Apuntes Semana 7 - 03/04/2025

Deylan Sandoval Sanchez - 2020234274
Escuela de Ingeniería en Computación
Instituto Tecnológico de Costa Rica

I. Redes Neuronales

A. Caso: Clasificación de MNIST

Se empieza con una base de datos que va a contener aproximadamente 60.000 muestras donde cada una son imágenes de 28x28 pixeles,cada pixel sera un feature por lo tanto se tiene 784 features, en cada imagen se encuentra un dígito escrito a mano, por lo que cada dígito va a ser diferente aunque sea el mismo numero, el objetivo va a ser tratar de hacer un clasificador de números escritos a mano

Regresión Logística:

Para hacerlo con regresión logística sabemos que esta recibe un vector de entrada con todos los features, para poder hacer que la imagen sea aceptada se va a aplastar toda la imagen para que convierta en una tira de caracteres, la cual se le va a dar a la regresión logística

¿De que tamaño va ser el input layer? Sera de 784, porque cada pixel va ser un feature de el input, en donde por cada uno va haber un peso asociado, es decir tambien hay 784 pesos(w)

Regresión multinomial:

En este caso sabemos que la clasificación binaria da solo resultados de 0 y 1, pero necesitamos una clasificación multiclase, porque debemos saber si es 1,2,3,4,5,6,7,8,9 o 0 Para resolver esto se va crear una clasificación binaria por cada uno de los dígitos, donde cada una se especializa si es su dígito correspondiente o no, se obtiene una codificación de ONE-HOT VECTOR

Cada pixel se debe ir pasando a todas las clasificaciones binarias para que puedan dar el resultado, llegando asi a pasar 784 pixeles a todas las clasificaciones binarias

Conversión de vector a matriz:

Se hace para que en lugar de calcular cada regresion lineal de forma vectorial, se cambian los vectores por matrices para hacer una sola operación y no N, donde N es el tamaño de la capa siguiente

Se convierte cada uno de los pesos en una matriz, donde se hace una fila por el tamaño de la siguiente capa y una columna para cada feature

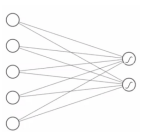

Fig. 1. Ejemplo 1.

En la Fig 1 se puede observar donde entrar 5 features a 2 salidas, ¿De que tamaño seria la matriz? Seria una matriz de pesos con 5 columnas y 2 filas.

¿Que sucede con el parámetro b? Este va a pasar a ser un vector, donde hay una dato por cada fila

II. Ejemplo 2 operaciones

x = [3,4,5,6]
W_1 = [3,2,4,5]
b_1 = 2
$W_1 \cdot$ x + b_1 = [3,4,5,6] $\cdot$ [3,2,4,5] +2 = 67 +2 = 69
sigmoid($W_1 \cdot$ x + b_1)
x = [3,4,5,6]
W_2 = [4,3,2,1]
b_2 = 6
$W_2 \cdot$ x + b_2 = [3,4,5,6] $\cdot$ [4,3,2,1] +3 = 40 +3 = 43
sigmoid($W_2 \cdot$ x + b_2)

III. Mismo ejemplo pero matrices

x = [3,4,5,6]
W = [[3,2,4,5],[4,3,2,1]]
b = [2,3]
xW^T +b
sigmoid(xW^T +b)
Resultado:

$$\begin{pmatrix}3 & 4 & 5 & 6\end{pmatrix}\begin{pmatrix}3 & 4\\ 2 & 3\\ 4 & 2\\ 5 & 1\end{pmatrix} + \begin{pmatrix}2 & 3\end{pmatrix} = \begin{pmatrix}69 & 43\end{pmatrix}$$

X, Representa una matriz de input, una fila por instancia. Permite computar múltiples samples a la vez

W, Matriz de pesos a excepción del bias, una fila para cada entrada
de la regresión y una columna por cantidad de regresión

b, Contiene un bias por cada regresión.

$$XW + b$$

Ahora en vez de hacer muchos ciclos por cada operación lo realizamos con opereciones de matrices que estan bien optimizadas

IV. Clasificación binaria(de nuevo)

Ahora para determinar si el evento ocurre o no, se va a crear una clasificación binaria como capa de salida.Se cuenta con las siguientes capas:

* Capa entrada
* Capa intermedia
* Capa de salida

La ultima capa es calculada basada en la interacción de la capa anterior La capa anterior ofrece no-linealidad

No linealidad nos permite atacar problemas complejos

Esta compuesto por capas(Hiper parametro)

Lo importante es que cada cada sea diferenciable

Si puedo derivar, puedo optimizar

Y en cada capa hay neruronas

V. Perceptron

Es una regresión logistica con otro loss function(Hinge Loss), propuesto por Frank Roseblant, se pensó que iba a resolver muchos problemas, pero llego el invierno de da AI, porque Minsky presento que tenia muchos problemas,como los requerimientos computacionales para la época, y que con los problema lineales no se puede resolver un XOR, lo cual es muy básico para la computación. Este problema años después se resolvió usando redes neuronales, porque si puede modelar problemas no-lineales

VI. Función de activación

En regresión logística lo llamábamos función no-lineal(sigmoid), esta función depende de la señal activa o no de la neurona, decide si deja pasar la información, la transforma o la bloque, existen varias funciones de activación que se debe saber cual elegir.

VII. Perceptron multicapa(MLP)

En este caso cada capa se va analizar con el dato de la capa anterior

h(0) = sigmoid(XW^0 +b^0)

h(1) = sigmoid(h(0)W^1 +b^1)

h(2) = sigmoid(h(1)W^2 +b^2)

h(n) = sigmoid(h(n-1)W^n +b^n)

No siempre se usa sigmoid, se usa en la salida cuando es un algoritmo de clasificación

A. Salida independiente

Si se dejan n salidas sigmoid, cada salida va a tener una salida independiente, por lo que se hace una distribución para generar un One-Hot vector, con 1 en el que tenga la probabilidad mas alta

Capa de salida:

$$h(n) = g(h(n-1)W^n + b^n) \tag{1}$$

Donde g(x) no es necesariamente una funcion sigmoid, es una funcion no-lineal

VIII. Función de costo

Con la funcion de perdida(L) se calculan los parametros W y b de cada una de las neuronas ¿Como se hacer?

$$\partial L/\partial W_i^j) \tag{2}$$

j= capa

i = neurona

IX. Maldición de la dimensionalidad

Se trata de que a medida que se aumentan las dimensiones el caso se vuelve mas complejo.

Ejemplo: hay que ir casa por casa para buscar una solución, con cada dimensión la cantidad de casa crece mucho, primero 1 dimensión solo es una fila de casas, 2 dimensiones son como un edificio y 3 dimensiones son como cuadras de que tiene edificios y en cada edificio hay que a todos los apartamento en ellos

X. Comportamiento Jerárquico

Se usa porque el conocimiento se hace igual que los humanos que aprenden cosas simples para irlo transformando en algo mas complejo.

Generando ganancias exponenciales en las funciones:

* Polinomios
* Composición de funciones que permiten rehusar funciones simples para otras de orden superior
* Representación compacta: poco pesos se pueden modelar funciones complejas
* Eg: Red neuronal que aproxime otro(Sigmoid)