Semana 02

Función de Costo

Esta FUNCION nos permite medir o desarrollar el desempeño

El **Aprendizaje automático** tiene que tener una medida de desempeño.

Función de Perdida: que es parte esencial de la función de costo. Nos sirve para medir el error de nuestro modelo,

cuando estamos realizando el aprendizaje(A este proceso se le suele llamar el entrenamiento) utilizando los datos para ajustar o p**ara entrenar el modelo.**

Tenemos acceso a las etiquetas reales de las instancias que estamos leyendo, que estamos utilizando , entonces tenemos la posibilidad de medir la predicción que da nuestro modelo y compararlo con la etiqueta real que tenemos.

La diferencia que haya entre nuestra predicción y la etiqueta real va a ser un error si nuestra etiqueta es igual a la etiqueta real (la etiqueta pre decida que predice nuestro modelo igual la etiqueta)entonces el error va a ser tiene que ser mínimo o debería de ser cero.

Entonces la **función de pérdida** nos sirve para medir esa diferencia ese error y este error pues se le llama pérdida.

Existen múltiples funciones de pérdida que pueden utilizarse para para hacer esta comparación, sin embargo una de las más utilizadas es esta que tenemos acá

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

que está basada en el cálculo de logaritmos.

Esta función de pérdida pareciera un poquito larga pero es muy sencilla al considerar que las etiquetas reales en clasificación binaria sólo tienen dos valores que serían 0 y 1 entonces veamos que tenemos en esta ecuación tenemos **y** tenemos **y** gorrito.

y en esta ecuación significa: la etiqueta real , en esta ecuación y el resto que hemos hablado en la anotación de la etiqueta real

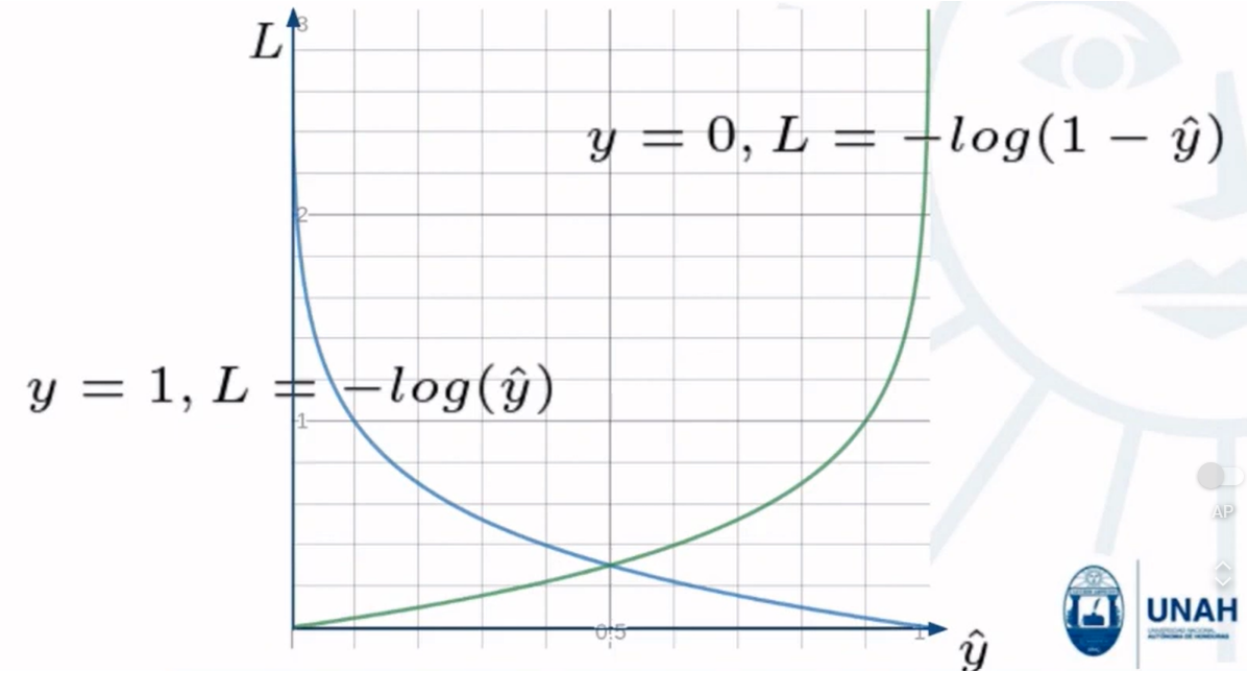
y gorrito es la predicción entonces la etiqueta real puede ser cero o puede ser 1. Si la etiqueta real es cero vemos que pasa ese término de aquí se

$$-(ylog(\hat{y}) \cdot$$

vuelve 0 acá tengo un número cero.

cuando y= 0 la pérdida es igual **a - logaritmo de 1- y gorrito ahora sí**

y=1 entonces qué



Grafica azul m es de la perdida con respecto a y gorrito.

y gor**rito es nuestra predicción y vean qué es lo que ocurre si la etiqueta real era uno** entonces si yo predigo un valor de 1 **significa que no hay error no hay pérdida ósea lo hicimos se hizo bien la** predicción por tanto la pérdida en este punto de cero ahora si yo me empiezo a alejar de uno y empiezo a hacer una predicción distinta de uno por ejemplo siempre digo 0.5 recordando que que se puede predecir 0.5 porque los valores que tenemos son valores de probabilidad **el sí predijo punto 0.5 el error es más** alto ahora si yo predigo 0

