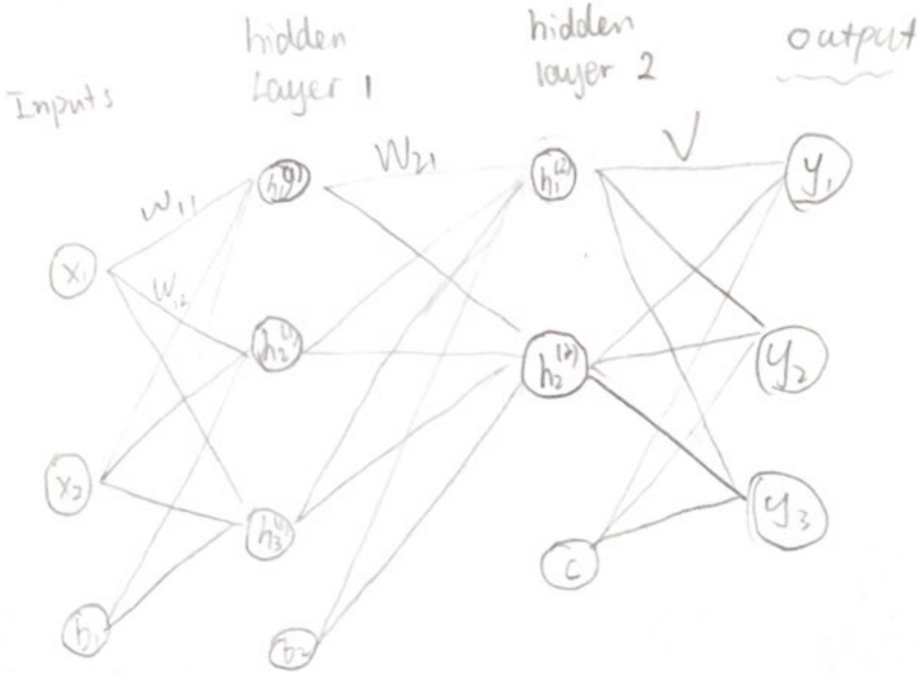
1 Feedforward : building a Relu 2 layer neural network

1. plot a network with:



2 mathematical equation:

$$a^{(1)} = \vec{x} \cdot \vec{w_j} + b_1$$

$$h^{(1)} = \max(0, a_j) = \max(0, a^{(1)})$$

$$a^{(2)} = \vec{h_1}^{(1)} \cdot \vec{w_{2j}} + b_2$$

$$h^{(2)} = \max(0, a^{(2)})$$

$$a^{(3)} = \vec{h}^{(2)} \cdot \vec{V} + c$$

$$\hat{y} = \mathrm{softmax}(a^{(3)})$$

2 Gradient Descent

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2$$
$$= (1-2x+x^2) + 100(y^2 - 2x^2y + x^4)$$
$$= 1-2x+x^2+100y^2-200x^2y+100x^4$$

$$\frac{f(x,y)}{\partial x} = -2+2x-(200y \cdot 2x)+400x^3$$
$$= -2+2x-400xy+400x^3$$

$$\frac{f(x,y)}{\partial y} = 200y - 200x^2$$

3. $L = (y_1 \log \hat{y}_1 + y_2 \cdot \log \hat{y}_2 + y_3 \log \hat{y}_3)$

$$\frac{\partial L(y,\hat{y})}{\partial C_s} = \mathbb{I}(s = \substack{\text{true} \\ \text{class}} - \hat{y}) \qquad s = 1,2,3$$

$$\frac{\partial L(y,\hat{y})}{\partial V_{ks}} = \left[\mathbb{I}(s = \substack{\text{true} \\ \text{class}} - \hat{y})\right] h_k^{(2)}, \quad k = 1,2$$

$$\frac{\partial L(y,\hat{y})}{\partial b_k^{(2)}} = \sum_s (\mathbb{I}(s = \substack{\text{true} \\ \text{class}} - \hat{y}) V_{ks} \mathbb{I}_{a_k > 0}$$

$$\frac{\partial L(y,\hat{y})}{\partial W_{jk}^{(2)}} = \sum_s (\mathbb{I}(s = \substack{\text{true} \\ \text{class}} - \hat{y}) V_{ks} \mathbb{I}_{a_k > 0} h_j^{(1)}, \quad j = 1,2,3.$$

$$\frac{\partial L(y,\hat{y})}{\partial b_j^{(1)}} = \sum_s (\mathbb{I}(s = \substack{\text{true} \\ \text{class}} - \hat{y}) V_{ks} W_{jk}^{(2)} \mathbb{I}_{a_k > 0} \mathbb{I}_{a_j > 0}$$

$$\frac{\partial L(y,\hat{y})}{\partial W_{ij}^{(1)}} = \sum_s \sum_k (\mathbb{I}(s = \substack{\text{true} \\ \text{class}} - \hat{y}) V_{ks} W_{jk}^{(2)} \mathbb{I}_{a_k > 0} \mathbb{I}_{a_j > 0} x_i \text{ for } i = 1,2$$