UNIT 3 : Neural Networks

LECTURE 8 : Introduction to FeedForward Neural Networks

Wooooooooo !!!!!!! We are back baby !!!!! Yeahhhhhh !!!!!

After the successful completion of Fundamentals of Statistics , your pal **Scooby** is back with a brand new set of handwritten lecture notes. I found that many of you liked them and appriciated the work by thanking me personally on my email. Which by the way is :

Scooby's email : **peepeepoopoo873@gmail.com**

Feel free to point out mistakes and message me anyways

For those who do not message me...
I don't care man!!! Whataever man !!!! Screw you guys !!!! Whatever!!!

(I do care, please message me. I was just pretending to be indifferent in the above line so that you can't hurt me emotionally.) Anyways.

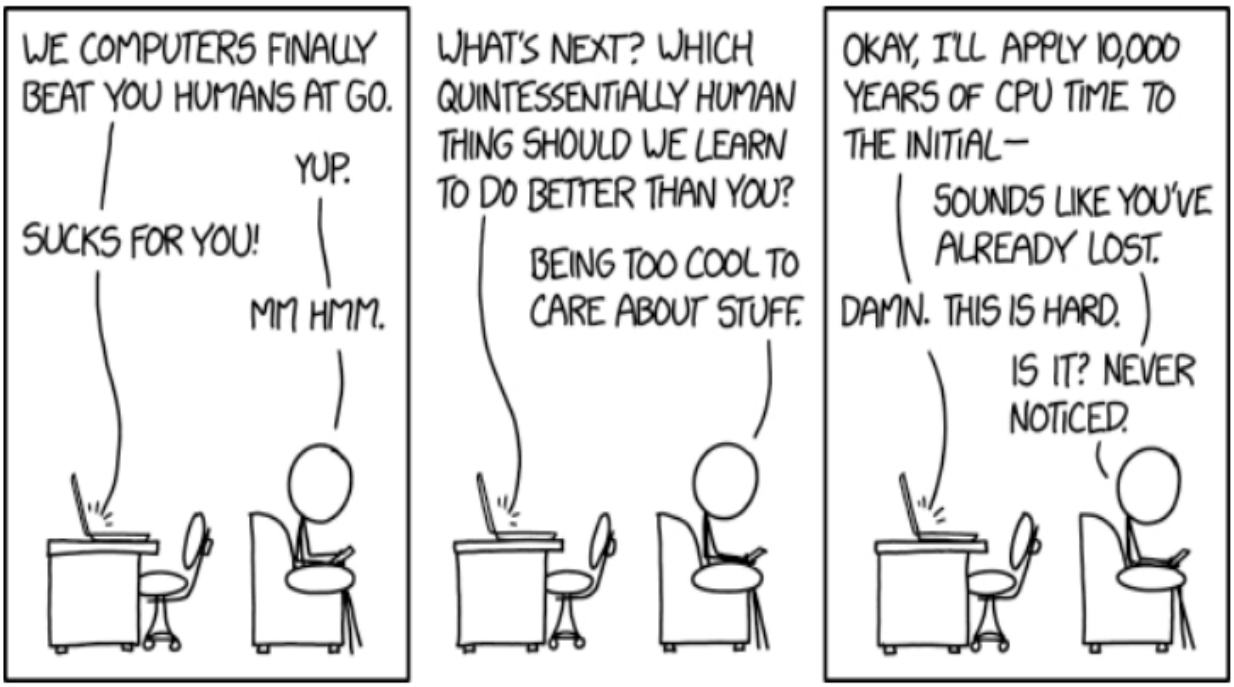
I tried to make these notes without any mistakes and cover most of the video material.

But as Yoda has rightly said : ' *Is human, To err* '

I however **DO NOT TAKE** any responsibilty if you fail your exams following my notes. So please check the actual lecture slides and videos if you suspect something is wrong, and mail me too.

The notes **DO NOT CONTAIN** solutions to any homework problems or in-lecture exercises. Only the problems and exercises solved in the videos are included in these notes.

Here is an interesting comics I found



Source : https://xkcd.com/1875/

I ran out of made up quotes so I will write very short stories instead. These may or may not resemble true events, but will definitely have a message that will help you along the journey, that is life.

STORY TIME : The Beast within

There was a man who was cursed by a witch. He turned into a warewolf at every full moon night. You know the usual Twilight stuff. But there was one problem. He was not hot. So chicks didn't find him sexy, like the Twilight heros and villians.

So he was ugly and he turned into a warewolf every fortnight. One evening he saw a beautiful chick near his farmhouse. (Yeah!! he lived in a farmhouse so what??). She was relly pretty with green eyes and blode hair. Her car had broken down. So he went up there to help her. The carburetor of the car had burnt off, and all the repair shops were far away, so he said that to the girl that she will have to wait till tommorow to fix the car.

He offered her to stay at his farmhouse for the night. But there was one condition she must obey. She should not at any cost see him after 12 in the night. The reason which he did not tell the chick was obviously about him turning into a warewolf as it was a full moon night that day.

She hesitatingly agreed. The night dawned and the guy turned into a warewolf. He started howling. The chick first resisted to leave the room as she remembered her promise but she could not any longer. She sneaked out of her room and saw the man transformed as a warewolf. The wolf caught her eye. But to his suprise the girl was also in a wolf fom. She took the form of a majestic she-wolf, with shiny silver fur all over her body, and a long tail. They both stared at each other. The guy wolf ran away. She went back to her room.

The next morning, she in her human form met the guy also in his human form. She said

' I have been cursed by the same witch who cursed you. I turn into a she-wolf just like you every full moon night. Yesterday when I saw you for the first time in your human form, you were just another guy. But when I saw your warewolf form at night, it's like we were meant to be together. What are the odds that my car would break down in front of your house. I believe that the Universe wants us to be together, What do you say ' she asked

The guy said

You know yesterday when I saw you in your human form, I found you as one of the most beautiful girls in the world. I would have given anything to be with you. But when later at night I saw you in your She-wolf form, it was gross. I can't stand hair on a woman's body, even if it's only twice a month. So here's the mechanic's number.....and..... Goodbye '

THE END
MORAL : An ugly person can have standards too, you Bigot

I hope these notes will help you

Let's get started

▷ Motivation

Let's recall the idea we discussed till now for non linear classification. We most generally had a d dimensional feature vector $x \in \mathbb{R}^d$ and we wanted to classify points. So we came up with the idea of using a hyperplane with equation : $\theta \cdot x = 0$ so as to classify the points according to the rule $h(x) = \text{Sign}(\theta \cdot x)$ but this approach was pretty limited because only linearly seperable data could be classified properly. So in order to incorporate non linearity, we came up with the concept of a transform $\phi(x) : \mathbb{R}^d \to \mathbb{R}^D$, where D was generally greater thand The data that wasn't seperable in lower (d) dimensions, was often seperable in higher (D) dimensions. & that enabled us to classify

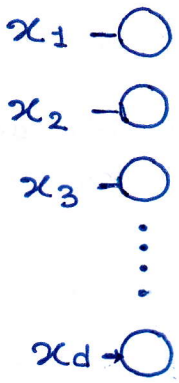
$$h(x) = (\theta \cdot \phi(x)) \qquad \theta \in \mathbb{R}^D \text{ (not the above } \theta)$$

Lets write this in a graphical way, to motivate the functioning of a neural network

> Node

The first thing that will be introduced here is a node. It will be represented graphically as a circle ○. A node will take certain (1 or more) inputs and give out one or more outputs. The input will come in from the Left side of the node and output will go to the right side of the node. The way we define the node depends on what the node does to the input in order to produce an output.

- So here we line up d nodes for d elements of a vector $x \in \mathbb{R}^d$ Each node stores the value of that particular element. We represent it by the following :



- Next step we did was to transform these vectors x using our transform $\phi(x)$ if you recall $\phi(x)$ was actually

$$\phi(x) = [\phi_1(x), \phi_2(x), \ldots\ldots\ldots \phi_D(x)]^T$$

D usually greater than d

ie $\phi(x)$ is a list of D multivariable functions

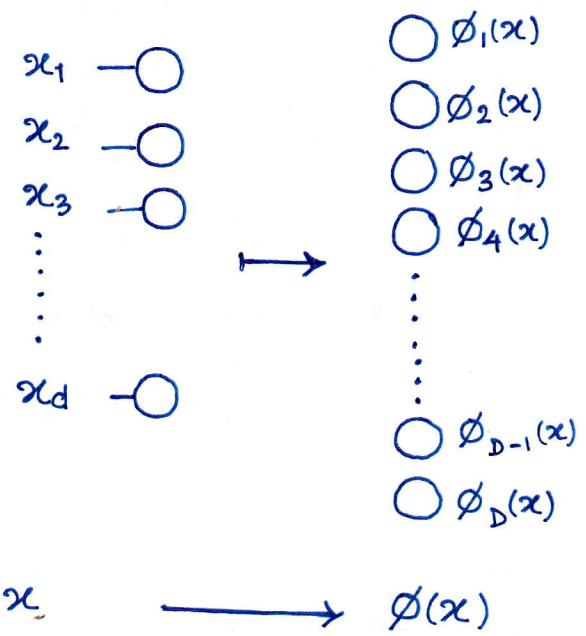
Now here comes the 2nd terminology

- To each node is associated a <u>transfer function</u>. It's a function that the particular node applied to the input. It's not very important piece of terminology, But this gives us an idea of what a node actually is

A node is a computational unit. It can take one or more inputs and operating on that input with the function it has associated. It gives one or more outputs.

The 1st line of nodes we drew for $x_1, x_2, x_3 \ldots\ldots x_d$ does nothing and return the same output as the input.. we can think of them as having associated to the identity function

- Lets continue to graphically represent our transform. For each node we will associate the ith function $\phi_i(x)$



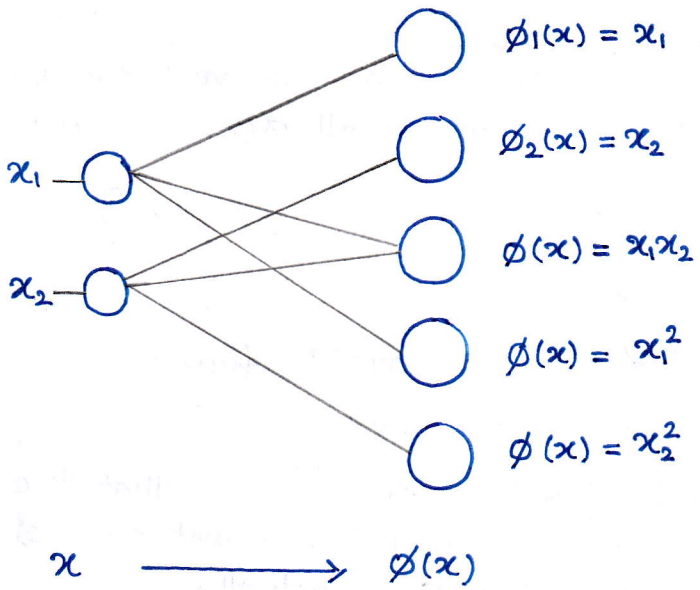
We have d nodes for x and D for $\phi(x)$ the outputs from the d nodes of x become the inputs for $\phi(x)$ nodes

Lets see how this will play out in an example

Example Suppose we have a 2 dimensional vector $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$