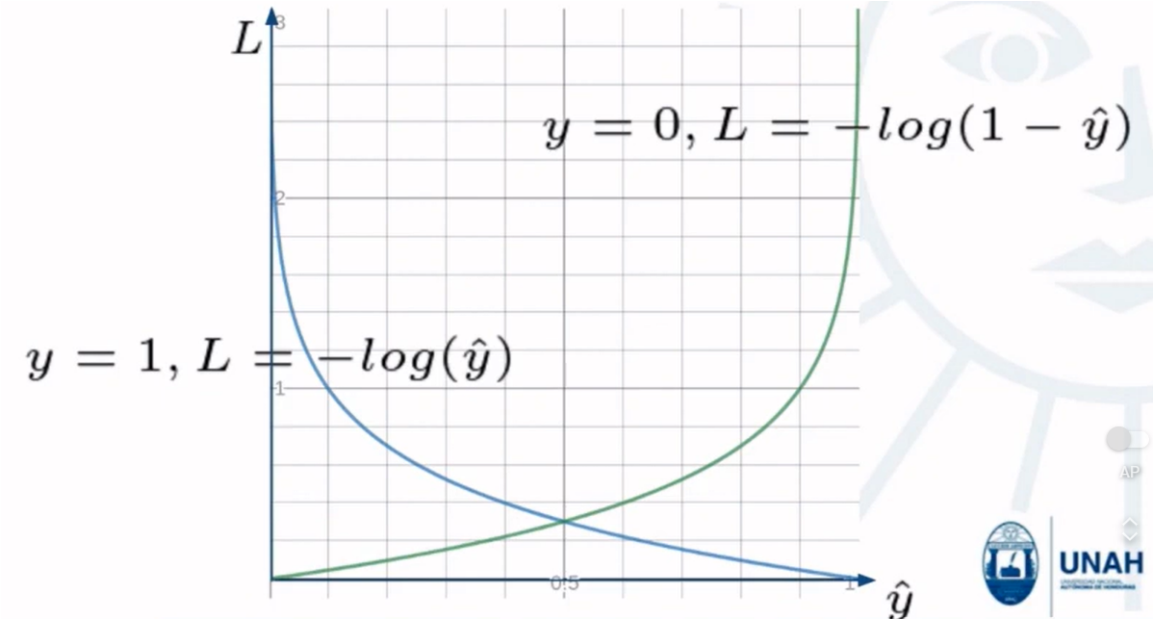
o sea que tengo un error estoy **completamente separado de lo que**realmente debió haber sido yo predije 0 **pero la etiqueta real es 1.**

Entonces al predecir cero **la pérdida va atendiendo a infinito** lo mismo con el otro lado si la etiqueta **real es 0 y la etiqueta predecída** yo predigo un 0 entonces el error de **la partida a 0.**

pero mientras yo más **me vaya separando de esa predicción o sea yo voy a acercándome a 1** es **decir alejándome de la etiqueta real mi** pérdida va a ir aumentando **hasta infinito**

vemos entonces que la función de pérdidas nos sirve para simplemente **medir la diferencia o el error que yo** tengo entre entre la etiqueta real y la predicción hecha por el modelo.

El hecho de que tenga esta forma también es muy útil a la hora de optimizar esta función vamos a ver qué va a ser necesario optimizarla.



función de costo

Permite tener una vision global del rendimiento del modelo en todo el conjunto de datos.

$$\hat{y}^{(i)} = \sigma(w^T x^{(i)} + b), 1 \leqslant i \leqslant m$$

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

el entrenamiento no se hace como una **sola instancia la función de pérdida** mide el error entre una etiqueta real y una etiqueta predecida , nuestro conjunto de entrenamiento va a tener más de una sola instancia o sea va a haber múltiples etiquetas y múltiples etiquetas predecidas y múltiples etiquetas reales por tanto entonces necesito una función que me dé una visión global del rendimiento de todo el modelo en todos los datos

entonces esto es la función de costo

la función de costos se representa por la letra J y es un promedio , la sumatoria de las pérdidas en todas las intancias de entrenamiento

y tenemos a esta y que representa y va desde 1 hasta **m que recuerdo recordando m indica todas las instancias de entrenamiento**

o sea que yo sumo todas las las pérdidas en todas las instancias de entrenamiento la dividido entre m y obtengo un promedio el promedio de las pérdidas es igual al costo j

si venga aquí tenemos la misma ecuación que teníamos antes acerca de cómo se calcula él el valor de la etiqueta pre decida utilizando la regresión logística este

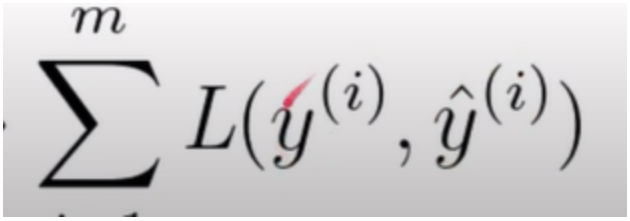
$$\hat{y}^{(i)} = \sigma(w^T x^{(i)} + b), 1 \leqslant i \leqslant m$$

valor de la etiqueta predecida es igual a la función sin molde de w transpuesta de x el producto punto de v w x x + b

sin embargo acá tenemos este super índice con él con él entre paréntesis con él y que ya lo habíamos mencionado anteriormente este súper índice indica la instancia va a ser

va a haber una etiqueta predecida para la instancia del 1, para la instancia otra para la instancia 2 y así hasta la instancia m vamos a tener m etiquetas predecidas y esas etiquetas predecidas que son estas que son

estas que se ven acá



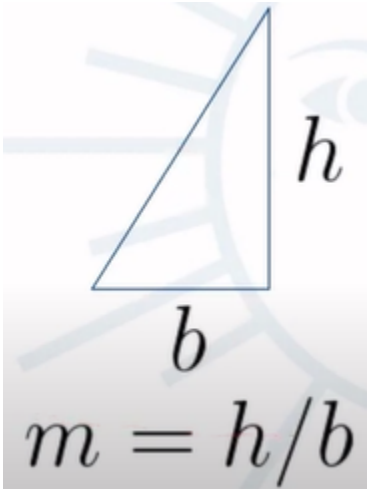
van a compararse con m etiquetas reales que yo voy a tener en el vector y mayúscula que es lo que tengo aquí, entonces esta pérdida va a haber m perdidas distintas L y de gorrito y y representa la pérdida en cada una de las instancias sumó estas pérdidas vean que estoy explicando esto solamente sumó estas pérdidas todas las dividida entre m y obtengo el costo

El costo y aparentemente o se muestra se suele mostrar como que depende del w y b porque depende de w en realidad porque las etiquetas reales ya están definidas o sea no es algo que va a cambiar lo que va a cambiar son las etiquetas predecidas y las etiquetas pre decididas no dependen de x porque aquí ya está definida los x son los atributos que están en el conjunto de entrenamiento sino que dependen de w y de b que son los verdaderos parámetros del modelo por eso decimos entonces que el costo depende de w y de b.

