Neural Network and Backpropagation Questions

Ying Wang

CDS, NYU

April 19, 2023

Question 1: Step Activation Function [1]

Suppose we have a neural network with one hidden layer.

$$f(x) = w_0 + \sum_i w_i h_i(x); \quad h_i(x) = g(b_i + v_i x),$$

where activation function g is defined as

$$g(z) = \begin{cases} 1 & \text{if } z \geqslant 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Which of the following functions can be exactly represented by this neural network?

- polynomials of degree one: $l(x) = ax + b$
- hinge loss: $l(x) = max(1 - x, 0)$
- polynomials of degree two: $l(x) = ax^2 + bx + c$
- piecewise constant functions

[1] From CMU

[Solution] Question 1: Step Activation Function

Suppose we have a neural network with one hidden layer.

$$f(x) = w_0 + \sum_i w_i h_i(x); \quad h_i(x) = g(b_i + v_i x),$$

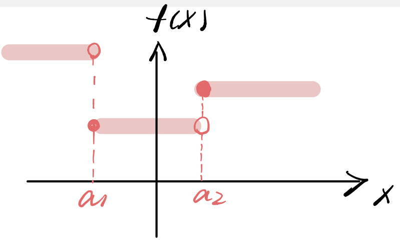
where activation function g is defined as

$$g(z) = \begin{cases} 1 & \text{if } z \geqslant 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Which of the following functions can be exactly represented by this neural network?

- polynomials of degree one: $l(x) = ax + b$ **No**
  If g can be identity function, then the answer is **Yes**
- hinge loss: $l(x) = max(1 - x, 0)$ **No**
- polynomials of degree two: $l(x) = ax^2 + bx + c$ **No**
- piecewise constant functions **Yes**
  $(-c) \cdot g(x - b) + (c) \cdot g(x - a)$ can represent $l(x) = c, a \leqslant x < b$.

[Solution] Question 1: Step Activation Function



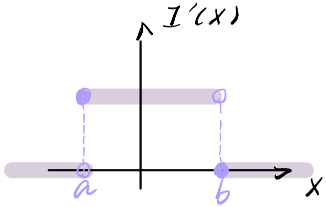
$$f(x) = \begin{cases} c_1 & x < a_1 \\ c_2 & a_1 \le x < a_2 \\ \cdots & \\ c_n & x \ge a_n \end{cases}$$

If we can represent any, $I(x) = c,\ a \le x < b$

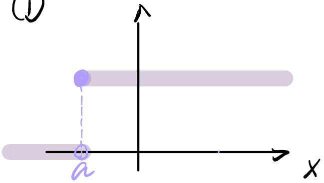
then any piecewise constant function can be represented by $f(x) = \sum I_i(x)$

Q: How to represent $I(x) = c,\ a \le x \le b$?

$I(x) = cI'(x)$ where $I'(x) = 1,\ a \le x \le b$

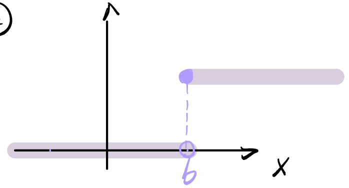


①



$I_1'(x) = g(x-a)$

②



$I_2'(x) = g(x-b)$

$\Rightarrow I'(x) = g(x-a) - g(x-b)$

$\Rightarrow I(x) = cg(x-a) - cg(x-b)$

Question 2: Power of ReLU [2]

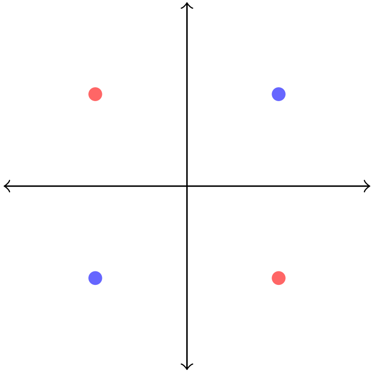
Consider the following small NN:

$$w_2^\top \textbf{ReLU}\left(W_1 x + b_1\right) + b_2$$

where the data is 2D, W_1 is 2 by 2, b_1 is 2D, w_2 is 2D and b_2 is 1D.

$$x_1 = (1,1)\ \ y_1 = 1; \quad x_2 = (1,-1)\ \ y_2 = -1; \quad x_3 = (-1,1)\ \ y_3 = -1; \quad x_4 = (-1,-1)\ \ y_4 = 1$$

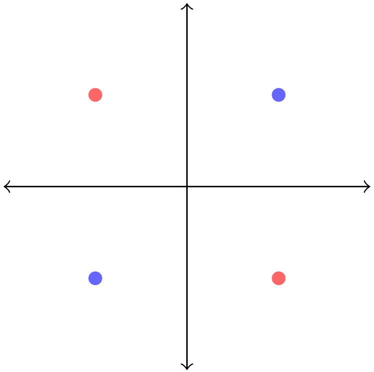
Find b_1, b_2, W_1, w_2 to solve the problem. (Separate points from class $y = 1$ and $y = -1$.)



