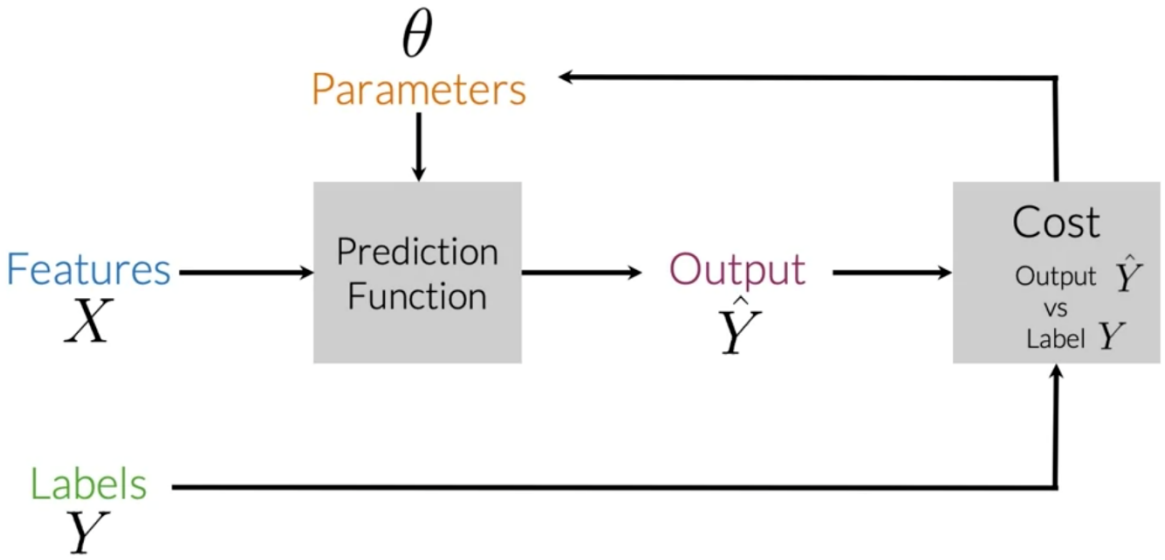
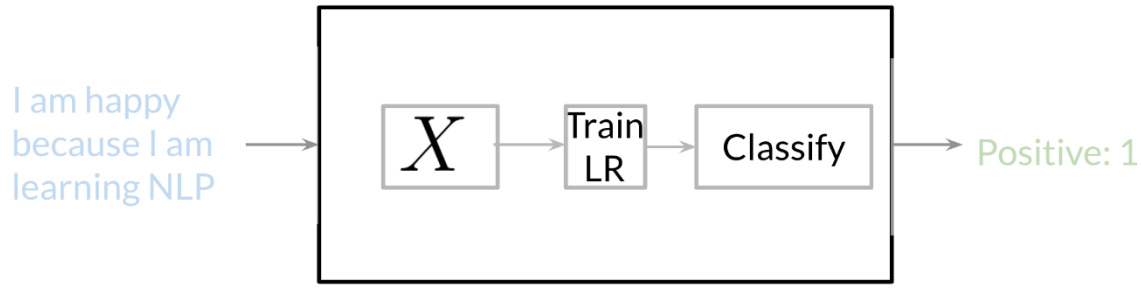
Supervised ML & Sentiment Analysis

Di supervised machine learning, Biasanya terdapat input X. *X adalah input feature yang digunakan untuk melakukan prediksi* , yhat. Setelah mendapatkan hasil prediksi, selanjutnya dilakukan komparasi dengan true labelnya Y. Hal ini akan memberikan nilai ***cost function*** yang digunakan untuk *update* ***parameters,theta*** **.**



Kalau misalnya pengen melakukan sentiment analisis di twitter, pertama tama kamu harus merepresentasikan sebuah teks menjadi fitur atau dijadikan sebagai (X), kemudian melatih input fitur yang telah didefinsikan yang selanjutnya akan digunakan untuk mengklasifikasikan teks



Note that in this case, you either classify 1, for a positive sentiment, or 0, for a negative sentiment.