Derivadas: un repaso

La derivada de una función es la pendiente de la **recta tangente a la curva de dicha** función en un punto determinado.

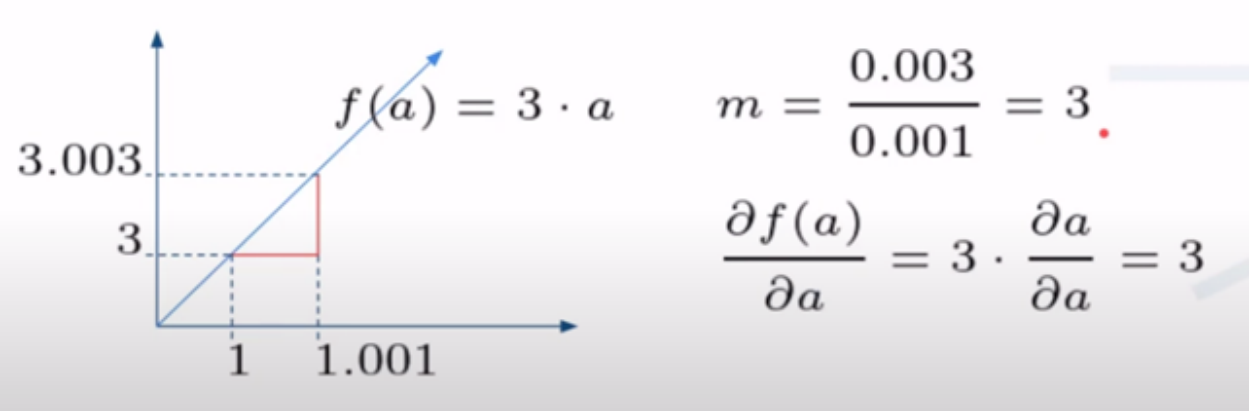
De acuerdo con esta definición entonces la derivada es un valor numérico que me da la inclinación de una recta tangente a una función en un punto específico. Por ejemplo si la hipotenusa este triángulo rectángulo fuese la recta tangente entonces la derivada sería la pendiente de esta inclinación de esta recta, la inclinación lo tendríamos dividiendo la altura entre la base en la derivada. entonces será igual h entre b m=h/b



está recto este sería tangente a una curva de una función que tiene un punto preciso y esa sería la derivada h entre b

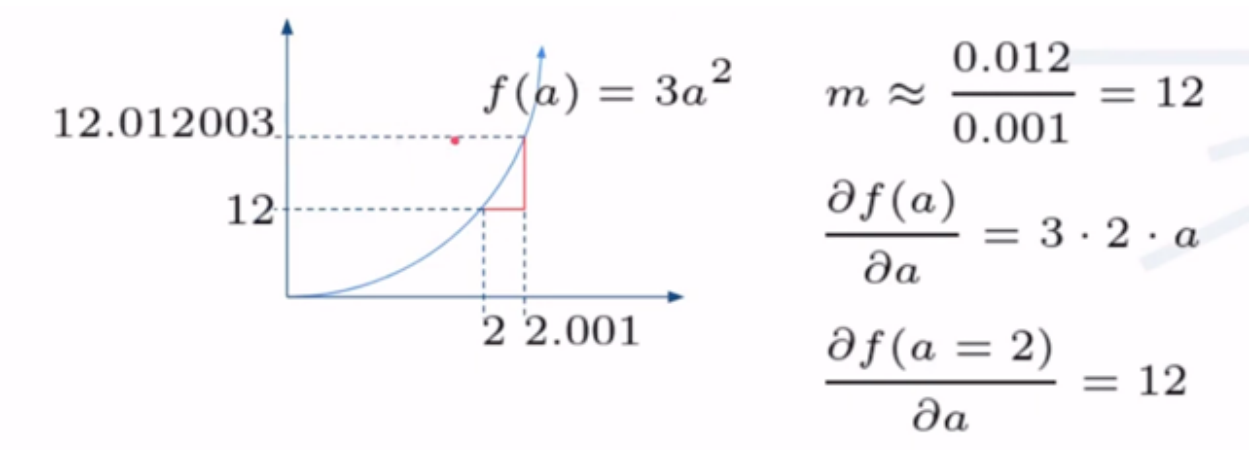
extendiendo este ejemplo a uno un poquito más aplicado, podemos ver los siguientes digamos que podemos encontrar la derivada de la función f(a)= 3a para hacerlo utilizando este concepto verdad el triángulo podríamos evaluar la derivada en un punto ahora cuando es igual a 1 observamos que esta gráfica no está hecha digamos a escala correcta pero el templo parece un poco grande para que se vea claro entonces empezamos en un punto uno quiere evaluar la derivada aquí para evaluar la derivada en ese punto yo podría entonces crear un pequeño triangulito para hacerlo e incremento un poquito verdad uno digamos el 0.01 y evaluó la altura de ese triángulo de acuerdo a la función en este punto ahora en 0 a 1.001 la función vale 3.003 entonces tengo este pequeño triángulo y para calcular la inclinación del de su poder no usar que sería la inclinación de la recta tangente a la curva en este punto observamos que en este caso esta

función lineal la la recta tangente a la curva es es la misma que la recta que la función porque una función lineal entonces la inclinación sería igual a la altura entre la base que sería igual a m 0.003 / 0.001 y eso me va a dar igual a 3 si yo calculo la derivada utilizando la fórmula de derivación analítica de manera analítica entonces yo puedo voy a llegar a la misma conclusión para la derivada de f(a) con respecto a ser igual a 3 por la derivada a con respecto a a que es 1, entonces sería igual a 3 obtengo entonces el mismo resultado de modo que otra vez la derivada simplemente en la inclinación de esta recta en este punto en este caso la inclinación vale 3



ahora miremos otro ejemplo que nos termina de ilustrar esta esta situación digamos que la función favor a la escuadra tica la función especial igual a 3 x al cuadrado en este caso yo pues quiero evaluar nuevamente la derivada la función en este caso cuando es igual a 2 puedo utilizar la misma idea incremento da un poquito de 0.001 y evaluo la función en este punto en este caso la función al evaluar la lo pueden comprobar con su calculadora 3 por 2.001 al cuadrado me da igual a esto a 12.0 12 0 0 3 en el caso que fuera de 2 verdad sería igual 3 por 4 12 tengo este pequeño triángulo y calculo la deriva calculó la inclinación ahora esta inclinación sería la inclinación de la hipotenusa de este triángulo pero ojo aquí vemos una cuestión en aproximadamente en la diapositiva

anterior era igual ahora yo tengo un aproximado porque porque no estoy **incluyendo este de 0.03 estoy quedándome** con la cifra más importante 0.002 entre 0.001 me da igual a 12



Debemos hacer un salto infinitesimal, infinitamente pequeño para obtener el valor correcto.