And suppose $\phi(x)$ is : $[x_1, x_2, x_1x_2, x_1^2, x_2^2]$

Lets see how the graph looks



Here $d = 2$ because $x \in \mathbb{R}^2$

$D = 5$ because $\phi(x) \in \mathbb{R}^5$

Note how, not all nodes are connected to all nodes before them

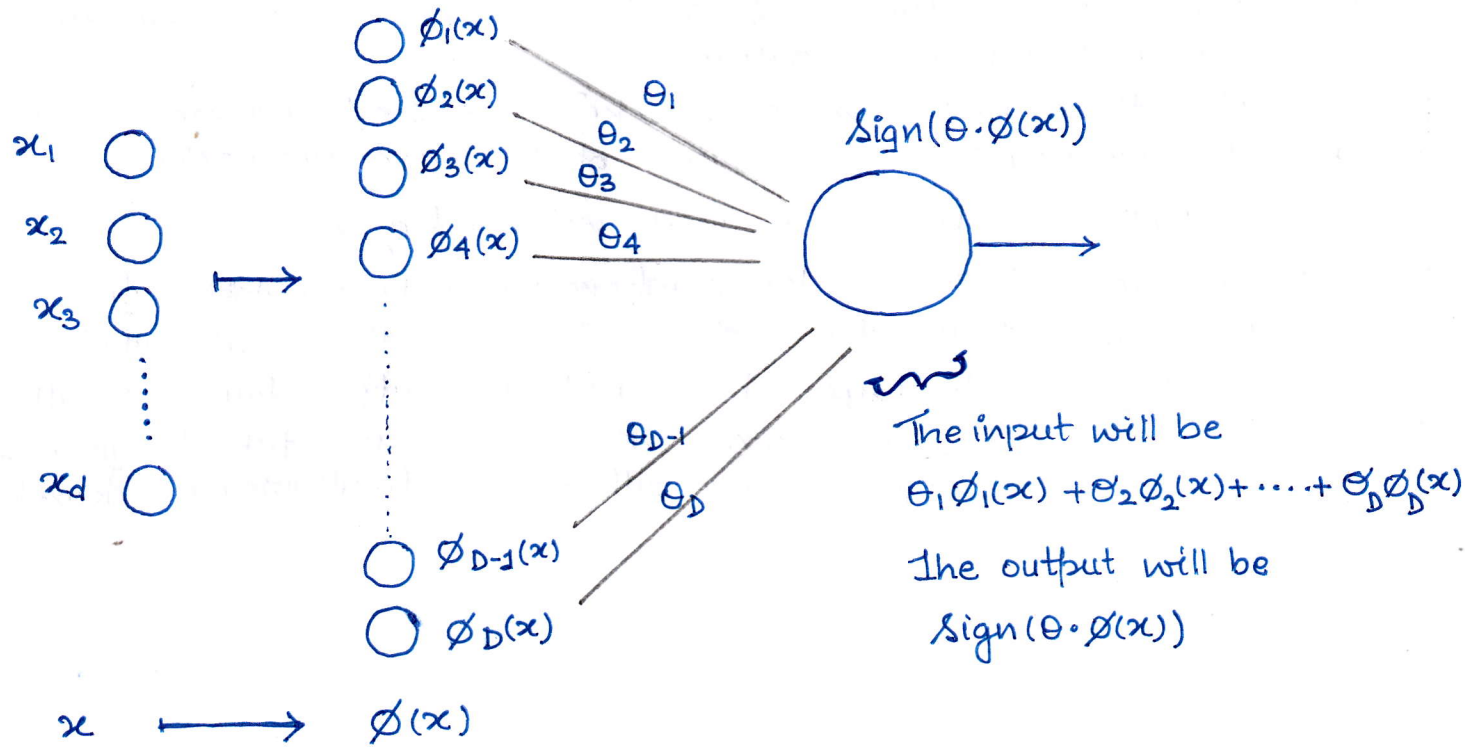
OK Lets move forward

Nowonce we have this feature representation we can take the Linear combination of all these to form $\theta \cdot \phi(x)$

We write the coefficient/ parameter θ_i over the output from $\phi_i(x)$ node. This means the output from $\phi_i(x)$ node should be weighed (ie multiplied) by θ_i to go as input to the final node.

This final node will have associated with it the classifier function that is $\text{Sign}(\theta \cdot \phi(x))$ for a general $x \in \mathbb{R}^d$

So the picture finally looks like



Now till now, we decided what $\phi(x)$ will be.. we were the ones who decided what the mappings for $\phi(x)$ will be, ie the functional form of ϕ_i's

But the topic of neural networks we now see, will itself try to optimize or find the best transforms for the problem at hand. Ie it will try to find the best feature transform or representation of the raw input x, that best suits our problem & data in hand

We are in a chicken egg situation. We want the algorithm to find the best transforms to get the best classification. Yet we will only be able to find these transforms when we have have the best classification. Yet the best classification will only be found once we have the best transforms and so on.

So Let's begin the study of....

NEURAL NETWORKS

▷ Neural Networks

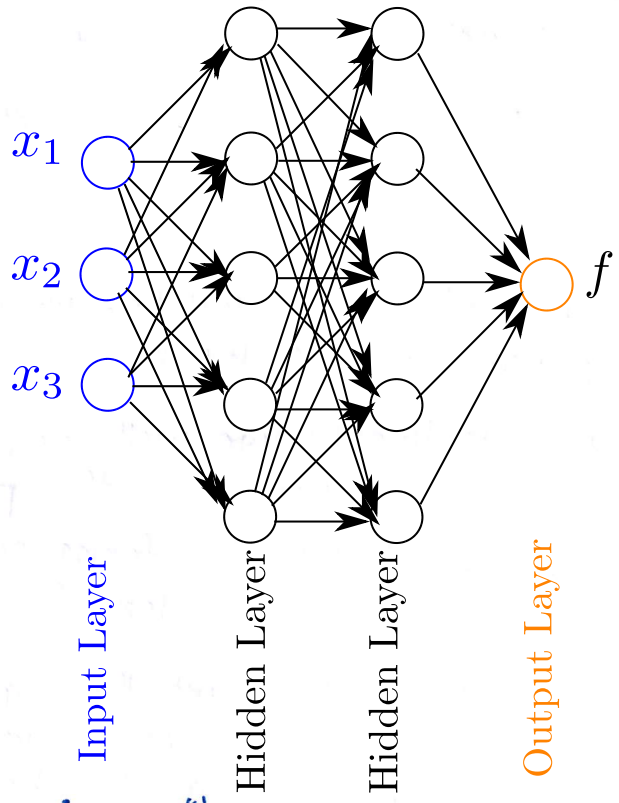
Now inspired from the model of a brain neuron scientists/engineers came up with the idea of making a brain like thingy for machines. And that is what we call a neural network. So in order to understand this we will need to learn some terminology, that will be helpful. Let's see

> Layer : we already know what a node is. Represented by 'O', node takes in one or more inputs and gives out one or more output. To each node is an associated function, denoted by f(). This function acts on the input to produce an output..

Now when these nodes are all stacked in a single column one above the other we say that it is a layer of the neural network

For eg. Below we have a network with 4 layers

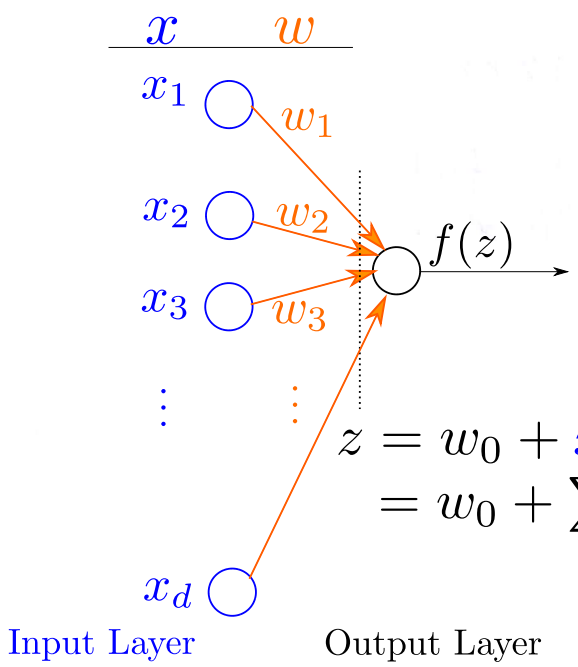
- Input Layer : Is the 1st layer, where our collected data is fed
- Output Layer : Is the layer that will give us the final output
- Hidden Layers : The layers bw input and output layer are all called hidden layers. These layers will take the input from behind and give output to the front. (Is that dirty? I don't know!)



Next piece of terminology is a unit

> Neural Network Unit : Is a primitive neural network component. A complete neural network can be thought of being made many neural network units It consists of an input layer, and an output layer. But the output layer gives only one output. This can be summarized as follows

The input layer will feed the input to the output node. This input is usually represented by the letter z. . The output node will act on this z to give us the output (single) f(z). The z usually is the weighted sum of different input's components weighed by the corresponding weights written over the arrows. Let's see an example of a unit.



$$z = w_0 + x \cdot w$$
$$= w_0 + \sum_{i=1}^{d} x_i w_i$$

> Activation function

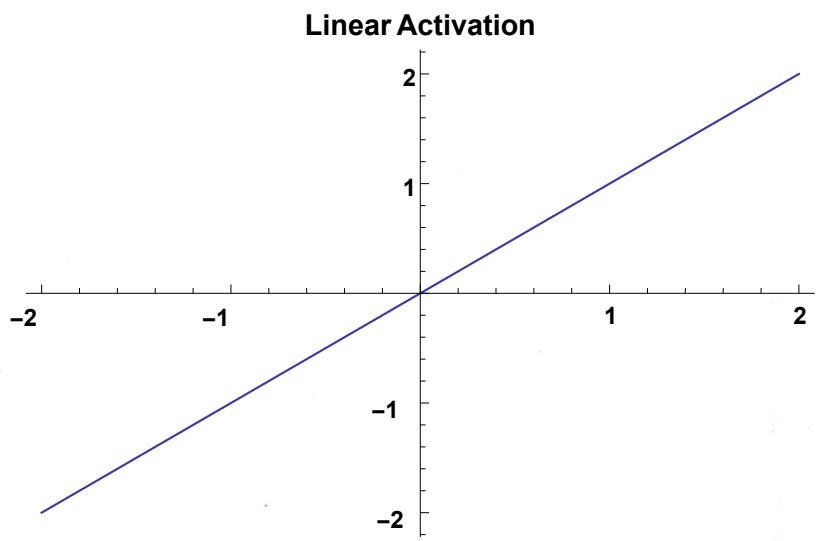
As discussed earlier, each node has with it associated an activation function denoted by f(z). This is usually (not always) a non-linear function. There are some very famous activation functions, let's look at some:

- Linear function

Defined by

$$f(z) = z$$

It just takes in the input z and gives out the same z

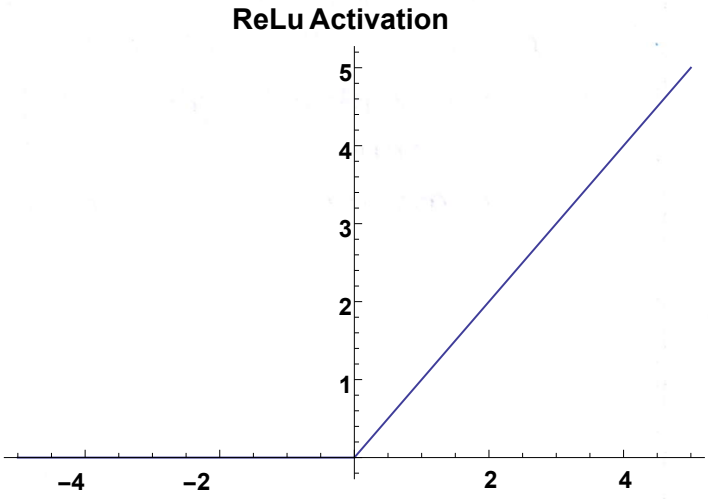


- ReLU : Rectified linear Unit

Defined by

$$f(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}$$

or equivalently $f(z) = \max\{z, 0\}$

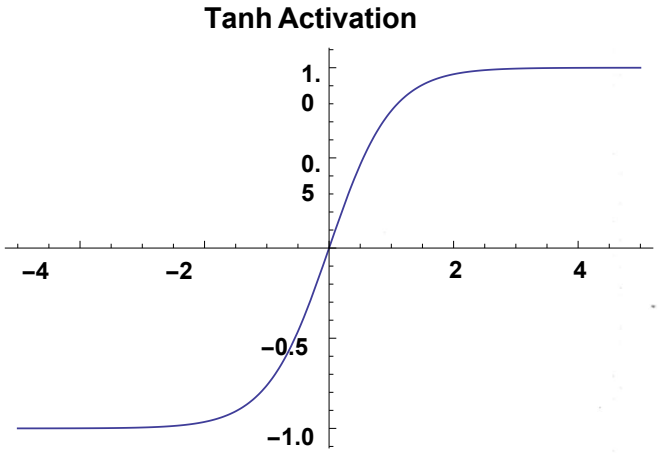


think in the case of SVM. The Loss function Looked like this with $z = y^{(i)}(\theta \cdot x^{(i)} + \theta_0)$