Vocabulary : Kumpulan kata Kata unik, dengan artian di dalam dataset yang ada, ambil only kata uniknya aja.

Sparse Representation : Ya itu dia boy jadi setiap Input / kalimat akan di representastikan menjadi vector, [1 / 0], nah karena ukuran kamus itu besar, maka dari itu pasti setiap inputan punya 0 yang banyak, sesuai dengan ukuran vocabularynya. Ini menyebabkan :

1. Training Jadi Lama
2. Prediksi juga jadi Lamir

Untuk address masalah Sparse Representation, dibentuk lah frequencies table

Frequencies Table : Nih kek gini Misal kita punya 2 kelas : dengan inputan SBG berikut

Positive tweets	Negative tweets
I am happy because I am learning NLP	I am sad, I am not learning NLP
I am happy	I am sad

Inputan ini akan dibentuk Frequencies Table, yang berbentuk seperti ini :

Vocabulary	PosFreq (1)	NegFreq (0)
I	3	3
am	3	3
happy	2	0
because	1	0
learning	1	1
NLP	1	1
sad	0	2
not	0	1

PosFreq → Frekuensi / banyaknya kata yang muncul di **vocabulary** dengan kelas Positif

NegFreq → Frekuensi / banyaknya kata yang muncul di **vocabulary** dengan kelas Negatif

Jadi ntar fitur yang digunakn untuk inputan datanya seperti berikut :

Kalimatnya : I am sad, I am not learning NLP → In ikan kalimat positif, nah kita kan pengen mepresentasikan kalimat tersebut kebentuk baru ,kek gambar dibawah

Vocabulary	PosFreq (1)
I	3
am	3
happy	2
because	1
learning	1
NLP	1
sad	0
not	0

I am sad, I am not learning NLP

$$X_m = [1, \sum_w freqs(w, 1), \sum_w freqs(w, 0)]$$

8

To encode the negative feature, you can do the same thing.

Vocabulary	NegFreq (0)
I	3
am	3
happy	0
because	0
learning	1
NLP	1
sad	2
not	1

I am sad, I am not learning NLP

$$X_m = [1, \sum_w freqs(w, 1), \sum_w freqs(w, 0)]$$

11

Nah Xm itu maksudnya , observasi ke -m

freqs: dictionary mapping from (word, class) to frequency

$$X_m = [1, \sum_w freqs(w, 1), \sum_w freqs(w, 0)]$$

Features of tweet m | Bias | Sum Pos. Frequencies | Sum Neg. Frequencies

Itu Rumues freqs(w,1) aps boy ?

Jadi maksud rumus yang ntu, untuk seluruh kata dalam inputan dihitung frekuensinya dengan kelas yang sesuai.

Jadi freqs(w,1) maksudnya jumlah total kata “w” di kelas 1 pada frequencies table. EZ

Jadi, observasi ke -m direpresentasikan dengan [1,8,11] . 1 bias, 8 Positive Feature, 11 negative Feature