Algoritmo de descenso de gradiente

un algoritmo de optimización que **utilizamos para minimizar una función** específica.

en nuestro caso **la función que queremos minimizar es la** función de costos

la funcion de costos j(w,b) nos servía para determinar o evaluar el desempeño de nuestro algoritmo con respecto a las etiquetas que contamos en nuestro conjunto de datos

sin embargo decíamos también que la función de costo depende de **w y b donde son los parámetros del modelo. y gor**rito en realidad sigma de w traspuesta de x(i)+b entonces estos parámetros necesitamos encontrarlos pero necesitamos encontrarlos de forma tal de que el costo sea mínimo, o sea yo quiero **parámetros que minimicen el costo**

$$J(w,b) = \frac{1}{m} \cdot \sum_{i=1}^{m} L(y^{(i)}, \sigma(w^T x^{(i)} + b))$$

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

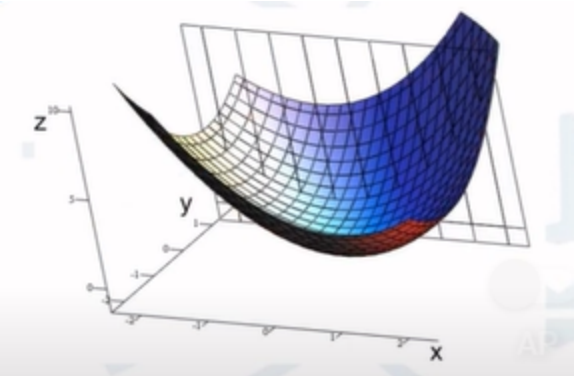
como encontrarlos **parámetros que minimicen el costo** ? la respuesta es mediante el algoritmo de **descenso de gradiente.**

en la grafica que aparece abajo podemos ver la imagen de una función de costo así como la **tenemos definida** con la pérdida igual a aquella función que utilizan

logaritmos vista anteriormente tiene una de las ventajas y es que a la hora de considerar múltiples variables en este caso dos variables w y b , su forma es convexa.

El algoritmo de descenso de gradiente es un algoritmo iterativo que funciona paso a paso empieza en un punto cualquiera de la gráfica, intenta paso a paso ir avanzando moviendo en este caso si yo fuera w y x fuera b moviendo cambiando los valores de w y b para que poco a poco vayamos buscando el valor donde el costo es minimo

el Algoritmo de descenso de gradiente no nos garantiza a encontrar el valor exacto o el punto exacto sino un punto aproximado **pero generalmente ese punto aproximado** es lo que es suficiente para nosotros



w como dijimos antes va a **tener el mismo tamaño que x y es donde x es el número y el tamaño de atributos**

por ejemplo en el caso de las imágenes de datos de 64 x 64 x 3 píxeles vamos a tener más de 12.000 dimensiones **esas 12 mil dimensiones son intratables** de forma analítica a resolver el problema así por tanto en el algoritmo de descenso de gradiente nos proporciona una solución computacionalmente a tratar tratable

también el Algoritmo de descenso de gradiente es util cuando ya tenemos funciones que no sean **convexas sino que de funciones que** tengan múltiples puntos mínimos

podríamos pensar como un valle con **montañas o mejor dicho un terreno que** tenga multiples partes que se undan verdad entonces ahi el algoritmo de descenso es mas util.

que buscar la derivada de igual 0 **porque la derivada de igual a 0 puede** ser igual a 0 en muchos puntos dependiendo de la complejidad de la funcion

EJEMPLO DE FUNCIONAMIENTO

repita {

$w_1 := w_1 - \alpha * \delta J/\delta w_1$

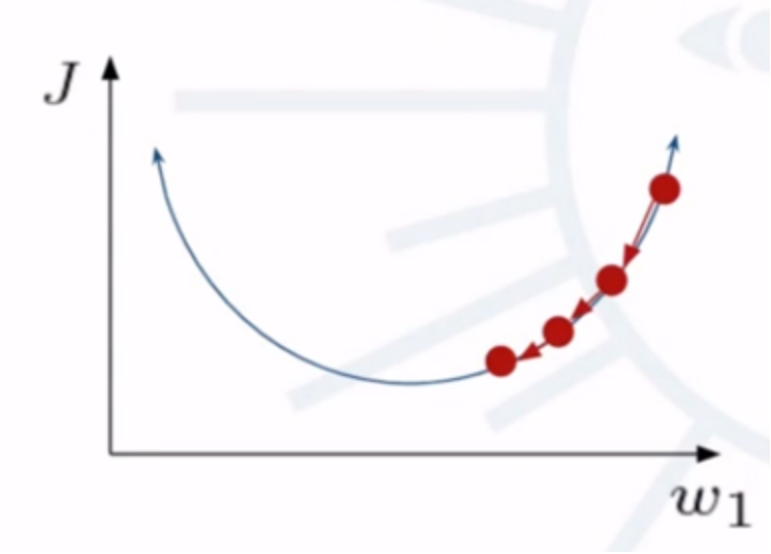
$w_2 := w_2 - \alpha * \delta J/\delta w_2$

...