- Tanh : Activation

defined by

$$\text{Tanh}(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

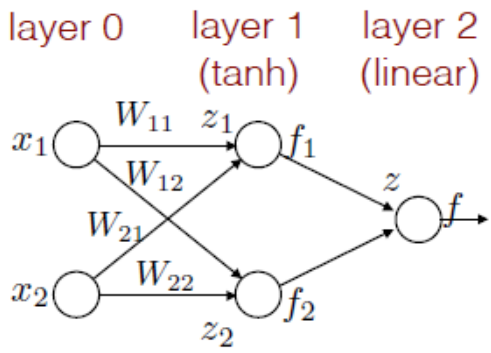


This can be seen as an alternative to Sign(z) where Sign() directly maps any z no matter how small or big to $\{+1, -1\}$. Tanh(z) is a softer version of this. It assigns output magnitudes in the interval $(-1, 1)$ according to the input magnitude and sign of z

Now lets see what happens when we add layes with these functions into the network

▷ One Hidden layer model

In order to see how a neural network works, we will simply see what happens step by step as we proceed in a single hidden layer model shown below



First let be clear that we understand the model above

→ Layer 0 is always the input raw data. we store two real values in the layer 0, which means our input vector x is two dimensional $x \in \mathbb{R}^2$. Ok so that is all about Layer 0

→ Next we see the hidden layer ie Layer 1. we are given that it has the transfer function tanh associated to each node. Ie $f_1(z) = \tanh(z)$ $f_2(z) = \tanh(z)$.

Now the input z_1 to the 1st node in 1st layer is the linear combination of the x_1 and x_2, weighted by W_{11} & W_{12} respectively. An intercept term will also be added for generality. so

$$z_1 = x_1 W_{11} + x_2 W_{21} + W_{01}$$

NOTE

W_{ij}

from to

Similiarly

$$z_2 = x_1 W_{12} + x_2 W_{22} + W_{02}$$

Ok that's the input for the 1st Layer

Now the nodes in layer 1 will operate on the inputs they get respectively so the output of 1st node in 1st Layer will be

$$f_1 = \tanh(z_1) = \tanh(x_1 W_{11} + x_2 W_{21} + W_{01})$$

So the output of 2nd node in 1st Layer will be

$$f_2 = \tanh(z_2) = \tanh(x_1 W_{12} + x_2 W_{22} + W_{02})$$

→ Th output will then go the final node weighted by some parameters W''s as inputs

The final node will give us the final output

$$f = z_1 W_1' + z_2 W_2' + W_0' \quad \text{for some } W'$$

Here we are given that the output layer is linear.

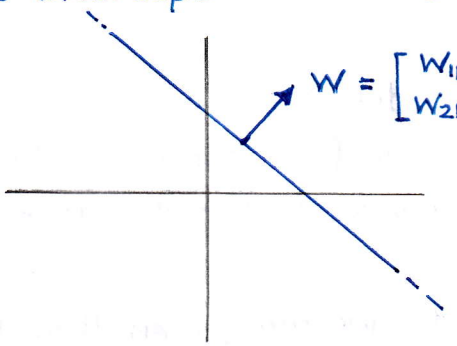
This may look simple mathematically but geometrically something beautiful is happening. Let's carefully analyse it.

> Geometrical Picture

Let's first look at the input to the 1st layer's 1's node

$$z_1 = x_1 W_{11} + x_2 W_{21} + W_{01}$$

This seems extremely familiar. Wonder if we have seen it somewhere. You guessed it right. $x_1 W_{11} + x_2 W_{21} + W_{01}$ is the equation of hyperplane (here line) in the $x_1 - x_2$ space with the normal vector $\mathbf{W} = [W_{11}, W_{21}]^T$ and intercept term W_{01}



So z_1 actually has to do something with this line (hyperplane generally)

And if you remember from lectures 1 & 2, we saw what the following means

Suppose we have the general ith point $x^{(i)} \equiv (x_1^{(i)}, x_2^{(i)})$

If $x^{(i)} W + W_{01} \equiv x_1^{(i)} W_{11} + x_2^{(i)} W_{21} + W_{01} = 0$

This means point $x^{(i)} \equiv (x_1^{(i)}, x_2^{(i)})$ lies on the line

If $x^{(i)} \cdot W + W_{01} \equiv x_1^{(i)} W_{11} + x_2^{(i)} W_{21} + W_{01} > 0$

This means point $x^{(i)} \equiv (x_1^{(i)}, x_2^{(i)})$ lies to the side of line towards which vector W points

If $x^{(i)} \cdot W + W_{01} \equiv x_1^{(i)} W_{11} + x_2^{(i)} W_{21} + W_{01} < 0$

This means point $x^{(i)} \equiv (x_1^{(i)}, x_2^{(i)})$ lies to the opposite side of the line, away from the direction towards which vector W points

Ok this is about the sign of $z^{(i)} = x_1^{(i)} W_{11} + x_2^{(i)} W_{21} + W_{01}$

Let's talk about it's magnitude. We also proved that the perpendicular distance of point $(x_1^{(i)}, x_2^{(i)})$ from the line $W \cdot x + W_{01}$ is

$$\frac{| W \cdot x^{(i)} + W_{01} |}{\| W \|}$$

remember that?

So now we have a complete interpretation of $z^{(i)}$

So $z^{(i)} = x_1^{(i)} W_{11} + x_2^{(i)} W_{21} + W_{01} = x^{(i)} \cdot W + W_{01}$

So $\frac{|z^{(i)}|}{\|W\|}$ is the distance of ith point from the line $W \cdot x + W_{01}$

So $|z^{(i)}|$ is $\|W\| *$ distance of ith point from the line

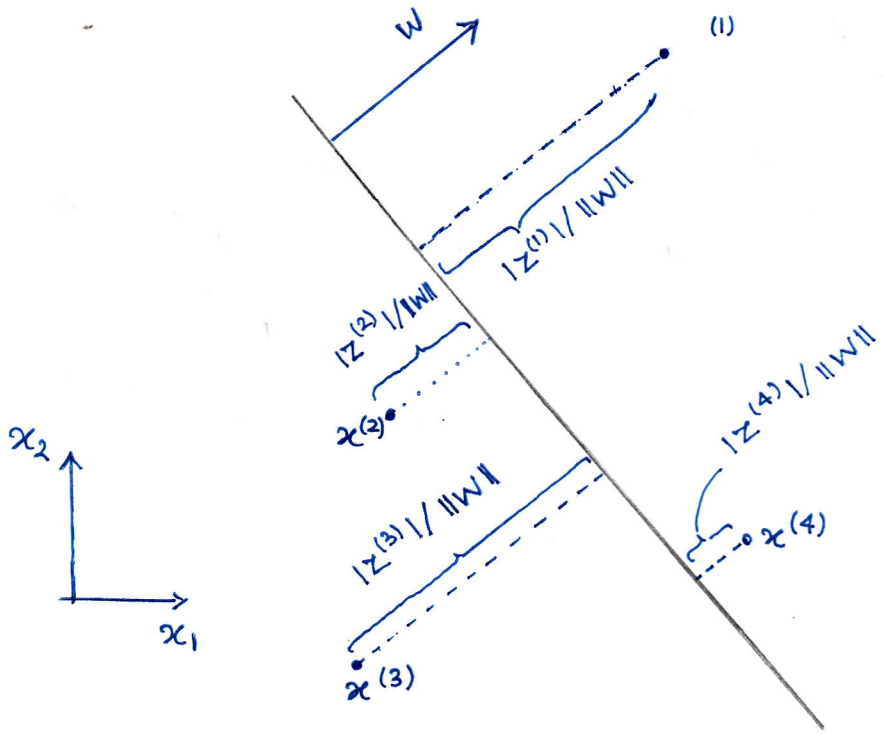
So $|z^{(i)}| \propto$ distance of ith point from the line.

Also we know what the sign of $z^{(i)}$ indicates

So $z^{(i)} \rightarrow$ postive is proportional to distance of ith point that lies to the same side towards which W vector points

$z^{(i)} \rightarrow$ negative is proportional to the distance point that lies to the opposite side towards which W vector points.

Let me show you in a figure



z_1 = positive
z_2 = negative
z_3 = negative
z_4 = positive

$z^{(1)} > z^{(4)}$ because $|z^{(1)}| > |z^{(4)}|$

$z^{(3)} < z^{(2)}$ because $|z^{(3)}| > |z^{(2)}|$

$|z^{(3)}| = |z^{(1)}|$ Because points are equidistant

$|z^{(2)}| > |z^{(4)}|$ Meaning 4 the point is far away the overall

$z^{(3)} < z^{(2)} < z^{(4)} < z^{(1)}$

So even if you don't want to be bothered with the nitty gritty details here's the idea, we need to remember

$\frac{Z^{(i)}}{\|W\|}$ = perpendicular distance of ith point with the information of the side on which ith point fall towards