Terdapat beberapa preprocessing yang dapat dilakukan dalam NLP, yaitu

- Tokenize
- stop words,
- punctuation,

- pengahpusan URL

- Stemming + porter stemmer

- Lowercasing

Stopword → Ya ini kek kata kata yang kurang relevant kek kata penghubung

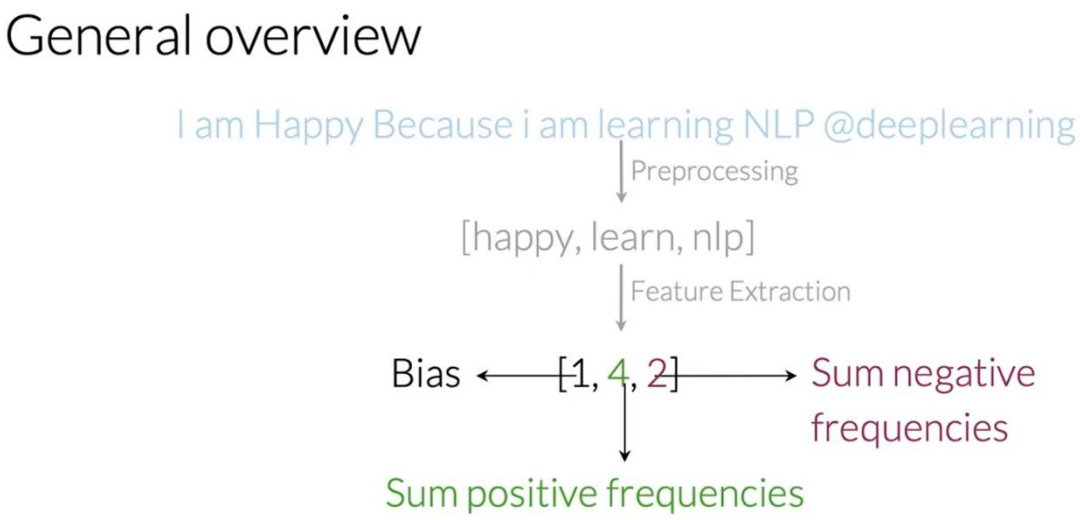
Punctuation → Tanda Baca

Stemming → ini ngubah sebuah kata kebentuk dasarnya, ini berperan bagus untuk menurunkan besarnya VOCABULARY yang dibentuk.

Porter Stemmer → missal ada kata 'danc' , diubah jadi dance :v

Penghapusan nya harus melihat dari pada TASK NLP yang sedang dikerjakan .

Jadi Di week 1 ini mostly yang dipelajari dan akan di implementasikan y aini :



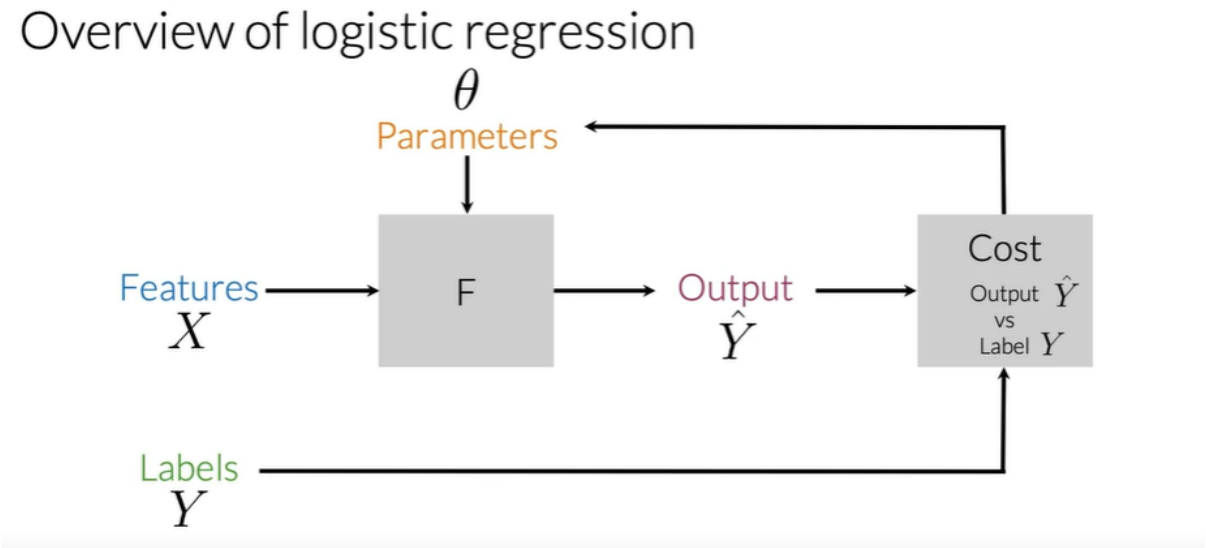
Liat kodingannya Kalau mau lebih jelas.