$w_{nx} := w_{nx} - \alpha * \delta J/\delta w_{nx}$

$b := b - \alpha * \delta J/\delta b$

}



el funcionamiento del algoritmo de descenso de gradiente se puede resumir en una sola ecuación muy **sencilla que funciona optimizando las** variables de forma independiente

w tiene nx elementos en nx componente w 1 w 2 hasta w en nx y tenemos **b que es un valor real**, sin embargo en todos la ecuación es la misma. El algoritmo de descenso de gradiente itera, repita una cantidad de veces buscando el punto mínimo.

entonces qué es lo que hace? el valor de un parámetro por ejemplo w1, lo actualiza de la le resta una constante alfa multiplicada por la derivada de la función en este caso la función del costo con respecto a ese parámetro.

veamos el valor de la derivada, el valor de la derivada de una función con respecto a ese parámetro me da la pendiente de la recta tangente **a la función en un punto específico**

yo parto de un punto por ejemplo e**l primero y yo encuentro la pendiente** de la recta tangente a la función en ese punto entonces , considerando solamente una una variable, entonces la pendiente de la recta tangente ese punto va a tener un valor positivo. este valor positivo se va a multiplicar por alfa que es un valor positivo y por un menos y lo que va a ocasionar es que w 1 se le reste un valor entonces a ese valor que teníamos antes se le va a restar un valor que nos va a llevar al siguiente punto .

yo solo puedo hacer dos cosas con w1 sumarle valores o restarle aquí en el ejemplo se le restó porque la derivada me está saliendo positiva una inclinación positiva

sin embargo si hubiesemos partido desde este punto la derivada en este punto la recta sería negativa y por tanto esto sería menos por menos sería un más entonces si yo parto desde acá le estaría sumando a la el valor de w1

la derivada la inclinación de la derivada me sirve para saber en qué dirección moverme si le tengo que sumar o si le tengo que restar a la a la variable al parámetro para acercarme aquí donde quiero llegar el punto mínimo.

En la siguiente iteración w1 ya valdría (el siguiente punto) sin embargo aquí la derivada sigue siendo positiva entonces le voy a restar otro valor, esta derivada positiva por este menos me va a ocasionar una resta y poco a poco me voy acercando.

otra cosa que observamos es que aquí el salto fue más grande sin embargo aquí el salto fue más pequeño y aquí el salto es más pequeño

por qué porque aquí la derivada de la más grande era más más inclinada aquí la derivada se va haciendo menos inclinada aquí se va haciendo menos inclinada y poco a poco se va acercando a cero

cuando más pequeña derivada más pequeño es el salto y esto es una ventaja porque cuando yo estoy cerca del punto mínimo quiero hacer saltos pequeñitos porque si hago un salto muy grande me puedo pasar al otro lado y entonces perdería el punto mínimo que desea buscar aquí viene entonces el juego entra en juego también el parámetro alfa,

el parámetro alfa que es constante me sirve para aumentar o reducir el tamaño del salto que yo estoy haciendo a este parámetro alfa se le llama el ritmo de aprendizaje o learning reat en inglés el learning reat o ritmo de aprendizaje y entre más alto es lo que va a ocasionar es que se hagan saltos más largos y entre más pequeños se hagan saltos más pequeños es bueno tener un un learning reat pequeño porque así yo me aseguro de llegar al punto mínimo.

sin embargo si el learning rate es muy pequeño que pasa que si yo parto tengo la suerte de partir de un punto un poco lejano al punto minimo entonces me va a tomar muchísimas iteraciones puede llegar a tomar muchísimas iteraciones llegar hasta el punto mínimo tantas que se consuman demasiado tiempo de procesamiento

bien entonces que el algoritmo descenso gradiente hacen lo mismo con w 2 doble

3 hasta w nx y hace lo mismo con b.

la idea es esencialmente la misma ir moviendo cambiando el ajustando el parámetro paso a paso acercándose hacia el punto minimo y este punto mínimo el tamaño del salto está en mitad determinado por el learning rain y **está determinado por la magnitud de la derivada y la dirección del salto está determinada por la x en la dirección de la derivada si es positiva o si es negativa**

cuando termine el algoritmo de descenso de gradiente?

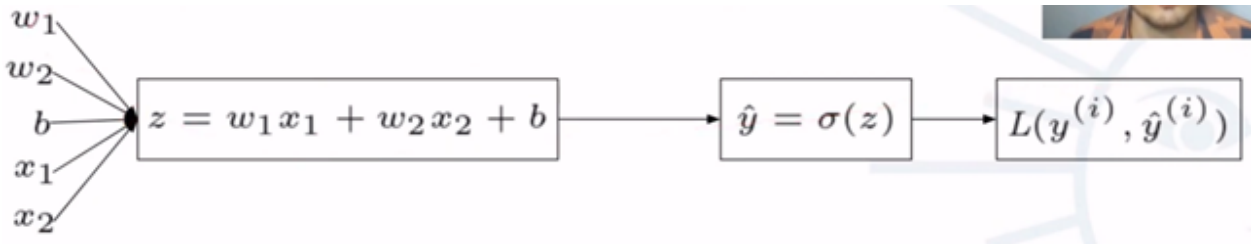
pues va a terminar cuando nosotros estemos satisfechos y ahí empecemos por ejemplo a considerar que ya estamos llegando a un punto de costo muy pequeño también va a terminar se suele definir un número de iteraciones fijos por ejemplo 1000 o 2000 y observar cuál es el comportamiento algoritmo en esas 2 mil iteracoiones

---

Grafo de Calculo de la Regresión Logistica: Propagación hacia delante y hacia atrás., retropropagacion

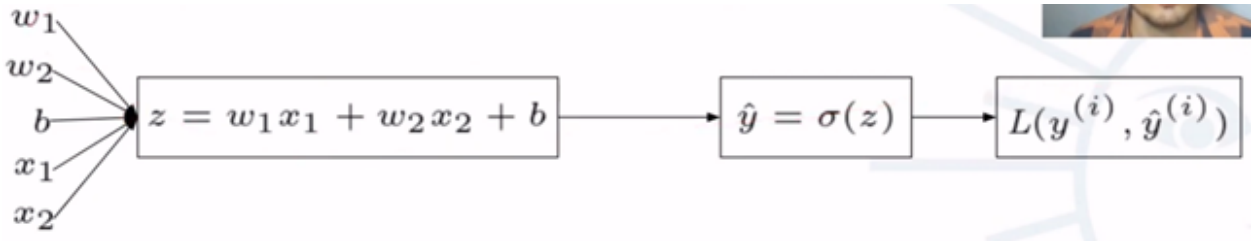
Grafo de cálculo
Herramienta gráfica que muestra los pasos necesarios para obtener un cálculo o un valor final buscado, las dependencias de dichos pasos. En el caso de la regresión logística sería

