th week Assignment
(Due date : 2023.05.14)

[필기]

1. 다음 중 딥러닝에 대한 설명이 옳으면 O, 틀리면 X를 하고, 그 이유를 설명하세요.

(30 points)

- (1) 퍼셉트론(Perceptron)은 OR 문제를 풀지 못하였다는 한계점이 존재하였다.
- (2) 은닉층(hidden layer) 하나로 XOR 문제를 풀 수 있다.
- (3) 은닉층(hidden layer)가 꽤 많은 어떤 심층 신경망(Deep Neural Network)에서 활성화함수로 시그모이드 함수를 이용하면, 역전파를 진행할 때 gradient가 항상 작아진다.
- (4) 은닉층(hidden layer)가 꽤 많은 어떤 심층 신경망(Deep Neural Network)에서 활성화함수로 하이퍼볼릭 탄젠트 함수를 이용하면, 역전파를 진행할 때 gradient가 항상 작아진다.
- (5) 과적합이 발생하는 이유는 모델의 일반화 성능이 좋지 않았다고 볼 수 있다.
- (6) 심층 신경망(Deep Neural Network)의 학습 과정에서 과적합 문제를 해결하기 위해 드롭아웃(Dropout)을 진행하였다면, 학습 과정과 테스트 과정에 모두 동일하게 적용해야 한다.
- (7) 배치 경사 하강법(Batch Gradient Descent)은 확률적 경사 하강법(Stochastic Gradient Descent)보다 계산 시간이 느리다는 단점이 있지만, 정확도 면에서는 높다는 장점이 있다.
- (8) 배치 경사 하강법(Batch Gradient Descent)을 이용하여 파라미터를 업데이트 하는 데 적합이 잘 되지 않는다면, 학습 데이터를 섞는 것이 때로는 효과적일 수 있다.
- (9) 딥러닝을 학습하기 위해 손실 함수(Loss function)로 크로스 엔트로피(CrossEntropy)를 사용하면, global optimum을 찾을 수 있다.

$$CrossEntropy = - \sum_{i=1}^n y_i \log \hat{y}_i$$

- (10) 딥러닝 알고리즘을 학습하기 위해 학습률(Learning rate)을 작게 설정할수록 더 좋은 solution을 제공한다고 보장할 수 있다.
- (11) 데이터를 표준화(Standardization)하게 되면 최솟값이 0이고, 최댓값이 1인 분포를 띄게 된다.
- (12) 심층 신경망(Deep Neural Network)에서 네트워크의 각 층을 넘어갈수록 활성화 함수를 적용하면서 분포가 달라질 수 있다.
- (13) 심층 신경망(Deep Neural Network)에서 은닉층을 더 깊게 쌓을수록 loss가 줄어든다.
- (14) 신경망의 옵티마이저(Optimizer) 중 하나인 RmsProp는 이전까지 기울기의 변화량을 고려하여 가중치를 업데이트한다.
- (15) 신경망의 옵티마이저 중 하나인 아다그라드(Adagrad)는 Epoch을 반복할수록 업데이트 되는 가중치가 줄어든다.

2. 딥러닝에서 활성화함수로 쓰기 올바른 경우면 O, 그렇지 않으면 X표 하세요.

(10 points)

(1) $f(x) = -0.8x + 1.2$

(2) $f(x) = \min\{2, x + 1\}$

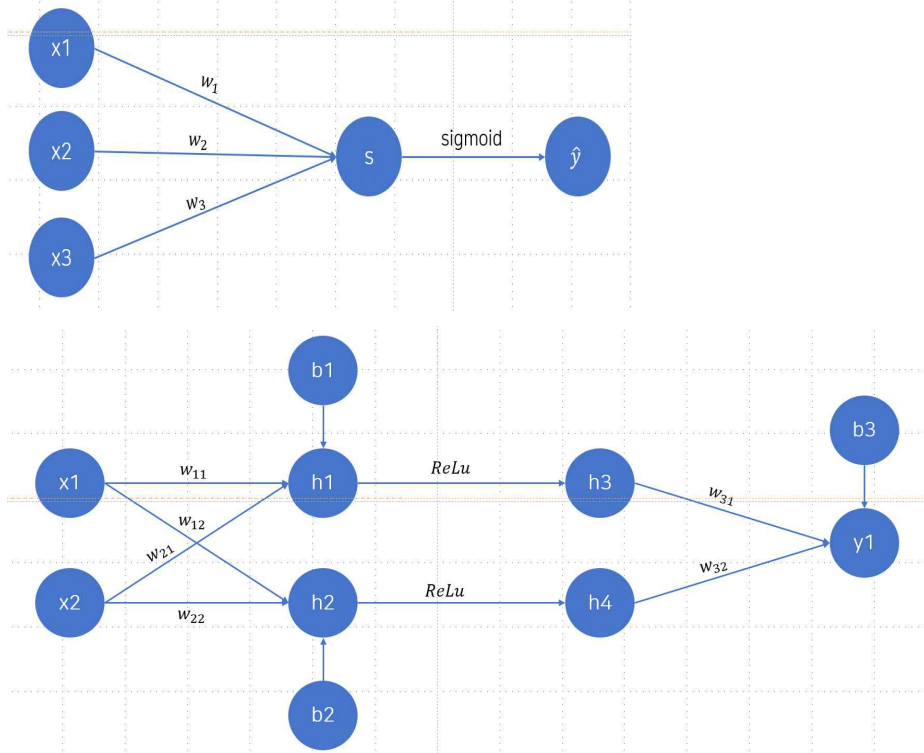
(3) $f(x) = \begin{cases} \max(x, 0.5x) & (x < 0) \\ \min(x, 0.5x) & (x \geq 0) \end{cases}$

