

5주차 - Programming Practice

matrix mismatch 오류로 고생을 많이 했다.

0번째 유닛의 delta값은 Theta_grad를 구하는데 제거할 필요가 있다는 것을(즉 없어야 mismatch 오류가 나지 않음) 이해하는데 시간이 오래 걸렸다.

예를들어 t번째 케이스의 은닉층 delta_2는 $h \times 1$ 벡터인데, $a_1(n+1 \times 1)$ 벡터를 곱해서

Theta1_grad($h \times n+1$) 행렬을 만들어야 한다. 0번째 유닛을 고려해서 delta_2가 $h+1 \times 1$ 벡터가 되어서는 안된다.

그리고 숙련된 상태가 아니라면 행렬계산을 복잡하게 생각하기 보단, 각 case에 대하여 for문을 이용하여 차원을 줄여 계산하는 방법이 오히려 시간이 덜 걸린다.

n = 입력층 유닛의 개수

h = 은닉층 유닛의 개수

r = 출력층 유닛의 개수

1. Neural Networks(Cost function, Gradient)

1. feedforward

$z(l)$: $s_l \times 1$

$a(l)$: $(s_l + 1 \times 1)$

$z(l)$ 로 $a(l)$ 을 구한 뒤에 $[1 ; a(l)]$ 을 대입하여 행을 추가한다.

1. backpropagation

delta

1. Regularization
2. Random Initialization
3. Gradient Checking