UNIT 3 : Neural Networks

LECTURE 9 : Feedforward Neural Network, Backpropogation & S.G.D

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/2173/

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 flying monkey

There was once a monkey who was born with some genetic defect, so he had wings. So while all the monkeys had a hard time jumping from one tree to another, Pintoe (that's the name of the monkey) easily flew around. Naturally this resulted in envy from the other monkeys, but they couldn't do anything.

But there was one monkey that was especially resentful of Pintoe, his name was Bakka. Bakka had this very hot monkey girlfriend with a shiny red ass (that's nothing vulgar, monkeys have red ass), who Pintoe stole from Bakka. Jacqueline Campbell (Bakka's ex) left Bakka, because Bakka could not afford all those air adventures that Pintoe could provide her, so she monkeybranched her way to Pintoe's heart. Her friend Shiela and Emily told her that she deserved someone much better than Bakka, because she was a queen. She deserved the best, and all that bullc**p your girlfriend's besties talk about you behind you back.

Anyway in order to get revenge, Bakka plotted a devious plan. He slowly started spreading the rumour that Pintoe was infact, a bird that disguised humself as a monkey in order to get the nuts.
'Which nuts?' asked one of Bakka's friend, to which he replied
'Deeeeezzzz nuts' pointing to the acorn nuts in the tree in which they searched for food.

The news spread. And all the monkeys started shooing away Pintoe, as they were sure now that he was a bird. And birds aren't allowed monkey food. Pintoe was not able to fetch for Jacqueline Campbell nuts, so she broke up with him, and got back to Bakka.

Jacqueline Campbell told Bakka that she always loved him. And he was the one for her. And it was his loss if he let her go. And she was confused about the relationship. He should take her back.

Bakka knew she was a golddigger, but he did not care, because of the big red butt. So they became a couple again.

Pintoe commited suicide by hanging himself (do you get the joke????)

Emily developed a jungle cocaine addiction.
Sheila married a convict, in her trip to jungle Las Vegas. She bares his child now. He(the husband) is in jail.

Bakka and Jacqueline Campbell lived happily ever after.

THE END
MORAL : In order to keep a girlfriend, you must constantly give her nuts.

I hope these notes will help you

Let's get started

▷ Backpropogation & Stochastic Gradient Descent

The topic is pretty simple, but pretty complicated. It's simple in a sense that the maths is everything we already know, but it's complicated in the sense that we have to take care of many indices, like the node which we are on, and the node to which we are going, and the layer which we are on. So we have to see that carefully.

But to start the topic off we can start with a very very simple example. This example is so simple that we won't face the 2nd problem of keeping tracks of all nodes & layers for the weight w's.

But this example will contain within it all the mathematical details to bring forward the idea of backpropogation & S.G.D. So let's begin

> The setup

We are talking about a very simple case here.

- The input we collect is $x \in \mathbb{R}$, it's not a vector but just a real number.
- No layer contains more than 1 node. Ie all layers contain only one node.
- There can be only one input to a node & one output from it (by construction).
- So there will be no problem of double indices. Because there is only 1 way in and one way out for everything. So

  z_i : Input to ith layer

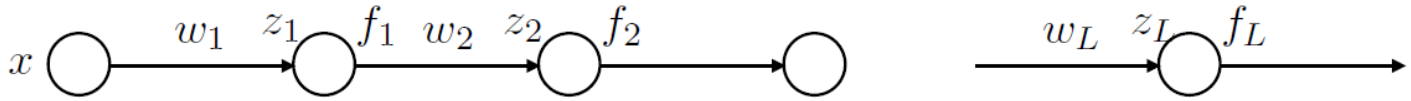
  f_i : Activation function of ith layer

  w_i : weight by which input from one node be multiplied to form z_i

- Also to make things simpler forget the bias term at each level.
- We will make the total number of layers equal to L so f_L will be the final output we get.
- Suppose for the sake of another simplification, All the activation functions related to all the layers are tanh activation.

Ie.
$$f_1 = \tanh(\,)$$
$$f_2 = \tanh(\,)$$
$$\vdots$$
$$f_L = \tanh(\,)$$

This looks something like



Ok now we are ready for the Math

The only things we need to know are these w's ie $w_1, w_2, w_3 \ldots\ldots, w_L$
Once we know these we will predict any ouput for a given x. And how would we find these w's. The answer is simple, by minimizing something
So just like before we will minimise the squared loss. If you recall, it looked something like

$$\frac{1}{2n} \sum_{i=1}^{n} \left(\text{Actual ith label} - \text{Predicted ith Label} \right)^2$$

where n = no. of data points in training set

We just like before denote the actual label by $y^{(i)}$
& the predicted label will be denoted by $f_L^{(i)}$
$f_L^{(i)}$ is the thing we get as the output in the final layer when we input the data point $x^{(i)}$ as an input in the 0th layer

So The Average Loss looks like

$$\frac{1}{2n} \sum_{i=1}^{n} \left(y^{(i)} - f_L^{(i)} \right)^2$$

This is just great, do you know why?
Because we can apply S.G.D to minimise the above function. Just a refresher when we have a minimization problem of the form

$$\min_{\theta} \frac{1}{n} \sum g_i(\theta)$$

where θ is a vector & g is a real valued multivariable function
the we can use SGD to apply the algorithm as follows

At kth iteration pick a random integer say j from the set

$$\{1, 2, 3, \ldots\ldots n\}$$

Perform the update :

$$\theta_{k+1} = \theta_k - \eta_k * \left[\nabla g(\theta)\right]_{@\ j\text{th point}}$$

So our problem is now simplified to following

we want to minimize $\frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)} - f_L^{(i)}\right)^2$

we will choose j at random from $\{1, 2, 3, \ldots\ldots n\}$

we will update :

$$\vec{w}_{k+1} = \vec{w}_k - \eta_k * \left[\vec{\nabla}\frac{(y - f_L(\vec{w}))^2}{2}\right]$$

Here we must note some things

→ $\vec{w}$ is a vector like $[w_1, w_2, \ldots\ldots w_L]^T$

→ The function above is to be minimised wrt these w's

→ $f_L()$ is actually a function of all these w's

So let's recall another terminology.

we used to call this term $\frac{(y - f_L(\vec{w}))^2}{2}$ as the loss for a point

Let's write it more properly

$$\text{Loss}(y, x, \vec{w}) = \frac{(y - f_L(x, \vec{w}))^2}{2}$$

f_L is actually a function of x too as it takes x as one of it's input. I didn't write that earlier because it dosen't matter as the minimization is done wrt $\vec{w}$

So Any way we want to minimise

$$\text{Objective} \equiv J(\vec{w}) \equiv \frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)} - f_L(x^{(i)}, \vec{w})\right)^2 \equiv \text{Average Loss.}$$

And we have decided that we will do it with SGD algorithm to get the step

$$\vec{w}_{k+1} = \vec{w}_k - \eta_k * \vec{\nabla} \frac{(y - f_L(x, \vec{w}))^2}{2}$$

which is equivalent to saying

$$\vec{w}_{k+1} = \vec{w}_k - \eta_k * \vec{\nabla} \, Loss(y, x, \vec{w})$$

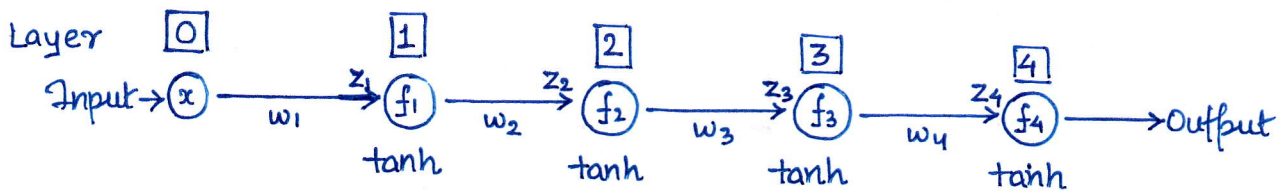
So it all boils down to finding

$$\vec{\nabla} Loss(y, x, \vec{w}) = \begin{bmatrix} \frac{\partial}{\partial w_1} \\ \frac{\partial}{\partial w_2} \\ \vdots \\ \frac{\partial}{\partial w_L} \end{bmatrix} Loss(y, x, \vec{w})$$

$$\vec{\nabla} Loss(y, x, \vec{w}) = \begin{bmatrix} \frac{\partial}{\partial w_1} \frac{(y - f_L(x, \vec{w}))^2}{2} \\ \frac{\partial}{\partial w_2} \frac{(y - f_L(x, \vec{w}))^2}{2} \\ \vdots \\ \frac{\partial}{\partial w_L} \frac{(y - f_L(x, \vec{w}))^2}{2} \end{bmatrix} = -(y - f_L(x, \vec{w})) * \begin{bmatrix} \frac{\partial}{\partial w_1} f_L(x, \vec{w}) \\ \frac{\partial}{\partial w_2} f_L(x, \vec{w}) \\ \frac{\partial}{\partial w_3} f_L(x, \vec{w}) \\ \vdots \\ \frac{\partial}{\partial w_L} f_L(x, \vec{w}) \end{bmatrix}$$

So we need to find $\frac{\partial}{\partial w_1} f_L(x, \vec{w})$, $\frac{\partial}{\partial w_2} f_L(x, \vec{w})$, ... & so on to.. $\frac{\partial}{\partial w_L} f_L(x, \vec{w})$

Now this is not as simple as it Looks. Lets suppose we have $L = 4$ ie a 4 Layered model that looks like below



x	$z_1 = w_1 x$	$z_2 = w_2 f_1$	$z_3 = w_3 f_2$	$z_4 = w_4 f_3$
	$f_1 = \tanh(z_1)$	$f_2 = \tanh(z_2)$	$f_3 = \tanh(z_3)$	$f_4 = \tanh(z_4)$

we are interested in the derivatives of $f_4(x, \vec{w})$

$f_4 = \tanh(z_4)$ we have $z_4 = w_4 f_3$ so

$f_4 = \tanh(w_4 f_3)$ we have $f_3 = \tanh(z_3)$ so

$f_4 = \tanh(w_4 \tanh(z_3))$ we have $z_3 = w_3 f_2$ so

$f_4 = \tanh(w_4 \tanh(w_3 f_2))$ we have $f_2 = \tanh(z_2)$ so

$f_4 = \tanh(w_4 \tanh(w_3 \tanh(z_2)))$ we have ..... & so on... we finally get..

..... we finally get.....

$$f_4(x, \vec{w}) = \tanh(w_4 \tanh(w_3 \tanh(w_2 \tanh(w_1 x))))$$

So this is the 'thingy' that we will have to take partial derivatives of in order to proceed forward. And I'll be a son of a gun, if i did it.

