Courses in this specialisation

Outline of the course

1) Neural Networks and Deep learning

Neural Networks has 2 steps

i) forward propagation ii) Backward propogation

Logistic Regression :

is an algorithm for Binary classification

$(x_1, y_1)\ (x_2, y_2) \ldots (x_m, y_m)$

$$X = \begin{bmatrix} | & | & | & | \\ x_1 & x_2 & x_3 \cdots & x_m \\ | & | & & | \end{bmatrix} \rightarrow n \times m \qquad Y = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \cdots & y_m \end{bmatrix} \rightarrow 1 \times m$$

We would Na probably expecting $n \times 1$ $X^T y$

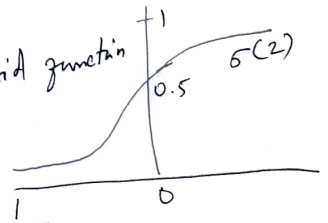
Given x, want $\hat{y} = P(y=1|x)$ $\quad 0 \le \hat{y} \le 1$

$x \in \mathbb{R}^{n_x}$ Parameters $w \in \mathbb{R}^{n_x}$, $b \in \mathbb{R}$

Output $\hat{y} = \sigma(w^T x + b)$ ← Sigmoid function

$z = w^T x + b$

$\sigma(z) = \frac{1}{1+e^{-z}}$



If z is large $\sigma(z) \approx \frac{1}{1+0}$

e^{-z} is very close to 0

So $\sigma(z)$ is close to 1

Conversely if z is small or $\sigma(z) = \frac{1}{1+e^{-z}} \approx \frac{1}{1+\text{big num}} \approx 0$

Cost function:

To train parameters w and b we need a cost function

$\hat{y} = \sigma(w^T x + b)$, where $\sigma(z) = \frac{1}{1+e^{-z}}$

Loss (error) function:

$L(\hat{y}, y) = -(y \log \hat{y} + (1-y)\log(1-\hat{y}))$

It helps to measure the accuracy of our algorithm by comparing $\hat{y}$ (output) & y (actual)

1) We would want the loss functions output value to be as small as possible.

If $y=1$ $L(\hat{y}, y) = -y \log \hat{y}$
$= -\log \hat{y} \Longrightarrow -\log 0 =$ infinity

If $\hat{y} = 0$
If $\hat{y} = 1$
$\log \hat{y} = 0$

So basically (1:1) if $y=1$ you want $\hat{y} = 1$

If $y=0$ $L(\hat{y}, y) = -\log(1-\hat{y})$

If $\hat{y} = 1 \Rightarrow \log(0) =$ infinity
$\hat{y} = 0 \Rightarrow \log(1) = 0$

Basically if $y=1$ we try to make $\hat{y}$ large
if $y=0$ we try to make $\hat{y}$ small

Loss function is measured on a single training example.

Whereas Cost function $J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$

$= \frac{-1}{m}\sum_{i=1}^{m}[y \log \hat{y} + (1-y)\log(1-\hat{y})]$

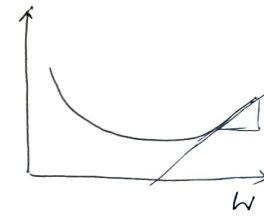
is the cost of your parameters w & b that minimizes cost J.

Average of loss function = Cost function

Gradient Descent:

The cost function measures how well your parameter w, b are doing

Gradient descent helps us to find w, b so that we can minimize $J(w,b)$ ~~optimally~~ efficiently.



Repeat {

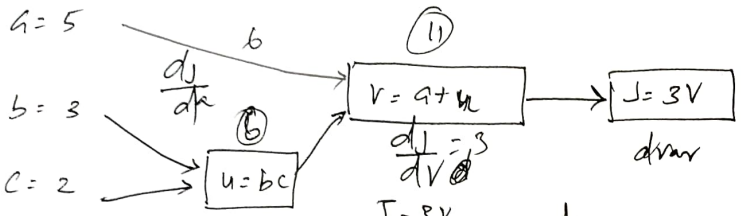
$w := w - \alpha \frac{dJ(w)}{dw}$

For simplicity we omit b so that we can deal with one variable. Then continuously minimize w

1) ~~Continuously subtract~~

1) Assign w to the difference between w and the derivative of the function $J(w)$ multiplied by a gradient descent constant α.

Computing derivatives



J = 3V

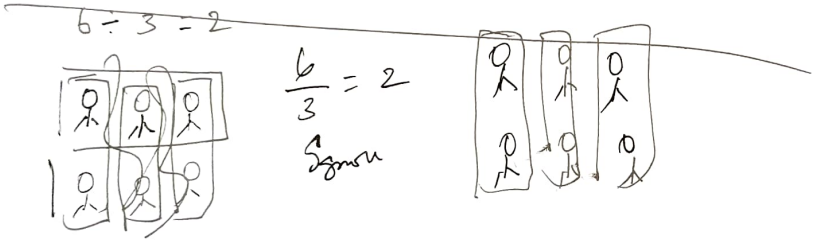
V = 11 → 11.001 } dJ/dV = .003/.001 = 3

J = 33 → 33.003 }

a = 5.00 → 5.001

V = 11.00 → 11.001

J = 33.003 dJ/da = .003/.001 = 3



$z = w^T x + b$ $\hat{y} = a = \sigma(z)$

$L(a,y) = -(y \log(a) + (1-y)\log(1-a))$

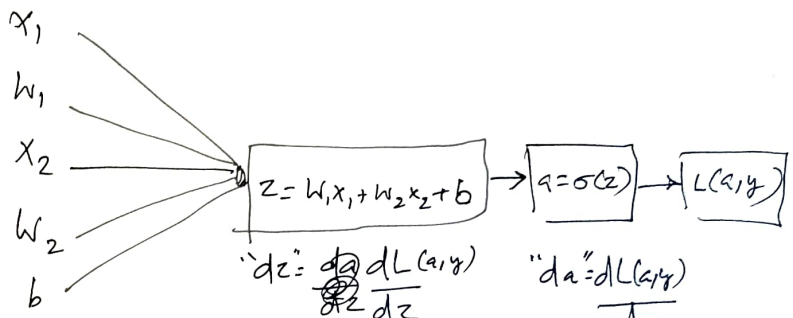
$\frac{dL(a,y)}{da} = \frac{1}{a} + \frac{1}{1-a}$