Ejemplo nx = 2 (sólo 2 atributos)



con n que igual 2 w va a tener dos dimensiones y x **va a tener 2 dimensión que son las que** están presentadas aca partimos entonces de w 1 w 2 de b, x1 y x2 sabemos que estos son parámetros del modelo todo esto sirve como entrada para el siguiente paso el cálculo de z el cálculo de z:

como ya sabemos la fórmula del z= w traspuesta por x +b. w transpuesta por x ya multiplicado ha hecho la expansión será igual a esto



después de haber calculado z pues lo que me queda es aplicarle la función sigmoide a z y obtener un valor de probabilidad y gorrito,

luego obteniendo este valor de probabilidad esta etiqueta presida yo puedo calcular la perdida

vemos este cálculo es para una sola instancia del conjunto entrenamiento sin embargo para obtener el valor final definitivo que sería el costo necesitaría hacer todo este proceso para las m instancias de modo que habiendo calculado la pérdida de las instancias podría yo calcular el costo.

Ahora la propagación hacia delante simplemente consiste en en el cálculo de estos valores hasta llegar al valor final.

sin embargo existe lo que se conoce como la **propagación hacia atrás** como vimos recientemente en el algoritmo de descenso de gradiente para poder actualizar los valores de los parámetros **dicho algoritmo necesita obtener los** valores de las

derivadas de la del costo con respecto a esos parámetros. para poder obtener esos valores de las derivadas nos sirve la propagación hacia atrás partiendo de la pérdida en el primer paso en la preparación hacia atrás sería :

1. calcular la derivada de la pérdida con respecto a y gorrito .(no con respecto a y porque no es un valor que nos interesa y es un valor que ya está el conjunto entrenamiento sin embargo **y** gorrito sí porque el que depende de nuestros parámetros. Por tanto la derivada de L con respecto al y gorrito conociendo la fórmula con logaritmos dada anteriormente la derivada sería ésta

$$\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$$

ahora lo que me interesa a mí es la derivada de L con respecto a a estos parámetros a los parámetros w y b para hacer eso entonces tengo que dar un paso más atrás lo primero que podría hacer en este siguiente paso sería

2. obtener la derivada de y gorrito con respecto a z que es la función que tengo aquí esta derivada y gorrito con respecto a z = a sigma de zeta por uno menos sigma de z.

$$\frac{\partial \hat{y}}{\partial z} = \hat{y}(1-\hat{y})$$

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} = \hat{y}(1-\hat{y})$$

y como sigma de z= y gorrito la puedo también escribir si y gorrito por 1 - y gorrito

ahora siempre estoy interesada en obtener la derivada de la pérdida de este la que yo puedo calcular en este momento pero me interesa la derivada de la pérdida en este caso ahora con una variable más atrás con respecto a z, entonces la derivada de la pérdida con respecto acepta para obtenerlo pues puedo aplicar la famosa regla de la cadena la derivada de la pérdida con respecto a z va a ser igual la derivada de la pérdida con respecto a la gorrito por la derivada de diego rito con respecto a Z

la derivada de la pérdida con respecto

al ego rito la calcula en el paso

anterior es ésta y de llegó recto con

respecto acepta a la acabo de calcular

es este entonces al multiplicar llegó

ritos por 1 - llegó rito por todo esto

que tenemos acá y simplificar nos queda

este valor llegó ritos por un número

sin embargo todavía no he llegado hasta

acá tengo que dar un paso más en el

3. siguiente paso entonces lo primero que voy a poder calcular es la derivada de z con respecto a estas variables x1 y x2 no me interesan porque están dadas en conjunto entrenamiento no interesan w 1 w 2 y b calculó estas derivadas de z con respecto a w 1 que con una función lineal bastante sencilla simplemente x1