But I am actually a human's son (actually two humans) I want a much neater way to do this and calculus provides us with a neater way

Observe that

$$f_4 = \tanh(w_4 f_3) \; ; \; f_3 = \tanh(w_3 f_2) \; ; \; f_2 = \tanh(w_2 f_1) \; ; \; f_1 = \tanh(w_1 x)$$

we can take help of

NOTE

You must forget about Neural Networks during this section to not get confused

➢ Chain rule : Mathematical Interlude

Now there are many forms of the famous chain rule in calculus useful for different scenarios but let me give you a general trick to find any rule. Let's call this the tree model. Here's how it goes:

- On top of our tree will be the function that we want to differentiate we will usually denote it by $f()$.
- Now f will be a function of some independent variables, these are the values that we can feed as input. These will be at the bottom of our tree.

  When we have 1 independent variable, I will usually denote it by the letter 't'. In that case our function will be $f(t)$ and we will be interested in the derivative $\frac{d}{dt} f(t)$. we have only one derivative so we are not interested in partial derivatives here.

  When we have more than 1 independent variables, I will denote them (if two) by 's' and 't'. If three then 'r', 's', 't' and so on

  2 indp. variables $f()$ will be $f(s,t)$
  3 indp. variables $f()$ will be $f(r,s,t)$

And we will be interested in partial derivatives now for example

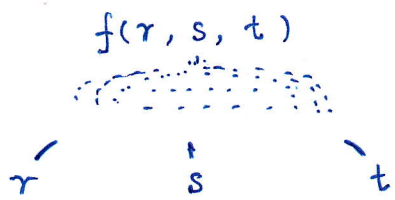
for $f(s,t)$ we have $\frac{\partial f(s,t)}{\partial s}$, $\frac{\partial f(s,t)}{\partial t}$

for $f(r,s,t)$ we have $\frac{\partial f(r,s,t)}{\partial r}$, $\frac{\partial f(r,s,t)}{\partial s}$, $\frac{\partial f(r,s,t)}{\partial t}$

So up till now we have

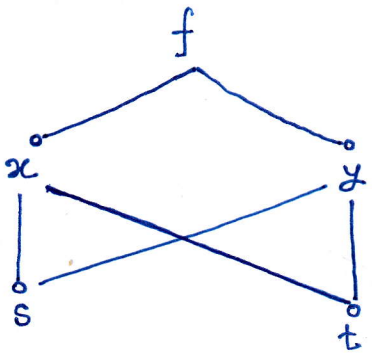
An upper layer of $f()$

A bottom layer of independent variables



Now we will see what comes in between these two layers

- In between the layers will come the intermediate variables. I will denote 1 layer of them with a, b, c. a, b, c will them selves be functions of variables <u>below</u> them. These are <u>not independent</u>. The only independent variables are the ones at the bottom layer. Let me just make some trees

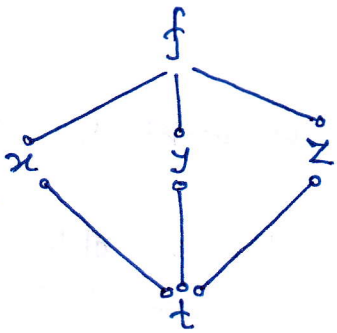


Top layer

Intermediate Layer. x and y are functions of variables below them ie $x \equiv x(t)$
$y \equiv y(t)$

Bottom layer : s and t are independent

Lets look at another tree more clarity

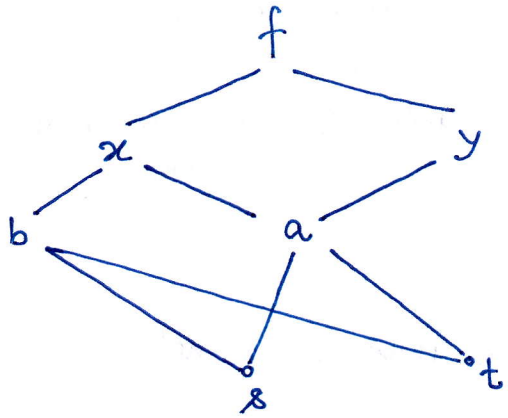


top layer

Intermediate layer $x \equiv x(t)$
$y \equiv y(t)$
$z \equiv z(t)$

Bottom layer : t is independent variable

Lets look at another tree much more clarity



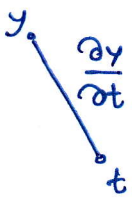
Top layer

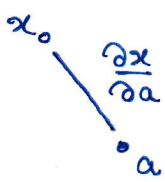
1st Intermediate layer
$x \equiv x(a,b)$; $y \equiv y(a,b)$

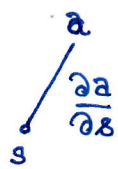
2nd Intermediate layer
$b \equiv b(s,t)$; $a \equiv a(s,t)$

Bottom layer : s, t are independent variables

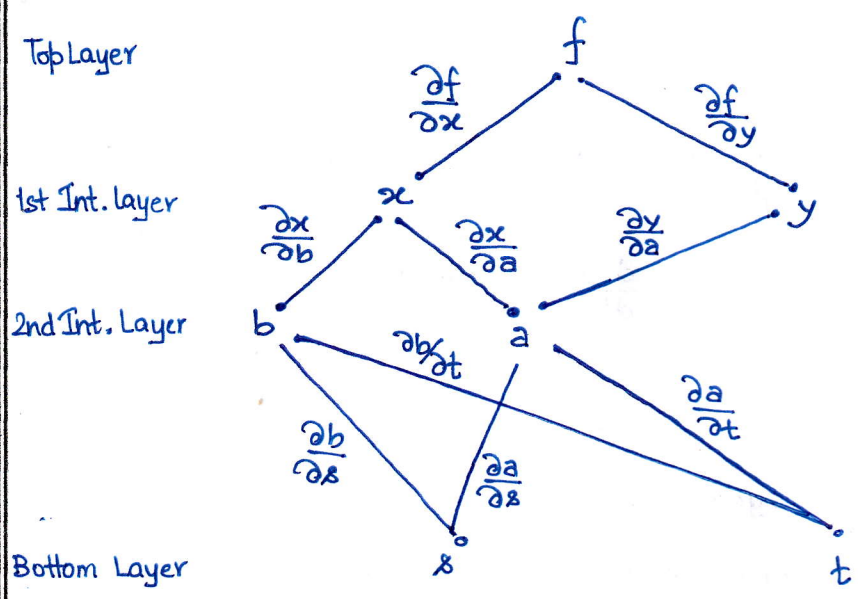
- Now to each branch we will associate a derivative. It would look like $\dfrac{\partial(\text{above thing})}{\partial(\text{below thing})}$ for eg







And our tree is complete. Now we can find any derivative, you name it
Let me first just draw one tree as an example



Here the top layer has f

$f \equiv f(x,y)$

x, y are inturn functions of a, b

$x \equiv x(a,b)$, $y \equiv y(a)$

a, b are inturn functions of s, t

$a \equiv a(s,t)$, $b \equiv b(s,t)$

s and t are independent variables

The rule to find the derivative of f wrt any independent variable is simple.

- Start from the top layer with the f and follow a path to the independent variable you wanna take the derivative of f with respect to
  And while traversing that path keep multiplying all the derivatives you encounter Lets say that's the path product you get for a single path.
- Add all these path products for all paths possible from the f at top layer to independent variable at bottom layer.

Now believe it on not we have all the chain rules that we can have
Let's see example by example what we have accomplished.

$\boxed{1}$ Simple chain rule for Single 1 variable function

We have $f(x)$ and x in turn is a function of t

So we have at the top layer f, single variable funct.

The intermediate Layer has x
x is a function of variable below it ie $x \equiv x(t)$

The bottom layer has the independent variable t.
We go from f to t and multiply

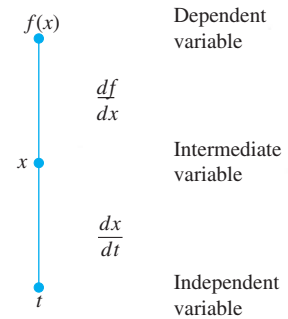
So

$$\boxed{\frac{df}{dt} = \frac{df}{dx}\frac{dx}{dt}}$$

This is the famous one variable chain rule

To find df/dt, we read down the route from f to t, multiplying derivatives along the way.

Chain Rule



Eg 1 Suppose $f(x) = e^x$ and $x(t) = \ln(t)$

So in order to find $\frac{df}{dt}$ we have two ways

way 1 : $f(x) \equiv f(x(t)) \equiv f(\ln(t)) = e^{\ln(t)} = t$ so $\frac{df}{dt} = 1$

way 2 : we have derived $\frac{df}{dt} = \frac{df}{dx}\frac{dx}{dt}$

$$= \frac{d}{dx}e^x \cdot \frac{d}{dt}\ln(t) = e^x \frac{1}{t} = e^{\ln(t)}\frac{1}{t} = t \cdot \frac{1}{t} = 1$$

Both give the same result.

2 Simple Chain rule : 1 more Layer

Suppose we are given $f(x)$ and x is a function of a, and a is a function of t

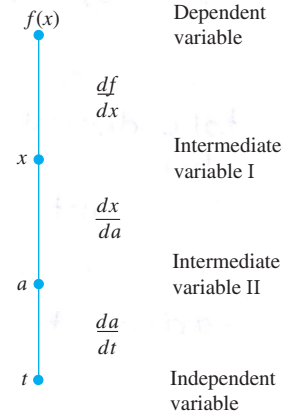
ie $f \equiv f(x)$; $x \equiv x(a)$; $a \equiv a(t)$

> The top layer has the function f the independent variable is t, so f will at last be a function of t. It's a single variable function we are interested in

$$\frac{df}{dt}$$

To find df/dt, we read down the route from f to t, multiplying derivatives along the way.

Chain Rule



> The 1st intermediate layer is x, where x is function of variable below ie $x \equiv x(a)$

> The 2nd intermediate layer is a, where a is a function of the variable below it ie $a \equiv a(t)$

> The bottom Layer has the independent variable t.