$= \frac{(1-a)+a}{a(1-a)} = \frac{1}{a(1-a)}$

$\frac{dL(a,y)}{da} = \frac{dL(a,y)}{da} \cdot \frac{da}{dz}$

$\frac{dL(a,y)}{da} = -\left(y\frac{1}{a} + (1-y)\frac{1}{1-a}\right)$

$= -\left(\frac{y}{a} + \frac{1-y}{1-a}\right)$



"dz" $= \frac{dL(a,y)}{dz} = \frac{dL}{da}\frac{da}{dz}$

$= a - y$ → Groucho

"da" $= \frac{dL(a,y)}{da} = -\frac{y}{a} + \frac{1-y}{1-a}$

$\frac{da}{dz} = a(1-a)$

$\Rightarrow \sigma(z) = \frac{1}{1+e^{-z}}$

$\frac{da}{dz} = \frac{d\,(\sigma(z))}{dz}$

$= \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right)$

$a = \frac{1}{1+e^{-z}}$

$a(1+e^{-z}) = 1$

$a + ae^{-z} - 1 = 0$

$\frac{da}{dz} = \frac{a + ae^{-z} - 1}{a(1+e^{-z})}$

$\frac{da}{dz} = a + a \cdot \frac{-1}{1+e^{-z}} \cdot e^{-z}$

$= a - ae^{-z}$

Your need to study calculus

Quotient Rule:

For a function $f(x) = \frac{g(x)}{h(x)}$, the derivative

$f'(x) = \frac{[g'(x) \times h(x)] - [g(x) \times h'(x)]}{h(x)^2}$

$\frac{[0 \times (1+e^{-z})] - [1 \times e^{-z}]}{(1+e^{-z})^2} = \frac{-e^{-z}}{1 + e^{-2z} + 2e^{-z}}$

$a = \frac{1}{1+e^{-z}}$

$a(1-a) = \frac{1}{1+e^{-z}} - \left(\frac{1}{1+e^{-z}}\right)^2$

$= \frac{1}{1+e^{-z}} - \frac{1}{1+e^{-2z}+2e^{-z}}$

Vectorization:

It gets rid of "for loops".

Logistic regression cost function:

$\hat{y} = \sigma(w^T x + b)$ where $\sigma(z) = \frac{1}{1+e^{-z}}$

Interpret $\hat{y} = p(y=1|x)$

Logistic regression with a neural network!

Shape (num_px, num_px, 3) where 3 is for 3 channels RGB

numpy shape of (3, 3, 3)

Eg: $\begin{bmatrix} [1,1,1]_{RGB} & [1,1,1]_{RGB} & [1,1,1] \\ [1,1,1] & [1,1,1] & [1,1,1] \\ [1,1,1] & [1,1,1] & [1,1,1] \end{bmatrix}$

Colour image matrix RGB

256 x 256, 256 x 256

shape (256, 256) 64 rows 64 Columns

64 x 64 x 3 = 64 Rows

a(1,1,1) ... a(1,64) ... a(64,1) ... a(64,64)

Common steps for pre-processing a new dataset are:

figure out the dimensions and shapes of the problem (m_train, m_test, num_px, ...)

Reshape the datasets such that each example is now a vector of size (num_px * num_px * 3, 1)

Standardize the data

Random Thoughts

image vector

X = 256 x 256 x 256

R G B

64 64 64

image1 [ ] [ ] [ ]

img2 .. .. ..

img3 .. .. ..

Height of image = 64 px

Width of img = 64 px

So the image is in colour then there is

64x64 for Red

64x64 for Blue

64x64 for Green

=> 64x64x3

So 1 image => 64 x 64 x 3

64x64 64x64 64x64

img1 = [ ] [ ] [ ]

Convert the image into 64x64x3 rows & then we will have

=> 12288 rows

& there are 209 images

X is the set of all images with 209 rows & 3 matrices of 64x64 flattened vector. Assume 5 images then we will have

$X = \begin{bmatrix} x^{(1)}_{(1,1)} & x^{(2)}_{(1,2)} & x^{(3)}_{(1,3)} & x^{(4)}_{(1,4)} & x^{(5)}_{(1,5)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x^{(1)}_{(12288,1)} & x^{(2)}_{(12288,2)} & x^{(3)}_{(12288,3)} & x^{(4)}_{(12288,4)} & x^{(5)}_{(12288,5)} \end{bmatrix}$ 1 — 12288

$W = [\ \ ]$

We calculate cost which is sum or avg of loss function

b

$1 \times 12288 \times 12288 \times 5 = 5^{1 \times 5}$ images represented as

$A = [a_1\ a_2\ a_3\ a_4\ a_5]$ either 0 or 1

$Y = [y_1\ y_2\ y_3\ y_4\ y_5]$

matrix

12288 X 5

Then W = 12288 size of (Y) = 5 $A = \sigma\left(\frac{1}{1+e^{-(w^T x + b)}}\right)$

Loss function $J = -y \log(a) + (1-y)\log(1-a)$ size (A) = 5

This gets applied to Y A

Cost fn $J = \frac{-1}{m} * \sum_{i=1}^{m} J(A,Y)$

$$A = \frac{1}{1 + e^{-(W^T X + b)}}$$

$W^T X = [w_1 \dots w_{12288}] \begin{bmatrix} x^1_{(1,1)} & x^2_{(2,1)} & x^3 & x^4 & x^5 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x^1_{12288,1} & x^2 & x^3 & x^4 & x^5 \end{bmatrix}$ th

$\Rightarrow w_0 \begin{bmatrix} x^1 \\ 1 \end{bmatrix} + w \begin{bmatrix} x^2 \\ 1 \end{bmatrix} + w \begin{bmatrix} x^3 \\ 1 \end{bmatrix} + w \begin{bmatrix} x^4 \\ 1 \end{bmatrix} + w \begin{bmatrix} x^5 \\ 1 \end{bmatrix}$

5 images

$W^T x + b = [1 \ 2 \ 3 \ 4 \ 5] + b$

5 value between 0 & 1

$A = [0 \ 1 \ 0 \ 0 \ 1]$

$dw: dw^T x \Rightarrow [A - Y]^T X$

Suppose Y = certain value A is certain value

when you change w by dw the $A - Y$ is certain value

$dw = A - Y$

That is initially Y = certain value

A = certain value

$A - Y = F$

You change w by dw then

$A - Y = F \pm$ the changes in w

Once you calculate A, remember our $m = 5$

$\Rightarrow A = [a_0 \ a_1 \ a_2 \ a_3 \ a_4]$

Then we need to apply the loss function

loss function has to be minimal or zero for our logistic regression to have worked.