Next Logistic regression :

Well ini mah namanya juga basic jadi sebisa gw biar makin paham videonya

Logistic Regression ini model statistic ya, sebenearnya ada beberapa hal yang perlu dipertimbangkan, namun kita abaikan dlu

Pertama pipeline yang akan dibuat di course ini ya kek gini :



Jadi kita masukin input x/ Features ke sebuah fungsi, fungsi ntar ngasilin output, output itu bakalan di hitung cost functionnya ,hasil cost function tadi bakaln update theta/ parametersnya untuk menyesuaikan kepada fungsi , F. Hal ini dilakukan sampe ya cost nya seminimal mungkin. Cost function itu LOSS Secara keseluruhan ya.

Nah fungsi F ini adalah si SIGMOID. Maksud rumus ini ya buat **output**, Belah kiri itu si X dan theta itu input nya, kalkulasinya belah kanan, x(i) itu observasi ke -I, theta ya updated parameter nya.

$$h(x^{(i)}, \theta) = \frac{1}{1 + e^{-\theta^T x^{(i)}}}$$

Cost Function di Logistic Regression (Ini untuk ngitung si **theta**)

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log h\left(x^{(i)}, \theta\right) + \left(1 - y^{(i)}\right) \log\left(1 - h\left(x^{(i)}, \theta\right)\right) \right]$$

$$\left[y^{(i)} \log h(x^{(i)}, \theta) \right.$$

Gambar 1

Ini maksudne apa? Ntar aku kasih contoh

$y^{(i)}$	$h(x^{(i)},\theta)$	
0	any	0
1	0.99	~0
1	~0	-inf

Y(i) itu label, h(x(i),theta) itu Prediksi, kuy coba dipahami

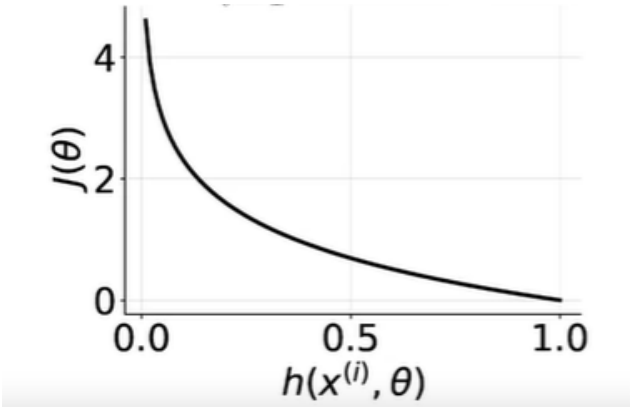
Jadi, kalau misalnya labelnya 0, dan prediksinya apapun itu maka nilai dari gambar 1, maka hasilnya bakalan 0 , iya dong 0 dikalikan apapun itu ya 0, teu kudu contoh-contoh lah.

Sedangkan kalau labelnya 1, dan prediksinya mendekati **true , hasilnya costnya bakalan mendekati 0.** Contoh : Karena ini materi CS,makenya log dengan basis 2 , so cara bacanya, 2 pangkat berapa yang hasilnya 0.87/0.99 ? Ez Rightt ?

1 * log(0.27) = **-1.8889686876113**

1 * log(0.99) = **-0.014499569695115**

Keliatan kalau misalnya dia jauh dari true label,i.e 1 ,costnya juga membesar 😊



Paham kan gambarnya ?

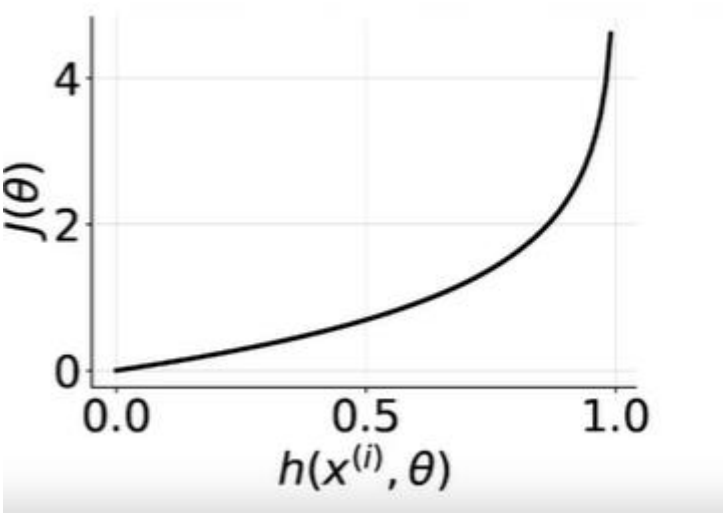
$$(1 - y^{(i)}) \log(1 - h(x^{(i)}, \theta))]$$