We go from f to the independent variable t at the bottom & multiply all the derivatives so we have

$$\boxed{\frac{df}{dt} = \frac{df}{dx}\frac{dx}{da}\frac{da}{dt}}$$

Eg 2 $f(x) = \ln(x)$; $x(a) = e^{a^2}$; $a(t) = \sqrt{t}$

way 1 $f(x(a)) = \ln(e^{a^2}) = a^2 = a(t)^2 = (\sqrt{t})^2 = t$ so $\frac{df}{dt} = 1$

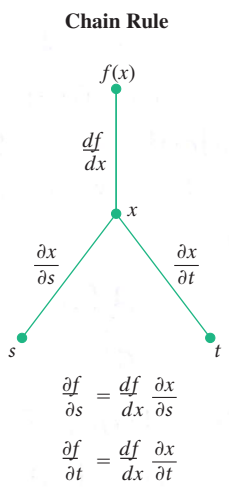
way 2 $\frac{df}{dx} = \frac{1}{x}$; $\frac{dx}{da} = 2a\,e^{a^2}$; $\frac{da}{dt} = \frac{1}{2\sqrt{t}}$ so we have $\frac{df}{dt} = \frac{1}{x} \cdot 2ae^{a^2} \cdot \frac{1}{2\sqrt{t}}$

$$= \frac{1}{e^{a^2}} \cdot 2\sqrt{t}\, e^{a^2} \cdot \frac{1}{2\sqrt{t}} = 1$$

$\boxed{3}$ Lets spice things a little bit

Let suppose we have function $f \equiv f(x)$
x in turn depends on s, t. where s, t are the inpedent variables. So f will finally be a function of s, t. So we will be interested in

$$\frac{\partial f}{\partial s} \quad \text{and} \quad \frac{\partial f}{\partial t}$$



The tree diagram looks like this
> The top layer is f
> The intermediate layer is $x \equiv x(s,t)$
> The bottom layer is a function of two independent variables s and t.

$$\frac{\partial f}{\partial s} = \frac{df}{dx}\frac{\partial x}{\partial s} \qquad \frac{\partial f}{\partial t} = \frac{df}{dx}\frac{\partial x}{\partial t}$$

$\boxed{\text{Eg 3}}$ we have $f(x) = x^2$; $x(s,t) = \sqrt{s^2 + t^3}$ we have to find $\frac{\partial}{\partial s} f$ and $\frac{\partial}{\partial t} f$ So lets begin

Way 1 : $f(x) = x^2 = x(s,t)^2 = s^2 + t^3$ so $\frac{\partial f}{\partial s} = 2s$, $\frac{\partial f}{\partial t} = 3t^2$

Way 2 : $\frac{\partial f}{\partial s} = \frac{df}{dx}\frac{\partial x}{\partial s} = 2x \cdot \frac{1 \cdot 2s}{2\sqrt{s^2+t^3}} = 2s$ ✓ checks out

$\frac{\partial f}{\partial t} = \frac{df}{dx}\frac{\partial x}{\partial t} = 2x \cdot \frac{1 \cdot 3t^2}{2\sqrt{s^2+t^3}} = 3t^2$ ✓ Checks out

So upuntil now we have done chain rule where we go from f to the independent variable and there is only one path to take. we will now see what happen when we have multiple patways. As I have told earlier we just add all these derivative products

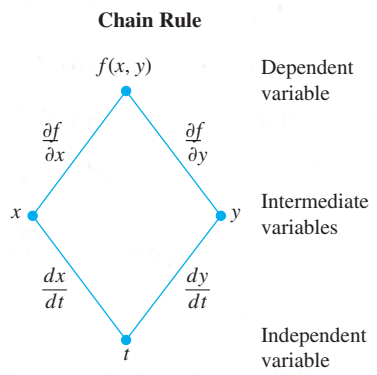
Lets start with a simple case

$\boxed{4}$ We have function $f \equiv f(x,y)$ and x, y are in turn functions of t. So $x \equiv x(t)$ & $y \equiv y(t)$

So even when f is a function of x an y, x and y are in turn dependent on the single independent variable t. So f is actually a function of t, ie f is a single variable function of t. So we are interested in

$$\frac{df(t)}{dt}$$

> The top layer is $f(x,y)$
> The intermediate Layer will be $x \equiv x(t)$, $y \equiv y(t)$
> The bottom layer will have the independent variable t.



$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$$

So in order to find $\frac{df}{dt}$ we will add all the derivative products going from different paths from f to t

from f to t via x : $\frac{\partial f}{\partial x}\frac{dx}{dt}$

from f to t via y : $\frac{\partial f}{\partial y}\frac{dy}{dt}$

we add them to find

$$\boxed{\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}}$$

$\boxed{\text{Eg 4}}$ $f(x,y) = x^{2/3} + y^{4/3}$; $x(t) = t^3$; $y(t) = t^{3/2}$

way 1 : $f(x,y) = x(t)^{2/3} + y(t)^{4/3} = (t^3)^{2/3} + (t^{3/2})^{4/3} = t^2 + t^2 = 2t^2 \therefore \frac{df}{dt} = 4t$

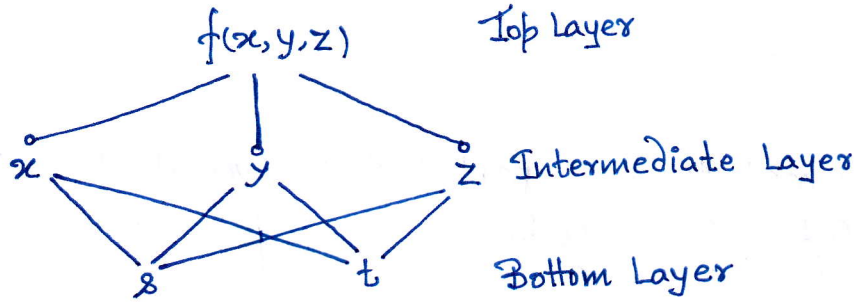
way 2 : $\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$

$$= \frac{2}{3}x^{-1/3} \cdot 3t^2 + \frac{4}{3}y^{1/3} \cdot \frac{3}{2}t^{1/2}$$

$$= \frac{2}{3}t^{-1} \cdot 3t^2 + \frac{4}{3}t^{1/2} \cdot \frac{3}{2}t^{1/2} = 2t + 2t = 4t$$

Same as above

$\boxed{5}$ Let say now we have a function $f(x,y,z)$, and x itself is a function of s and t ie $x \equiv x(s,t)$ and y itself is a function of s and t and so is z, so $y \equiv y(s,t)$ and $z \equiv z(s,t)$. Here s,t are independent variables. The tree looks like



Now can you guess what are the derivatives we are interested in?
You guess it right $\frac{\partial f}{\partial s}$ $\frac{\partial f}{\partial t}$

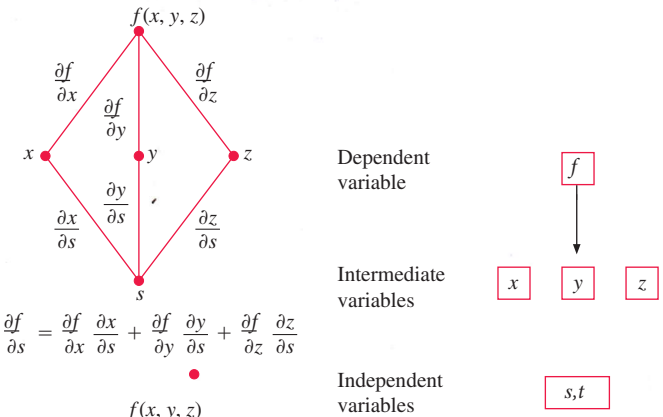
So in order to find those we can find all the paths that lead from f to that independent variable.

If we want to find

$\frac{\partial f}{\partial s}$ we see three paths that lead from f to s

- f to s via x
- f to s via y
- f to s via z

Therefore



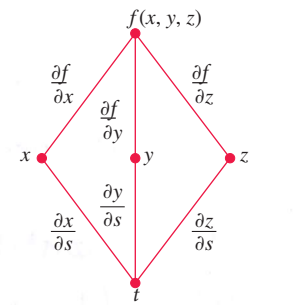
$$\frac{\partial f}{\partial s} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial s} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial s}$$

Similiarly to find

$\frac{\partial f}{\partial t}$ we see three paths that lead from f to t

- f to t via x
- f to t via y
- f to t via z

Therefore



$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial t}$$

$\boxed{\text{Eg}}$ 5 $f(x,y,z) = x^2 + \ln(y) + \sqrt{z}$; $x(s,t) = \sqrt{s^2+t}$; $y(s,t) = e^{s+t}$; $z(s,t) = (s^2+t^2)^2$

Way 1: $f(x,y,z) = (\sqrt{s^2+t})^2 + \ln(e^{s+t}) + \sqrt{(s^2+t^2)} = s^2+t+s+t+s^2+t^2$

$= 2s^2 + s + t^2 + 2t$

Suppose want to find $\frac{\partial f}{\partial s} = 4s+1$ from above

Way 2: $\frac{\partial f}{\partial s} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial s} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial s}$

$$= 2x \cdot \frac{1 \cdot 2s}{2\sqrt{s^2+t}} + \frac{1}{y} e^{s+t} + \frac{1}{2\sqrt{z}} 2(s^2+t^2)(2s) = 2s + 1 + 2s = 4s+1$$

Same..

So that's the end of this interlude. We have the skillset to find any derivative of any function with any number of layers (intermediate) and any no of connections. Let's get back to our problem.

> Getting back to our problem

we had the $L = 4$

$f_4(x, \vec{w})$

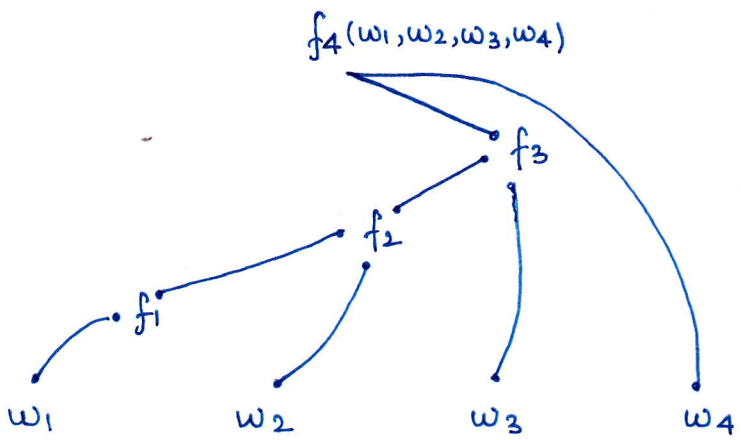
Here we wanted to find partial derivatives of $f_4(x, \vec{w})$

$\frac{\partial f_4}{\partial w_1}$, $\frac{\partial f_4}{\partial w_2}$, $\frac{\partial f_4}{\partial w_3}$, $\frac{\partial f_4}{\partial w_4}$

Here we have 4 independent variables w_1, w_2, w_3, w_4

and we have the independent variables (or functions) f_1 f_2 f_3

Let's see how our tree diagram will look



: Independent variables

Let me first write all the relations

$f_4 = \tanh(w_4 f_3)$ —— Ⓐ

$f_3 = \tanh(w_3 f_2)$ —— Ⓑ

$f_2 = \tanh(w_2 f_1)$ —— Ⓒ

$f_1 = \tanh(w_1 x)$ —— Ⓓ

And now using our chain rule we can find any derivatives

$\frac{\partial}{\partial w_4} f_4$: that's simple there's only one path

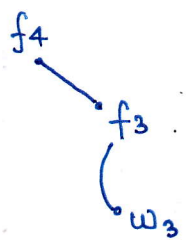
From Ⓐ we know $f_4 = \tanh(w_4 f_3)$

$$\frac{\partial f_4}{\partial w_4} = \left(1 - \tanh^2(w_4 f_3)\right) \cdot f_3$$

Note

$$\frac{d}{dz} \tanh(z) = 1 - \tanh^2(z)$$

$\frac{\partial}{\partial w_3} f_4$: that won't be too difficult either we have the branch from f_4 to w_3