Log function $J(w, b) = -(y \log a + (1-y) \log(1-a))$

How did we get this

The probability that

~~$P(Y|x) = 1$~~

$Y = 1 \quad P(A|x) = 1$

The chance the $Y = 1$ provided the probability of A is 1 & the chance that $Y = 0$

$Y = 0 \quad P(A|x) = 1 - A$

$Y = 1 \quad P(A|x) = A$

$Y = 0 \quad P(A|x) = 1 - A$

$z = W^T x + b$

$A = \frac{1}{1 + e^{-z}}$

Combining the two equations

$A^Y + (1 - A)^{(1-Y)}$

$Y \log A + (1 - Y) \log(1 - A)$

$J(w, b) = [y_0 - a_0 \ y_1 - a_1 \ y_2 - a_2 \ y_3 - a_3 \ y_4 - a_4]$

Cost of $J(w, b) = \frac{-1}{m} \sum_{i=1}^{m} J(w, b)$

We have to minimize the cost

Cost is found with A that depends on W, X, b

Y is the actual value

$dw = \frac{1}{m} X \cdot [A - Y]^T$

$X = \begin{bmatrix} x^1_{1,1} & x^2_{1,2} & x^3_{(1,3)} & x^4_{(1,4)} & x^5_{(1,5)} \\ | & | & | & | & | \\ x^1_{12288,1} & x^2_{(12288,2)} & x^3_{(12288,3)} & x^4_{(12288,4)} & x^5_{(12288,5)} \end{bmatrix}$ $[A - y]$

for eg:

~~[0 0 1 1 1]~~

for eg: $A = [0 \quad 0.5 \quad 0.75 \quad 1]$

All gradient of img 1 will be 0

Shape of w and dw will be the same img 2 will be same. There will be 12288 dws for so many images

$dw = \begin{bmatrix} x^1_{(1,1)} \\ \vdots \\ x^1_{12288} \end{bmatrix}$ $\quad dw = \frac{1}{m} \times \begin{bmatrix} | & | & | & | & | \\ dw_1 & dw_2 & dw_3 & dw_4 & dw_5 \\ | & | & | & | & | \end{bmatrix}$

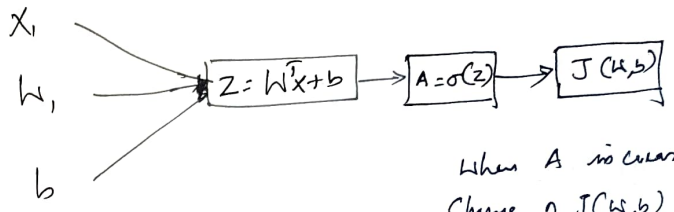
$\Rightarrow \frac{1}{m} \times \left(0 \begin{bmatrix} a^1 \\ | \\ x^1_{12288} \end{bmatrix} + 1 \begin{bmatrix} x^2_{(1,2)} \\ \vdots \\ x^2_{(12288,2)} \end{bmatrix} + 0.5 \begin{bmatrix} | \\ a^3 \\ | \end{bmatrix} + 0.75 \begin{bmatrix} | \\ x^4 \\ | \end{bmatrix} + 1 \begin{bmatrix} | \\ x^5 \\ | \end{bmatrix} \right)$

$\Rightarrow \frac{1}{m} \begin{bmatrix} | \\ dw \\ | \end{bmatrix}_{12288}$ which becomes $w = w - \alpha\, dw$

$b = \frac{1}{m} \sum_{i=1}^{m} (A - Y) =$

~~find for~~ the new w, b, As the iteration with old X, Y

forward + Backward propagation



When A increase by dA Change of $J(w,b)$ wrt w

$\frac{dJ(w,b)}{dA} = \frac{d}{dA} \frac{-1}{m} \sum_{i=1}^{m} y \log a + (1-y) \log(1-a)$

When Z increases by dz A increases. The rate of Change of A wrt Z is

$\frac{dA}{dz} = \frac{d\left(\frac{1}{1+e^{-z}}\right)}{dz}$

$\Rightarrow \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right)$

$\frac{dz}{dw} = \frac{d}{dw} w^T x + b$

$= dw^T x + b$

$\frac{dw}{dw} = \frac{dJ}{dA} \times \frac{dA}{dz} \times \frac{dz}{dw}$

$= \frac{dJ}{dw}$

$\frac{dJ}{db} = \frac{dJ}{dA} \times \frac{dA}{dz} \times \frac{dz}{db}$

$= \frac{dJ}{db}$

initialize parameters

loop :

Neural Network

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_m \\ | & | & & | \end{bmatrix}$$

$n \times m$

$$\begin{bmatrix} 1 & 3 & & 6 \\ 2 & 4 & \cdots & 7 \\ 3 & 5 & & 8 \end{bmatrix}$$

$Z = WX + b$ $\quad m = 3 \times 6$

$$W^{[1]} = \begin{bmatrix} - w_1^{[1]} - \\ - w_2^{[1]} - \\ - w_3^{[1]} - \end{bmatrix} \Rightarrow \begin{bmatrix} \theta_1^{[1](1)} & \theta_1^{[1](2)} & \theta_1^{[1](3)} \\ \theta_2^{[1](1)} & \theta_2^{[1](2)} & \theta_2^{[1](3)} \\ \theta_3^{[1](1)} & \theta_3^{[1](2)} & \theta_3^{[1](3)} \end{bmatrix}$$

$Z^{[1]} = W^{[1]} x + b^{[1]}$

$A^{[1]} = \sigma(Z^{[1]})$

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ 1 & 3 & 6 & 9 \\ 2 & 4 & 7 & 10 \\ 3 & 5 & 8 & 11 \end{bmatrix}^T \quad \begin{bmatrix} w_1^{[1](1)} & w_1^{[1](2)} & w_1^{[1](3)} \\ w_2^{[1](1)} & w_2^{[1](2)} & w_2^{[1](3)} \\ w_3^{[1](1)} & w_3^{[1](2)} & w_3^{[1](3)} \\ w_4^{[1](1)} & w_4^{[1](2)} & w_4^{[1](3)} \end{bmatrix}^T \quad m=3$$

$$Z^{[1]} = \begin{bmatrix} 1 \times w_1^{[1](1)} + 2 \times w_1^{[1](2)} + 3 \times w_1^{[1](3)} \\ 3 \times w_2^{[1](1)} + 4 \times w_2^{[1](2)} + 5 \times w_2^{[1](3)} \\ 6 \times w_3^{[1](1)} + 7 \times w_3^{[1](2)} + 8 \times w_3^{[1](3)} \\ 9 \times w_4^{[1](1)} + 10 \times w_4^{[1](2)} + 11 \times w_4^{[1](3)} \end{bmatrix} = Z^{[1]} \begin{bmatrix} \hat{y} & \hat{y} & \hat{y} \end{bmatrix}$$