If positive : ith point is towards the side where W points

If negative : ith point is towards opposite side where W points

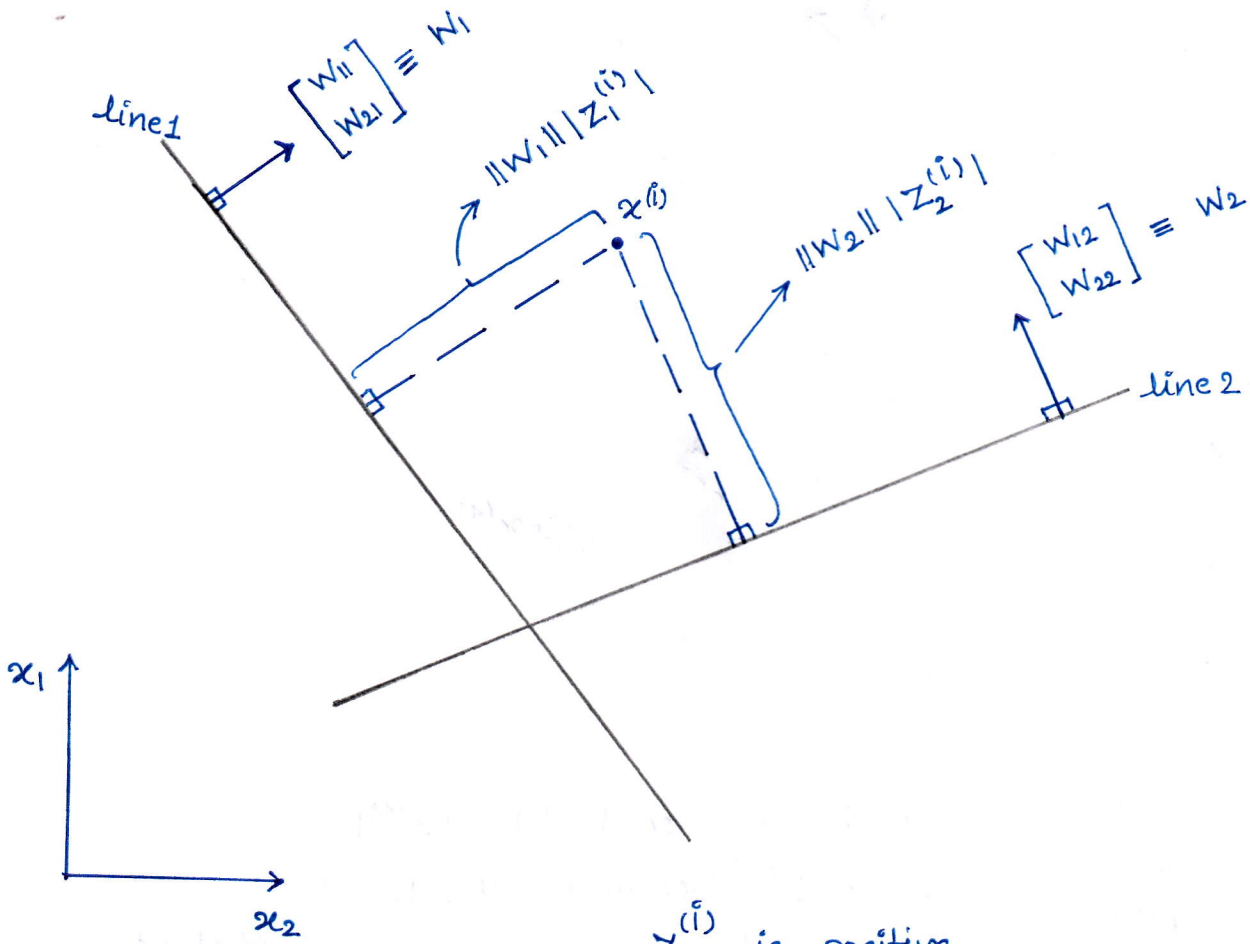
Ok now we are ready to see what happens in the 1st Layer geometrically

We assume the $W_{11}, W_{12}, W_{21}, W_{22}, W_{01}, W_{02}$ are known

So for a given ith point $x^{(i)} \equiv (x_1^{(i)}, x_2^{(i)})$

$Z_1^{(i)} = x_1^{(i)} W_{11} + x_2^{(i)} W_{21} + W_{01}$ will be the $\|W_1\|$ times the perpendicular distance of ith point from the line $x_1 W_{11} + x_2 W_{21} + W_{01}$

$Z_2^{(i)} = x_1^{(i)} W_{12} + x_2^{(i)} W_{22} + W_{02}$ will be the $\|W_2\|$ times the perpendicular distance of ith point from the line $x_1 W_{12} + x_2 W_{22} + W_{01}$



$Z_1^{(i)}$ is positive

$Z_2^{(i)}$ is positive

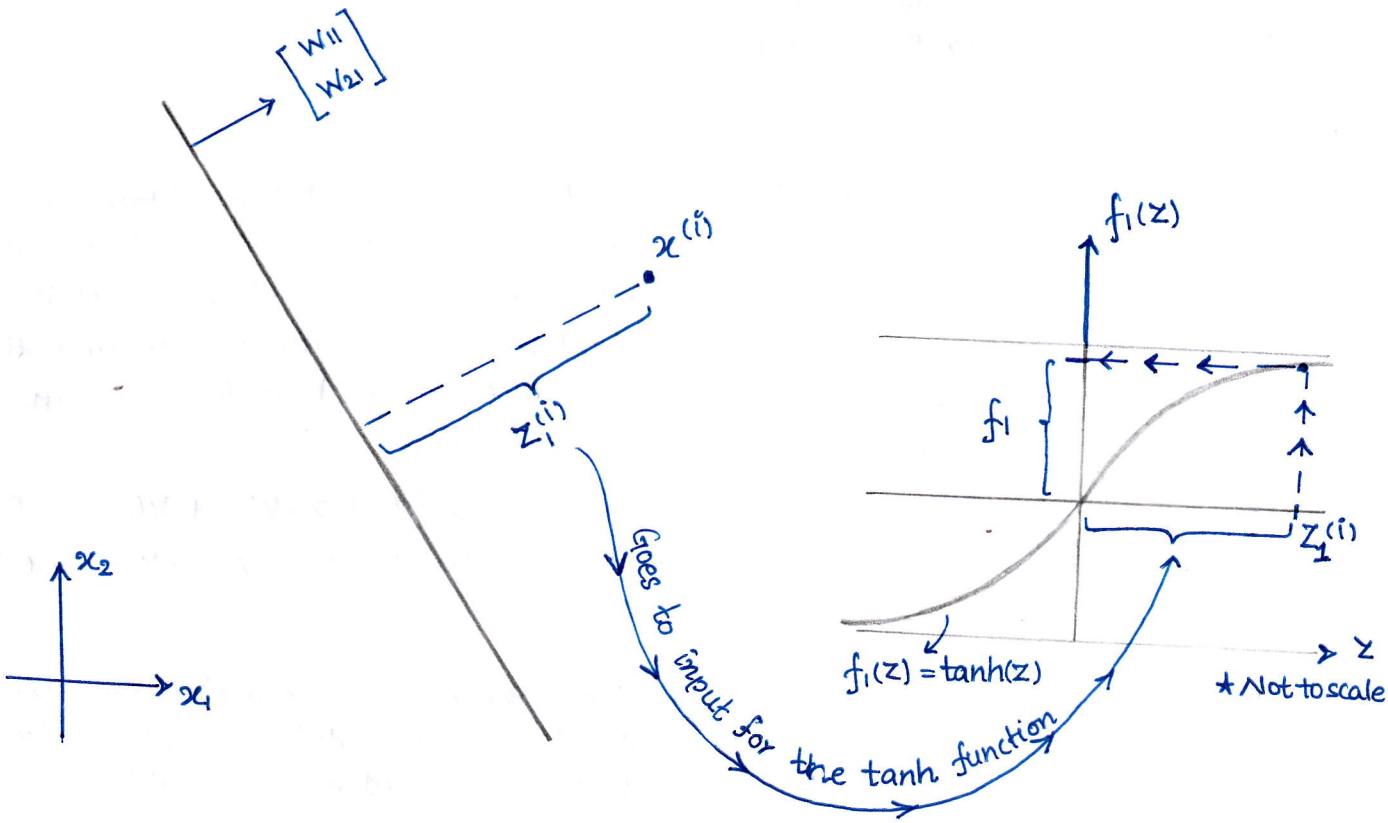
Now let's see what happens when these inputs are fed to the 1st layer
let's see this for the case of z_1 when it goes to 1st node of 1st layer

$z_1 \longrightarrow$ goes to $\longrightarrow$ O $f_1 \longrightarrow$ output is $f_1(z_1)$

Here we assume that the 1st layer has the tanh associated with it.

$$f_1 = \tanh(z_1) = \tanh(x_1 W_{11} + x_2 W_{21} + W_{01})$$

Here's how this looks graphically. For ith point



* Here we assume $\|W_1\| = 1$ for simplicity else the distance will be $z_1^{(i)}/\|W_1\|$

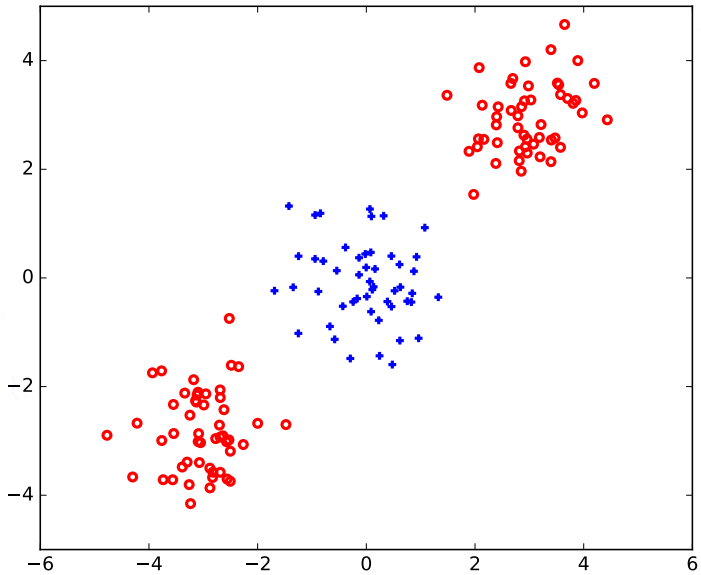
Similiarly we can see what happens in node 2 in the 1st layer

$z_2^{(i)} \rightarrow$ goes as input to tanh() $\rightarrow$ $\tanh(z_2^{(i)})$ is the output f_2

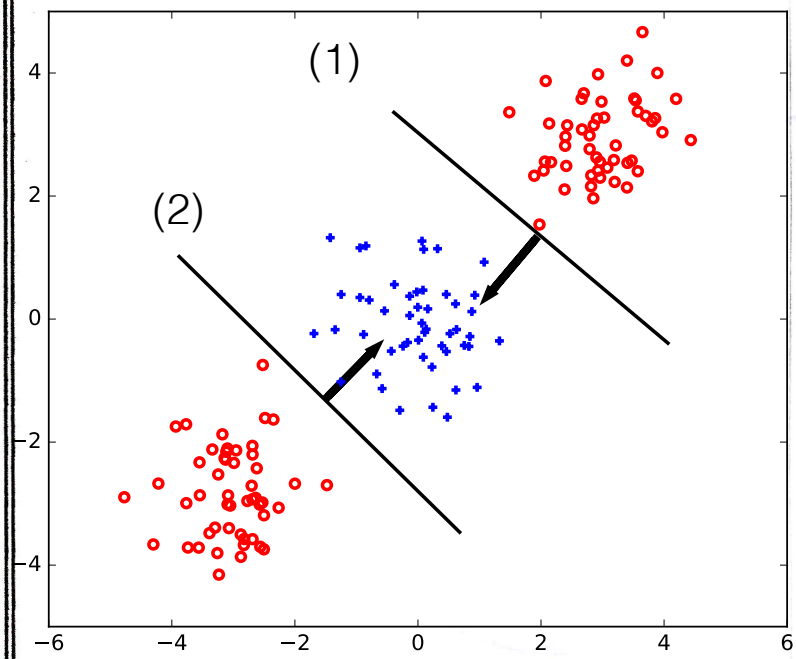
So with these ideas in mind let's see some cases, where data is passed on from one layer to other. We will have $x \in \mathbb{R}^2$ ie $x \equiv (x_1, x_2)^T$

We will see how original two dimensional points in $x_1 - x_2$ space transform to points in $f_1 - f_2$ space

So we will start with these points in the x_1-x_2 space. Clearly the points are not linearly separable in the x_1-x_2 plane But we hope that after passing through some layers they become seperable



We also assume that we somehow know $W_{11}, W_{12}, W_{21}, W_{22}, W_{01}, W_{02}$ and we then can draw the line (generally hyperplane) on the x_1-x_2 space using these values