So $\frac{\partial f_4}{\partial w_3} = \frac{\partial f_4}{\partial f_3} \frac{\partial f_3}{\partial w_3}$ from Ⓐ & Ⓑ we have

$$\frac{\partial f_4}{\partial f_3} = \left(1 - \tan^2(w_4 f_3)\right) \cdot w_4$$

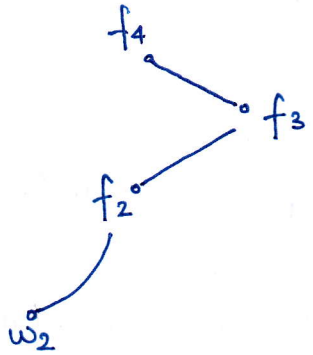
$$\frac{\partial f_3}{\partial w_3} = \left(1 - \tan^2(w_3 f_2)\right) \cdot f_2$$

So we have

$$\frac{\partial f_4}{\partial w_3} = [1-\tan^2(w_4 f_3)]\cdot w_4 \cdot [1-\tanh^2(w_3 f_2)]\cdot f_2$$

Next we can find

$\frac{\partial f_4}{\partial w_2}$: that's not difficult either look below 1 branch



$f_4 = \tanh(w_4 f_3)$ — Ⓐ
$f_3 = \tanh(w_3 f_2)$ — Ⓑ
$f_2 = \tanh(w_2 f_1)$ — Ⓒ

& from chain rule we have

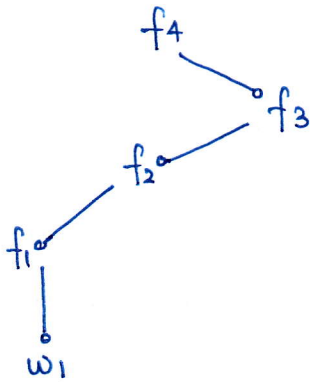
$$\frac{\partial f_4}{\partial w_2} = \frac{\partial f_4}{\partial f_3}\frac{\partial f_3}{\partial f_2}\frac{\partial f_2}{\partial w_2}$$

Ⓐ Ⓑ Ⓒ

$$[1-\tan^2 h(w_4 f_3)]\cdot w_4 \cdot [1-\tan^2 h(w_3 f_2)]\cdot w_3 \cdot [1-\tan^2 h(w_2 f_1)]\cdot f_1$$

AND FINALLY LET ME, NOW ELABORATE WHAT WE HAVE BEEN DOING ALL ALONG BY FINDING

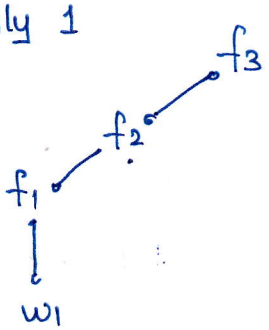
$\frac{\partial f_4}{\partial w_1}$ we will have single branch



the formula will be

$$\frac{\partial f_4}{\partial w_1} = \frac{\partial f_4}{\partial f_3}\underbrace{\frac{\partial f_3}{\partial f_2}\frac{\partial f_2}{\partial f_1}\frac{\partial f_1}{\partial w_1}}$$

But look at this part carefully. Suppose I told you to find $\frac{\partial f_3}{\partial w_1}$ what would we do? We will see all the branches that go from f_3 to w_3 and multiply derivatives. There's only 1



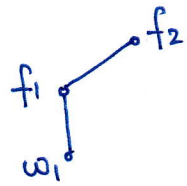
This is obviously a subpart of the upper branch & we can say from our rule that

$$\frac{\partial f_3}{\partial w_1} = \frac{\partial f_3}{\partial f_2}\underbrace{\frac{\partial f_2}{\partial f_1}\frac{\partial f_1}{\partial w_1}}$$

Now look at this part carefully

If I asked you to find

$\frac{\partial f_2}{\partial w_1}$ we will see all branches from f_2 to w_1. we have only 1



This will give us

$$\frac{\partial f_2}{\partial w_1} = \frac{\partial f_2}{\partial f_1}\frac{\partial f_1}{\partial w_1}$$

So lets now rewrite $\frac{\partial f_4}{\partial w_1}$ with this point of view in mind.

→ you want to find $\frac{\partial f_4}{\partial w_1}$ go to the previous layer ie go back

take derivative (partial) of f_4 wrt f_3 and derivative (partial) of f_3 wrt w_1

So $$\frac{\partial f_4}{\partial w_1} = \frac{\partial f_4}{\partial f_3}\frac{\partial f_3}{\partial w_1}$$

→ Now we can easily find $\frac{\partial f_4}{\partial f_3}$ because $f_4 = \tanh(w_4 f_3)$

But now we have to find $\frac{\partial f_3}{\partial w_1}$. No worries from the 3rd layer go back to the second layer to find $\partial f_3/\partial f_2$

$$\frac{\partial f_3}{\partial w_1} = \frac{\partial f_3}{\partial f_2}\frac{\partial f_2}{\partial w_1}$$

→ Now we can easily find $\frac{\partial f_3}{\partial f_2}$ because $f_3 = \tanh(w_3 f_2)$

But now we have to find $\frac{\partial f_2}{\partial w_1}$ again go a layer back to find $\frac{\partial f_2}{\partial f_1}$ so we have

$$\frac{\partial f_2}{\partial w_1} = \frac{\partial f_2}{\partial f_1}\frac{\partial f_1}{\partial w_1}$$

→ Now that's no problem both of them can be found because

$$f_2 = \tanh(w_2 f_1) \quad \& \quad f_1 = \tanh(w_1 x)$$

And this is the basic idea of backpropogation

To find

$\frac{\partial f_4}{\partial w_1}$ → we go one layer back → $\frac{\partial f_4}{\partial w_1} = \frac{\partial f_4}{\partial \ _3} \frac{\partial f_3}{\partial w_1}$

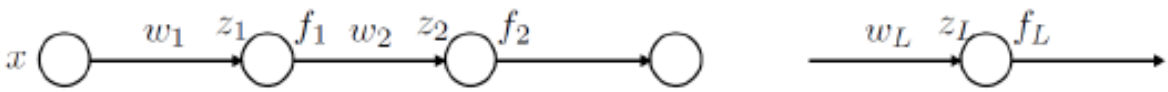
→ we go another layer back → $\frac{\partial f_4}{\partial w_1} = \frac{\partial f_4}{\partial f_3} \frac{\partial f_3}{\partial f_2} \frac{\partial f_2}{\partial w_1}$

→ we go another layer back → $\frac{\partial f_4}{\partial w_1} = \frac{\partial f}{\partial f_3} \frac{\partial f_3}{\partial f_2} \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial w_1}$

& finally we get our answer by going back & back untill we are satisfied.

And we can also see this idea for our general model

Remember this



So here we have the

$$\text{Loss} \equiv f_L(\vec{w})$$

And we will have the following relations

$$f_L = \tanh(w_L f_{L-1})$$
$$f_{L-1} = \tanh(w_{L-1} f_{L-2})$$
$$f_{L-2} = \tanh(w_{L-2} f_{L-3})$$
$$\vdots$$
$$f_2 = \tanh(w_2 f_1)$$
$$f_1 = \tanh(w_1 x)$$

So we can hop back wards to find $\frac{\partial f_L}{\partial w_i}$ where $w_i \in \{w_1, w_2, w_3 \ldots\ldots w_L\}$

So suppose we want $\frac{\partial f_L}{\partial w_1}$, this is how we go step by step

$\frac{\partial f_L}{\partial w_1} = \frac{\partial f_L}{\partial f_{L-1}} \frac{\partial f_{L-1}}{\partial w_1}$ we know $\frac{\partial f_L}{\partial f_{L-1}} = [1 - \tan^2(w_L f_{L-1})] \cdot w_L$ we just need

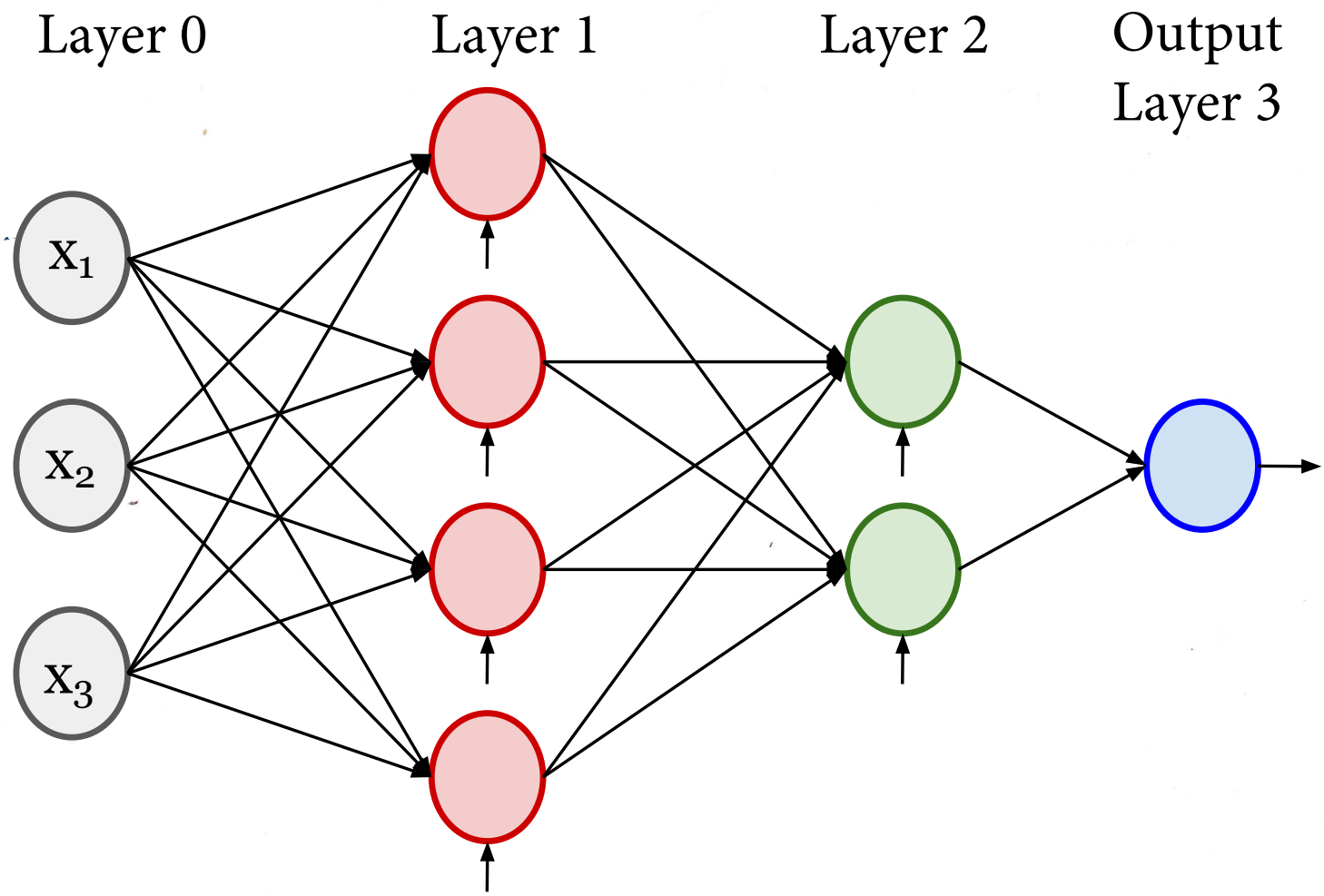
$\frac{\partial f_{L-1}}{\partial w_L} = \frac{\partial f_{L-1}}{\partial f_{L-2}} \frac{\partial f_{L-2}}{\partial w_1}$. we know $\frac{\partial f_{L-1}}{\partial f_{L-2}} = [1 - \tan^2(w_{L-1} f_{L-2})] \cdot w_{L-2}$ we just need

$\frac{\partial f_{L-2}}{\partial w_1} = \frac{\partial f_{L-2}}{\partial f_{L-3}} \frac{\partial f_{L-3}}{\partial w_1}$ & we can keep going on & that's BACKPROPOGATION

➤ Back propogation: the most general case

We have a little idea of how to backpropogate, but to me it still seems like a scary task, when we will have many layers and nodes and activation fuction and many w's

So let's slay this dragon finally. First let me show you what neural network we will be working with



So in order to define/analyse different thing we will need terminology. Lets begin.