$Z^{[1]} = W^{[1]} X + b$

$A^{[1]} = \sigma(Z^{[1]})$

$Z_1^{[1]}$ $Z_2^{[1]}$ $Z_3^{[1]}$

$A_1^{[1]}$ $A_2^{[1]}$ $A_3^{[1]}$

Shape of $X = 3 \times 4$ $\quad n=m$

Shape of $W = 4 \times 3$

$Z = 1 \times 4$

$A = 1 \times 4$

$W^2 = 4 \times 1$

$A W^2$

Activation function :

Similar to sigmoid function or sigmoid activation function

Given x:

$z^{[1]} = W^{[1]} x + b$

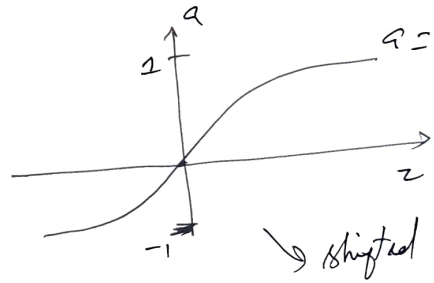
$a^{[1]} = \sigma(z^{[1]})$ $\quad g(z^{[1]})$

$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$

$a^{[2]} = \sigma(z^{[2]})$ $\quad g(z^{[2]})$

"g" could be a non-linear function which could be a sigmoid function

$a = \frac{1}{1+e^{-z}}$



$a = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

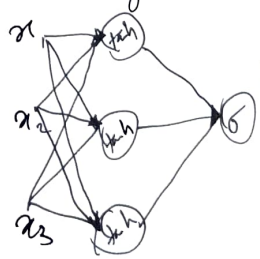
↳ shifted function of sigmoid

tanh function is superior

The exception is the output layer when the result is binary

$g^{[1]}(z^{[1]}) = \tanh(z^{[1]})$

$g^{[2]}(z^{[2]}) = \sigma(z^{[2]})$

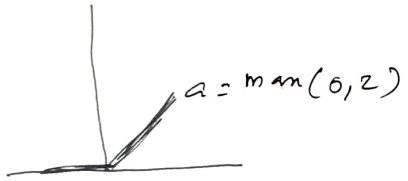


With very high or low Z values the slope of the function becomes very smaller.

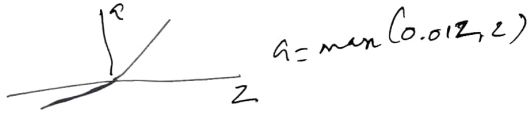
When the slope of the function becomes very small, it will slow down the gradient descent.

The alternative is Rectified Linear Unit

ReLu



Leaky ReLu



Forward Propagation for layer l

Input $a^{[l-1]}$

Output $a^{[l]}$, cache $(z^{[l]})$

$z^{[l]} = W^{[l]} \cdot a^{[l-1]} + b^{[l]}$

$a^{[l]} = g(z^{[l]})$

Backward Propagation for layer l

→ Input $da^{[l]}$

→ Output $da^{[l-1]}$, $dW^{[l]}$, $db^{[l]}$

$dz^{[l]} = da^{[l]} \times g^{[l]'}(z^{[l]})$

$dW^{[l]} = dz^{[l]} \cdot a^{[l-1]}$

$db^{[l]} = dz^{[l]}$

$da^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$

$\frac{da}{dz}$ $z = Wx + b$

$\frac{dL(a,y)}{da}$ $a = \sigma(z)$

$L(a,y)$

Rate of change of $L(a,y)$ wrt a is $\frac{dL(a,y)}{da}$

$dz^{[l]} = \frac{da}{dz}$

Rate of change of a (da) with respect to z $= \frac{da}{dz}$

$= \frac{d(\sigma(z))}{dz}$

$= \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right)$

$dw = dw \cdot x$

What are hyperparameters?

Parameters $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]} \ldots$

=

Hyperparameters: Learning rate α

# of iterations for gradient descent.

# of hidden layers

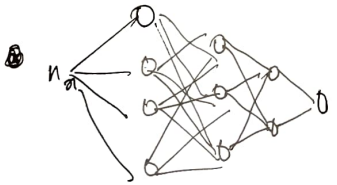
# of hidden units

Choice of activation function.

Often we store the value for $n^{[l]}$ in an array called layers

$n^{[l]} \rightarrow$ layer = dims $[n_0, 4, 3, 2, 1]$

$a^{[0]}$ $a^{[1]}$ $a^{[2]}$ $a^{[3]}$ $a^{[4]}$



4, 1

$X \Rightarrow Z_1^1 = W^1 X + b^1 \rightarrow A = \sigma(Z^1) \rightarrow Z_2^2 = W^2 A + b^2$

$\frac{dA^2}{dZ^2}$ $A^2 = \sigma(Z^2)$

$\frac{dA^2}{dZ^2} \frac{d}{dZ^2} \frac{1}{1 + e^{-(W^2 A + b^2)} Z^2}$ $\frac{dL}{dA^2}$ $L(A^2, Y)$

rate of change of dZ^2 wrt W dA^2

Image num_px, num_px, 3

example 4, 4, 3

x_1 [[[1 2 3 4; 2 3 4 1; 3 4 1 2; 4 1 2 3], [ ], [ ]]

4x4 4x4 4x4

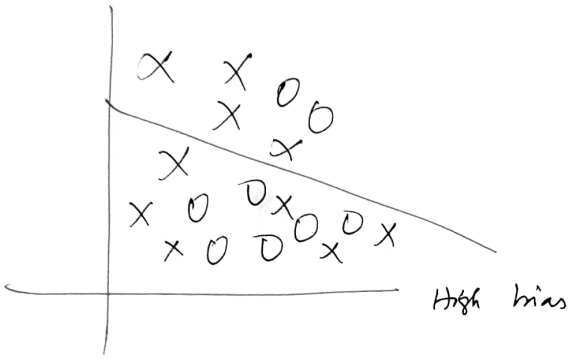
[[ ], [ ], [ ]]

4x4 4x4 4x4

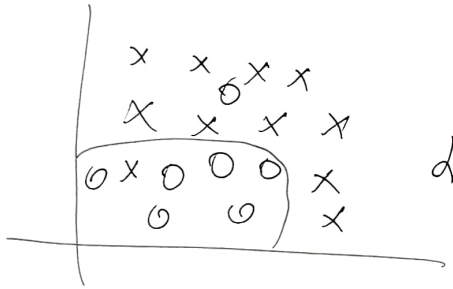
[[ ], [ ], [ ]]

Here $m = 3$

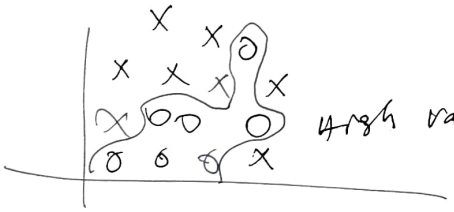
Bias / Variance :



High bias



Just right



High variance

High bias? Look at the training set

Solution → Bigger network, hidden units
run tests longer
(Maybe NN architecture that is suited)

High variance?: Look at dev set performance.

