CSDS 440: Assignment 5

Shaochen (Henry) ZHONG, `sxz517`

Mingyang TIE, `mxt497`

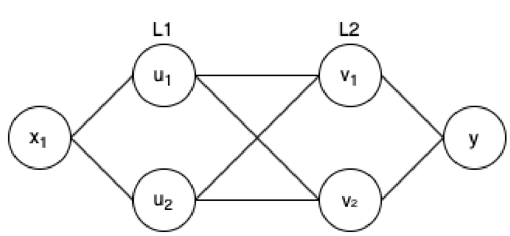
Due on 10/09/2020, submitted early on 10/02/2020
Fall 2020, Dr. Ray

Problem 22

I asked TA, couple of my classmates, and consulted online resources. It seems there are two ways to answer this question. One is to calculate the gradient between each layer and show that without the $sign()$ function the weight will stop updating, but it does not utilized the hint of decision boundary.

So a conceptual argument can be formed on the idea of if we drop the $sign()$ function, a neuron's activation function will simply be its input – which is a linear function of all pervious neurons. This suggests the output of the ANN will also be a linear function of the input, as combination of linear function is always linear. Thus the ANN will have a linear decision boundary, which voids the purpose of having a deep network (as we can rewrite the whole network with just a single layer), and such network won't be useful for solving a problem which requires a non-linear decision boundary.

Below is the gradient calculation approach.



Assume we have the above ANN of two hidden layers, each with m number of nerons, a common square loss will be:

$$
\begin{aligned}
L(w) &= \frac{1}{2}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \\
&= \frac{1}{2}\sum_{i=1}^{m}(y_i - sign(w \cdot x_i))^2
\end{aligned}
$$

For the sake having something differentiable, we drop the $sign()$ function and make it as:

$$
L(w) = \frac{1}{2}\sum_{i=1}^{m}(y_i - w \cdot x_i)^2
$$

Therefore, for the gradient of L_2 and output, we have:

$$
\begin{aligned}
\frac{\partial L_2}{\partial w} &= \sum_{i}^{m}(y_i - w \cdot x_i)(-x_i) \\
&= \sum_{i}^{m}(w \cdot x_i^2 - x_i \cdot y_i)
\end{aligned}
$$

We furter differentiate it to get the gradient of L_1 and L_2:

$$
\begin{aligned}
\frac{\partial L_1}{\partial w} &= \frac{\partial}{\partial w}(\sum_{i}^{m}(w \cdot x_i^2 - x_i \cdot y_i)) \\
&= \sum_{i}^{m} x_i^2
\end{aligned}
$$

Similiarily, for gradient of input and L_1

$$
\frac{\partial L_{\text{input}}}{\partial w} = \frac{\partial}{\partial w}\sum_{i}^{m} x_i^2 = 0
$$

Since the gradient is 0, the weight of the input will not be updated. Note this is just a 2-layer ANN, an arbitrary ANN with more layers but without the $sign()$ function might also encounter this problem and make the weight updating mechanism ineffective.