receta con respecto a w 2 es x 2 y la

derivada de zeta con respecto a b es 1

con estas ya puedo aplicar la regla de

la cadena nuevamente la derivada de l

con respecto a w 1 va a ser simplemente

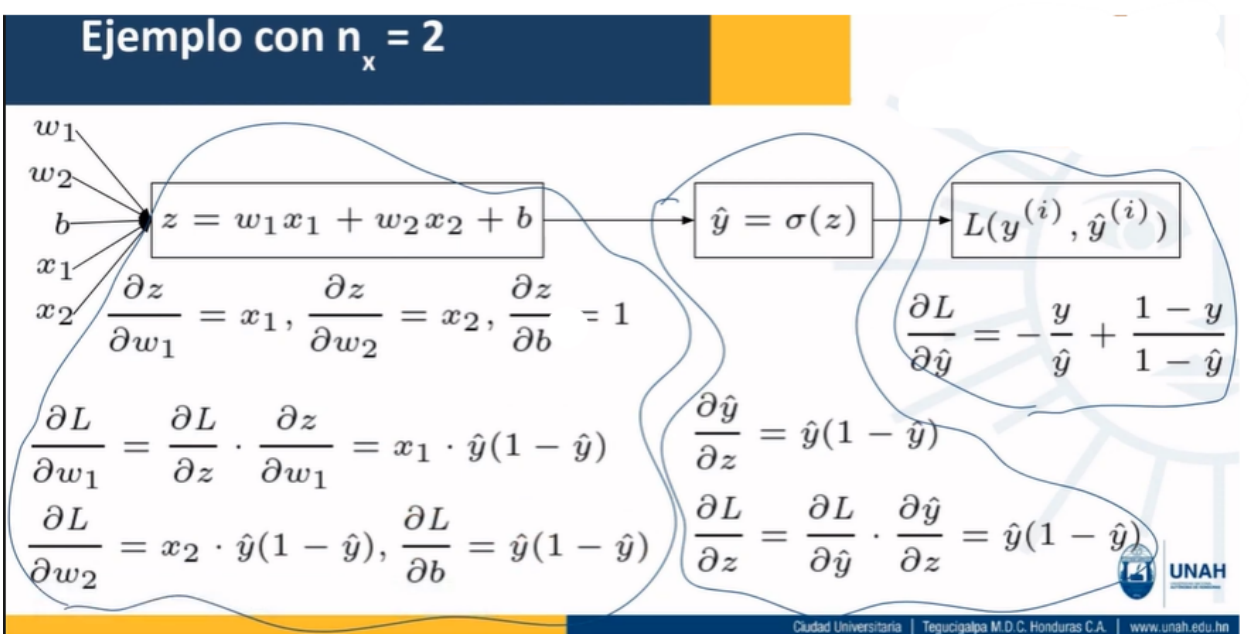
la derivada de l con respecto a z

que la calcula en el paso anterior por

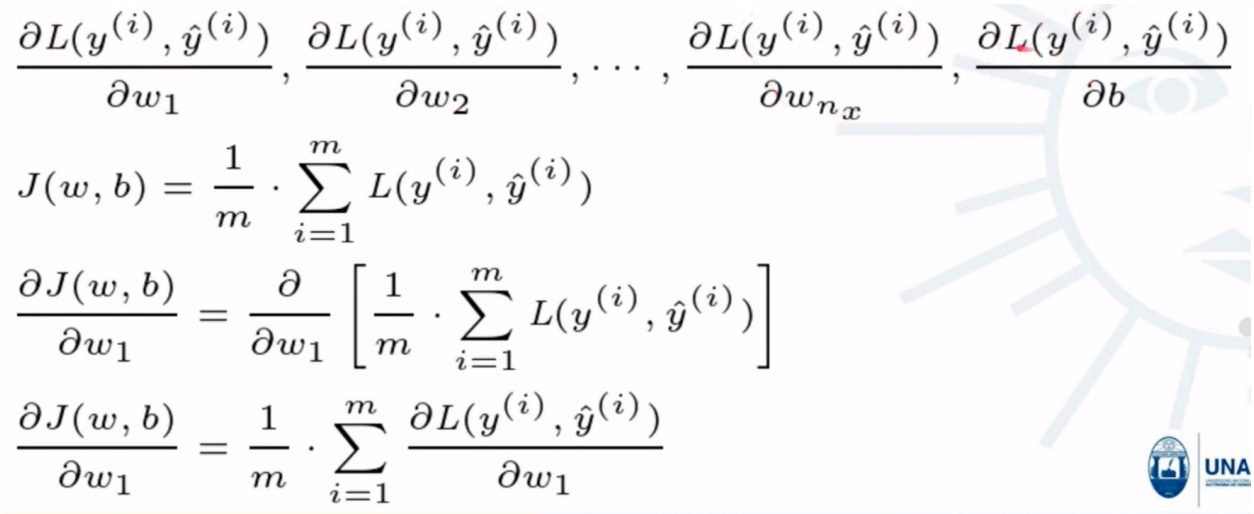
la derivada de z con respecto a w 1 que

es simplemente x 1 por tanto el

resultado es simplemente x 1 por diego
ritó menos 1 llevó ritmo de forma
similar va a ocurrir con la derivada de
l con respecto a w 2 va a ser igual a x2
x 1 menos llegó por x llegó ritó por 1 -
diego ruiz y finalmente la deriva
con respecto a ello va a ser igual 1 por
llegó rito por 1 - hierba
con esto entonces este valor
este valor y este valor son los valores
de las derivadas parciales que yo
necesito casi casi porque porque son
derivadas de l con respecto a los
valores a los parámetros sin embargo lo
que nos interesaba era la derivada lo
que nos interesa al final es la derivada
de aj con respecto a estos parámetros
sin embargo para eso necesitamos
considerar las m instancias esas m
instancias las vamos a considerar despues.



Regresión Logística con m instancias



Lo que necesitamos saber es el costo, por esa razon vamos a obtener la derivada del costo

Para conocer el costo simplemente debemos calcular el promedio de las perdidas.