Solution: More data, Regularization
Maybe more appropriate NN architecture

Neural Network Regularization:

$$\frac{\lambda}{2m} \sum_{l=1} \| W^{[l]} \|^2$$

$$\| W^{[1]} \|^2 + \| W^{[2]} \|^2 + \| W^{[3]} \|^2 + \dots$$

$$\| W^{[l]} \|^2 = \sum_{i=1}^{n^{[l-1]}} \sum_{j=1}^{n^{[l]}} \left(W_{ij}^{[l]} \right)^2$$

frobenius norm

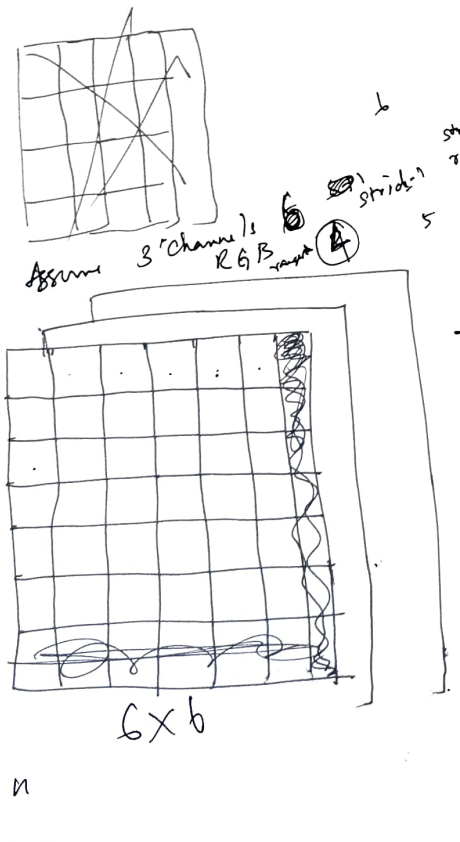
Convolutional Neural Networks:

Edge detection example:

A 6×6 convolved with 3×3 matrix produces a 4×4

To select a 2×2 slice at the upper left corner of a matrix a_prev (5,5,8), you would do

```
a_slice_prev = a_prev[0:2, 0:2, :]
                      ↓      ↓   ↓
                   Height  Width  filters or Channels
                   0-2     0-2    All
```



Stride = 3 range = 2

Stride = 4 range = 1

stride = 2 range = 2

Convolve pseudo code / Algorithm

Steps:

1) Iterate over every sample of m
2) Select the ~~slice~~ shape of the input matrix that matches the filter matrix shape
3) Convolve and output

Now the algorithm:

$$n_H = \frac{n_{H\_prev} - f + 2p}{stride} + 1$$

$$n_H \times stride = n_{H\_prev} - f + 2p + stride$$

$$n_{H\_prev} = n_H \times stride + f - 2p - stride$$

$$= stride(n_H - 1) + f - 2p$$

$\Rightarrow$ When $n_H = 0$

~~for h~~

n_{H_prev} starting position: 0 1

n_{H_prev} ending position: 3 4

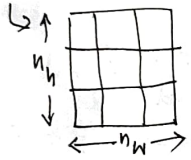
h = 1
$h = 1 + stride = 2$
$= 3 - 1 = 2$
h = 2
$= 2 + stride = 2$
$= 4$
h = 3
$= 4 + stride = 2$
$= 5$

Convolutional layer backward pass:

Computing dA:

This is the formula for computing dA with respect to the cost for a certain filter W_c and a given training example.

$$dA+ = \sum_{h=0}^{n_H} \sum_{w=0}^{n_W} W_c \times dZ_{hw}$$



This is an important topic that needs to be covered.

Object Detection:

Anchor Boxes:

YOLO algorithm:

Triplet images:

$$\mathcal{L}(A,P,N) = \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha$$

X — $z = Wx + b$ — $y = \sigma(z)$

W

Back propagation

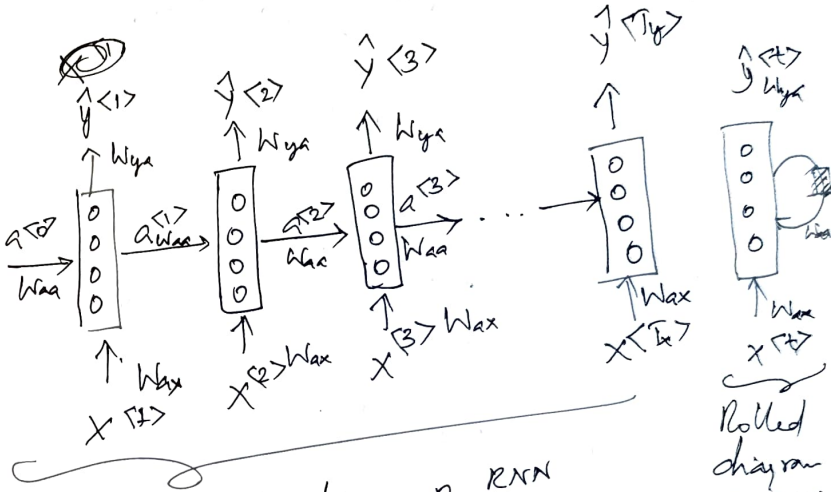
$$X = \begin{bmatrix} 1 & 5 & 7 \\ 3 & 8 & 2 \\ 4 & 9 & 5 \end{bmatrix} \qquad W = \begin{bmatrix} 4 & 5 & 6 \\ 8 & 7 & 10 \\ 2 & 4 & 3 \end{bmatrix}$$

$z = Wx + b$

$a = \sigma(z)$

RNN:

