Local Minima 2,1,2

Fig. 1. 1,1,1 NN

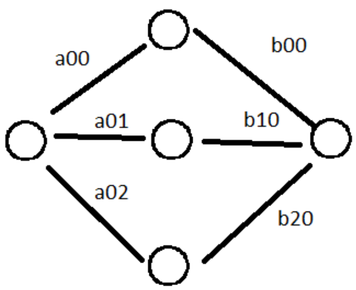

Fig. 2. 1,3,1 NN

***Abstract*—Trying to find out if there are some interesting local minimas in 2,1,2 neural network. 2 inputs, 1 hidden relu, 2 outputs.**

I. Introduction

We assume throughout the discussion that inputs are positive, so $x > 0$. It is clear that linear nns do not have local minima. The next obvious step is to investigate non linear relu nns. Looking at 1,1,1, it is clear that there is no local minima, it behaves like linear until $a00 > 0$ and outputs 0 otherwise.

Two directions forward:

- Increase the number of hidden neurons
- Increase the number of inputs

II. 1,n,1: On increasing the size of hidden neurons

Intuitively, when all weights are positive or negative, this behaves like a linear netowrk or always zero. The case that is to be investigated is when some $a0i > 0$ while some $a0j < 0$. The major observation which distinguishes 1 input from multiple inputs is that these neurons will behave exactly the same for each example xi in 1,n,1.

So, if neuron n is off for example xi, it will be turned off for example xj because all examples are positive which implies any neuron only turns off when corresponding $a < 0$. Now, because of this,when any neuron is turned off, 1,n,1 behaves like 1,n-1,1 in that ”region”. The set where any neuron is off is actually a connected region (this is true for any n,n,n network, but in this case it is actually half of the space i.e $a < 0$). Now, local in these regions there can not be any local minima because they all behave linearly inductively. The only case which needs to be checked is that global minimas for these regions can be actually local minima to the entire space. This is equivalent to saying ”Does 1,n,1 make an improvement on the loss over 1,n-1,1?”. 1,n,1 actually contains all the global/local minimas of 1,n-1,1 and the only way it can create new local minimas is if it comes up with a better global minima on input examples.

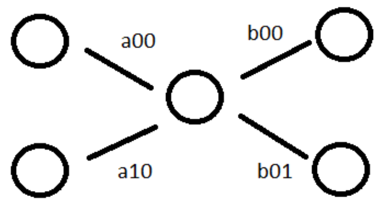

Fig. 3. 2,1,2 NN

III. 2,1,2: Increasing the number of inputs

1,1,1 does not have a local minima. Increasing the number of inputs immediately lead to a local minima. The simplest case 2,1,2 has a family of minimas.

Denote by ri the output of relu unit on example xi. Then, we can easily see

$$ri = 0 \iff a00 * xi0 < -(xi1 * a10) \quad (1)$$

$$r0 = 0 \iff a00 * x00 < -(x01 * a10) \quad (2)$$

$$r1 = 0 \iff a00 * x10 < -(x11 * a10) \quad (3)$$

Equation 1 and 2 are lines in (a00,a01) plane. This leads to four regions. For input (1,0) and (1,1), the lines are (with a00 as x and a01 as y)

$$r0 = 0 \iff x < 0 \quad (4)$$

$$r1 = 0 \iff x < -y \quad (5)$$

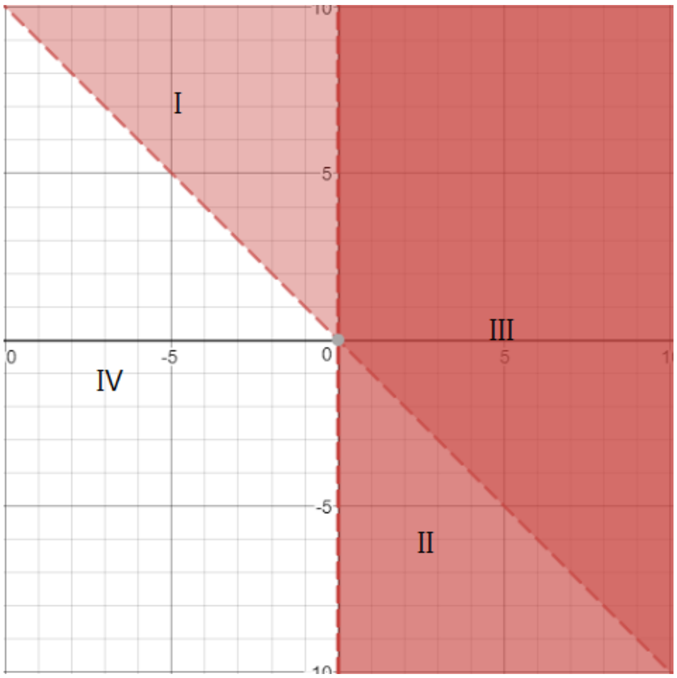

Fig. 4. 2,1,2 Loss NN

Region I ($x < 0$ and $x > -y$) has neuron working only for example x1, region II ($x > 0$ and $x < -y$) has neuron working only for example x0, region III ($x > 0$ and $x > -y$) has the neuron working for both examples and region IV has neuron turned off for both examples.

Loss in Region I will always have a non zero component from example x0, so the minimum loss there will be $>= norm(y0)$. In fact (given b00,b01), the output from the neuron for example x1 which leads to minima is unique, let it be r(for x0, it will be 0).

$$a00 * x10 + x11 * a10 = r \tag{6}$$

This is an equation of a line. a00 and a01 which satisfy this equation and lie in region I are local minimas (again, given b00,b01, changing the b00,b01 will give another line of local minimas, the loss on all such lines is same.)

The natural question is would gradient descent fall in these local minimas. If you start from region I, would always lead to these minimas? Would starting from region III, safeguard convergence to global minima?

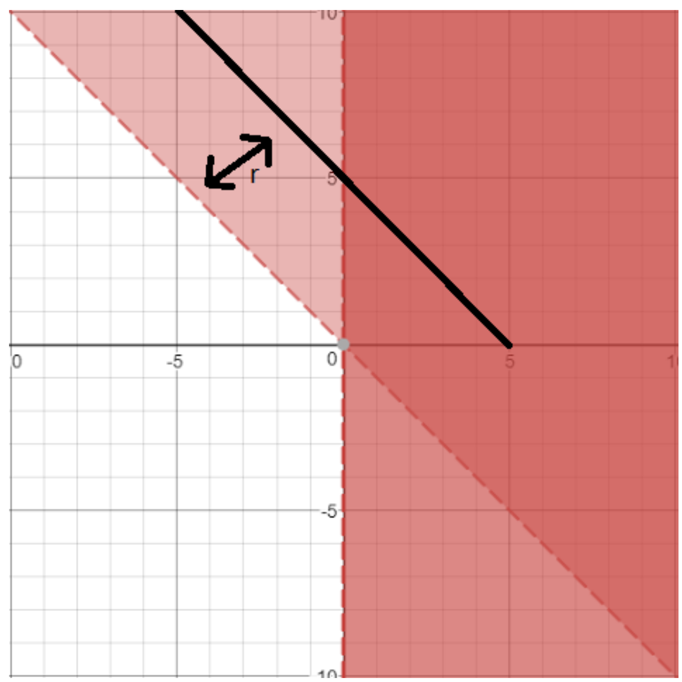

Fig. 5. 2,1,2 local minima

ACKNOWLEDGMENTS