Yang ini gimans boy? Asal usul nya kurang tau, tapi intinya rumus nanti keliatan kalau digabung, sekarang kita coba contoh dulu ya , ini sebenarnya untuk ngitung loss untuk yang labelnya 0

$y^{(i)}$	$h(x^{(i)}, \theta)$	
1	any	0
0	0.01	~0
0	~1	-Inf

Liat yang (1- y(i)) , kalau misalnya 1, berarti 0 * log (…) hasilnya ya 0 bang. Nah sekarang kita coba baca yang labelnya 0, misal nih kalau predict, hasil log nya (0.22) atau (0.99).

Alasan menggunakan (1- hasil Prediksi) dikarenakan apabila, nilai log itu mendekati 0 , ya dia bakalan membesar, coba aja log(0.2), nah makanya ekivalennya dia dijadikan log(1 – predictednya) sehingga kita dapet nilai loss functionnya lebih kecil



1-y(i)*log(1 – predicted) = (1 – 0) * log(1 – 0.22)

= 1 * log (0.88)

= 1 * **-0.18442457113743**

$$\left[y^{(i)} \log h\left(x^{(i)}, \theta\right) + \left(1 - y^{(i)}\right) \log\left(1 - h\left(x^{(i)}, \theta\right)\right)\right]$$

Nah jadi contoh aja lah ya, untuk jelasin gabungan dari rumus iniaja ya :

Misal nih, kita ambil nilai

Y = 1 , prediksinya (0.98)

J = 1*log(0.98) + (1 – 1) * log(1 – 0.98)

J = 1 * - 0.029146345659517 + 0

J = - 0.029 (Kecil kan)

Y = 1 , prediksinya (0.08)

J = 1*log(0.08) + (1 – 1) * log(1 – 0.08)

J = 1 * -3.6438561897747 + 0

J = -3.64 (gede kan)

Y = 0 , prediksinya (0.08)

J = 0*log(0.08) + (1 – 0) * log(1 – 0.08)

J = 0 + 1 * log (0.92)

J = 1 * -0.12029423371771

J = -0.120

Y = 0 , prediksinya (0.98)

J = 0*log(0.98) + (1 – 0) * lo g(1 – 0.98)

J = 0 + 1 * log(0.02)