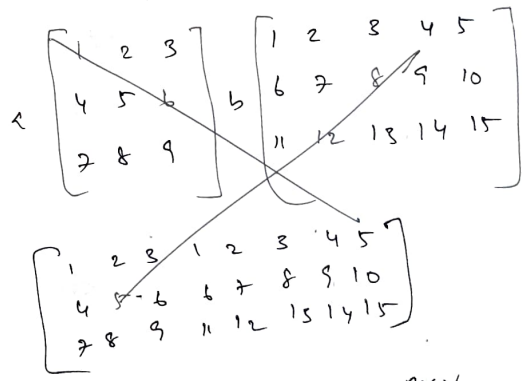
Recurrent Neural Networks



Unrolled diagram of RNN

Rolled diagram of RNN

The RNN scans through the data from left to right



Vanishing gradients with RNN

RNN

x W_{ax}

$[\quad]$ $[\quad]$ $+b$

a W_{aa}

$[\quad]$ $[\quad]$

Which is

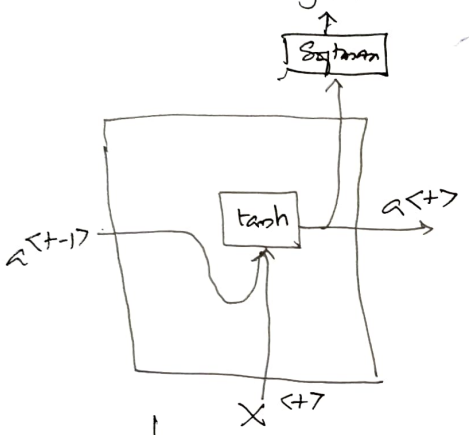
$\hat{y}$ W_{yy}

$[\quad]$ $[\quad]$

Gated Recurrent Unit (GRU)

RNN unit

$$a^{<t>} = g(W_a [a^{<t-1>}, x^{<t>}] + b_a)$$



↓ Hidden layer of RNN unit

GRU (Simplified)

c = memory cell

$c^{<t>} = a^{<t>}$

$\tilde{c}^{<t>} = \tanh(W_c [c^{<t-1>}, x^{<t>}] + b_c)$ ⇒ Candidate for replacing $c^{<t>}$

Gamma $\Gamma_u = \sigma(W_u [c^{<t-1>}, x^{<t>}] + b_u)$ → is always 0 or 1

↳ "update"

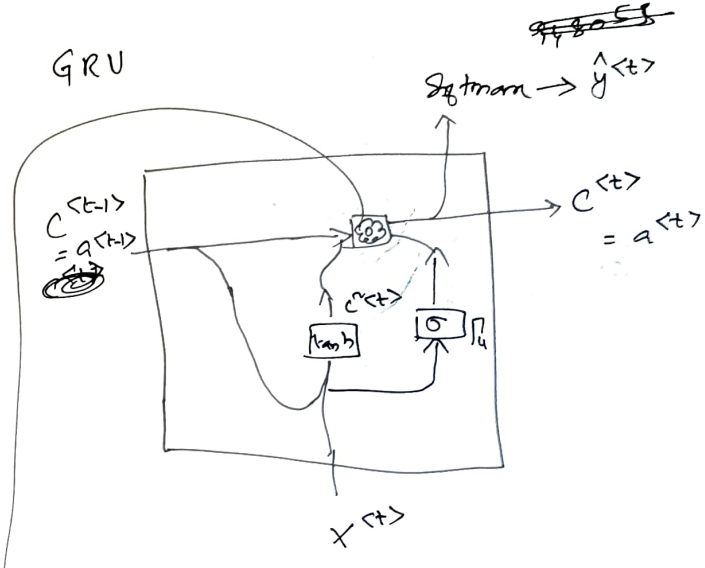
$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

Element wise multiplication

↳ If Γ_u or the gate is 1 then set $c^{<t>} = \tilde{c}^{<t>}$

else set $c^{<t>} = c^{<t-1>}$

GRU



$c^{<t>}$ can be a vector

Long Short Term Memory – LSTM

LSTM Unit

Architected Schmidhuber

long short term memory

$$\tilde{c}^{<t>} = \tanh(W_c [a^{<t-1>}, x^{<t>}] + b_c)$$

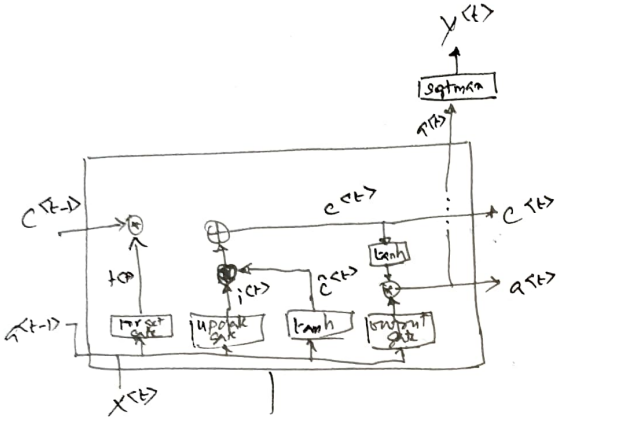
Update Gate $\Gamma_u = \sigma(W_u [a^{<t-1>}, x^{<t>}] + b_u)$

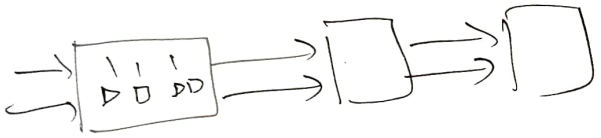
Forget Gate $\Gamma_f = \sigma(W_f [a^{<t-1>}, x^{<t>}] + b_f)$

Output Gate $\Gamma_o = \sigma(W_o [a^{<t-1>}, x^{<t>}] + b_o)$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

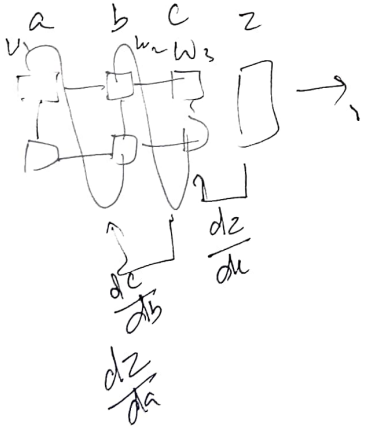
$$a^{<t>} = \Gamma_o * c^{<t>}$$





GRU is a simpler model, computationally faster

LSTM more powerful & flexible



Embeddings:

Word2vec

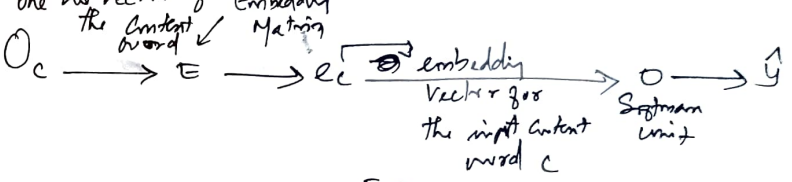
Skip Grams:

Mikolov et al

x → y

Context c ("orange") → Target t (juice)

6257 4834



$e_c = E O_c$

Softmax: $P(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$

Probability of different target words given the input context word

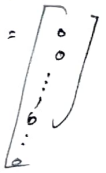
$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$$

"θ_t" is the parameter associated with output "t"

Loss function for Softmax

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{10000} y_i \log \hat{y}_i$$

y is a one hot vector



The target will be a one hot vector

How to sample context 'c'?