- As usual the layer that stores the raw data will be called layer 0. This is the layer with 3 nodes in our case that stores x_1, x_2, x_3. Therefore input $x \in \mathbb{R}^3$
- Now we will need to keep track of nodes as well as layers that we are in
  So superscript relates to the layer we are talking about
  subscript relates to the nodes we are talking about
- Weights : we will denote weight by

  w^{l}_{jk} is the weight for the connection <u>from</u> k th neuron in $(l-1)$th layer <u>to</u> j th neuron in l th layer

  $w^{\text{to layer}}_{\text{to, from}}$ is the general form. I write them above the connecting arrows in the figure

Note : its from k to j written as j, k don't get confused

> Bias : Earlier we did not include the bias term, but here we are talking about the general case so we define

b_j^l : bias of jth neuron in lth layer

If you see closely, you will see one vertical arrow below every neuron. Thats were I write the bias term

> Activation : To each neuron is associated an activation function

a_j^l : is used to denote the activation of jth neuron in lth layer

Note that the weights $w_{j,k}^l$ and b_j^l are Real valued <u>numbers</u>, while the a_j^l is a Real valued function. It is the same $f_L, f_{L-1}, f_{L-2} \ldots$ in the 1d simple example

It will take input what it gets from previous layers & operate on into give the output

Ie

$$a_j^l = f\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

Bias term for jth node in lth layer

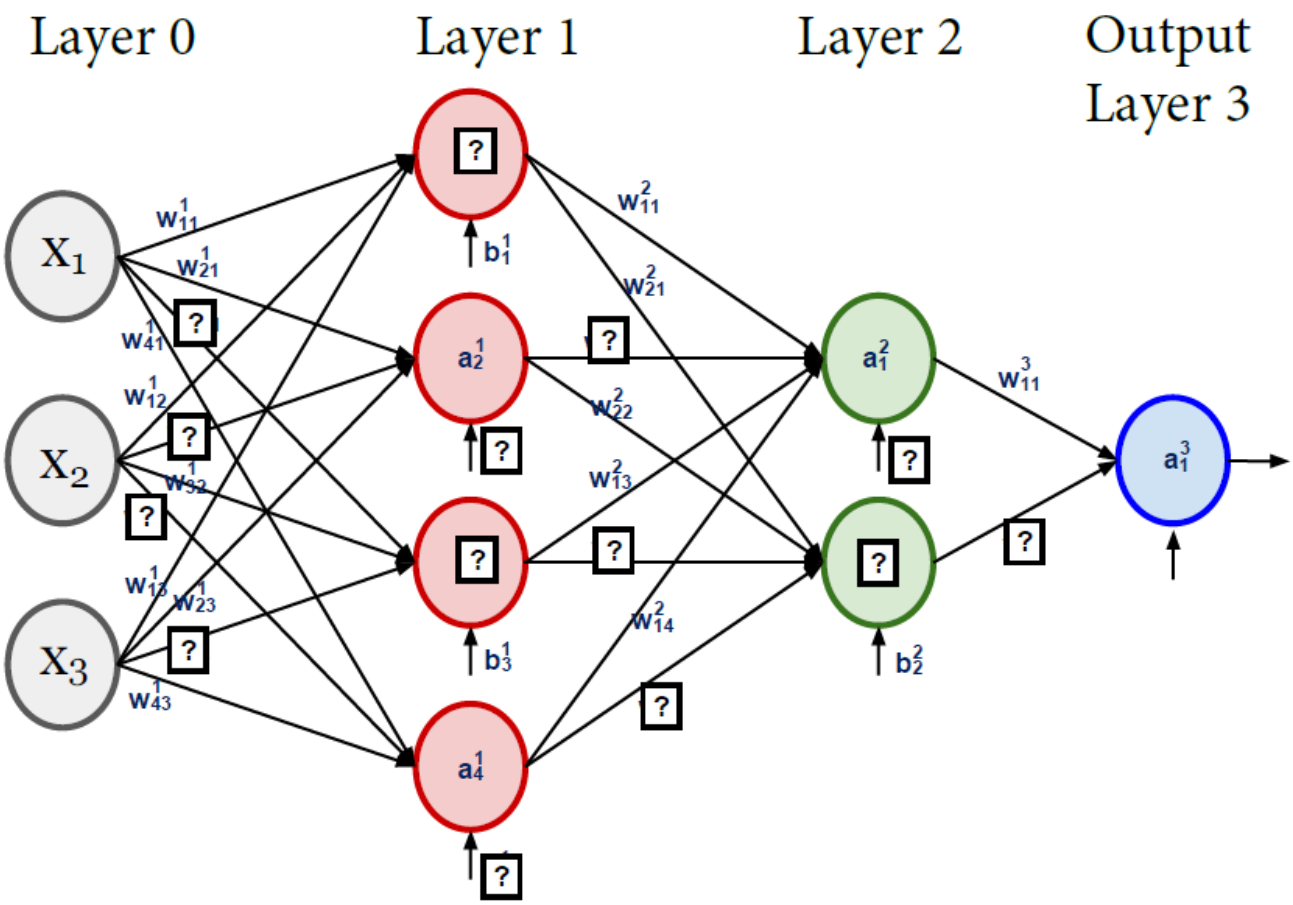
This is Real output of jth node in lth layer

Some function like tanh() ReLU()

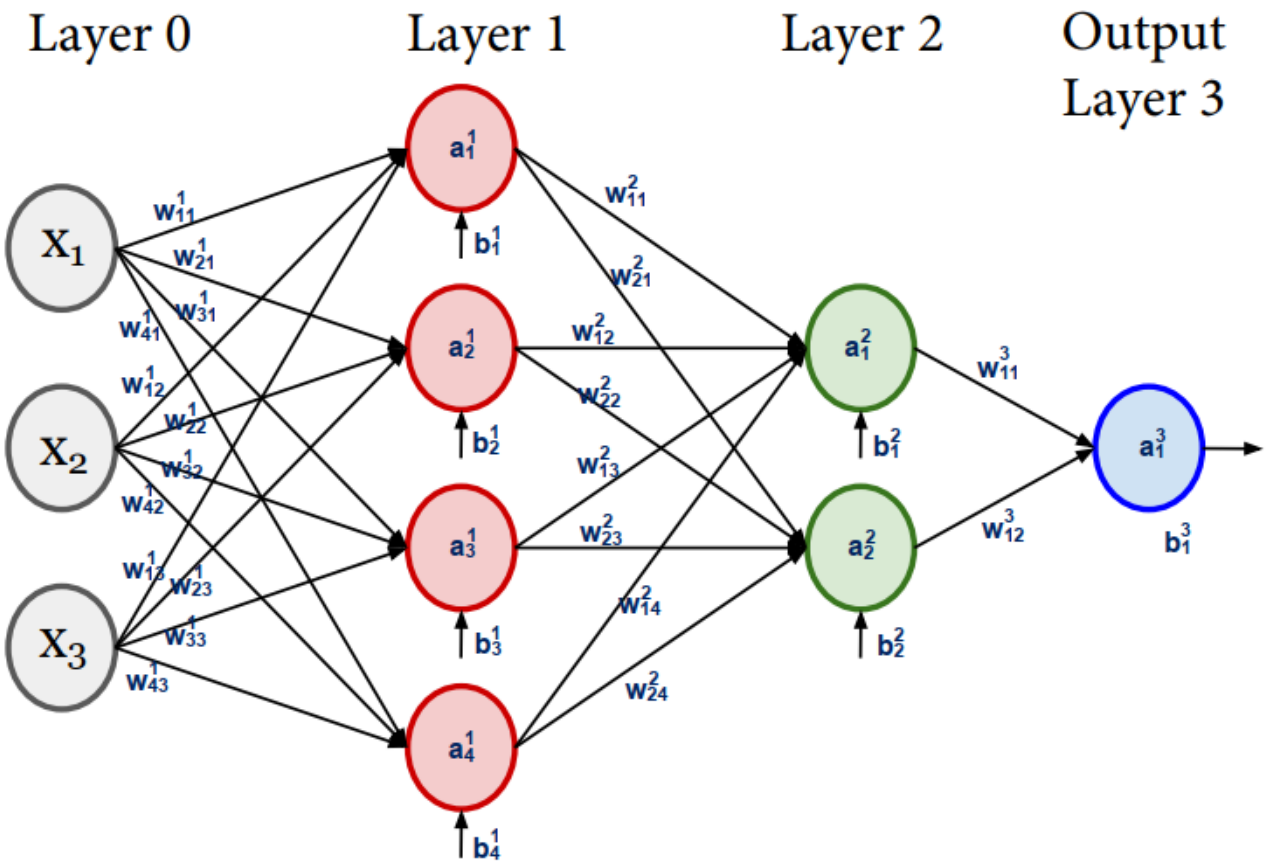
Sum over all the nodes from l-1 th layer that provide input to jth node in lth layer

Output of kth node in l-1th layer weighted by thing on the connecting arrow

Here I have filled some terminology. Can you guess the other ?



The overall picture looks like this



Now we have everything we need to find the partial derivatives. And we plan to use some loss, and then apply S.G.D. So just like before we can do everything as earlier and finally we will end up finding partial derivatives in order to feed the algorithm.