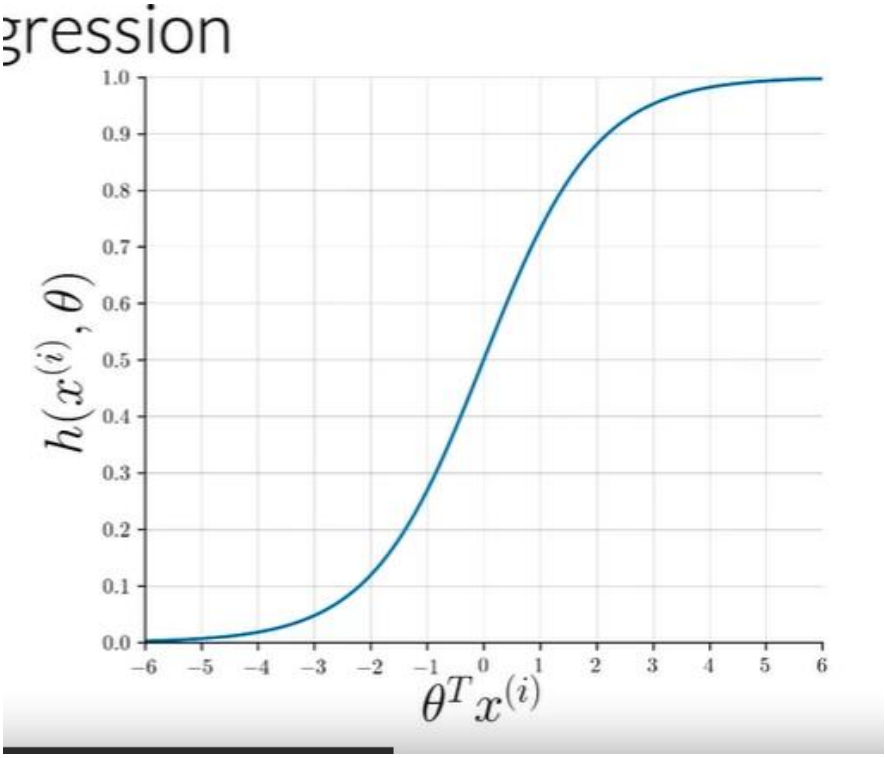
J = 1 * -5.6438561897747

J = -5.6438561897747

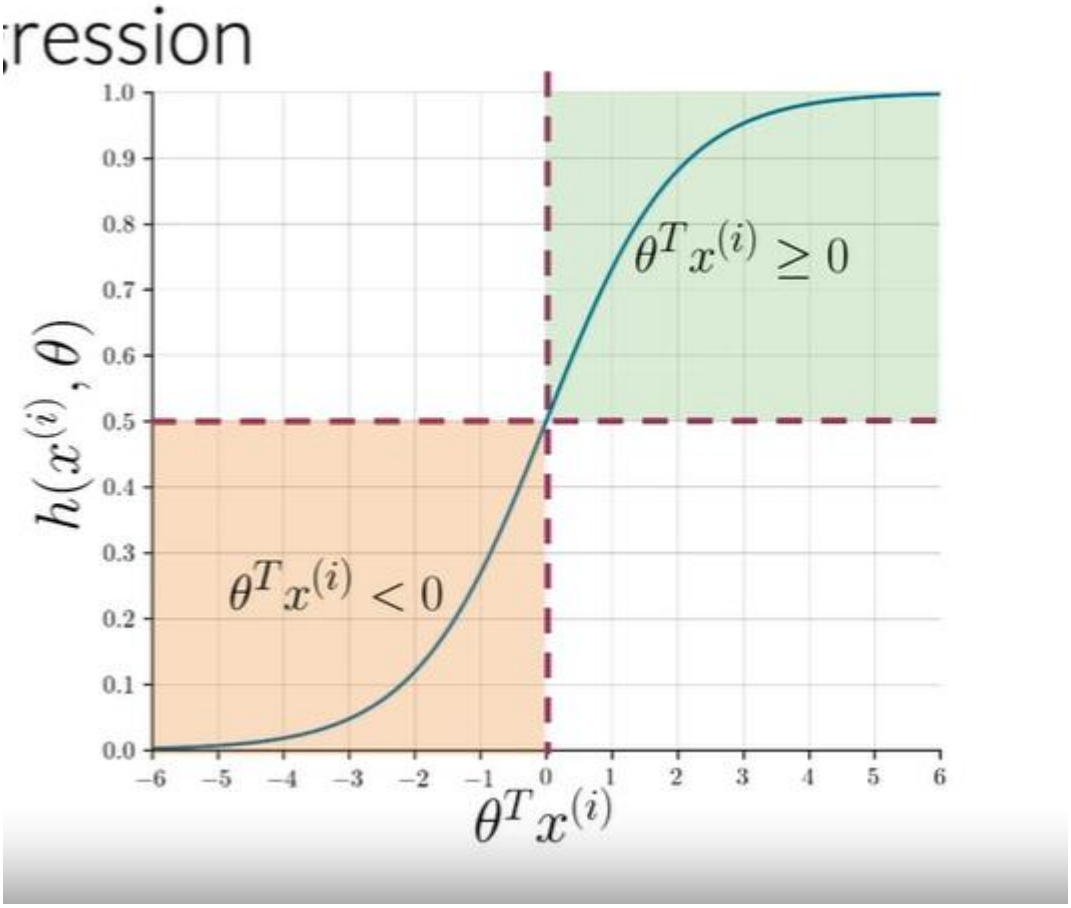
Sekarang paham kan ? Kalau ndak paham ntar tanya aja gw