Sample uniformly random from a training corpus

Negative Sampling:

I want a glass of orange juice to go along with my cereal

we are going to predict

Orange Juice is it a Context target

Context	word	target?
Orange	Juice	1
Orange	king	0

Logistic regression model

$P(y=1|c,t) = \sigma(\theta_t^T e_c)$

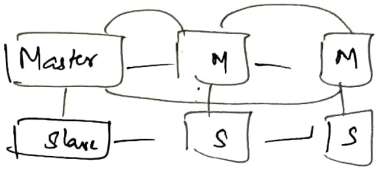
Global word Vectors:

↳ Global Vectors for word Representation

I want a glass of orange juice to go along with my cereal

Ot A = P

It B = R



Cosine similarity:

To measure how similar two words are, we need to measure the degree of similarity between two embedding vectors for the two words. Given two vectors u and v, cosine similarity is defined as follows:

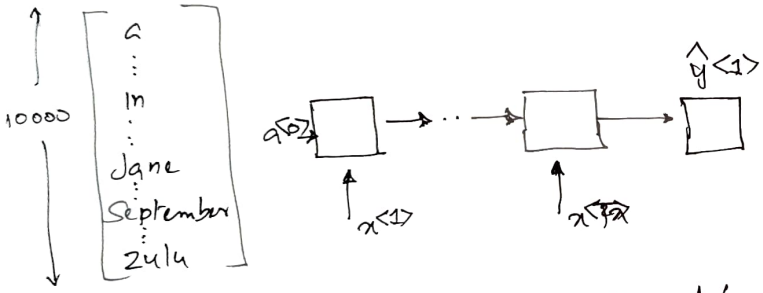
$$\text{Cosine Similarity}(u,v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2} = \cos(\theta)$$

Sequence to Sequence Models:

Beam Search:

Jane visit l'Afrique en septembre

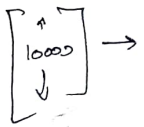
⟶ Jane is visiting Africa in september



Beam search "B" is beam width. When b=3 beam search will consider 3 possibilities at a given time.

b = 3

⇒



Bleu Score:

Bilingual Evaluation Understudy

Bleu score on bigrams

↳ pairs of words appearing next to each other.

Attention Model:

Context is sum of the different features at different time steps weighted by attention weights.

$$\sum_{t'} \alpha^{<1,t'>} = 1 \qquad c^{<1>} = \sum_{t'} \alpha^{<1,t'>} a^{<t'>}$$

Sum of all attention weights should be 1

$$c^{<t>} = \left(\overrightarrow{a}^{<t>}, \overleftarrow{a}^{<t>}\right)$$

$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t=1}^{T_x} \exp(e^{<t,t'>})}$$

$b = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$ $c = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$ $d = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$

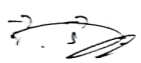
$b + c = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$

$a = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ $c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\begin{bmatrix} -1 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

-2

4



$\vec{r} \cdot \vec{s}$

10

$|r| = 5$ $|s| = \sqrt{161}$

$\frac{2}{12} = \frac{1}{6}$

$\vec{r} \cdot \vec{s} = 10$

$\frac{10}{25} \begin{bmatrix} 3 \\ -4 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{6}{5} \\ \frac{-8}{5} \\ 0 \end{bmatrix}$

$a = \begin{bmatrix} 5 \\ 0 \\ 12 \end{bmatrix}$ $b = \begin{bmatrix} 0 \\ 5 \\ 12 \end{bmatrix}$

$a + b = \begin{bmatrix} 5 \\ 5 \\ 24 \end{bmatrix}$ $|a+b| = 25+25+576 = 626$

$|a| = 25$ $|b| = 25+144 = 169$

$a = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$ $b = \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}$

$a = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ $b = \begin{bmatrix} 1 \\ 1 \\ 6 \end{bmatrix}$ $c = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ $\frac{1}{2}$

$a = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ $b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$ $c = \begin{bmatrix} -3 \\ 1 \\ -2 \end{bmatrix}$

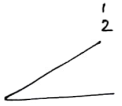
$+ \quad + \quad -$

$\begin{matrix} 3 \\ -1 \\ 2 \end{matrix} + \begin{matrix} -3 \\ 1 \\ -2 \end{matrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

11:55

12:59

2



$\frac{3}{2}$

Calculus:

Area of a Circle πR^2

Difference in radius from one ring to next

Total radius $dr_0 + dr_1 + dr_2 + dr_3 + dr_4$

$= r$

Ans $r = 3$ Circumference = height of the strip $= 2\pi r$

- $2\pi(0.0)(0.1)$
- $2\pi(0.1)(0.1)$
- $2\pi(0.2)(0.1)$

$\simeq 2\pi(2.9)(0.1)$

What is a derivative?

$(x+dx)^2$

$(x+dx)^3$

$(x+dx)(x+dx)(x+dx)$

$\Rightarrow x^3 + 3x^2dx + 3x\,dx^2 + dx^3$

ignore

$3x^2dx$

$x^3 + 3x^2dx$

$(x+dx)(x+dx)$

$x^2 + xdx + xdx + dx^2$

$x^2 + 2x\,dx$

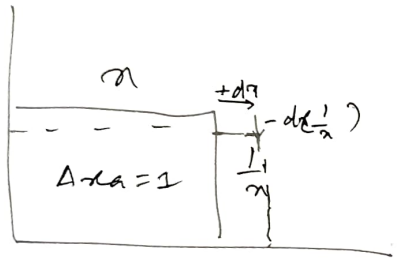
$f(x) = \frac{1}{x}$

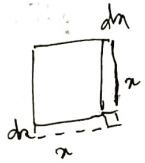
$$\frac{d\left(\frac{1}{x}\right)}{dx} = ?$$

$\frac{1}{2} + \frac{1}{2}$

$\frac{1}{2} + \frac{1}{4}$

$\frac{2+1}{4} = \frac{3}{4}$





$x\,dx + x\,dx + dx^2$

$2x\,dx$

$\frac{1}{x}$; $f(x^{-1}) = -x^{-2}$; $= -\frac{1}{x^2}$

$x \propto -d\left(\frac{1}{x}\right)$

Change in Area

$(x+dx) \times -d\left(\frac{1}{x}\right)$

$-x\,d\left(\frac{1}{x}\right) = -dx\,d\left(\frac{1}{x}\right)$

$= -\frac{1}{x^2}$?

