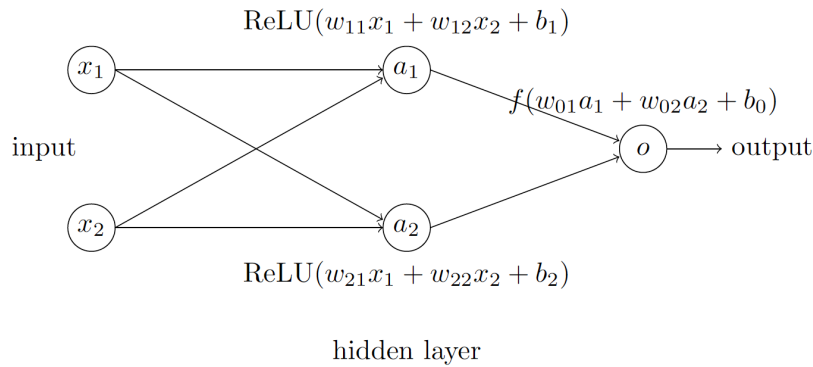


Machine Learning Assignment 5 (Neural Networks)

Due: 17, May

1. [25pts] Multi-layer Perception

Suppose that we apply neural networks on a problem which has boolean inputs $x \in \{0,1\}^p$ and boolean output $y \in \{0,1\}$. The network structure example is showed as below. In this example we set $p = 2$, single hidden layer with 2 neurons, activation function $ReLU(u) = u$ if $u > 0$ otherwise 0, and an additional threshold function (e.g., $f(v) = 1$ if $v > 0$, otherwise $f(v) = 0$) for output layer.

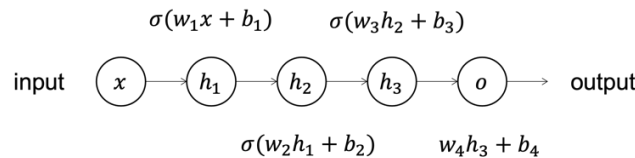


(1) [5pts] Using the structure and settings of neural network above, show that such a simple neural network could output the function $x_1 XOR x_2$ (equals to 0 if $x_1 = x_2$ and otherwise 1), which is impossible for linear models. State the values of parameters (i.e., w_{ij} and b_i) you found.

(2) [20pts] Now we allow the number of neurons in the hidden layer to be more than 2 but finite. Retain the structure and other settings. Show that such a neural network with single hidden layer could output an arbitrary binary function $h: \{0,1\}^p \mapsto \{0,1\}$. You can apply threshold function after each neuron in the hidden layer.

2. [10pts] Gradient explosion and gradient vanishing

As shown in the figure below, the neural network has three hidden layers, each with only one neuron, and the activation function is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. Let the input be $x = 3$. Use backpropagation to calculate the gradient, and experience the gradient explosion and gradient vanishing issues [1].

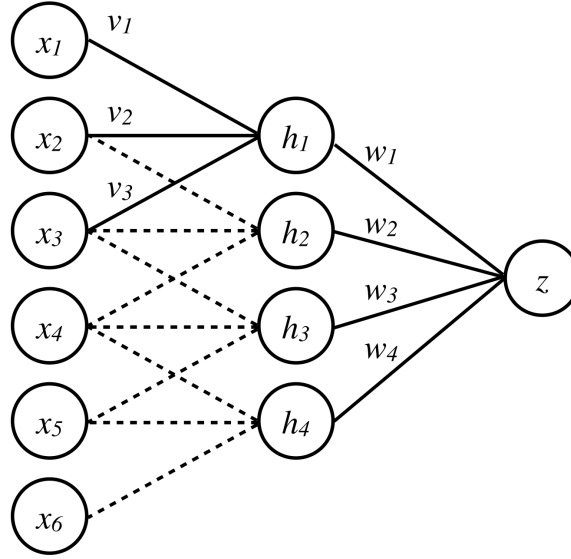


(1) [5pts] If $w_1 = 100, w_2 = 150, w_3 = 200, w_4 = 200, b_1 = -300, b_2 = -75, b_3 = -100, b_4 = 10$, calculate the gradient $\frac{\partial o}{\partial b_1}$.