Suppose the w's that we obtain gives us the lines as shown in the left figure How we find w's is the topic of the next Lecture but we for now assume that w's are given there fore we can draw these two lines

line 1 : $x_1 W_{11} + x_2 W_{21} + W_{01} = 0$

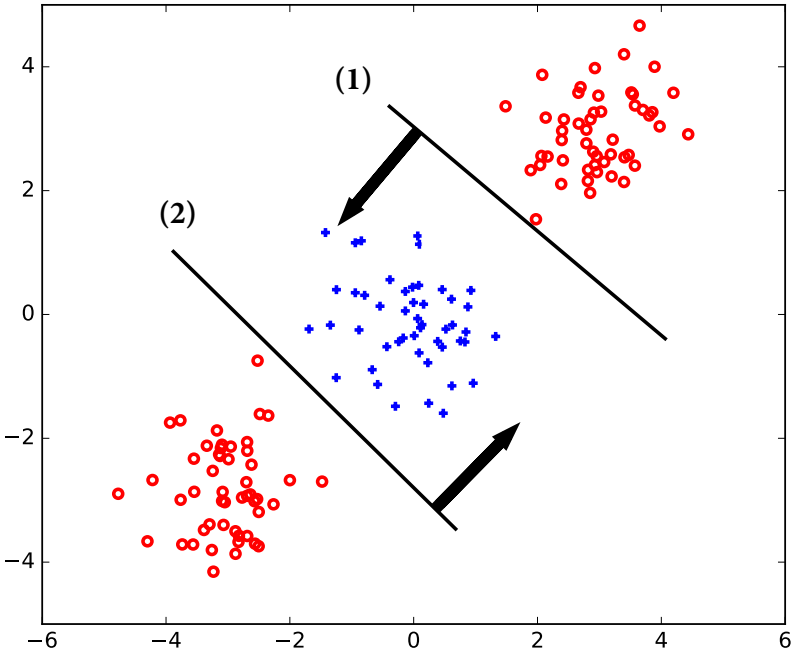
line 2 : $x_1 W_{12} + x_2 W_{22} + W_{02} = 0$

Now the plan is simple. we want to see how these different points transfer to the f_1-f_2 plane using different activations

Now we will see what happens when z1 and z2 are fed to layer 1 We will see three cases, where layer 1 will have three activation functions respectively

> Linear
> ReLU
> Tanh

First we will see how three different points transform, and this wll tell us how all the points transform.



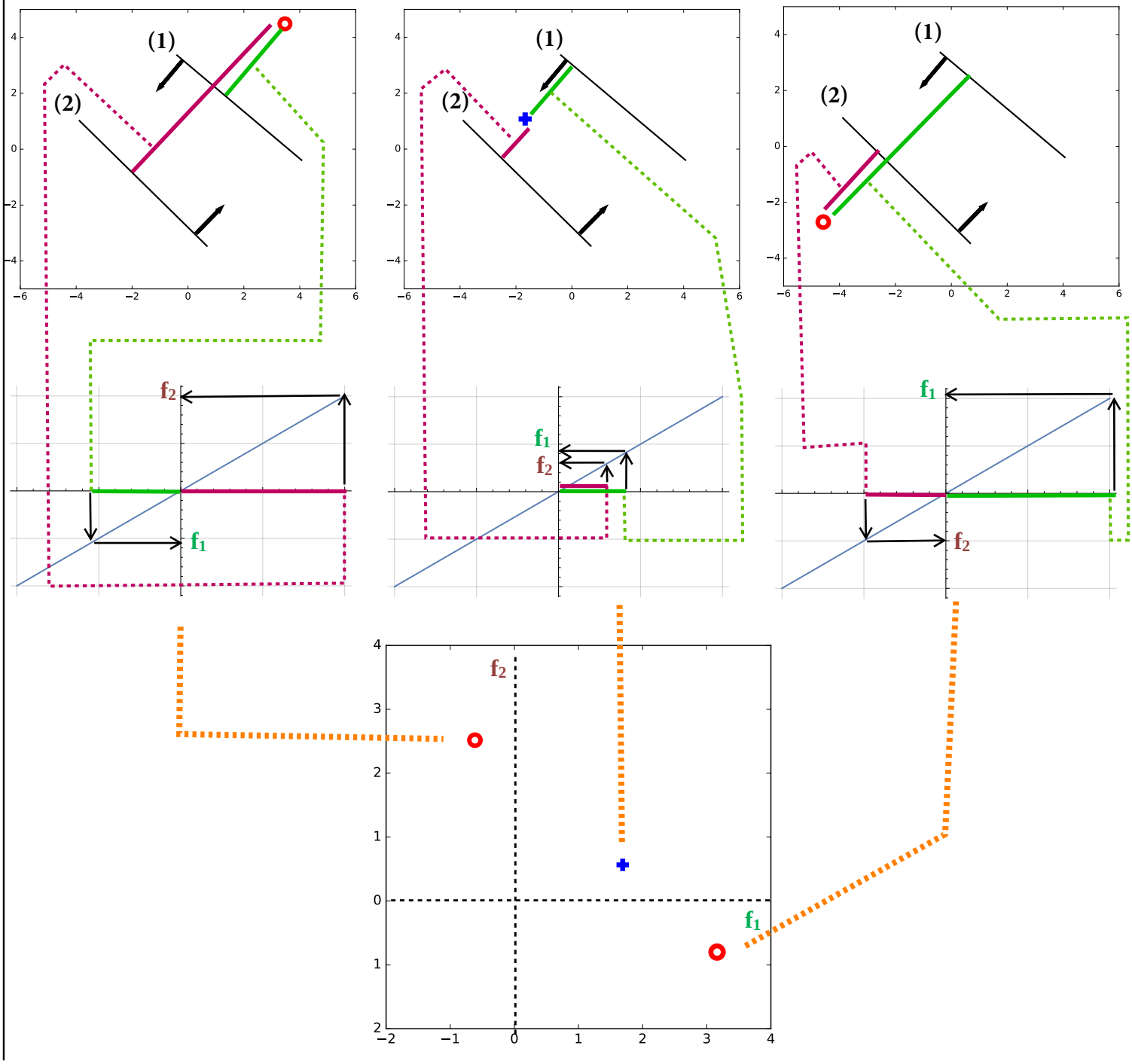
We see the case of three points and how they transform after passing the Linear Activation layer ie we see how each point in x1-x2 space transforms to f1-f2 space

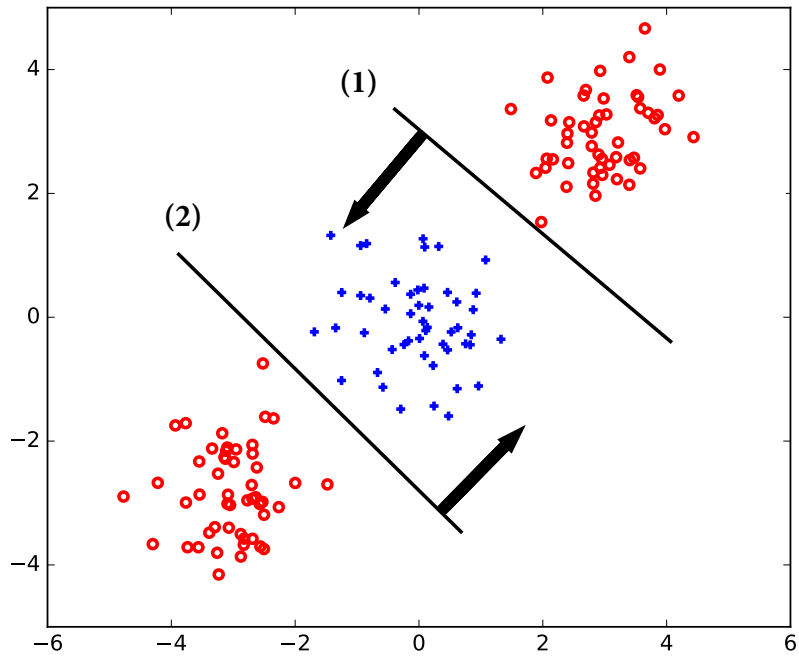
Note that whenever a point is in the same side as w vector of a particular line(1 or 2), its z is positive, otherwise it's negative

