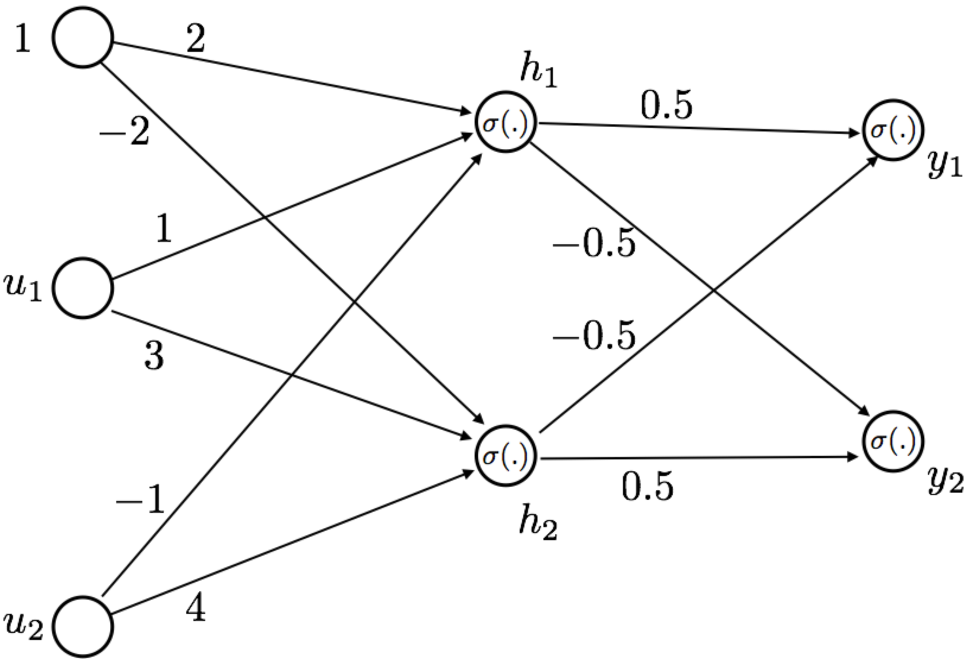
CS/ECE/ME532 Period 21 Activity

Estimated time: 10 mins for P1, 10 mins for P2a, 20 mins for P2b, 20 mins for P2c

1. Consider the neural network with three input nodes, two hidden nodes, and two output nodes shown below. The numbers by the edges are the weights that are applied to the output of the corresponding nodes. Use the ReLU activation $\sigma(z) = \max\{0, z\}$. Suppose the values at the input nodes are $u_1 = 4$, $u_2 = -2$. Find the values of the hidden nodes and the output nodes.



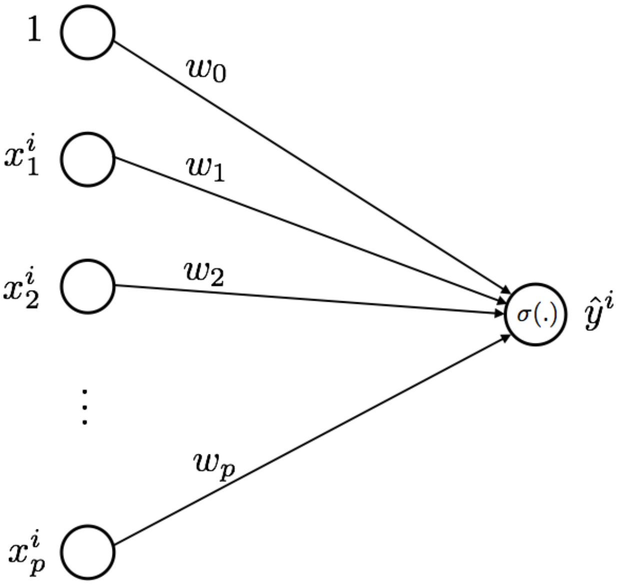
SOLUTION: $h_1 = \max\{0, 2(1) + 4 - (-2)\} = 8$
$h_2 = \max\{0, -2(1) + 3(4) + 4(-2)\} = 2$
$y_1 = \max\{0, 0.5(8) - 0.5(2)\} = 3$
$y_2 = \max\{0, -0.5(8) + 0.5(2)\} = 0$

2. We use the single neuron shown in the figure for classification. Here x_j^i is the j-th feature in the i-th training sample and the output is $\hat{y}^i = \sigma\left(\sum_{j=0}^{P} w_j x_j^i\right)$.