The output in our case is a_1^3 so we will be interested in

$$\frac{\partial a_1^3}{\partial w_{j,k}^l} \quad \& \quad \frac{\partial a_1^3}{\partial b_j^l}$$

Note that w's and b's are the parameter to be learned

Also note that a_1^3 is actually a real valued function

So instead of getting a complicated formula why don't I just find one derivative to see how things will work out. And so I choose to find

$$\frac{\partial a_1^3}{\partial w_{22}^1}$$

Just a random choice

Let's begin

> Finding $\dfrac{\partial a_1^3}{\partial w_{22}^1}$

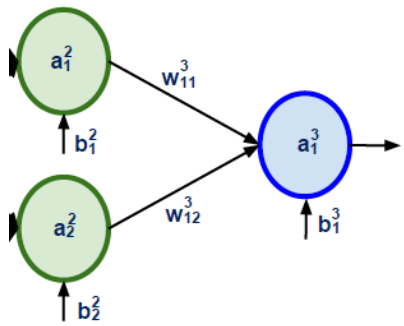
11> As I have stated repeatedly a_1^3 is actually a function. Lets write for the sake of clarity as

$$a_1^3 \equiv f^3\left(w_{11}^3 a_1^2 + w_{12} a_2^2 + b_1^3 \right) \quad \text{——— (A)}$$

$f^3_{()}$ is the functional form associated to the 3rd layer. Think of it as Tanh() or ReLU() if you want.

Now I wrote Eq (A) just to give you the idea of what we plan to do next. Just like we did earlier, we will say that $\dfrac{\partial a_1^3}{\partial w_{22}^1}$ is not so easy to find, but from our tree interpretation of derivatives earlier I can break it into

$$\frac{\partial a_1^3}{\partial w_{22}^1} = \frac{\partial a_1^3}{\partial a_1^2}\frac{\partial a_1^2}{\partial w_{22}^1} + \frac{\partial a_1^3}{\partial a_2^2}\frac{\partial a_2^2}{\partial w_{22}^1}$$



we can easily find these parts from Eq (A)

But now we are stuck with finding these two terms

$$\frac{\partial a_1^2}{\partial w_{22}^1} \quad \& \quad \frac{\partial a_2^2}{\partial w_{22}^1}$$

Let's try to find this one. Lets first see a_1^2.

a_1^2 will be something like

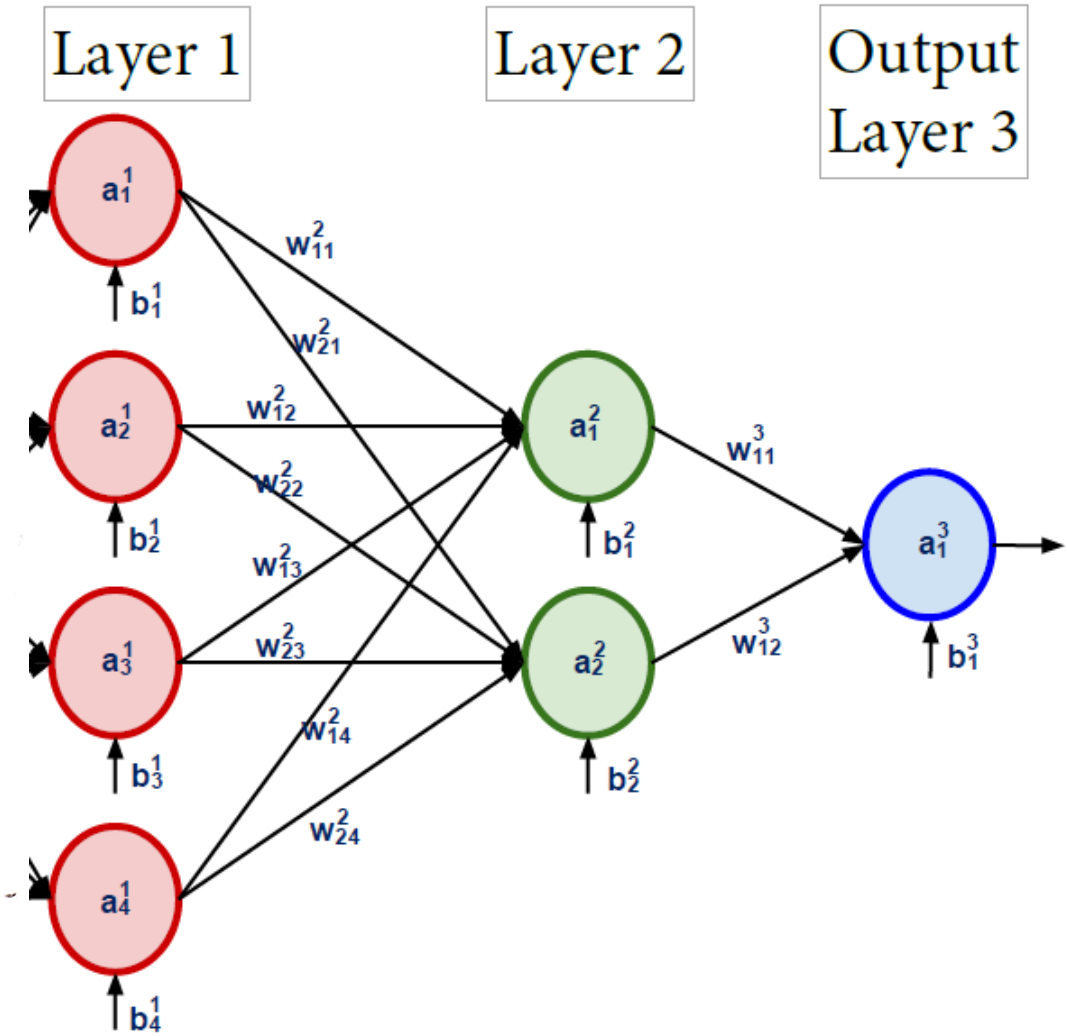
$$a_1^2 = f^2\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1 + w_{13}^2 a_3^1 + w_{14}^2 a_4^1 + b_1^2 \right) \text{ — (B)}$$

$f^2_{()}$ is the functional form of 2nd layer's activation function.
Think of it as Tanh() or ReLU()

And again I will use the derivative trick

I see a_1^2 takes in input 4 functions from previous layer

$a_1^1, a_2^1, a_3^1, a_4^1$ so we write



So we can write

$$\frac{\partial a_1^2}{\partial w_{22}^1} = \frac{\partial a_1}{\partial a_1^1}\frac{\partial a_1^1}{\partial w_{22}^1} + \frac{\partial a_1^2}{\partial a_2^1}\frac{\partial a_2^1}{\partial w_{22}^1} + \frac{\partial a_1^2}{\partial a_3^1}\frac{\partial a_3^1}{\partial w_{22}^1} + \frac{\partial a_1^2}{\partial a_4^1}\frac{\partial a_4^1}{\partial w_{22}^1}$$

These 4 terms can be easily calculated from (B) in prev page

The thing that we will be stuck with, is the four terms

$$\frac{\partial a_1^1}{\partial w_{22}^1}, \quad \frac{\partial a_2^1}{\partial w_{22}^1}, \quad \frac{\partial a_3^1}{\partial w_{22}^1}, \quad \frac{\partial a_4^1}{\partial w_{22}^1}$$

But I want to emphasize certain important points here

Look at the terms $\frac{\partial a^1}{\partial w_{22}^1}, \quad \frac{\partial a_3^1}{\partial w_{22}^1}, \quad \frac{\partial a_4^1}{\partial w_{22}^1}$

These all will be zero ..... because....

In a_1^1, a_3^1, a_4^1 functions, the input will never have w_{22}^1 term

a_1^1 will look like : $f^1(w_{11}^1 x_1 + w_{12}^1 x_2 + w_{13}^1 x_3 + b_2^1)$
a_3^1 will look like : $f^1(w_{31}^1 x_1 + w_{32}^1 x_2 + w_{33}^1 x_3 + b_3^1)$
a_4^1 will look like : $f^1(w_{41}^1 x_1 + w_{42}^1 x_2 + w_{43}^1 x_3 + b_4^1)$

$f^1()$ is the functional form of the activation of 1st layer.
Think of it as Tanh() or ReLU().

Do you see the term w_{11}^2 in any of the above the only term that will have w_{11}^2 is

$$a_2^1 \equiv f^1(w_{21}^1 x_1 + \underline{w_{22}^1 x_2} + w_{23}^1 x_3 + b_2^1) \quad \text{—— Ⓒ}$$

and from this we can find $\frac{\partial a_2^1}{\partial w_{22}^1}$, easily and therefore we can find

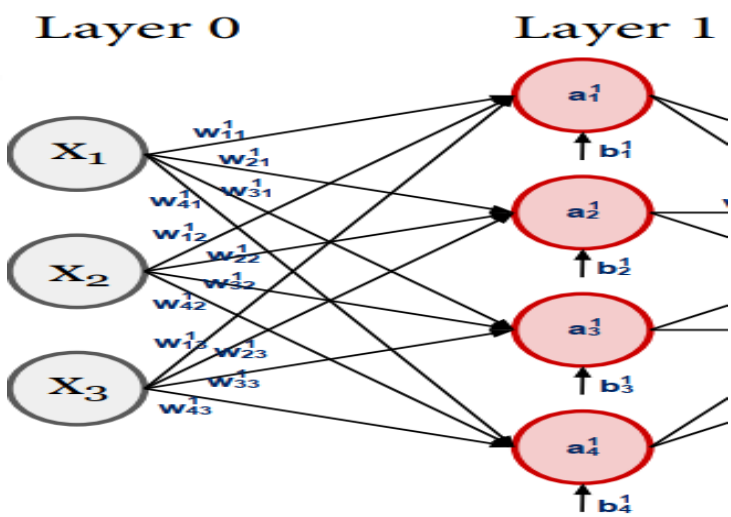
$$\frac{\partial a_1^2}{\partial w_{22}^1} = \underbrace{\frac{\partial a_1^2}{\partial a_1^1}}_{0} \frac{\partial a_1^1}{\partial w_{22}^1} + \frac{\partial a_1^2}{\partial a_2^1} \frac{\partial a_2^1}{\partial w_{22}^1} + \underbrace{\frac{\partial a_1^2}{\partial a_3^1}}_{0} \frac{\partial a_3^1}{\partial w_{22}^1} + \underbrace{\frac{\partial a_1^2}{\partial a_4^1}}_{0} \frac{\partial a_4^1}{\partial w_{22}^1}$$