(4) $f(x) = \begin{cases} \min(x, 0.5x) & (x < 0) \\ \min(x, 2x) & (x \geq 0) \end{cases}$

(5) sigmoid 함수를 활성화 함수로 쓸 경우, 어떤 문제점이 있는 지 쓰세요.

또한, 이 문제점을 해결하기 위해 어떤 활성화함수를 쓰면 좋은 지 설명하세요.

3. 아래 두 신경망에 대하여 물음에 답하세요. (30 points)



(1) (a)의 신경망에서 \hat{y} 를 w_1, w_2, w_3 을 이용하여 나타내세요.(bias는 없습니다.) (4 points)

(2) (a)의 신경망에서 loss function을 MSE로 정할 때, $\frac{\partial L}{\partial x_1}, \frac{\partial L}{\partial x_2}, \frac{\partial L}{\partial x_3}$ 을 각각

$w_1, w_2, w_3, \hat{y}, y$ 을 이용하여 나타내세요.(bias는 없습니다.) (8 points)

$$L = \frac{1}{n} \sum (y - \hat{y})^2$$

(3) (b)의 신경망에서 주어진 값이 아래와 같다. (y 는 실제 값입니다.)

x_1	x_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	b_1	b_2	b_3	y
3.0	-2.0	1.0	-1.5	-1.0	1.5	1.0	-0.5	1.5	-2.0	1.0	4.5

$h_1, h_2, h_3, h_4, \hat{y}$ 의 값을 모두 구하세요. (5 points)

(4) (3)의 결과에서 Mean Squared Error의 값을 구하세요. (3 points)

(5) (b)의 신경망에 대하여 Gradient Descent를 이용하여 가중치를 update하려고 합니다.

아래 물음에 답하세요.(loss function은 MSE이고, learning rate는 0.02입니다.)

(10 points)

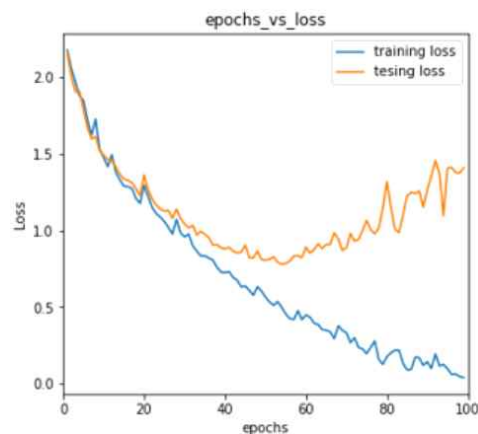
a. 가중치 w_{31} 를 update 하세요.

b. a에서 update한 가중치 w_{31} 와 같은 방법으로, w_{11} 과 w_{12} 를 update 하세요.

4. 아래와 같은 상황에서 문제를 어떻게 해결하면 좋을지 설명해 보세요. (10 points)

(1) Epoch을 거듭하면 거듭할수록 training loss와 test loss가 아래와 같은 상황이 되었습니다.

다. 어떻게 하는 것이 좋을까요? (6 points)



(2) 앞서, Sigmoid 함수를 활성화함수로 이용하면, 신경망을 깊게 쌓았을 때, 기울기 소실 (Vanishing Gradient) 문제가 발생한다는 점을 배웠습니다. 하지만, 반대로 기울기가 폭발적으로 증가해버릴 수 있습니다. 이를 기울기 폭주(Gradient Exploding)이라고 하는데, 이처럼, 기울기 소실 또는 기울기 폭주를 막을 수 있는 방법은 어떤 것들이 있을까요?