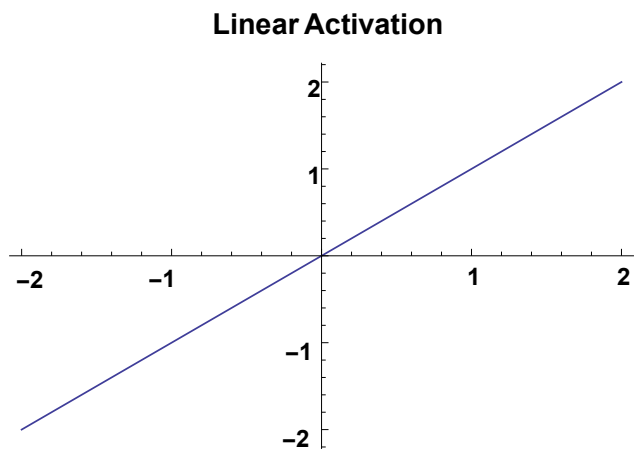
1st Point

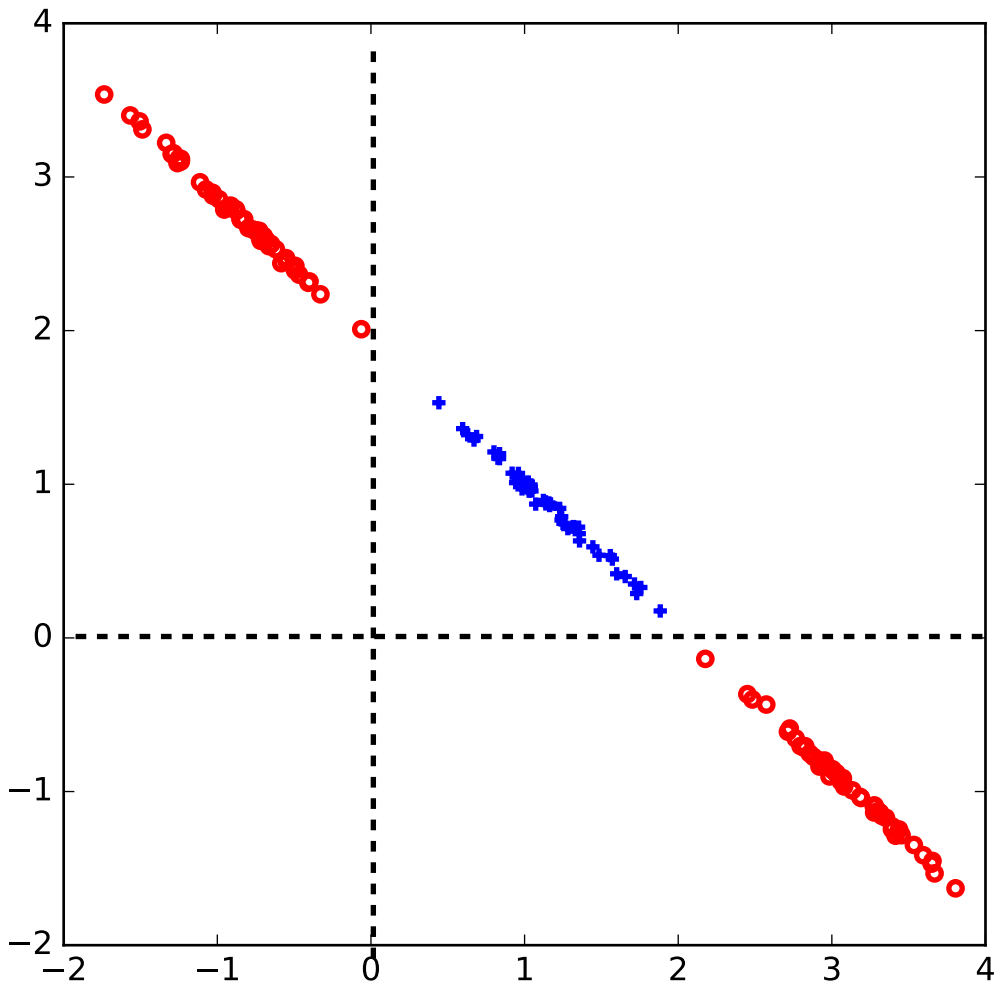
2nd Point

3rd Point



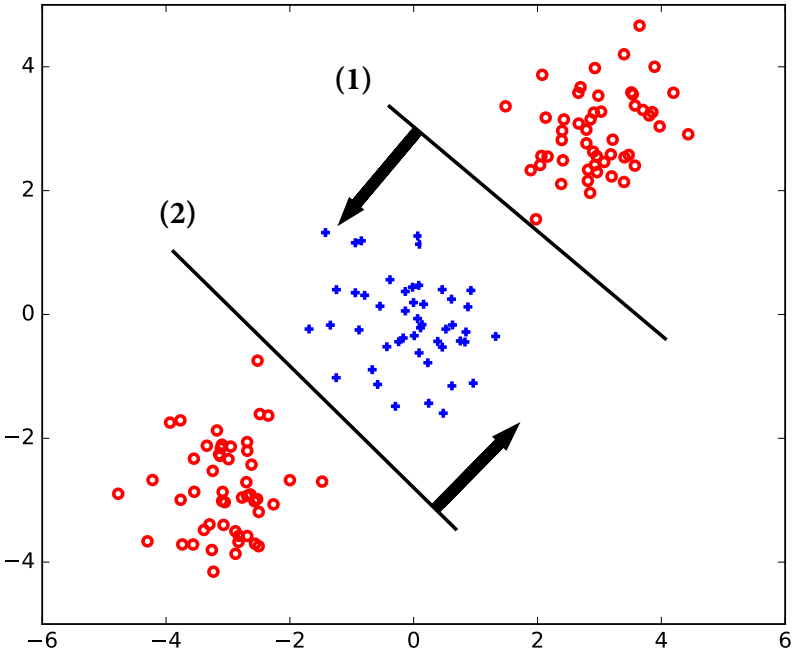






Although we passed the data point through the Linear activtion layer we don't have any special progress. The data is still not seperable in the new f1-f2 space.

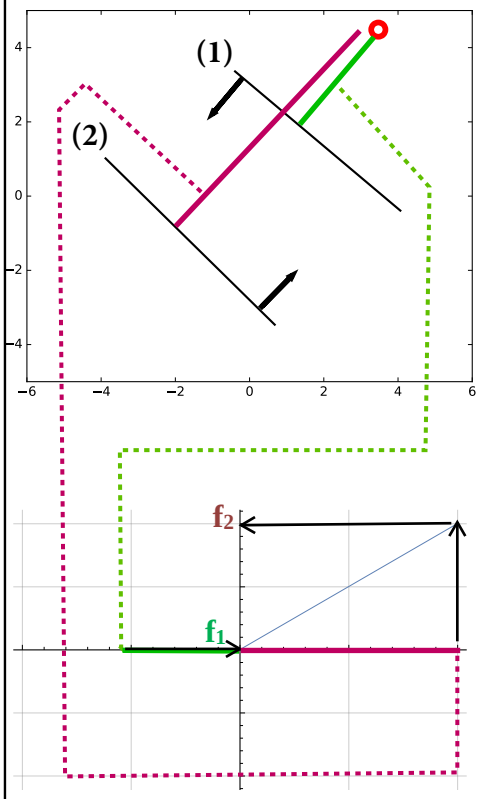
Lets try to see what ReLU activation does.



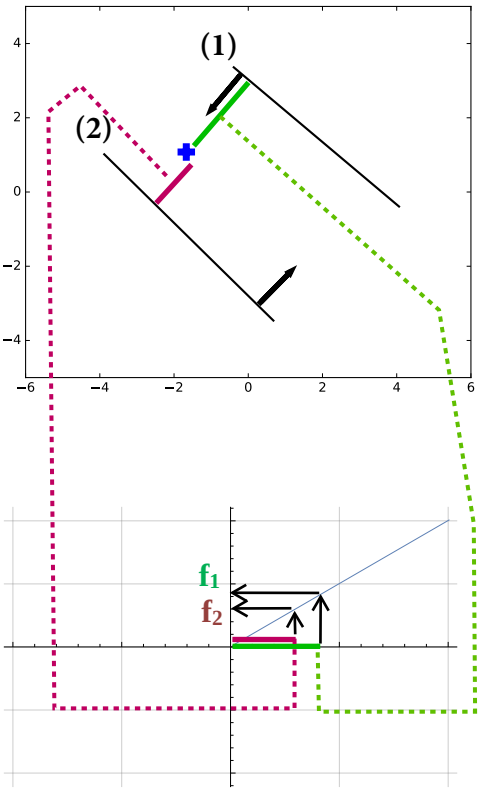
We see the case of three points and how they transform after passing the ReLU Activation layer ie we see how each point in x1-x2 space transforms to f1-f2 space

Note that whenever a point is in the same side as w vector of a particular line(1 or 2), its z is positive, otherwise it's negative