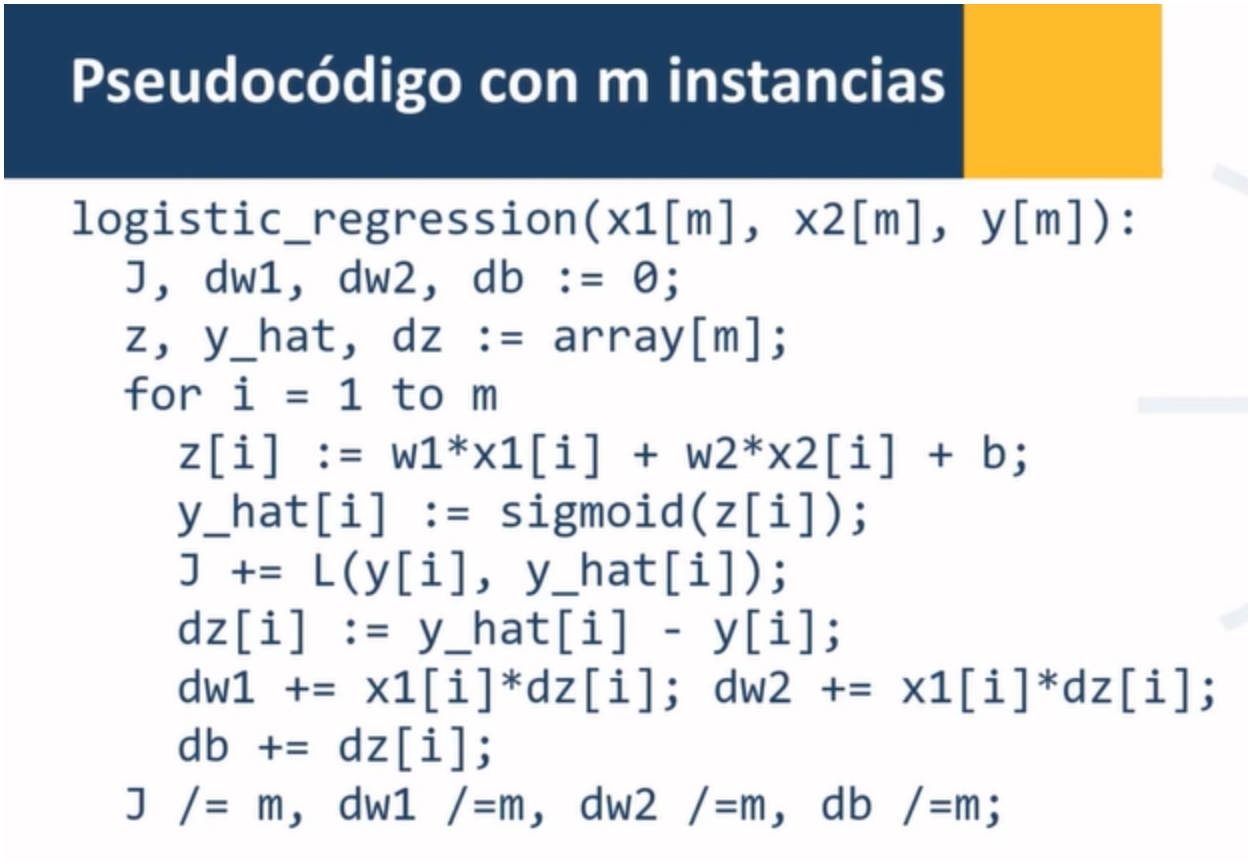
Veamos que pasa con las derivadas

DErivada de la perdida con respecto a w1

derivada del costo con respecto a los parámetros es igual a simplemente la la sumatoria o el promedio de la **derivada de la pérdida con respecto a ese parametro, en los diferentes instancias.**

Para calcular el costo calculo el promedio de las perdidas

Para calcular la derivada del costo con respecto a un paramentro calculo el promedio de las derivadas de las perdidas con respecto a este parametro



REgresion logitica, recibe 3 arreglos empiezan en uno y terminan en m para que coincidan con el esquema matematico que tenemos.

Guarda todos los atributos 1 de las instancias desde la 1 hasta m

En y estan las etiquetas.

2. variables iniciadas en 0

3. areglos para guardar las instancias, inicializados como arreglos de tamaño m, se aume que empiezan en 0
4. ciclo para considerar las m intancias, tenemos dos pasos la propagacion hacia delante (Se resumia en las primeras tres lineas)

   El calculo de z para la i esima instancia, w1, w2, y b seran aleatorios al principio y despues se ajustaran

   Luego se calcula y gorrito, funcion sigmoide de z.

   luego J ira acumulando la suma de las perdidas

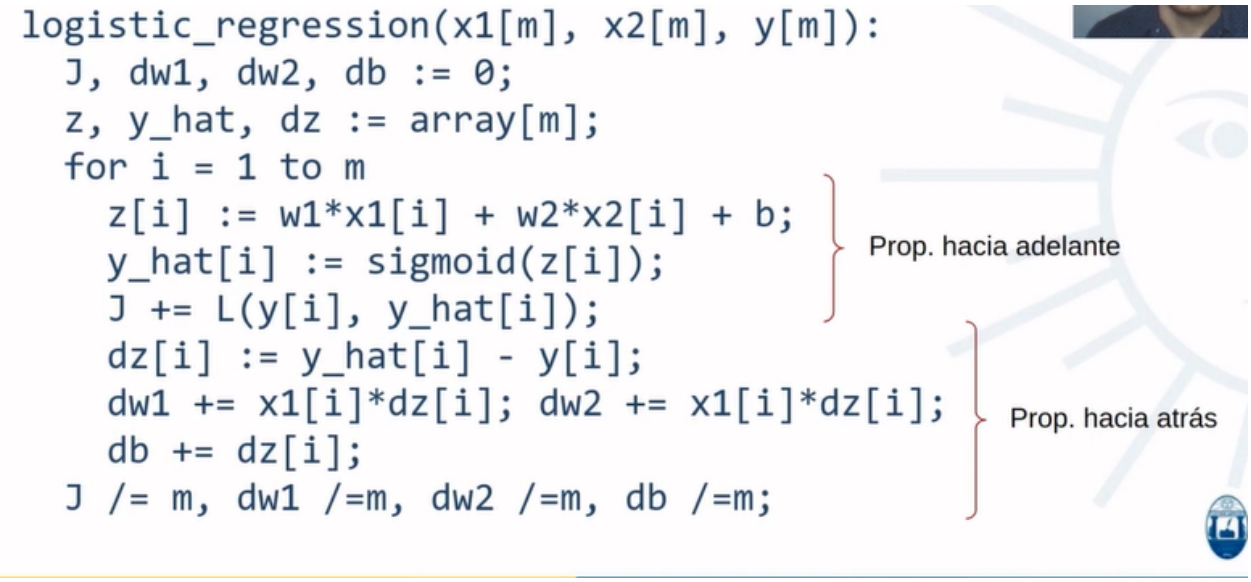
1. W1, w2, b . Inicialmente vana a ser aleatorios y se van a ir ajustando
2. j acumula la suma de las perdidas
3. Propiamente hacia atras no necesita calcular derivadas
4. dw1 y db quieren obtener el costo con respecto a los parametros

y hacia atras acumulo los calores de las sumas dentro del ciclo.

Fuera del ciclo:

j/m para obtener el costo, dw1/m para obtener la derivada del costo con respecto a w1, w2 y

Costo y derivadas del costo con respecto a los parametros considerando las m intancias de entrenamiento


logistic_regression(x1[m], x2[m], y[m]):
J, dw1, dw2, db := 0;
z, y_hat, dz := array[m];
for i = 1 to m
z[i] := w1*x1[i] + w2*x2[i] + b;
y_hat[i] := sigmoid(z[i]);
J += L(y[i], y_hat[i]);
dz[i] := y_hat[i] - y[i];
dw1 += x1[i]*dz[i]; dw2 += x1[i]*dz[i];
db += dz[i];
J /= m, dw1 /=m, dw2 /=m, db /=m;
Prop. hacia adelante
Prop. hacia atrás