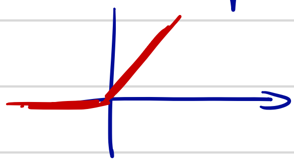
Question 1: Multilayer Perception.

$a = \phi\left(\sum_j w_j x_j + b\right)$

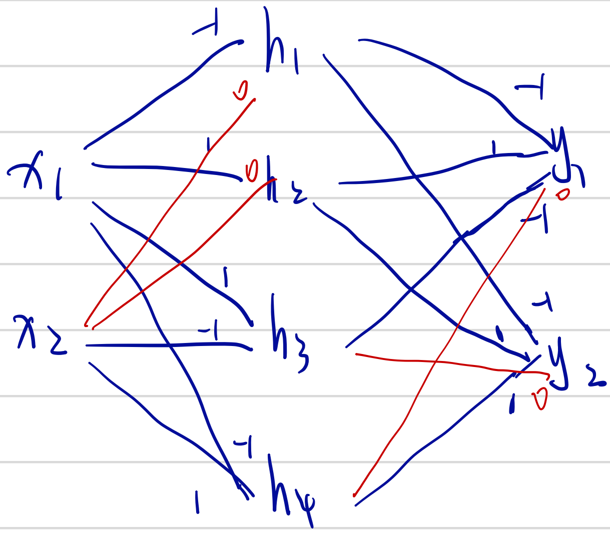
ReLU: rectified linear unit.



$y = \max(0, z)$

request: $y_1 = \min(x_1, x_2)$

$y_2 = \max(x_1, x_2)$



$h_1 = -x_1$

$h_2 = x_1$

$h_3 = x_1 - x_2$

$h_4 = x_2 - x_1$

$\vec{x} = [x_1 \ x_2]$

$\vec{y} = [y_1, y_2]$

$$\boxed{\vec{W}^{(1)} = \begin{bmatrix} -1 & 1 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \qquad \vec{b}^{(1)} = \vec{0}}$$

$$\vec{h} = \vec{W}^{(1)T} \cdot \vec{x} + \vec{b}^{(1)}$$

$$\boxed{\vec{W}^{(2)} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \\ -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \vec{b}^{(2)} = \vec{0}}$$

$$\vec{y} = \vec{W}^{(2)T} \cdot \vec{h} + \vec{b}^{(2)}$$

Explain why my solution works

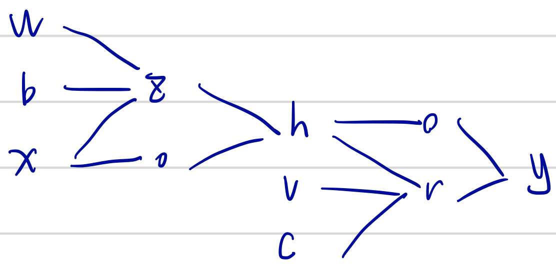
In my solution, I used the hidden layer to express positive x_1, negative x_1, x_1-x_2, x_2-x_1.

Next, y_1 is $\min(x_1, x_2)$. If $x_1 > x_2$, x_1-x_2 will be bigger than 0, $y_1 = -\phi(x_1-x_2) + x_1 = x_2 - x_1 + x_1 = x_2$. If $x_1 < x_2$, $x_1 - x_2$ will be smaller than 0, then $y_1 = -\phi(x_1-x_2) + x_1 = x_1$

For y_2 is $\max(x_1, x_2)$ If $x_1 > x_2$, $x_2 - x_1$ will be smaller than 0, $y_2 = \phi(x_2-x_1) + x_1 = 0 + x_1 = x_1$. If $x_1 < x_2$, $x_2 - x_1$ will be bigger than 0, $y_2 = \phi(x_2-x_1) + x_1 = x_2 - x_1 + x_1 = x_2$. Now, the results for every condition are correct. Besides, when x_1, and x_2 are negative, this method also works. For example, when x_1 is negative, $-x_1$ is positive, and x_1 is zero after activation function. But we set the weight for $-x_1$ is -1, and 1 for the x_1, so the sum of $-\phi(-x_1) + \phi(x_1)$ is always value of x_1.

