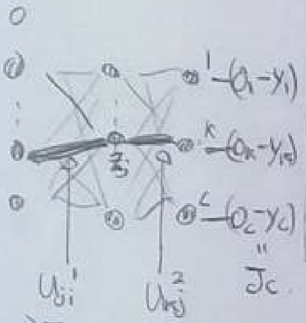
*기계학습 7주차 A4 1Page 정리 201610409 컴퓨터공학과 김성수

Training MLP.

*Gradient Descent ⟶ SGD.

- $U^1 = U^1 - \rho \frac{\partial J}{\partial U^1}$
- $U^2 = U^2 - \rho \frac{\partial J}{\partial U^2}$

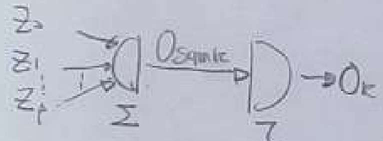


U^1, U^2 초기화
repeat
  X 순서 shuffle
  for(X 샘플)
    Fowarding을 통해 output
    $\frac{\partial J}{\partial U^1}, \frac{\partial J}{\partial U^2}$ 계산
    U^1, U^2 갱신
until: 멈춤조건

*$\frac{\partial J}{\partial U^2}$

$$\text{MSE } J = \sum_{q=1}^{c} J_q = \sum_{q=1}^{c} \frac{1}{2}(y_q - O_q)^2$$

$$\frac{\partial J}{\partial u^2_{kj}} = \frac{\partial}{\partial u^2_{kj}}\left(\frac{1}{2}(y_k - O_k)^2\right) = -(y_k - O_k)\frac{\partial O_k}{\partial u^2_{kj}}$$



$$= -(y_k - O_k)\frac{\partial \tau(O_{sumk})}{\partial O_{sumk}} \cdot \frac{\partial O_{sumk}}{\partial u^2_{kj}}$$

$$= \underbrace{-(y_k - O_k)\,\tau'(O_{sumk})}_{\delta_k \text{로 let}} \cdot z_j$$

*$\frac{\partial J}{\partial U^1}$

$$\text{MSE } J = \sum_{q=1}^{c} J_q = \sum_{q=1}^{c} \frac{1}{2}(y_q - O_q)^2$$

$$\frac{\partial J}{\partial u^1_{ji}} = -\sum_{q=1}^{c}(y_q - O_q)\frac{\partial O_q}{\partial u^1_{ji}} \rightarrow \frac{\partial O_q}{\partial z_j} \cdot \frac{\partial z_j}{\partial u^1_{ji}}$$

$$= -\sum_{q=1}^{c}(y_q - O_q)\,\tau'(O_{sumq}) \cdot \frac{\partial O_{sumq}}{\partial z_j} \cdot \tau'(z_{sumj}) \cdot \frac{\partial z_{sumj}}{\partial u^1_{ji}}$$

$$= -\sum_{q=1}^{c}\underbrace{(y_q - O_q) \cdot \tau'(O_{sumq})}_{\delta_q \text{로 let.}}\, u^2_{qj} \cdot \tau'(z_{sumj})\, x_i$$

$1 \le k \le c$ $\quad 0 \le j \le p,\ 1 \le k \le c$

$$\therefore \frac{\partial J}{\partial u^2_{kj}} = -(y_k - O_k)\,\tau'(O_{sumk})\, z_j = -\delta_k z_j$$

$1 \le j \le p$

$$\therefore \frac{\partial J}{\partial u^1_{ji}} = -\tau'(z_{sumj})\, x_i \sum_{q=1}^{c}(y_q - O_q)\,\tau'(O_{sumq})\, u^2_{qj}$$

$$= -\tau'(z_{sumj})\sum_{q=1}^{c}\delta_q u^2_{qj} \cdot x_i = -\eta_j x_i$$

$0 \le i \le d,\ 1 \le j \le p.$

⟶ 오류 역전파 알고리즘
출력층의 오류를 위 식을 이용하여,
역방향으로 전파하며 Gradient를 계산하는 역할

* Practical Considerations.

*아키텍처
⟶ 은닉층, 은닉 노드 개수를 정해야 함
⟶ 적절한 규제 (Regularization) 기법으로 Overfit 방지.

*초기값.
⟶ 가중치를 초기화.
⟶ 적절한 값을 부여함으로써 학습 성능↑

*학습률
⟶ 처음 = 끝 학습률이 같은 방식
⟶ 처음 > 끝 학습률을 줄이는 방식.

*활성함수
⟶ 로지스틱 시그모이드, tanh 함수의 경우,
은닉층의 개수를 늘리면 Vanishing Gradient ⟶ 가중치 매개변수 업데이트 X 현상
⟶ ReLU 등을 활용하여 해결.