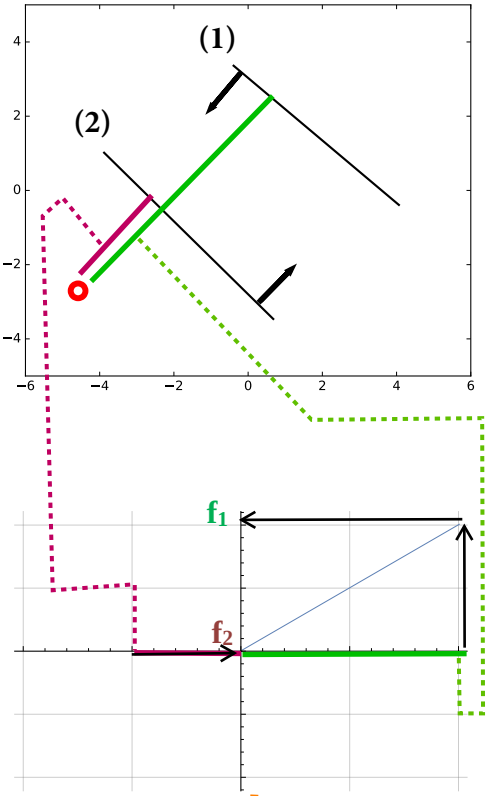
1st Point

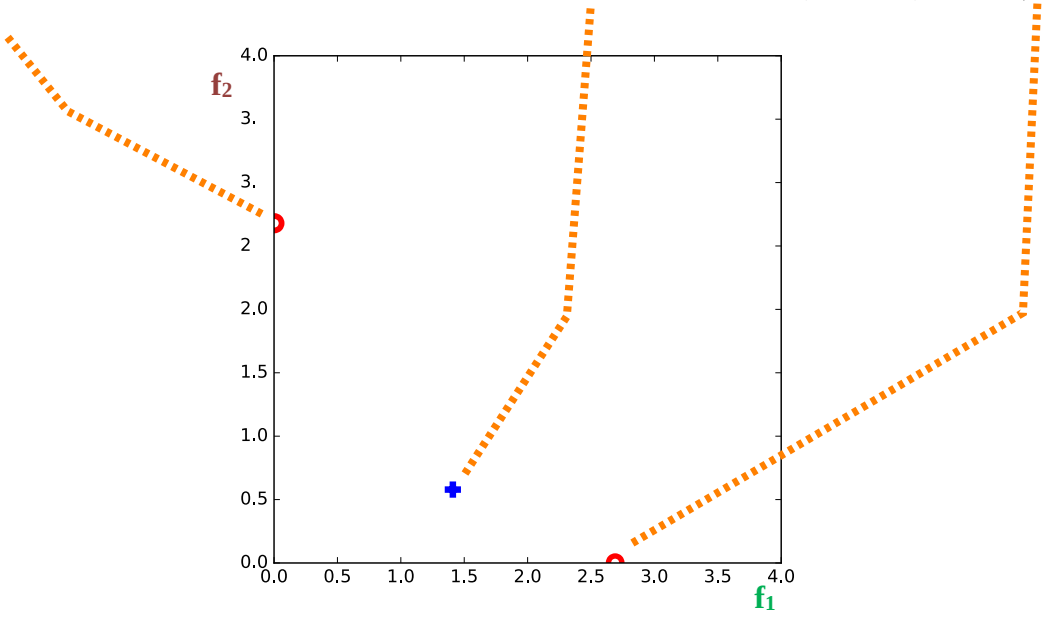


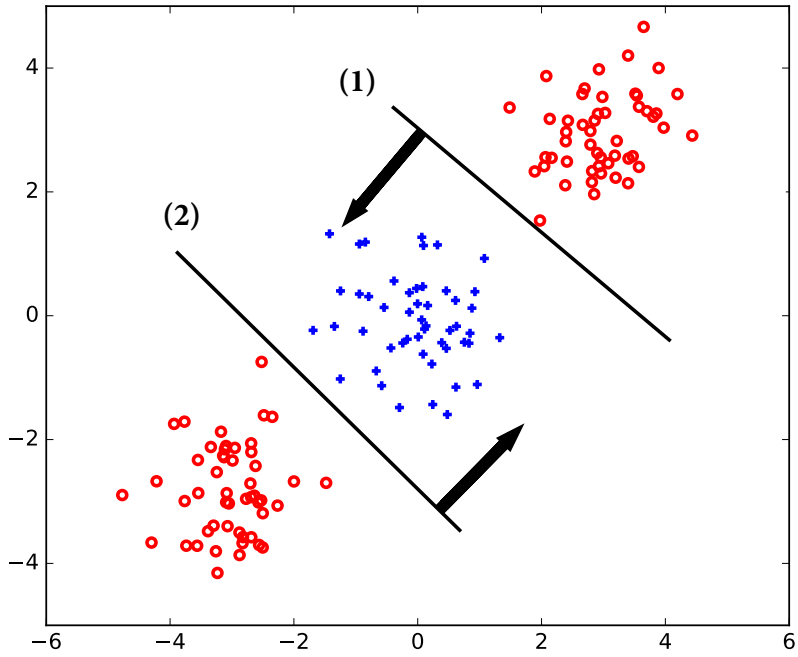
2nd Point

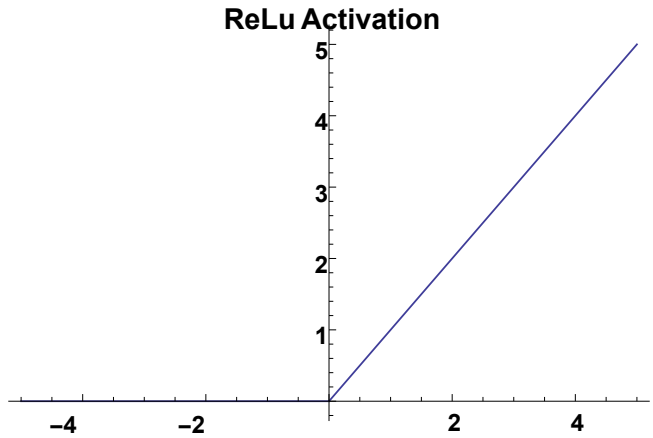


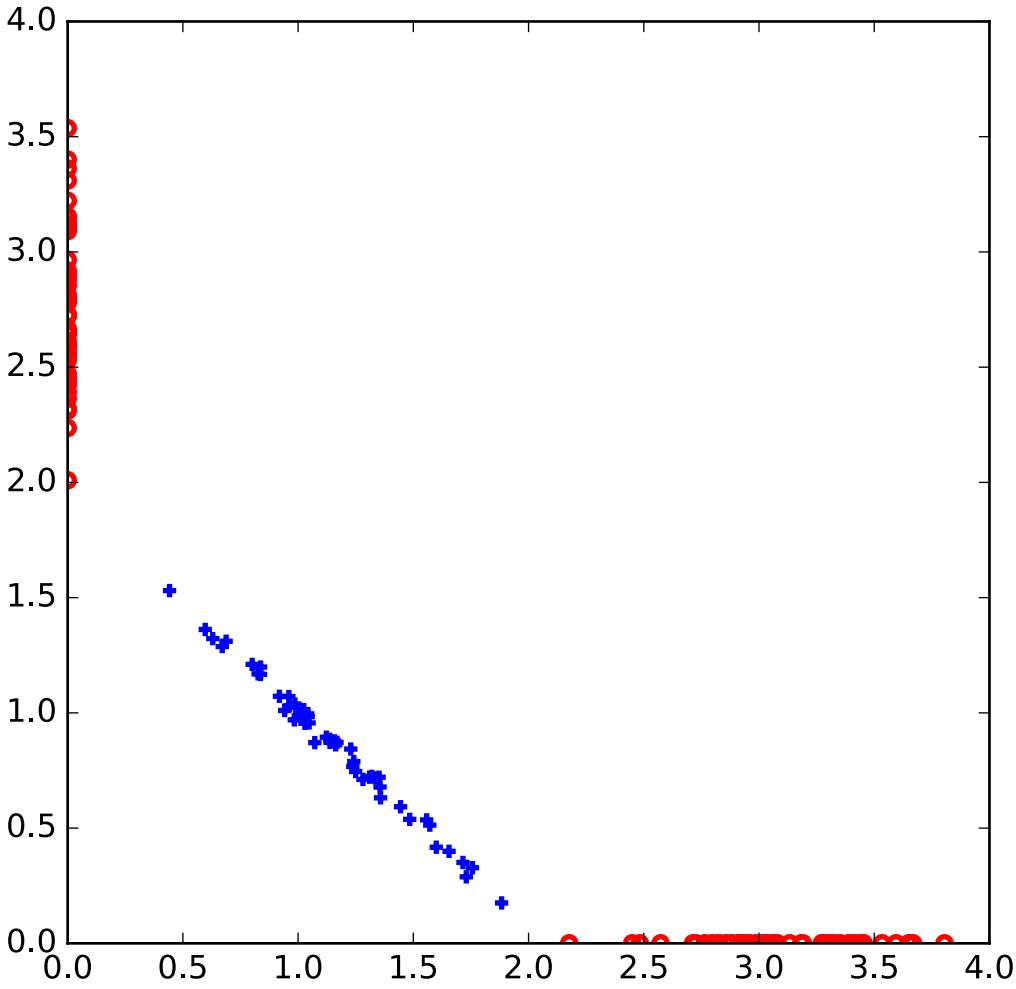
3rd Point





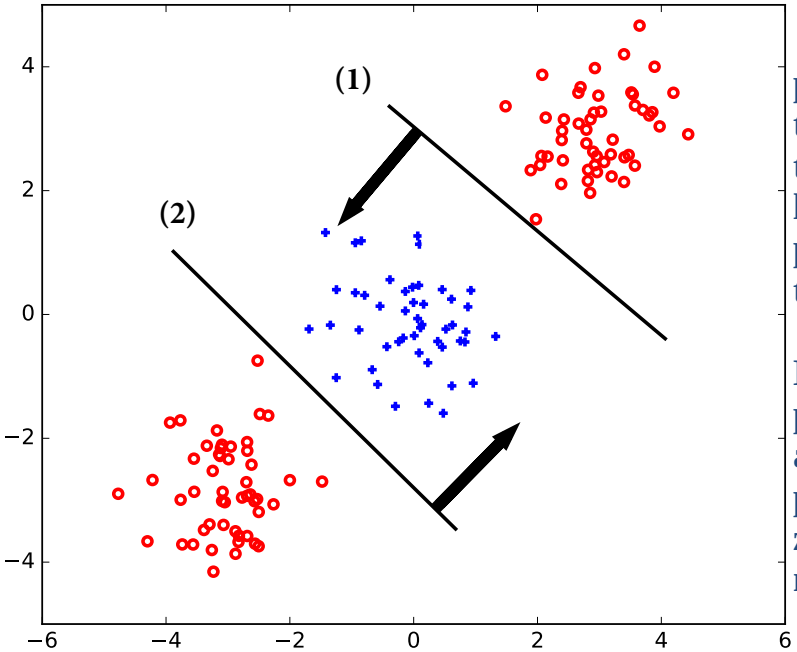






The picture that we got here change a little when we use the ReLU activation but still we won't be able to linearly separate the data in this new f1-f2 space.

Let try Tanh Activation next.



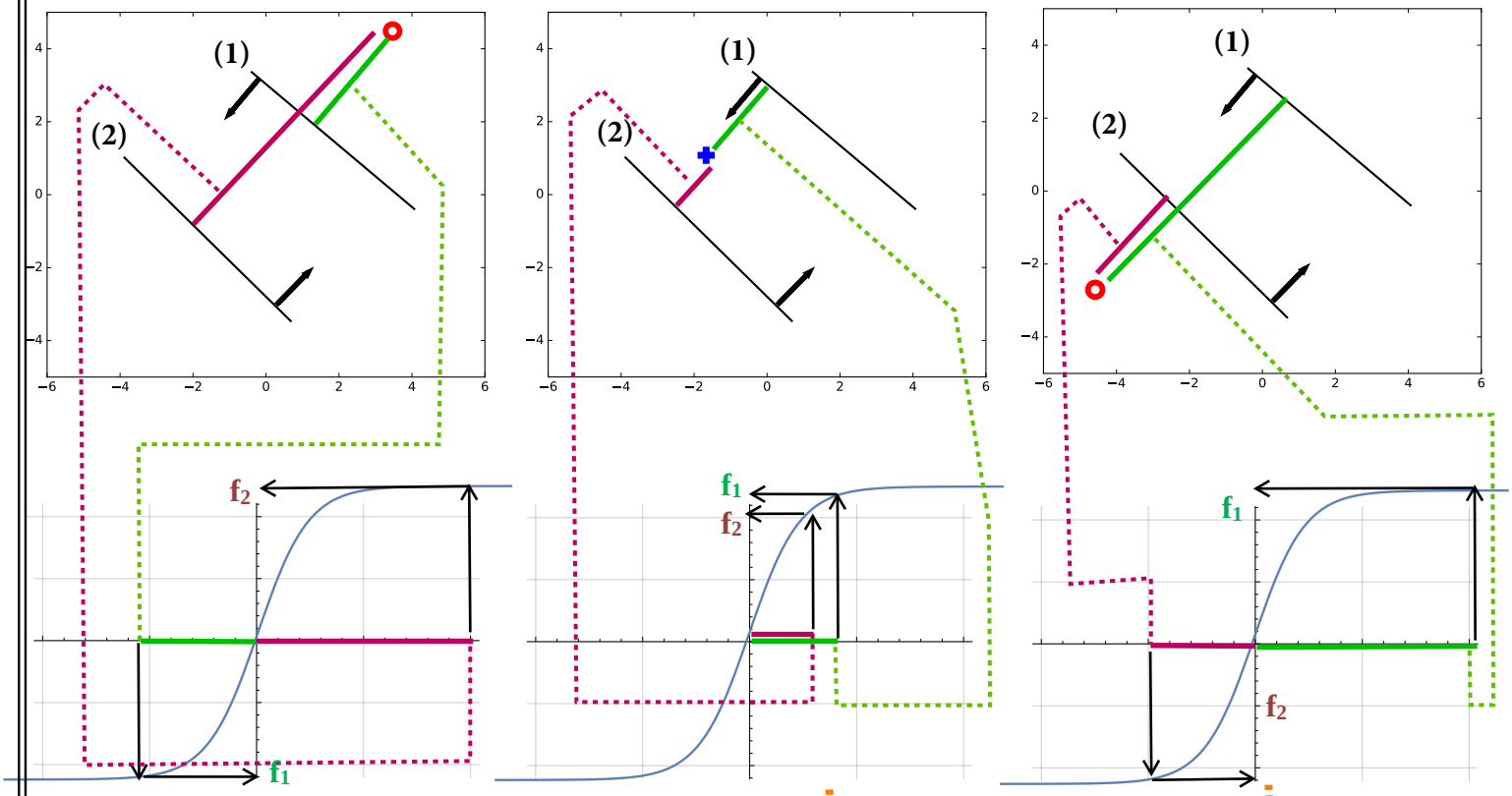
We see the case of three points and how they transform after passing the Tanh Activation layer ie we see how each point in x1-x2 space transforms to f1-f2 space

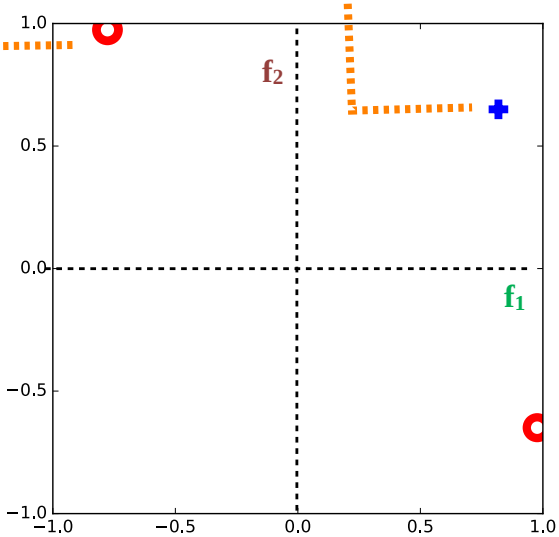
Note that whenever a point is in the same side as w vector of a particular line(1 or 2), its z is positive, otherwise it's negative

