Local Minima 2,1,2

Fig. 1. 1,1,1 NN

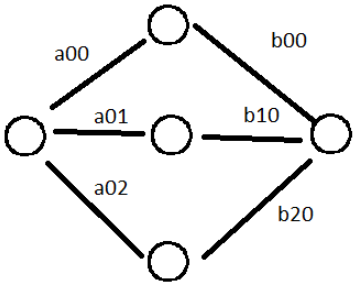

Fig. 2. 1,3,1 NN

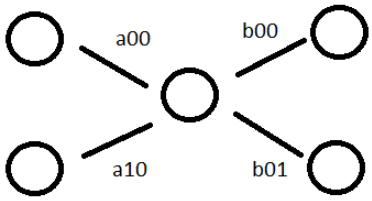

Fig. 3. 2,1,2 NN

***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.

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)

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

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

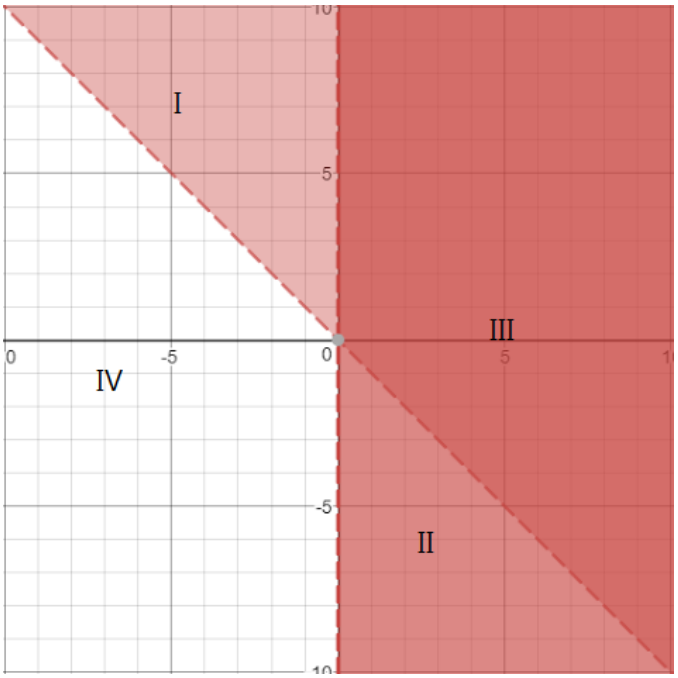

Fig. 4. Neuron behavior

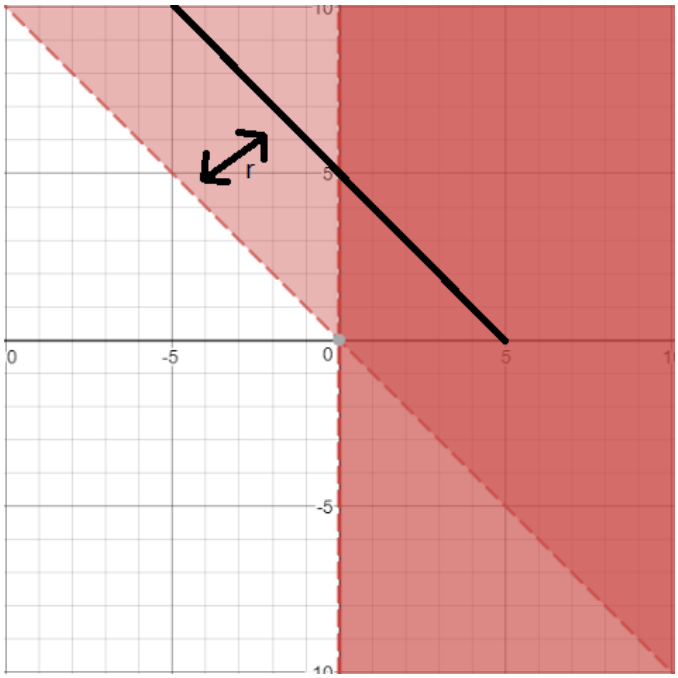

Fig. 5. 2,1,2 local minima

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?

IV. 2,2,2 CASE

4 shows the behavior of a relu neuron. 2,2,2 has two relu neurons and we can imagine them to be two points. This allows us to divide the analysis into 6 independent cases (I will refer to global minima for the all cases as G_{all}). 6 shows the running example for analysis. Output from neuron 1 for both examples (k_1, r_1). Output from neuron 2 for both examples (k_2, r_2). Expected output: (y_1, y_2) and (y_3, y_4).

1) **Both lie in region III**: This is similar to linear neural network. Global minima in this region is G_{all} .