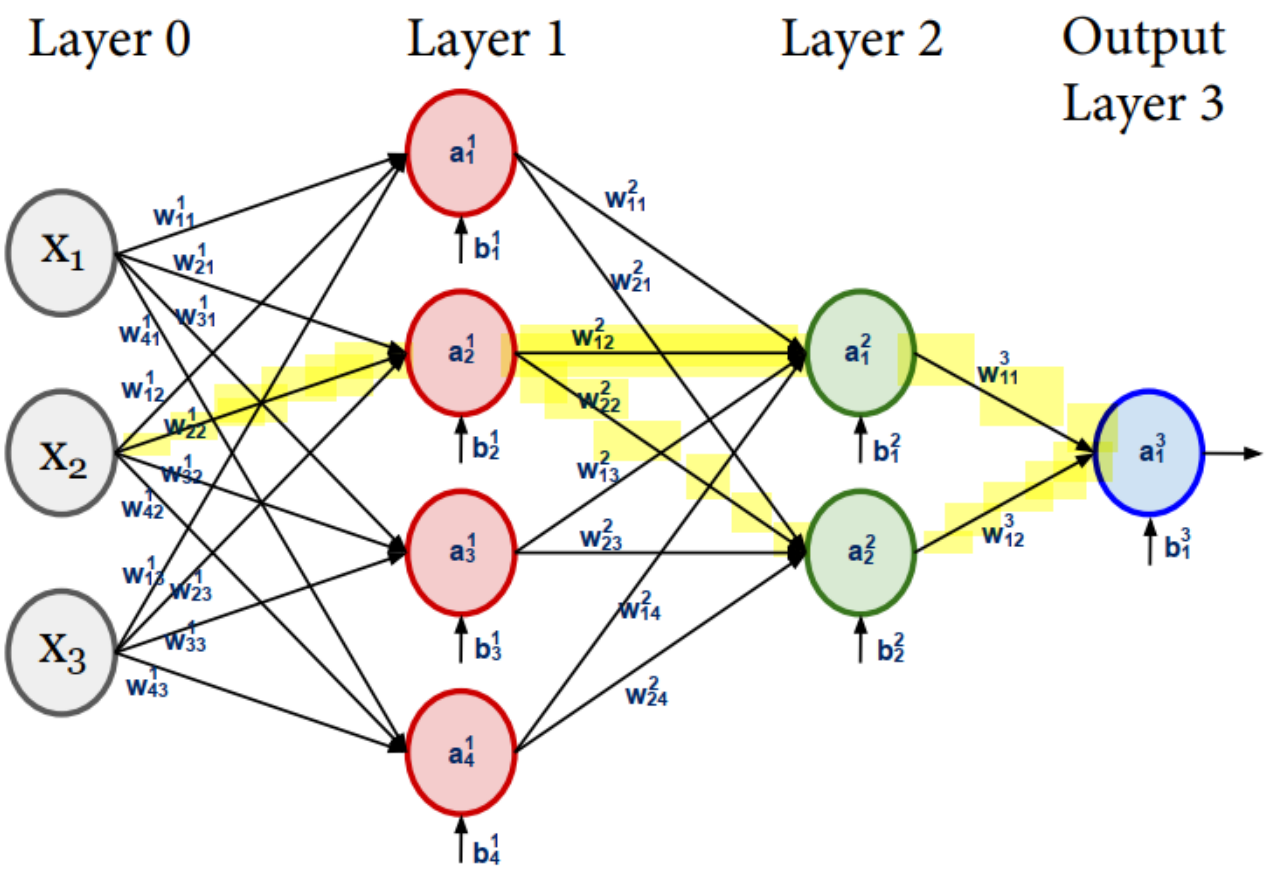
$$= \frac{\partial a_1^2}{\partial a_2^1} \frac{\partial a_2^1}{\partial w_{22}^1}$$

Can be calculated from Ⓑ

Can be calculated from Ⓒ

This can also be seen from the image below look how a_1^1, a_3^1, a_4^1 are <u>not</u> related to w_{22}^1





So here's the full picture to find

$$\frac{\partial a_1^3}{\partial w_{22}^1} = \frac{\partial a_1^3}{\partial a_1^2}\frac{\partial a_1^2}{\partial w_{22}^1} + \frac{\partial a_1^3}{\partial a_2^2}\frac{\partial a_2^2}{\partial w_{22}^1}$$

This can be found

$$\frac{\partial a_1^2}{\partial w_{22}^1} = \underbrace{\frac{\partial a_1^2}{\partial a_2^1}\frac{\partial a_2^1}{\partial w_{22}^1}}_{\text{This can be found}}$$

This can be found

$$\frac{\partial a_2^2}{\partial w_{22}^1} = \underbrace{\frac{\partial a_2^2}{\partial a_2^1}\frac{\partial a_2^1}{\partial w_{22}^1}}_{\text{This can be found}}$$

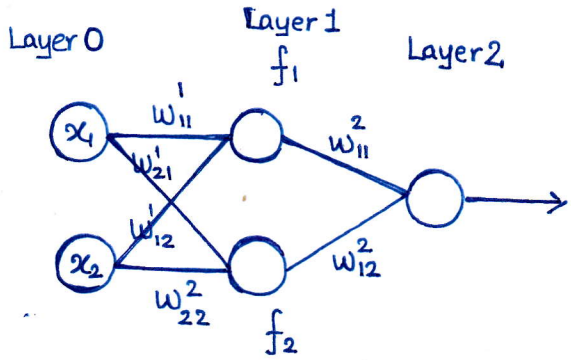
You can see how we backpropate through braches that contain the term w_{22}^1 highlighted above

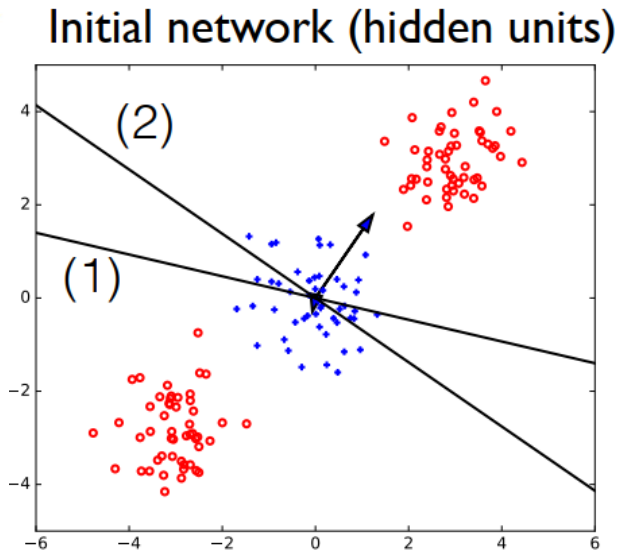
Similiarly we can find other derivatives

▷ Overcapacity

Prof T. now shows us some examples of how the SGD works to classify the points

Here we have the same data points as earlier. we will pass the data through one Layer & train the network ie find w's. The network looks like

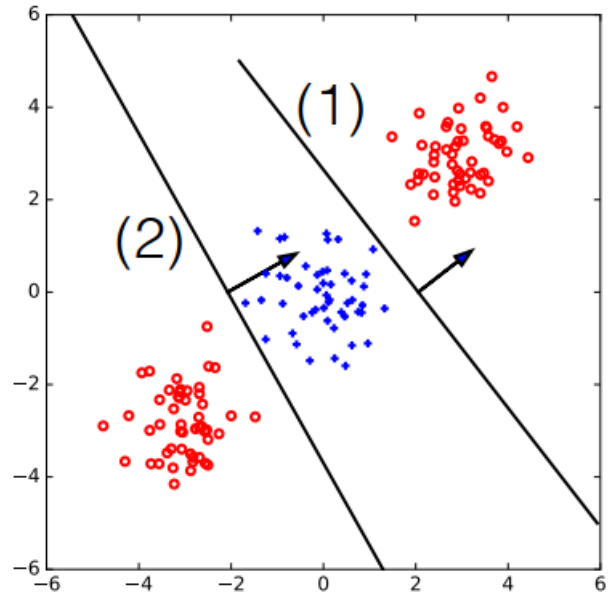




Prof T. wanted to find all these w's & biases (not shown) And he initialised the w vectors in the S.G.D code to some random values as seen by the random lines in the figure

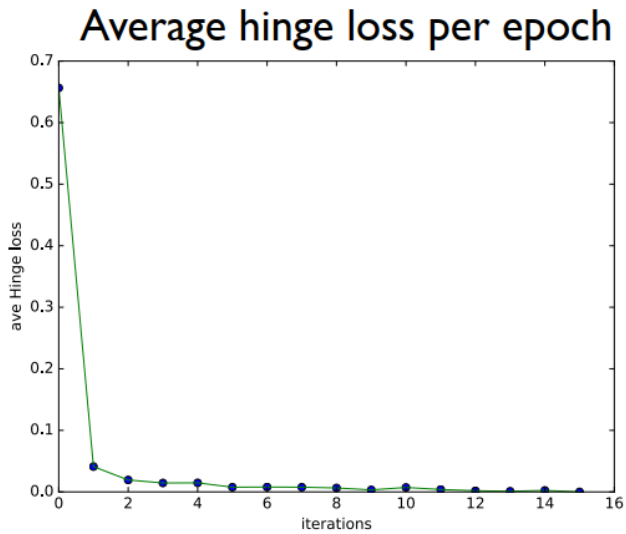
After certain iterations we see that in $x_1 - x_2$ space, the algorithm was able to find the lines that made the loss equal to zero

This is shown in adjoining figure :→



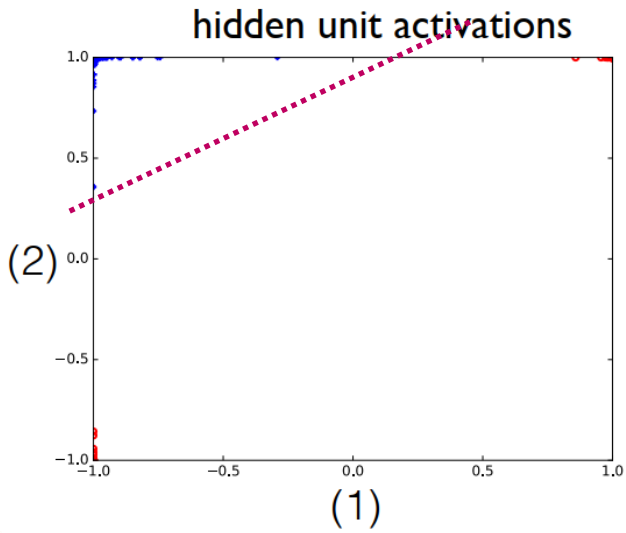
The loss Prof T. chose for this problem was hinge loss.