$$-\frac{1}{m}\sum_{i=1}^{m}$$

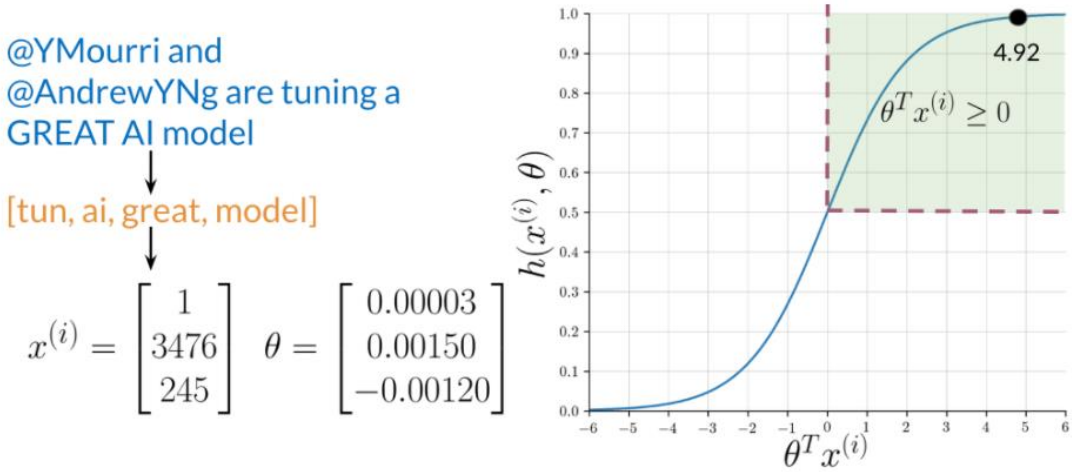
Untuk bagian depan itu Cuma Nambahin semua, ambil rata ratanya terus kaliin Minus, biar hasilnya positif 😊.



Ini Fungsi Sigmoid. Liat kan dari sana, untuk h(x(i),theta) itu hasil prediksinya, yang x axisnya mah Cuma ***dot product*** antara si parameters dengan input features , nah disini ada hal yang penting, coba diliat kalau misalnya dot product , yang x axis >= 0 pasti dia positif , kalau <= 0 dia negatif, inget ini kalau threshold y axis nya Cuma 0.5 kalau berbeda ya beda lah. NIh gambar biar jelas :