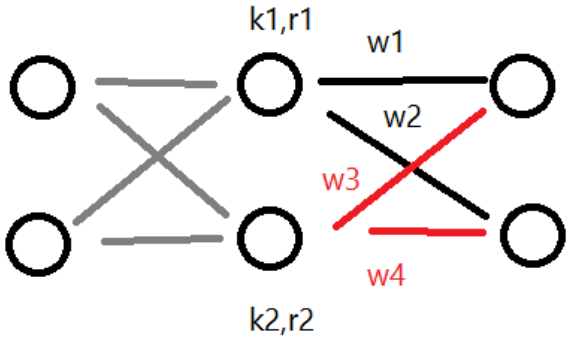

Fig. 6. 2,2,2 local minima

2) **One lies in region III and another lies in region IV**: This implies one is turned on for both examples and other is turned off for both. This is essentially equal to network with one **linear** neuron (2,1,2). Global minima in this region is equal to global minima for 1 neuron network.

$$L = (k_2 * w_3 - y_1)^2 + (k_2 * w_4 - y_2)^2 + (r_2 * w_3 - y_3)^2 + (r_2 * w_4 - y_4)^2$$

Local minimas (since it is linear 2,1,2) in this region are same as global minima in this region. This will be different than G_{all} when 1-neuron network does not give zero loss (ie when the outputs are linearly independent).

3) **One lies in region I/II and another lies in region IV**: This means one neuron is turned off for one example

and other neuron is turned off for both. Similar to the local minima case discussed in 2,1,2.

$$L = (k_1 * w_1 - y_1)^2 + (k_1 * w_2 - y_2)^2 + (y_3)^2 + (y_4)^2$$

4) **One lies in region I/II and another lies in region III**: In this case, one neuron is turned off for one example ($r_1 = 0$) and other is on for both examples.

$$L = (k_1 * w_1 + k_2 * w_3 - y_1)^2 + (k_1 * w_2 + k_2 * w_4 - y_2)^2 + (r_2 * w_3 - y_3)^2 + (r_2 * w_4 - y_4)^2$$

$$\frac{dL}{dw_1} = 2(k_1 * w_1 + k_2 * w_3 - y_1) * k_1 = 0 \implies k_1 * w_1 + k_2 * w_3 = y_1$$
$$\frac{dL}{dw_2} = 2(k_1 * w_2 + k_2 * w_4 - y_2) * k_1 = 0 \implies k_1 * w_2 + k_2 * w_4 = y_2$$
$$\frac{dL}{dw_3} = 2(k_1 * w_1 + k_2 * w_3 - y_1) * k_2 + 2(r_2 * w_3 - y_3) * r_2 = 0 \implies r_2 * w_3 = y_3$$
$$\frac{dL}{dw_4} = 2(k_1 * w_2 + k_2 * w_4 - y_2) * k_2 + 2(r_2 * w_4 - y_4) * r_2 = 0 \implies r_2 * w_4 = y_4$$

That is all derivatives equal to zero implies loss should be zero. So, local minima in this region is same as global minima in this region. And this global minima is equal to G_{all}.

5) **One lies in region I/II and another lies in region II/I**: In this case, one neuron is turned off for one example ($r_1 = 0$) and other is off for other different example ($k_2 = 0$).

$$L = (k_1 * w_1 - y_1)^2 + (k_1 * w_2 - y_2)^2 + (r_2 * w_3 - y_3)^2 + (r_2 * w_4 - y_4)^2$$

$$\frac{dL}{dw_1} = 2(k_1 * w_1 - y_1) * k_1 = 0 \implies k_1 * w_1 = y_1$$
$$\frac{dL}{dw_2} = 2(k_1 * w_2 - y_2) * k_1 = 0 \implies k_1 * w_2 = y_2$$
$$\frac{dL}{dw_3} = 2(r_2 * w_3 - y_3) * r_2 = 0 \implies r_2 * w_3 = y_3$$
$$\frac{dL}{dw_4} = 2(r_2 * w_4 - y_4) * r_2 = 0 \implies r_2 * w_4 = y_4$$

Same as before, that is all derivatives equal to zero implies loss should be zero. So, local minima in this region is same as global minima in this region. And this global minima is equal to G_{all}.

6) **Both lie in region I/II**: In this case, both neurons are turned off for one example ($r_1 = 0$ and $r_2 = 0$) and turned on for other example.

$$L = (k_1 * w_1 + k_2 * w_3 - y_1)^2 + (k_1 * w_2 + k_2 * w_4 - y_2)^2 + (y_3)$$
$$L \geq (y_3)^2 + (y_4)^2$$

$$\frac{dL}{dw_1} = 2(k_1 * w_1 + k_2 * w_3 - y_1) * k_1 = 0 \implies k_1 * w_1 + k_2 * w_3 = y_1$$
$$\frac{dL}{dw_2} = 2(k_1 * w_2 + k_2 * w_4 - y_2) * k_1 = 0 \implies k_1 * w_2 + k_2 * w_4 = y_2$$
$$\frac{dL}{dw_3} = 2(k_1 * w_1 + k_2 * w_3 - y_1) * k_2 = 0 \implies k_1 * w_1 + k_2 * w_3 = y_1$$
$$\frac{dL}{dw_4} = 2(k_1 * w_2 + k_2 * w_4 - y_2) * k_2 = 0 \implies k_1 * w_2 + k_2 * w_4 = y_2$$

All local minimas are global minimas in this region. Also, the global minima for this region would be local minima for all cases, since this region's loss will be grater than norm of 2nd example's expected result.

This finishes the characterization of all local minimas for 2,2,2 neural network (other than 2,1,2 case). Local minimas (different from global minima) can only exist in case 2, 3 and 6.

ACKNOWLEDGMENTS