(2) [5pts] If $w_1 = 0.2, w_2 = 0.5, w_3 = 0.3, w_4 = 0.6, b_1 = 1, b_2 = 2, b_3 = 2, b_4 = 1$, calculate the gradient $\frac{\partial o}{\partial b_1}$.

3. [25pts] CNN

Consider this **one-dimensional** convolutional neural network architecture.

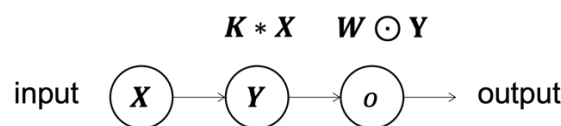


In the first layer, we have a one-dimensional convolution with a single filter of size 3 such that $h_i = \sigma(\sum_{j=1}^3 v_j x_{i+j-1})$. The second layer is fully connected, such that $z = \sigma(\sum_{i=1}^4 w_i h_i)$. The hidden units and output unit's activation function $\sigma(x)$ is the logistic (sigmoid) function with derivative $\sigma'(x) = \sigma(x)(1 - \sigma(x))$. We perform gradient descent on the loss function $L = (y - z)^2$, where y is the training label for x .

- [5pts] What is the total number of parameters in this neural network? Recall that convolutional layers share weights. There are no bias terms.
- [10pts] Compute $\frac{\partial R}{\partial w}$
- [10pts] Compute $\frac{\partial R}{\partial v_i}$

4. [30pts] CNN

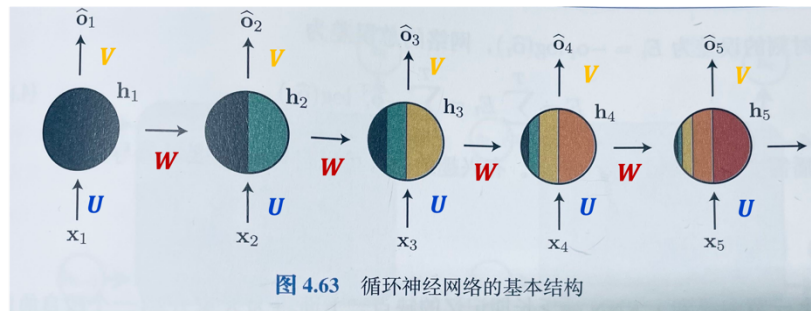
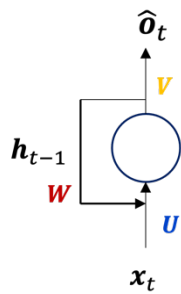
As shown in the figure below, the neural network consists of a convolutional layer and a fully connected layer, with input as $\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$, convolutional kernel (filter) as $\mathbf{K} = \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}$, convolution result as $\mathbf{Y} = \mathbf{K} * \mathbf{X} = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix}$, and output as $o = \sigma(w_{11}y_{11} + w_{12}y_{12} + w_{21}y_{21} + w_{22}y_{22})$, σ is sigmoid function. The sample label is z , and E is the mean squared error ($E = \frac{1}{2} \|z - o\|^2$). Find the derivative of the convolutional kernel [2].



Where, “*” is convolution, “⊙” is Hadamard product which is element-wise product.

5. [30pts] RNN: BPTT

Provide the detailed derivation process of the BPTT (Backpropagation Through Time) algorithm, ensuring that the symbols are consistent with those in the Lecture PPT.



Reference

- [1] 王贝伦, “习题 4.24,” 出处 机器学习, 东南大学, 2021.
- [2] 王贝伦, “习题 4.25,” 出处 机器学习, 东南大学, 2021.