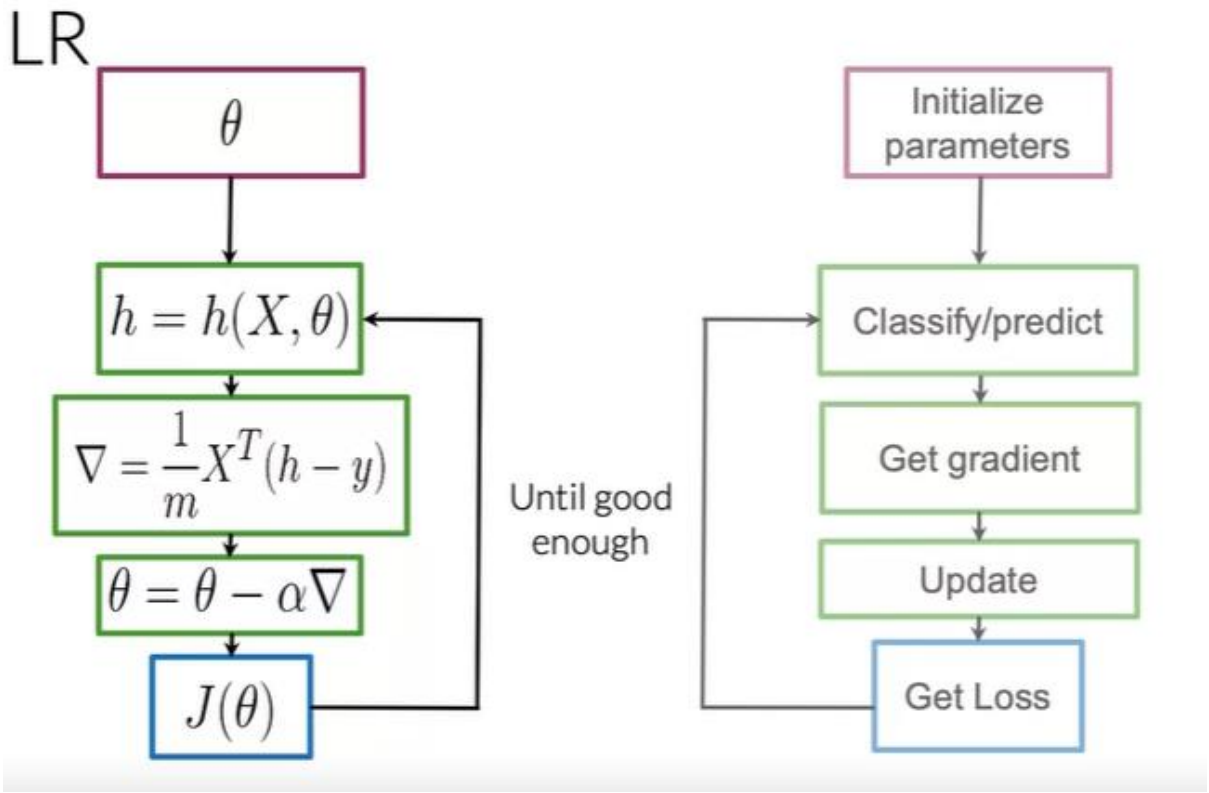
Contoh biar paham kek gimana model yang akan dibuat di week 1



Paham kan dot product ? Kalau ndak ya gitu lah ah Riet aing mah…

Trainingnya Gimans Boy ?

Gini boy



Nah nah gimans boy ? Gini awalnya inisiasi dlu tuh theta, habis itu di kalkulasi dah costfunctionya. Step selanjutnya itu isian dari pada kalkulasi costfunction. Nah yang segitiga kebawah, itu dia ngitung turunan, I dontknow from where, nah selanjutnya hasil turunan tadi itu digunakan untuk update si theta. Itu “a” kecil maksudnya learning rate ya. Hasilnya akan digunakan untuk mengkalkulasi J(theta), cost function yang baru.

Untuk Testingnya gimans ?

- X_{val} Y_{val} θ

$h(X_{val}, \theta)$

$pred = h(X_{val}, \theta) \geq 0.5$

$$\begin{bmatrix} 0.3 \\ 0.8 \\ 0.5 \\ \vdots \\ h_m \end{bmatrix} \geq 0.5 = \begin{bmatrix} 0.3 \geq 0.5 \\ 0.8 \geq 0.5 \\ 0.5 > 0.5 \\ \vdots \\ pred_m \geq 0.5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ pred_m \end{bmatrix}$$

Pada saat Testing cukup pahamin aja, luaran model kalian itu apa? Moslty untuk klasifikasi pasti diskrit kan ? Nah tapi untuk mendapatkan nilai diskrit, hasil prediksinya pasti berbentuk kontinu. Misal

Untuk data Pertama, Labelnya = 1, Prediksi Model 0.96

Data kedua, Labelnya = 0, prediksi Model 0.043

Kan di Fungsi sigmoid itu ada yang Namanya, **Threshold, 0.5 misalnya, nah itu dikonversi**

Untuk gradient descent, Ntaran, Terlalu deepweb mathnya, Aku update kalau udah