2. Neural Network Basics (2)

2019년 1월 14일 월요일 오전 12:33

3. $\dfrac{\partial J}{\partial x}$?

$$\begin{cases} J = CE(y, \hat{y}) \\ \hat{y} = \text{softmax}(\overbrace{hW_2 + b_2}^{\text{let's say this as } z_2}) \\ h = \sigma(xW_1 + b_1) \end{cases}$$

(cf)

$$\begin{cases} y, \hat{y} : M \times D_y \\ x : M \times D_x \\ W_1 : D_x \times H \\ b_1 : 1 \times H \\ h : M \times H \\ W_2 : H \times D_y \\ b_2 : 1 \times D_y \end{cases}$$

$$\frac{\partial CE(y,\hat{y})}{\partial x} = \frac{\partial CE(y,\hat{y})}{\partial z_2} \cdot \frac{\partial z_2}{\partial x}$$

$$= (\hat{y} - y) \cdot \frac{\partial z_2}{\partial x} \quad (\because \text{2번 문제})$$

$$= (\hat{y} - y) \cdot \frac{\partial z_2}{\partial h} \cdot \frac{\partial h}{\partial x}$$

$$= (\hat{y} - y) W_2^T \cdot \frac{\partial h}{\partial x}$$

$$= \left[\{(\hat{y} - y) W_2^T\} \circ \{h(1-h)\} \right] \cdot W_1^T \quad : M \times D_x$$

4. \# of params?

$$n(W_1) + n(W_2) + n(b_1) + n(b_2)$$

$$= (D_x \times H) + (H \times D_y) + (H) + (D_y)$$

5. (programming part – sigmoid implementation)

6. (programming part – gradient checker)

<Central difference approximations>

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

매우 작은 $h > 0$ 를 이용해 gradient를 위와 같이 근사 가능.
근사한 값과 chain rule로 계산한 gradient와
차이값을 확인해서 gradient를 제대로 구했는지 어느정도 check 가능.

7. (programming part – neural net)