(4 points)

5. 아래는 어떤 심층 신경망(Deep Neural Network)의 구조를 대략적으로 나타낸 것입니다. 물음에 답하세요. (20 points)

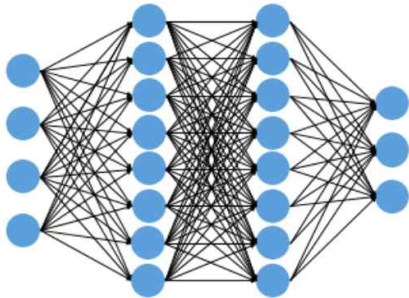
(1) 입력층(Input layer)의 차원이 10이고, 출력층(Output layer)의 차원이 3인 퍼셉트론(Perceptron)에 존재하는 모든 매개변수(Parameters)의 개수는 얼마인가요?

(3 points)

(2) 입력층(Input layer)의 차원이 2이고, 첫 번째 은닉층(Hidden layer)의 차원이 6, 그 다음 출력층(Output layer)의 차원이 3인 심층 신경망의 구조를 그리세요. (5 points)

(3) (2)의 신경망에 존재하는 모든 파라미터의 개수는 얼마인가요? (5 points)

이제, 아래와 같은 신경망을 참고해주세요.



(4) 데이터가 총 100개가 있을 때, 입력층과 첫 번째 은닉층 사이의 가중치 행렬의 크기, 첫 번째 은닉층과 두 번째 은닉층 사이에서의 가중치 행렬의 크기를 쓰세요. (7 points)

[실기]

6. PyTorch를 이용하여 다중 선형 회귀를 구현해 보려고 합니다. 아래의 순서를 따라서 완성해 보세요. (50 points)

(1) x1에는 0.2부터 1까지의 실수 중에서 랜덤으로 100개를 저장하고, x2는 0.1부터 0.9까지의 실수 중에서 랜덤으로 100개를, x3은 0.1부터 1까지의 실수 중 랜덤으로 100개를 저장하세요. y는 $x1 + 2x2 + 2x3 + 2$ 에 추가적으로 0부터 1까지의 실수 중 랜덤으로 하나를 뽑아서 noise를 주세요. 이 모든 과정은 Numpy를 이용하면서 진행하며, 최종적으로 x1, x2, x3의 shape는 (100, 3)이고, y는 (100)이어야 합니다. (10 points)

(2) x1, x2, x3, y를 Numpy에서 tensor의 형태로 바꾸어 주세요. (5 points)

(3) batch size를 32로 두고, 학습을 진행하려고 합니다. 그 전에, TensorDataSet를 이용하여 DataSet의 형태로 만들어 주고, DataLoader에 사용하고자 하는 데이터셋을 담으세요. (배치가 적절하게 섞이도록 dataloader 옵션에 shuffle = True를 추가해 주세요.)

(10 points)

(4) CustomLinearRegression 클래스에 nn.Module을 상속받고, 만들 모델을 클래스에 적으세요. 만들 모델은 입력층과 출력층 사이에 하나의 은닉층이 존재하며, 차원은 4입니다.

활성화 함수(Activation Function)은 sigmoid를 사용하세요. (10 points)

(5) 모델을 설정하고, optimizer는 확률적 경사 하강법(SGD)를 선택하고, learning rate는 0.001로 한 다음에, 학습을 시작해 주세요. Epoch는 200회 반복하며, 10번마다 Loss 값을 출력하도록 구현하세요. Loss Function은 MSE를 사용합니다. (15 points)

7. PyTorch를 이용하여 MNIST 데이터에 있는 손글씨를 분류해보려고 합니다. 아래의 순서를 따라서 완성해 보세요. (50 points)

(1) train_dataset과 test_dataset에 있는 dataset을 DataLoader에 담으세요. 이때, batch size는 64로 설정하세요. DataLoader 과정에서 오류가 뜬다면, drop_last = True를 추가해보세요. (10 points)

(2) MNIST 데이터는 0부터 9까지의 손글씨로 적힌 숫자가 무엇인지를 맞추도록 한 데이터셋입니다. 아래는 nn.Module을 상속받은 클래스 MLP에 대한 구조일 때, MLP 클래스를 알맞게 완성하세요. (15 points)

```
MLP(  
    (fc1): Linear(in_features=784, out_features=128, bias=True)  
    (relu): ReLU()  
    (fc2): Linear(in_features=128, out_features=64, bias=True)  
    (fc3): Linear(in_features=64, out_features=10, bias=True)  
    (softmax): Softmax(dim=1)  
)
```

(3) loss function은 CrossEntropyLoss로, 옵티마이저는 Adam을 이용하며, 옵티마이저의 파라미터는 자유롭게 설정하세요. train 데이터를 이용하여 epoch을 20번 돌려 학습한 후, test 데이터를 이용해서 검증하는 단계까지 완성하세요. (25 points)