$-x\,dx^{-1}$

$(x+dx)\left(\frac{1}{x} - d\left(\frac{1}{x}\right)\right)$

$\frac{x \times 1}{x} - x\,d\left(\frac{1}{x}\right) + \frac{1}{x}dx - dx\,d\left(\frac{1}{x}\right)$

$x \times \frac{1}{x} - x\,d\left(\frac{1}{x}\right) + \frac{1}{x}dx - dx \cdot d\left(\frac{1}{x}\right) = 1$

$1 - x\,d\left(\frac{1}{x}\right) + \frac{1}{x}dx - dx \times d\left(\frac{1}{x}\right) = 1$

$\frac{dy}{dx} : -x\,d\left(\frac{1}{x}\right) + \frac{1}{x}dx = 1$

$\frac{dy}{dx} = -\frac{1}{x^2}$

$\frac{dy}{dx} = \frac{1}{x} - x$

$(x+dx)\left(\frac{1}{x} - d\left(\frac{1}{x}\right)\right) = 1$

$= x \times \frac{1}{x} - x \times d\left(\frac{1}{x}\right) + dx \times \frac{1}{x} - dx \times d\left(\frac{1}{x}\right)$

$= 1 - x \times d\left(\frac{1}{x}\right) + \frac{dx}{x} - dx \times d\left(\frac{1}{x}\right)$

$x \times d\left(\frac{1}{x}\right) + \frac{dx}{x} = \frac{1}{x^2}$?

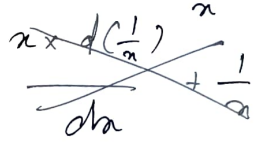
$x \times dx^{-1} + \frac{dx}{x}$

$\frac{1}{x}$

x^{-1}

$-1x^{-2}$ is the derivation

$\Rightarrow \frac{1}{x^2}$



$x \times \frac{1}{x} - x \times d\left(\frac{1}{x}\right) + \frac{1}{x}dx - dx \cdot d\left(\frac{1}{x}\right) = 1$

$\Rightarrow 1 - x \times d\left(\frac{1}{x}\right) + \frac{1}{x}dx - 1 = 1$

$dy : -x \times d\left(\frac{1}{x}\right) + \frac{1}{x}dx = 1$

$\rightarrow x \times d\,x^{-1} + \frac{1}{x}dx$

$f = x \times \frac{1}{x}$

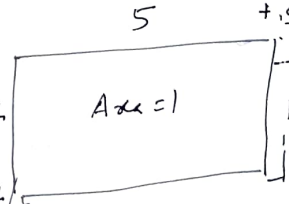
$dy = (x+dx) \times \left(\frac{1}{x} - d\left(\frac{1}{x}\right)\right)$

$\frac{dy}{dx} = -\frac{1}{x^2}$

$\frac{1}{5} - \frac{1}{25} = \frac{5-1}{25} = \frac{24}{25}$

$\frac{1}{5} - \frac{1}{30} = \frac{6-5}{30} = \frac{1}{30}$

Suppose the width increase by 6



$5 \times \frac{1}{5} = 1$

$\frac{1}{5} - \frac{1}{30} = \frac{1}{6}$

$\frac{1}{5} - \frac{(-2)}{15} = \frac{1}{5} + \frac{2}{15} = \frac{3+10}{15}$

$\frac{5-1}{30} = \frac{4}{30} = \frac{2}{15}$

$\frac{1}{5} - x = \frac{1}{6}$

$-x = \frac{1}{6} - \frac{1}{5} = \frac{5-6}{30}$

$-x = -\frac{1}{30}$

$x = \frac{1}{30}$

$\frac{1}{5} - \frac{4}{30}$

$\frac{30-20}{150} = \frac{10}{150} = \frac{1}{15}$

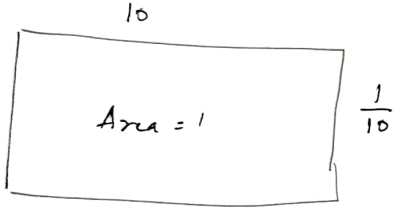
5 | 30.5 | 1.1 | 6

$(x+dx)\times -d\left(\frac{1}{x}\right) + \left(\frac{1}{x} - d\left(\frac{1}{x}\right)\right)\times dx$

$-x\,d\left(\frac{1}{x}\right) - dx\,d\left(\frac{1}{x}\right) + \frac{1}{x}dx - d\left(\frac{1}{x}\right)dx$

$-x\,d\left(\frac{1}{x}\right) - 2dx\,d\left(\frac{1}{x}\right) + \frac{1}{x}dx$

What is the derivative of $\frac{1}{x}$



$\frac{1}{10} - x = \frac{1}{12}$

$\frac{1}{10} = \frac{1}{12} + x$

$x = \frac{1}{10} - \frac{1}{12}$

$\frac{6-5}{60} = \frac{1}{60}$

2 | 10, 12
5 | 5, 6
6 | 1, 6

$\frac{1}{10} - \frac{1}{60}$

$\frac{6-1}{60} = \frac{5}{60} = \frac{1}{12}$

10 | 10, 60
1, 6

$\frac{1}{10} - x = \frac{1}{11}$

$x = \frac{1}{10} - \frac{1}{11} = \frac{11-10}{110}$

$= \frac{1}{110}$

$\frac{1}{10} + \frac{1}{110}$

10 | 10, 110
11 | 11, 1

$\frac{11+1}{110} = \frac{12}{110}$

What is mean :-

Average of the dataset

$X = \{x_1, x_2, x_3 \dots x_n\}$

$M = \frac{1}{n}\sum_{i=1}^{n} x_i$

What is ~~~~ :- Standard deviation

The average distance of points or data from the mean.

Eg :- $\{1, 2, 3, 4, 5\}$

$M = \frac{1+2+3+4+5}{5} = \frac{15}{5} = 3$

So standard deviation σ would be : $\{-2, -1, 0, 1, 2\}$

The values have to be absolute

$\Rightarrow \frac{|-2| + |-1| + |0| + |1| + |2|}{5} = \frac{6}{5} = 1.2$

on an average they are all 1.2 away from the mean

So what is normalising

$\frac{x-M}{\sigma} \Rightarrow$

4-M