[2] From Harvard

[Solution] Question 2: Power of ReLU

$$w_2^\top \,\mathbf{ReLU}\left(W_1 x + b_1\right) + b_2$$



One choice is

$$W_1 = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}, b_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$w_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, b_2 = -1$$

[Solution] Question 2: Power of ReLU

x	$h(x)$	ReLU$(h(x))$	$\hat{y}$
$(1,1)$	$(2,-2)$	$(2,0)$	$2-1=1$
$(-1,1)$	$(0,0)$	$(0,0)$	$0-1=-1$
$(1,-1)$	$(0,0)$	$(0,0)$	$0-1=-1$
$(-1,-1)$	$(-2,2)$	$(0,2)$	$2-1=1$

Question 3: Backpropagation [3]

Suppose we have a one hidden layer network and computation is:

$$h = \text{RELU}(Wx + b1)$$
$$\hat{y} = \text{softmax}(Uh + b_2)$$
$$J = \text{Cross entropy}(y, \hat{y}) = -\sum_i y_i \log \hat{y}_i$$

The dimensions of the matrices are:

$$W \in \mathbb{R}^{m\times n} \quad x \in \mathbb{R}^n \quad b_1 \in \mathbb{R}^m \quad U \in \mathbb{R}^{k\times m} \quad b_2 \in \mathbb{R}^k$$

Use backpropagation to calculate these four gradients

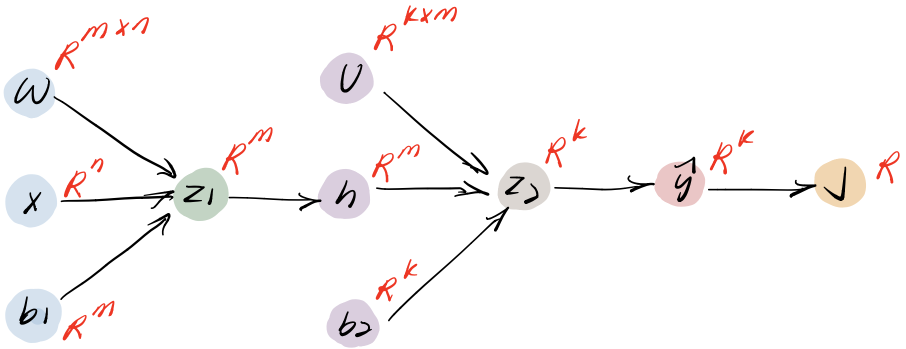
$$\frac{\partial J}{\partial b_2} \quad \frac{\partial J}{\partial U} \quad \frac{\partial J}{\partial b_1} \quad \frac{\partial J}{\partial W}$$

[3] From Stanford

[Solution] Question 3: Backpropagation

$$z_2 = Uh + b2 \quad \delta_1 = \frac{\partial J}{\partial z_2} = \hat{y} - y$$
$$\frac{\partial J}{\partial b_2} = \delta_1$$
$$\frac{\partial J}{\partial U} = \delta_1 h^T$$
$$\frac{\partial J}{\partial h} = U^T \delta_1$$
$$z_1 = Wx + b_1 \quad \delta_2 = \frac{\partial J}{\partial z_1} = U^T \delta_1 \circ 1\{h > 0\}$$
$$\frac{\partial J}{\partial b_1} = \delta_2$$
$$\frac{\partial J}{\partial W} = \delta_2 x^T$$

[Solution] Question 3: Backpropagation



$$J = -\sum_i y_i \log \frac{e^{z_{2i}}}{\sum_j e^{z_{2j}}} = -\sum_i y_i \left(z_{2i} - \log \sum_j e^{z_{2j}}\right)$$

$$\frac{\partial J}{\partial z_{2k}} = -y_k + \sum_i y_i \frac{e^{z_{2k}}}{\sum_j e^{z_{2j}}} = -y_k + \hat{y}_k \sum_i y_i = -y_k + \hat{y}_k \Rightarrow \frac{\partial J}{\partial z_2} = \hat{y} - y$$

$\sum_i y_i = 1$ (y is one-hot vector)