

1 Neural networks

1.1

a

$$\frac{\partial l}{\partial u} = \frac{\partial l}{\partial a} \times \frac{\partial a}{\partial u} = (W^{(2)})^T \frac{\partial l}{\partial a} \cdot * H(u)$$

b

$$\implies \frac{\partial l}{\partial a} = z - y$$

c

$$\frac{\partial l}{\partial W^{(1)}} = \frac{\partial l}{\partial u} \times \frac{\partial u}{\partial W^{(1)}} = \frac{\partial l}{\partial u} \times x^T$$

d

$$\frac{\partial l}{\partial b^{(1)}} = \frac{\partial l}{\partial u} \times \frac{\partial u}{\partial b^{(1)}} = \frac{\partial l}{\partial u}$$

e

$$\frac{\partial l}{\partial W^{(2)}} = \frac{\partial l}{\partial a} \times \frac{\partial a}{\partial W^{(2)}} = \frac{\partial l}{\partial a} \times h^T$$

1.2

$$W_n^{T+1} = W_n^T - \eta g \implies \frac{\partial l}{\partial W_n} = 0$$

Hence there will be no learning

1.3

$$u = W^{(1)}x + b^{(1)}$$

$$a = W^{(2)}u + b^{(2)}$$

$$\implies W^{(2)}(W^{(1)}x + b^{(1)}) + b^{(2)}$$

$$\implies W^{(2)}W^{(1)}x + W^{(2)}b^{(1)} + b^{(2)}$$

Since $a = Ux + v$

$$U = W^{(2)}W^{(1)}$$

this has K x D dimensions

$$v = W^{(2)}b^{(1)} + b^{(2)}$$

this has K x 1 dimensions

2 Kernal Methods

2.1

$$J(w) = \min_w \sum_{n=1}^N (l(w^T \phi(x_n), y_n)) + \frac{\lambda}{2} \|w\|_2^2$$

Given we can assume : $s = (w^T \phi(x_n), y_n)$

$$\implies \frac{\sum_n \partial l(s, y)(x_n)}{\partial s} + \lambda w = 0$$

$$\implies w = -\frac{1}{\lambda} \frac{\sum_n \partial l(s, y)}{\partial s} \phi(x_n)$$

$$\implies \text{Hence } w^* = \sum_n \alpha_n \phi(x_n) \quad \text{where } \alpha_n = -\frac{1}{\lambda} \frac{\sum_n \partial l(s, y)}{\partial s}$$

2.2

$$J(w) = \min_w \sum_{n=1}^N (l(w^T \phi(x_n), y_n)) + \frac{\lambda}{2} \|w\|_2^2$$

$$\text{Since } w = \sum_n \alpha_n \phi(x_n)$$

$$\implies w = \phi^T \alpha$$

$$J(w) = l(\phi w, y) + \frac{\lambda}{2} \|\phi^T \alpha\|_2^2$$

$$\implies l(\phi \phi^T \alpha, y) + \frac{\lambda}{2} (\alpha^T \phi \phi^T \alpha)$$

Now we know that $K = \phi \phi^T$

$$\implies J(w) = l(K\alpha, y) + \frac{\lambda}{2} \alpha^T K \alpha$$