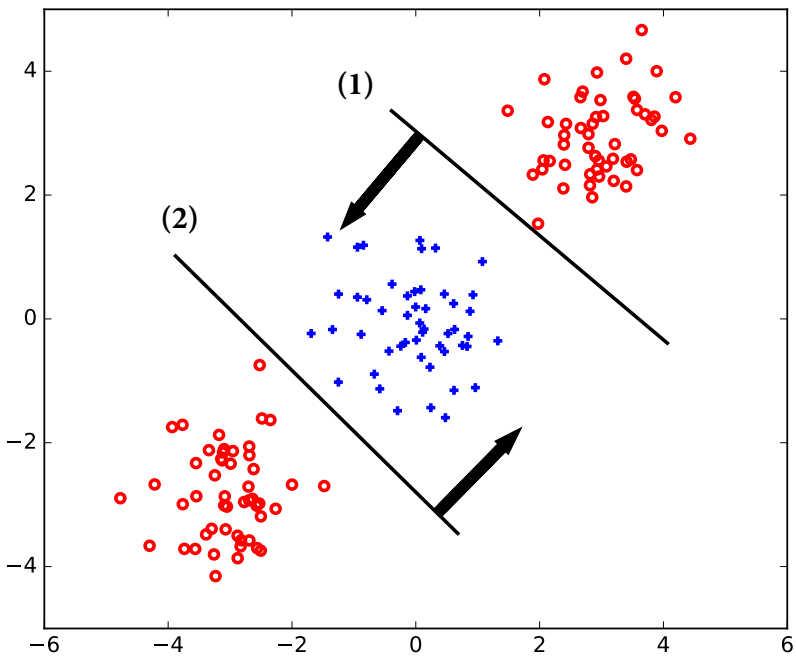
1st Point

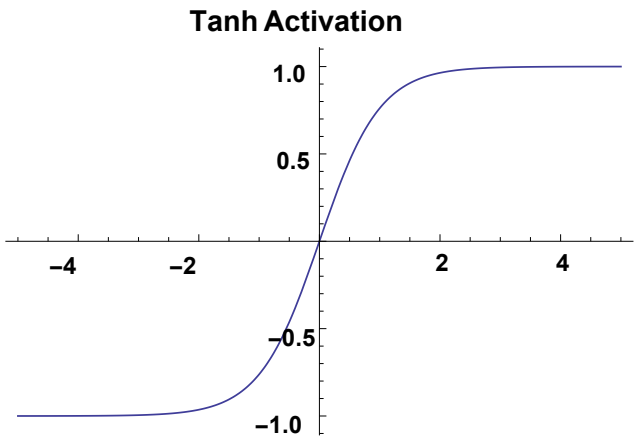
2nd Point

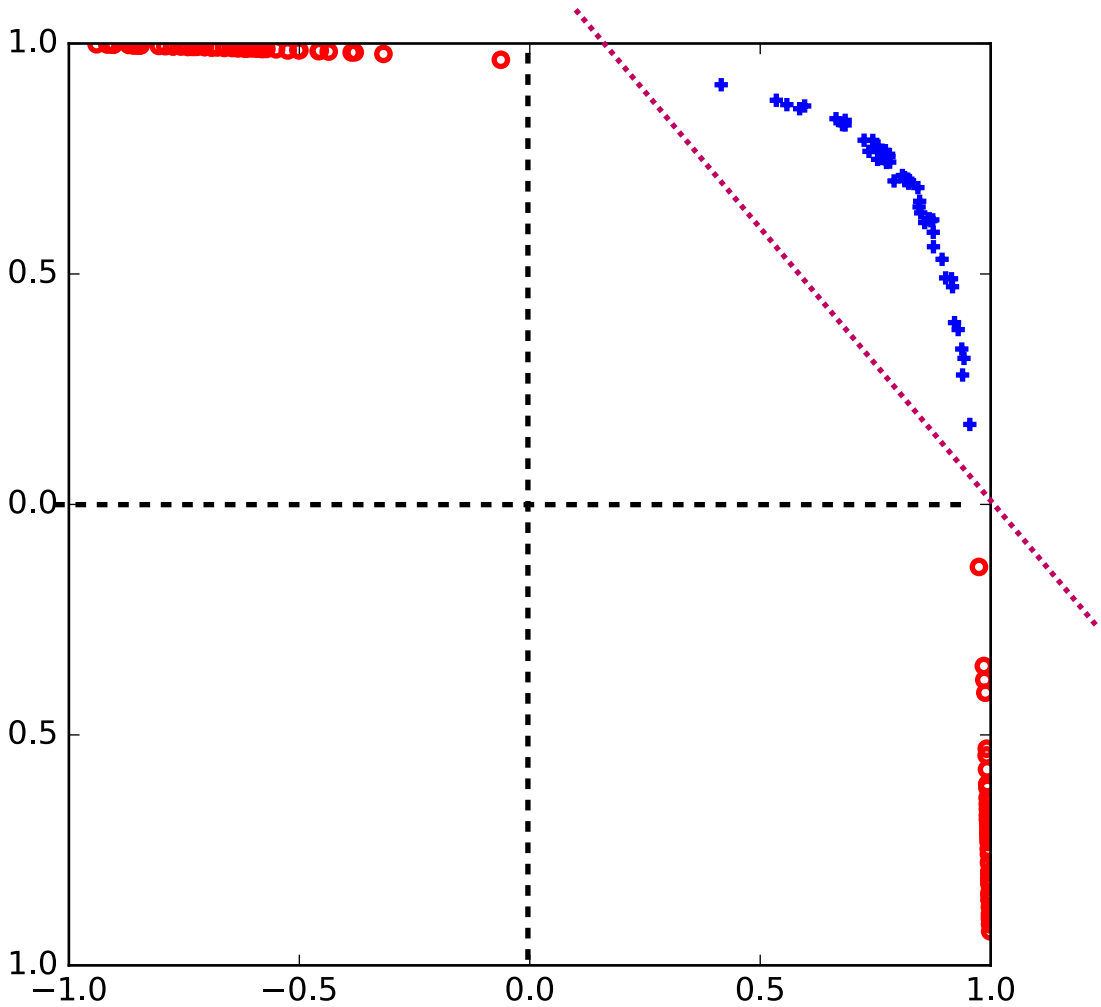
3rd Point











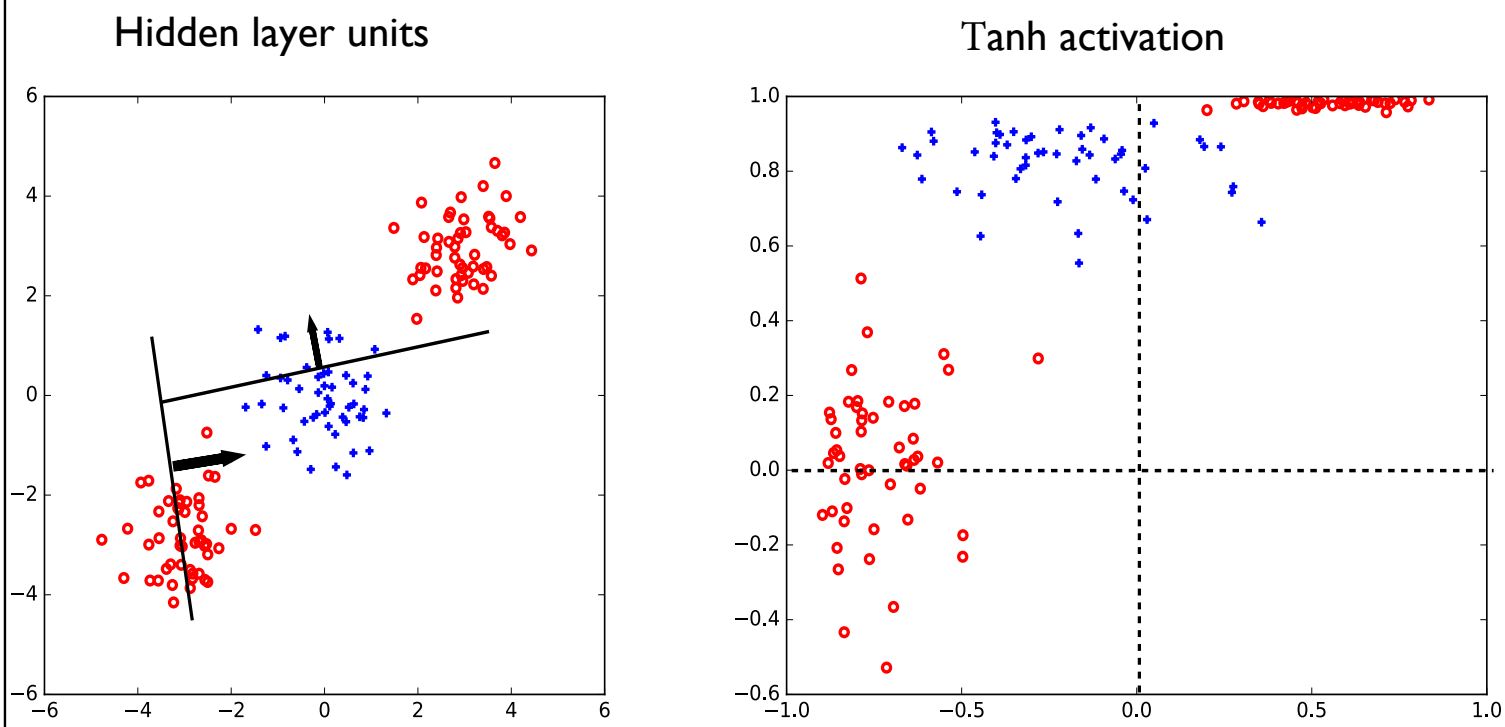
AAAAHHHHAAA!!!!!!

We can clearly separate the data in the f1-f2 space if we use the Tanh Activation function so that's a victory for us. If fact have drawn one of many possible choices.

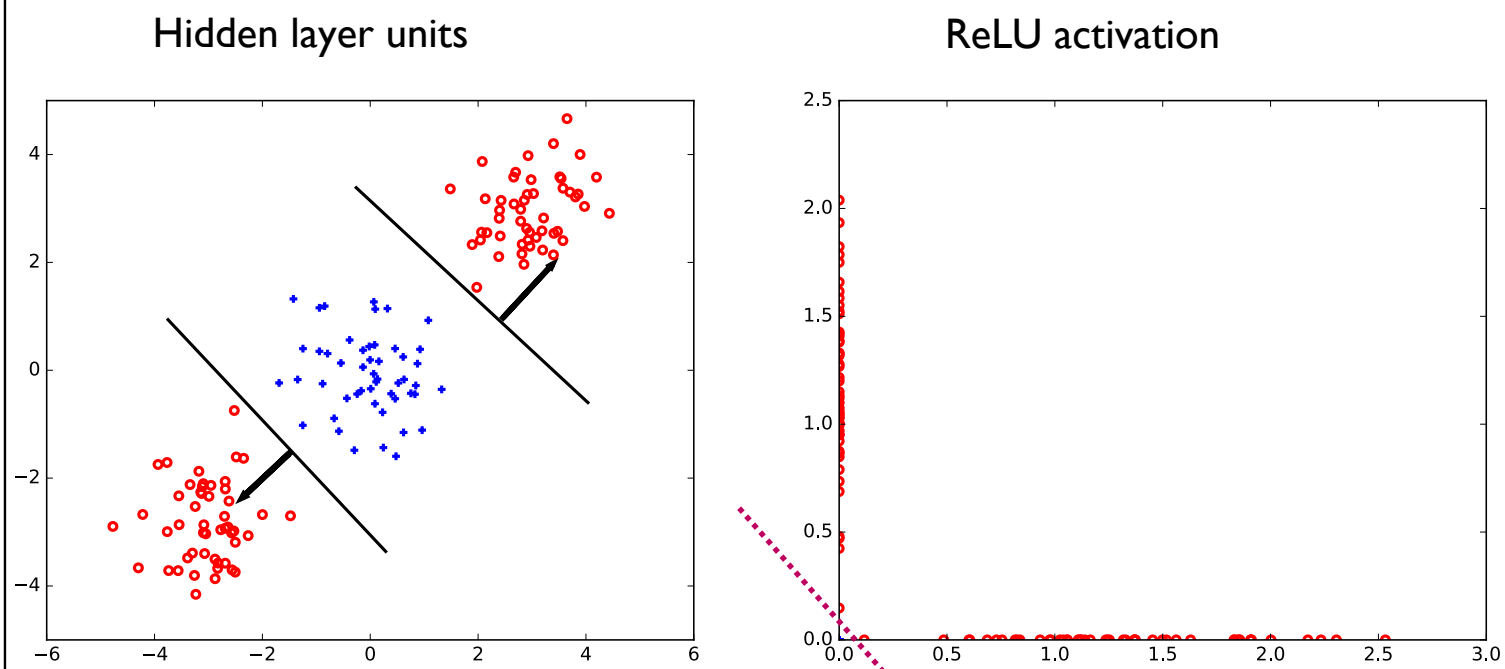
Yeahhh!!!! WE ROCKKKK.....PARTYYYYY!!!!!!

But it's not always a straightforward task to find some activation function that will do our job.

For example below Prof. J used the same Tanh activation function but with different set of W's for the lines. We clearly see that the data is now not separable in the f1-f2 space.



Not only that by changing the orientation of the W vectors below, Prof J. was able to separate the data in the f1-f2 space of the ReLU activation function, that we earlier saw were not separable.
All the blue points now are mapped to the origin (0,0) in the f1-f2 space. Because z_1 , z_2 are negative for all blue points according to the new W's (new line 1 and 2 below), there fore ReLU gives (f1, f2) = (0,0) for all such points



I didn't understand the last part properly so..... FIN