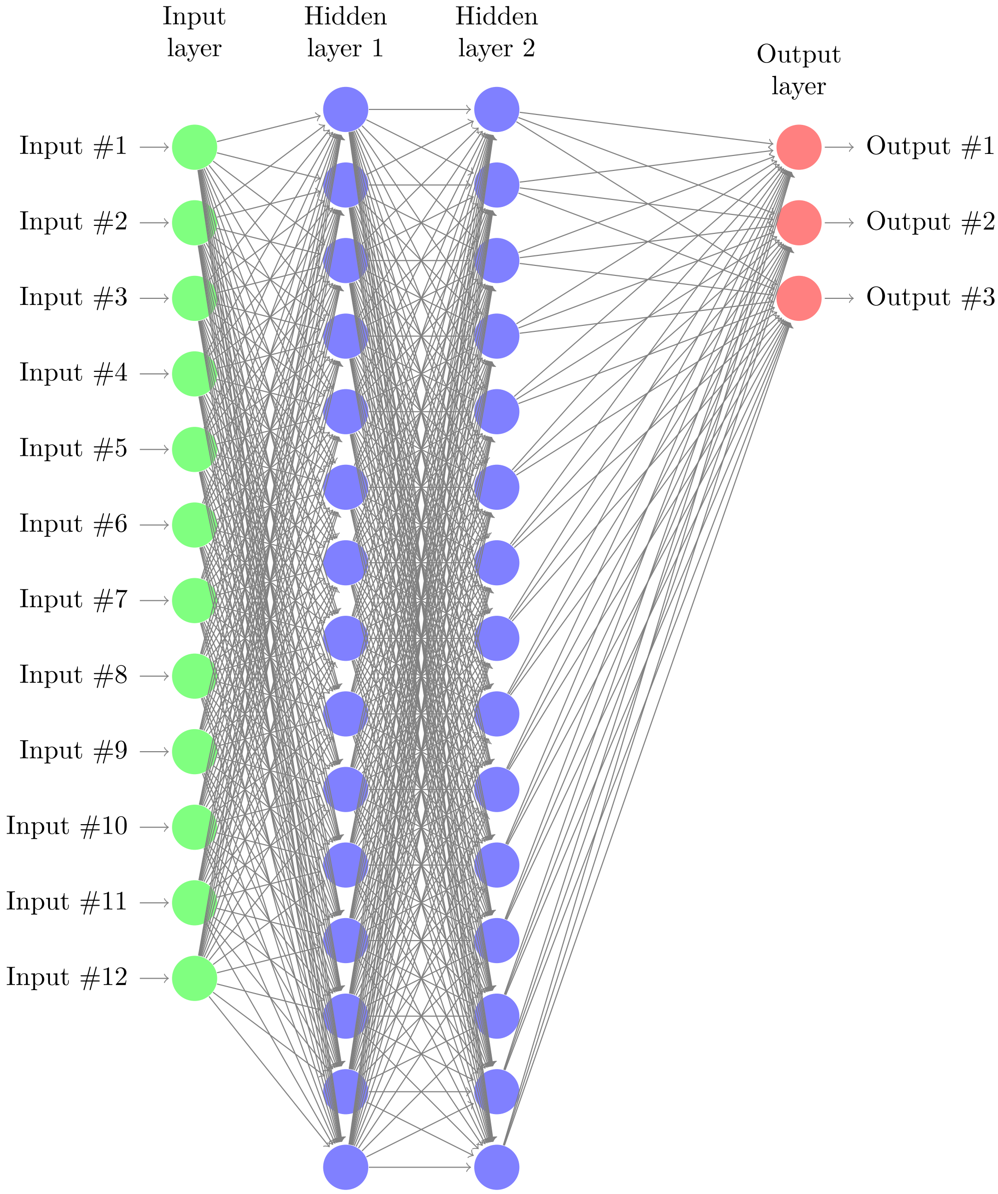
Machine Learning
Sommersemester2020
Exercise 8

Ciheng Zhang (3472321) zch3183505@gmail.com
Gang Yu(3488292) HansVonCq@gmail.com
Huipanjun Tian (3471607) Thpjpyl5111217@gmail.com

June 27, 2020

Ciheng Zhang(3472321)
Gang Yu(3488292)
Huibanjun Tian(3471607)

1 Formalizing Neural Networks



$$F(x) = {W_3}^T\sigma({W_2}^T\sigma({W_1}^T x + b_1) + b_2) + b_3$$

Assume the width of hidden layers is 15. So the W_1 is the weight between input layer, and the dimension is $12 * 15$ and the input dimension is $12 * 1$. and the bias b_1 is $15 * 1$. Then the matric W_2 is the weight between hidden layer1 and hidden layer2. the dimension of W_2 is $15 * 15$.Then the matric W_3 is the weight

Ciheng Zhang(3472321)
Gang Yu(3488292)
Huibanjun Tian(3471607)

between Hidden layer2 and output layer. The dimension is $15 * 3$. the bias between hidden layer1 and hidden layer2 is b_2 and the dimension is $15 * 1$. The bias between hidden layer2 and output layer is $3 * 1$
its a multinouli problem so we choose the Cross Entropy function as loss function:

$$loss = -\Sigma(z_i - log\Sigma e^{z_j})$$

Then we want to punish the missclassification for class -1, So we add the weight for this loss function:

$$loss = -\Sigma(1 + \alpha_i)(z_i - log\Sigma e^{z_j})$$

and for example we let $\alpha = [1, 0, 0]$. So we can punish the missclassification for class -1.

2 Backpropagation by Hand

1.

$$w_1^{(o)} max(w_{11}^{(h)} * x_1 + w_{21}^{(h)} * x_1, 0) + w_2^{(o)} max(w_{12}^{(h)} * x_2 + w_{22}^{(h)} * x_2, 0) = -0.25$$

2.

$$L = (f_w(x) - y)^2 = 1.5625$$

3.

$$\frac{dL}{dw_{11}} = \frac{dL}{df_w(x)} \frac{df_w(x)}{do_1} \frac{do_1}{dg} \frac{dg}{dw_{11}}$$

$$= 2(f_w(x) - y) * w_1^{(o)} * deReLU * x_1 = -1.75 * 1 * 1 * 2 = -3.5$$

Then we update the weight:

$$w_{11}^h = w_{11}^h - \eta \frac{dL}{dw_{11}} = 0.85$$