Sensitivity Based Adversaries

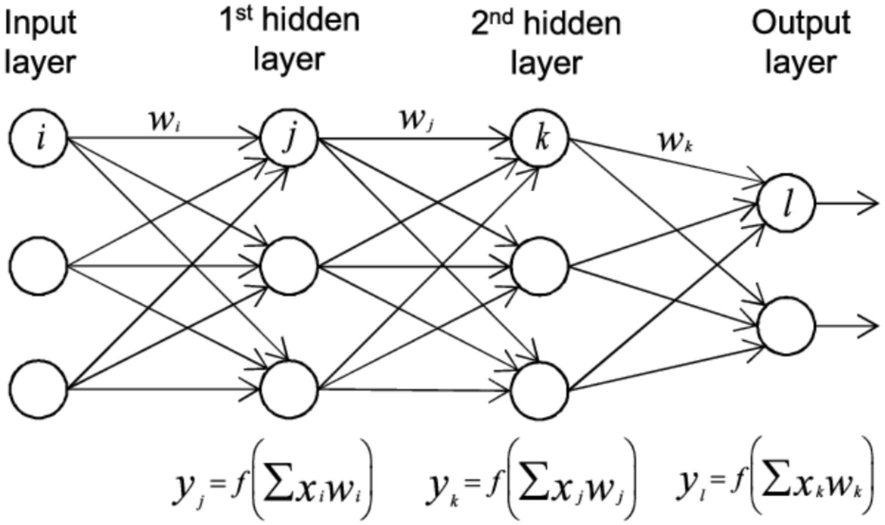
Neha Nagarkar

September 10, 2020

1 Introduction

Neural Networks are known to perform well on images but their performance is affected by adversaries. In simple terms, Adversaries are nothing but the input images perturbed by adding noise. In order to understand why the neural networks are so much susceptible to noise, first we need to understand their behavior. So, I have tried to perform sensitivity analysis for a neural network.

For this research project I have used MNIST data and I have trained a simple neural network with 2 hidden layers with a dropout, in order to make sure the network does not overfit. The accuracy of this network comes out to be 98.52%. The network looks something like shown below.


Input layer
1st hidden layer
2nd hidden layer
Output layer
i
j
k
l
w_i
w_j
w_k
$y_j = f\left(\sum x_i w_i\right)$
$y_k = f\left(\sum x_j w_j\right)$
$y_l = f\left(\sum x_k w_k\right)$

Number of neurons in each layer.
Input layer: 784
1^{st} hidden layer:512

2^{nd} hidden layer:512
output layer: 10

The output of every layer is an input to next layer except the output layer. Hence, the output of each neuron is given by a and calculated as follows.

$$z = f(x) = \sum w_i * x_i + b_i \tag{1}$$

a=g(z) Where g is an activation function.

I have used these concepts to calculate the output and compare the actual output with the calculated value as follows:

$$E = abs(Y - a_k) \tag{2}$$

Then Using chain rule,

$$\frac{\partial E}{\partial z_3} * \frac{\partial z_3}{\partial a_2} * \frac{\partial a_2}{\partial z_2} * \frac{\partial z_2}{\partial a_1} * \frac{\partial a_1}{\partial z_1} * \frac{\partial z_1}{\partial x_1} \tag{3}$$

I got this equation to backtrack the error to the input and find the most sensitive neuron for the given input image and given network.

Using this sensitive neuron, we can control the output of the neural network by modifying the input corresponding to this neuron.This is useful when we want to enhance the performance of neural network.