The stochastic gradient descent update for the weights at step t is

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \alpha_t \nabla f^{i_t}(\boldsymbol{w}^{(t)})$$

where α_t is the step size and $f^{i_t}(\boldsymbol{w}^{(t)})$ is the loss function associated with training sample i_t



a) Write the expression for $\hat{y}^i$ in terms of the weights w_j and the input x_j^i for the ReLU activation function $\sigma(z) = \max\{0, z\}$.

b) Suppose squared-error loss is used to find the weights, that is, the loss function is $f(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n}(\hat{y}^i - y^i)^2$ where y^i are the labels for feature sample $\boldsymbol{x}^i$.

i. Find the gradient of $f(\boldsymbol{w})$ with respect to w_j assuming the ReLU activation function.

ii. Write out psuedo-code for implementing SGD to learn the weights $\boldsymbol{w}$ given n training samples (features and labels) $\{\boldsymbol{x}^i, y^i, i = 1, 2, \ldots n\}$ assuming the ReLU activation function.

c) Now suppose we use ridge regression for the loss function

$$f^i(\boldsymbol{w}) = \frac{1}{2}(\hat{y}^i - y^i)^2 + \lambda \sum_{j=0}^{P} w_j^2$$

and we use the logistic activation function $\sigma(z) = (1+e^{-z})^{-1}$. Derive the gradient for the update step, $\nabla f^{i_t}(\boldsymbol{w}^{(t)})$ and write the update equation for $\boldsymbol{w}^{(t+1)}$.

SOLUTION:

a) We have

$$\hat{y}^i = \sigma\left(w_0 + \sum_{m=1}^{p} w_m x_m^i\right)$$

$$= \begin{cases} 0, & w_0 + \sum_{m=1}^{p} w_m x_m^i < 0 \\ w_0 + \sum_{m=1}^{p} w_m x_m^i, & w_0 + \sum_{m=1}^{p} w_m x_m^i \geq 0 \end{cases}$$

b) Now define $f^i(\boldsymbol{w}) = \frac{1}{2}(\hat{y}^i - y^i)^2$, so $f(\boldsymbol{w}) = \sum_{i=1}^{n} f^i(\boldsymbol{w})$. By the chain rule

$$\frac{df(\boldsymbol{w})}{dw_j} = \sum_{i=1}^{n} \frac{df^i(\boldsymbol{w})}{d\hat{y}^i}\frac{d\hat{y}^i}{dw_j}$$

where

$$\frac{df^i(\boldsymbol{w})}{d\hat{y}^i} = (\hat{y}^i - y^i)$$

and

$$\frac{d\hat{y}^i}{dw_j} = \begin{cases} 0, & w_0 + \sum_{m=1}^{p} w_m x_m^i < 0 \\ x_m^i, & w_0 + \sum_{m=1}^{p} w_m x_m^i \geq 0 \end{cases}$$

Then, at iteration t:

- Choose i_t uniformly at random from $\{1, 2, \ldots, n\}$
- Set step size α_t
- Update weights

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \alpha_t \nabla f^{i_t}(\boldsymbol{w}^{(t)})$$

  where

$$\nabla f^{i_t}(\boldsymbol{w}^{(t)}) = \begin{cases} 0, & w_0^{(t)} + \sum_{m=1}^{p} w_m^{(t)} x_m^{i_t} < 0 \\ (\hat{y}^{i_t} - y^{i_t})\boldsymbol{x}^{i_t}, & w_0^{(t)} + \sum_{m=1}^{p} w_m^{(t)} x_m^{i_t} \geq 0 \end{cases}$$

c) Note that

$$\frac{df^{i_t}(\boldsymbol{w}^{(t)})}{dw_k} = (\hat{y}^{i_t} - y^{i_t})\frac{d\hat{y}^{i_t}}{dw_k} + 2\lambda w_k$$

where we follow the steps in The Backpropagation Algorithm for Training Neural Networks video to obtain

$$\frac{d\hat{y}^{i_t}}{dw_k} = \sigma\left(\sum_{j=0}^{P} w_j x_j^{i_t}\right)\left[1 - \sigma\left(\sum_{j=0}^{P} w_j x_j^{i_t}\right)\right] x_k^{i_t}$$

$$= \hat{y}^{i_t}\left(1 - \hat{y}^{i_t}\right) x_k^{i_t}$$

Define $\delta^{i_t} = (\hat{y}^{i_t} - y^{i_t})\hat{y}^{i_t}\left(1 - \hat{y}^{i_t}\right)$ so we can write

$$\nabla f^{i_t}(\boldsymbol{w}^{(t)}) = \delta^{i_t}\boldsymbol{x}^{i_t} + 2\lambda\boldsymbol{w}^{(t)}$$

and thus the update is

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \alpha_t\delta^{i_t}\boldsymbol{x}^{i_t} - 2\alpha_t\lambda\boldsymbol{w}^{(t)}$$