Question 2: Backprop

a)



h, y . same size

h, x same size

b) vector form

$$\bar{L} = 1$$

$$\bar{y} = \bar{L} \cdot \frac{dL}{dy} = \bar{y}$$

$$\bar{r} = \bar{y} \cdot \frac{dy}{dr} = \bar{y} \cdot \phi'(r)$$

$$\bar{h} = \bar{y} \cdot \frac{dy}{dh} + \bar{r} \cdot \frac{dr}{dh} = \bar{y} \cdot \frac{dy}{dr} \cdot \frac{dr}{dh} + \bar{r} \cdot V^T = \bar{r} \cdot V^T + \bar{r} \cdot V^T = 2\bar{r}V^T$$

$$\boxed{\bar{V} = \bar{r} \cdot \frac{dr}{dV} = \bar{r} \cdot h^T}$$

$$\boxed{\bar{c} = \bar{r} \cdot \frac{dr}{dc} = \bar{r}}$$

$$\bar{z} = \bar{h} \cdot \frac{dh}{dz} = \bar{h} \cdot \phi'(z)$$

$$\boxed{\bar{W} = \bar{z} \cdot \frac{dz}{dW} = \bar{z} \cdot x^T}$$

$$\boxed{\bar{b} = \bar{z} \cdot \frac{dz}{db} = \bar{z}}$$

$$\bar{x} = \bar{z} \cdot \frac{dz}{dx} + \bar{h} \cdot \frac{dh}{dx} = \bar{z} \cdot W^T + \bar{h}$$

3.

should complete the following table.

	# Units	# Weights	# Connections
Convolution Layer 1	290400	34848	105415200
Convolution Layer 2	186624	307200	223948800
Convolution Layer 3	64896	884736	149520384
Convolution Layer 4	64896	663552	112140288
Convolution Layer 5	43264	442368	74760192
Fully Connected Layer 1	4096	177209344	177209344
Fully Connected Layer 2	4096	16777216	16777216
Output Layer	1000	4096000	4096000

Assume: ignore pooling layer.

fully connected layer

output unit

weight } input unit × output unit

connection

convolution layer

(# output × copies)

kernel Dims × kernel # $\quad$ $k^2 J$

kernel Dims × output units # $\quad$ $k^2 W H J$

Convolution layer 1: $2 \times 55 \times 55 \times 48 = 290400$ $\quad$ $11 \times 11 \times 48 \times 3 \times 2 = 34848$

$2 \times 11 \times 11 \times 3 \times 55 \times 55 \times 48 = 105415200$

Convolution layer 2: $5 \times 5 \times 48 \times 128 \times 2 = 307200$

$5 \times 5 \times 48 \times 128 \times 2 \times 27 \times 27 \times 2 = 223948800$

Convolution layer 3: $3 \times 3 \times 128 \times 192 \times 2 \times 2 = 884736$

$3 \times 3 \times 128 \times 192 \times 13 \times 13 \times 4 = 149520384$

Convolution layer 4: $3 \times 3 \times 192 \times 192 \times 2 = 663552$

$3 \times 3 \times 192 \times 192 \times 13 \times 13 \times 2 = 112140288$

Convolution layer 5: $3 \times 3 \times 192 \times 128 \times 2 = 442368$

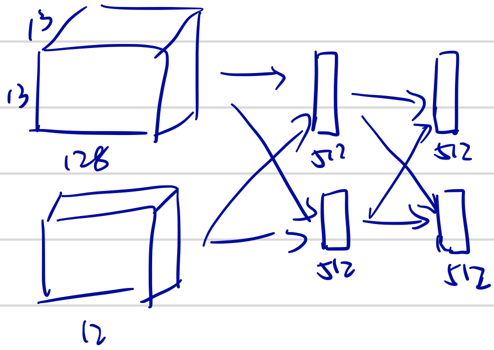
$3 \times 3 \times 192 \times 128 \times 13 \times 13 \times 2 = 74760192$

Full connected 1: $13 \times 13 \times 128 \times 2 \times 2048 \times 2 = 177209344$

Full connected 2: $2048 \times 2 \times 2048 \times 2 = 16777216$

output: $2048 \times 2 \times 1000 = 4096000$

b) i) To reduce the memory usage, we need to reduce the output unit or parameters. From the question 1 a)-, we can easily find the fully connected layer 1 required too many parameters, which is about 2 million. So, I would add a new pooling layer between Fully connected layer 1 and Fully connected layer 2. Or, we can dense the densed layers to 512.



, then we only need $13 \times 13 \times 128 \times 512 \times 4$
$= 44,302,336$

ii) To reduce connection, I would reduce the kernal filters, increase the stride of kernal, increase max-pooling. Then the width, height and depth of output layers should decrease. And the connections will decrease either.