The adjoining figure shown, we see how the error approaches zero after about 12 iterations of the SGD iteration



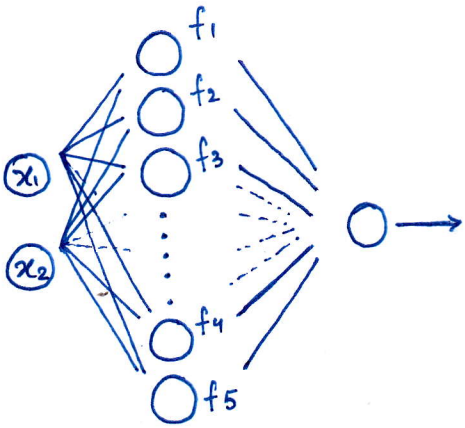
We would like to know what happens in the $f_1 - f_2$ space

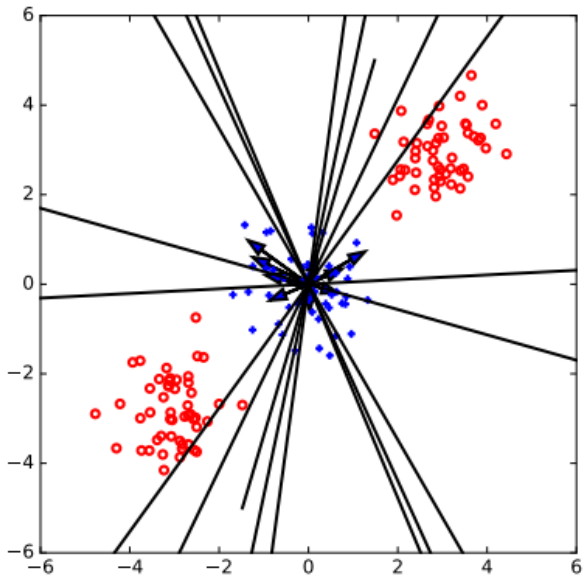
we see how the picture evolves in the f_1-f_2 space. The data now is clearly linearly seperable & we see how the SGD trained a proper model



But what if now for some reason Prof T. decided to insert 10 hidden units instead of two in the hidden layer?

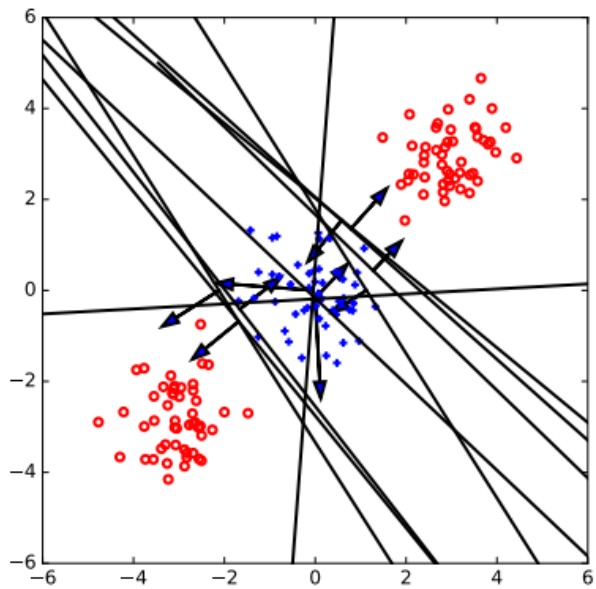
The network looks like





Again Prof T. Randomly initializes all the 10 lines w's as shown & train the model with SGD

The SGD obviously trains the model & after certain iteration we get the picture shown



Many of the hidden units end up doing something useful for the classification task ie they look similiar to the hidden unit solutions earlier with the case of two hidden units

But some other are doing random things.

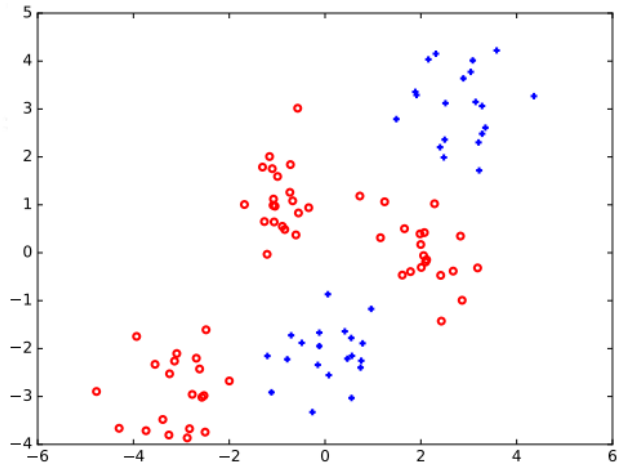
This relates to what is called overcapacity

We gave the problem more units (10) than were actually needed (2) to solve the problem therefore some became redundant.

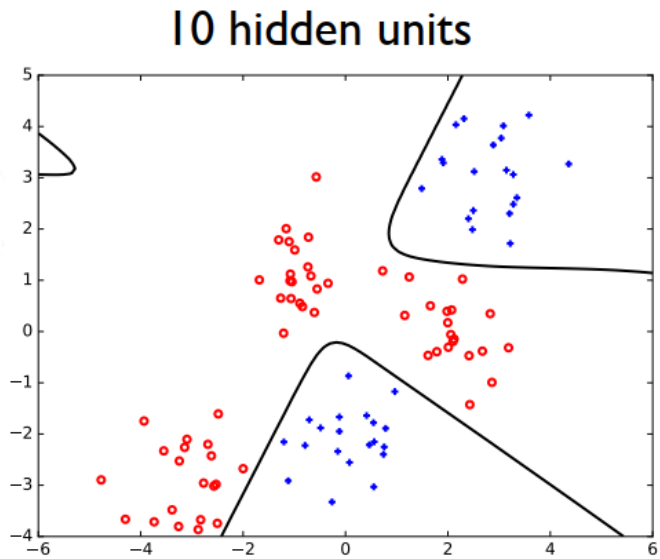
ie not all hidden units contributed usefully

But overcapacity is not always a bad thing

Now Prof T tackels a slightly difficult problem.
We clearly see or feel that in the adjoining figure, only two lines will not be enough ie only two hidden units will not be enough



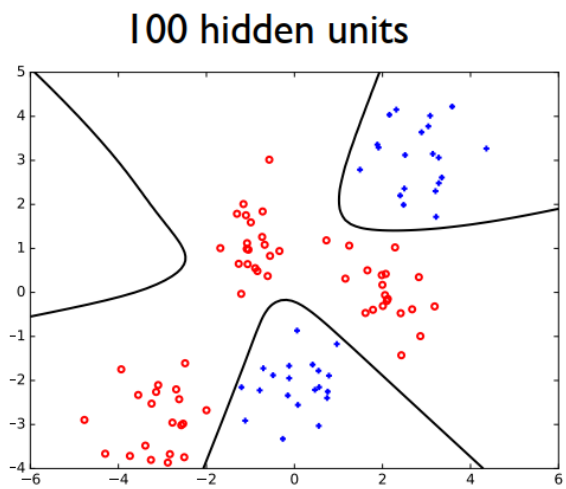
Here 10 units are useful as we see how the points are now classified properly



And not only that with 100 hidden units, we still get a proper classification that should be the case.

But we see that their is some odd part of the decision boundary at the top left corner, which seems unnecessary for the data.

It came because all the hidden units were initialised by setting the offset parameter to 0.
There was nothing to change that, and therefore the effect persisted in the final solution.

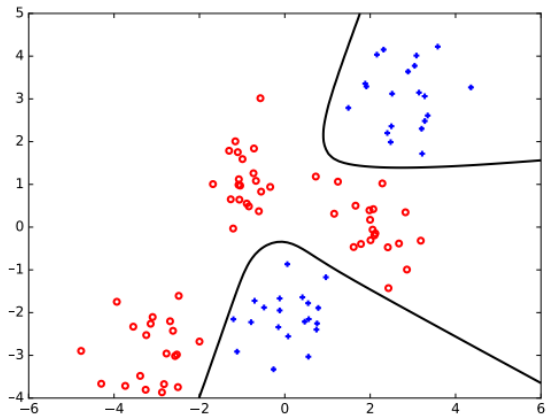


However....

If Prof T. initialised the offset parameter to non zero values

we get rid of that unnecessary part as shown in the adjoining figure.

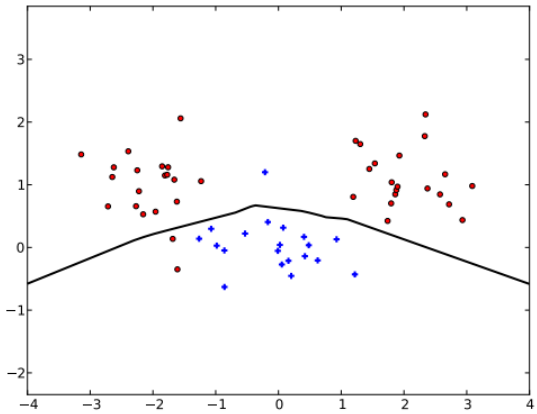
100 hidden units
(random offset initialization)



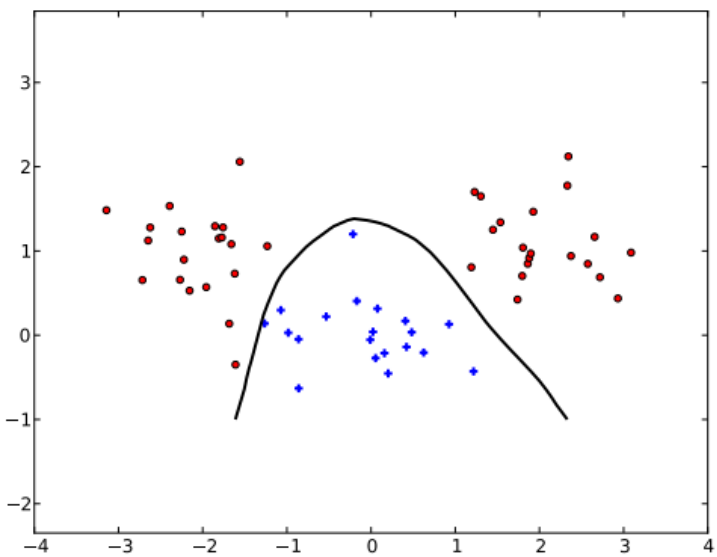
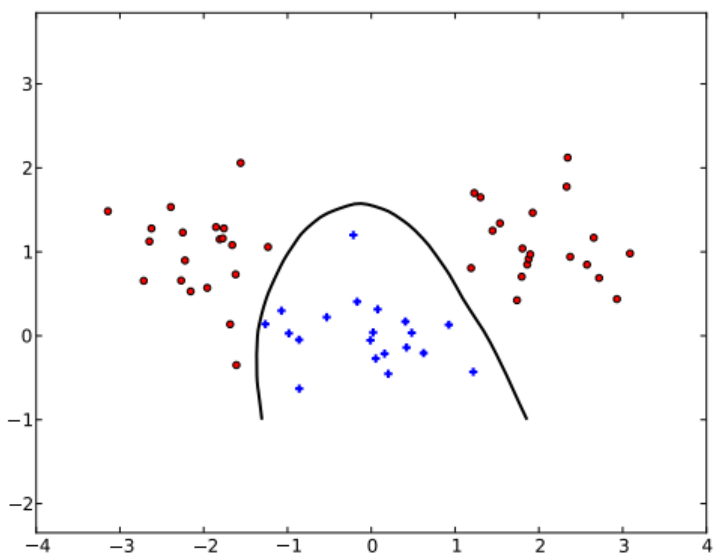
we now look at an example where over-capacity is helpful.

Here we see what happens when we trained our SGD with ReLU activation & 10 hidden units. Although most of the points are properly classified, some are not. we can now give our network over capacity by tweaking the number of units to 100 or even 500 these are shown in the figures below. Now we get the correct classification.

10 hidden units



100 hidden units



500 hidden units