

Documento de shMessages

David Santiago y Jean Paul Ariza Calderón

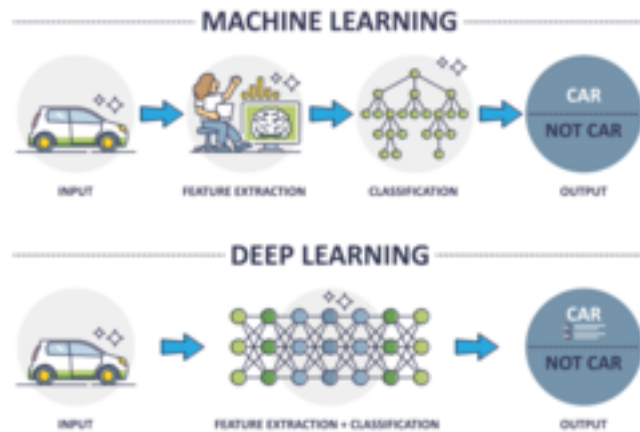
September 2023

Contents

En esta sección, veremos una pequeña introducción a lo que serían las Redes Neuronales, Deep Learning y la Inteligencia Artificial.

1 Introducción al Deep Learning

El deep learning es un subdominio del Machine Learning que resuelve el problema de Representation Learning introduciendo representaciones que se expresan en términos de otras representaciones más simples. Se fundamenta en la construcción en conceptos complejos a partir de conceptos mas simples.



En este caso le ponemos los datos en raw o bruto, para que el algoritmo pueda escoger aquellos que son importantes y resolver el problema con la Red Neuronal. Cuando tenemos la extracción de datos para poder saber las características de un coche mediante un Algoritmo de Deep Learning, ese Algoritmo se encarga de realizar el proceso, y nos devuelve la predicción por ejemplo si es un Coche o No, o en casos mas complejos, si es un coche deportivo o un SUV.

Una cosa importante es que necesitamos una cantidad robusta de datos para poder, entrenar y pulir nuestros modelos de Redes Nueronales, la mayoría de Técnicas Tradicionales de Machine Learning, son buenas en algún u otro aspecto a la hora de hacer la resolución del problema.Si tenemos un conjunto de datos pequeño, una técnica de Machine Learning Tradicional funcionaria mejor, entre mas cantidad de conjunto de datos, mejor nos rendirá el Machine Learning.Pero en otros casos, por ejemplo cuando debemos de realizar un algoritmo mas complejo y mas preciso podemos disponer de Redes Neuronales y Deep Learning.

1.1 Aplicaciones y Clasificaciones del Deep Learning

Ha conseguido resultados excelentes en campos donde la extracción de características fuerán difíciles de hacer.

1. Anuncios Digitales y de Marketing
2. Clasificación de Imágenes
3. Reconocimiento de Audio
4. Traducción automática Avanzada
5. Conversión de Texto a Voz
6. Conducción Autonoma
7. Simulación de Fotos y Videos

1.2 Aprendizaje supervisado y no supervisado

1.3 Aprendizaje Supervizado

Los Algoritmos que se basan en Aprendizaje Supervizado son los mas utilizados, es la tarea de Aprendizaje Automático que mapea una entrada a una salida en pares de entrada-salida de ejemplo. La función resultante es utilizada posteriormente para predecir valores a partir de ejemplos de datos no etiquetados. La función hipótesis es la función la cual se encuentra en la caja negra para poder realizar las predicciones.

1.3.1 Tipos de Aprendizaje Supervizado

1. Regresión:Este tipo de valores intentan predecir valores continuos, por ejemplo el coste de un incidente de seguridad, el posible output, es un rango de valores muy alto.Por lo general si tenemos una representacion gráficaa, los puntos de incidente deberian estar dispersados en un plano cartesiano o por ejemplo- haciendo cierta tendencia.Con un eje x y un eje y.Un Algoritmo de Regresión intentará realizar una funcion recta la cual corta en el 0 en ciertos casos y se va a poder adapatar o ajustar a

los puntos tratando de pasar por la mayoría de puntos de ellos. Con esto podemos predecir futuros incidentes en base a esa variable de entrada.

2. Clasificación: Por otro lado tenemos la Clasificación, en este caso intenta predecir valores discretos, en este ejemplo si tenemos un problema el cual queremos clasificar un animal en un conjunto por ejemplo en si es animal o no es animal o sea 1 o 2. Por ejemplo si tenemos un correo de spam o no spam, entonces podemos extraer el numero de signos de exclamación por ejemplo, suponiendo que los correos no legitimos hipoteticamente tiene mas signos de exclamación y así podemos predecir los siguientes datos en base a una gráfica. Van a construir una funcion hipótesis la cual va a parecer a una linea, la cual divide la area en 2 en el ejemplo dado, pone si un correo es spam o no spam en base a los resultado graficados.

1.4 Aprendizaje No Supervizado

El Aprendizaje No Supervizado es la tarea del Aprendizaje Automático de Inferir una Función que describa un conjunto de datos in etiquetar, o sea no haber clasificado los datos de entrenamiento si no que la máquina misma lo haga. Este tipo de Algoritmos, va a agrupar los conjuntos de datos en grupos, mas no una linea que los divida.

1.5 Introducción a las Redes Neuronales Artificiales

Una Red Neuronal Artificial es un tipo de Algoritmo de Machine Learning que esta inspirado en las redes neuronales biológicas. Estas Redes Neuronales Artificiales es el componente principal de Deep Learning, y la mayoría de tiempo todas las neuronas tienen diferentes capas y casi nunca van a ser iguales. Estas Neuronas se introdujeron por primera vez en 1943 por el neuropsicológico Warren McCulloch y el matemático Walter Pitts. Cuando surge el paper de McCulloch y Pitts empiezan a darle vueltas a este tema y se empiezan a plantear algoritmos inteligentes. En 1960 no tiene mucho impulso ya que se llegaban a conclusiones las cuales decian que las redes neuronales no iban a llegar a ningun lado por culpa de el atascamiento en el tema. En 1990 tampoco se hablan mucho de Deep Learning, ya que el vector Machine era mas preciso, y un poco mas simple matematicamente hablando. En 2012 vuelve a tener impulso gracias a Geoffrey Hinton ganando el reto de ImageNet con una Red Neuronal Convolutional.

Las RNA resurgen gracias a la gran cantidad de datos disponibles, ya que podriamos entrar a la era dorada de esta disciplina, esta disciplina ha estado en empuje por factores que no se encontraban antes, se podría decir que todos los factores se han alineado, tambien como el poder computacional que no teniamos hace 60 o 50 años, no disponiamos ordenadores buenos que podrian entrenar estas neuronas o inteligencias artificiales. También se podría decir que los Algoritmos de las Inteligencias Artificiales han mejorado muchisimo teniendo en

cuenta los Algoritmos de los años 80, han tenido cambios muy pequeños pero el impacto ha mejorado muchísimo.

La primera Neurona de McCulloch y Pitts corresponde con la primera neurona artificial de la historia, se caracteriza porque es una neurona que recibe valores binarios, o sea su input era de (1,0) y su output o salida era de (1,0), y cada neurona filtra su información binariamente, o sea con un uno o cero. La neurona de McCulloch y Pitts, estaba denominada por dos funciones, la cual tenía una fórmula z que realizaba una agregación, y la función a que es una función de activación, la cual toma una decisión basándose en la z .

Un ejemplo de una neurona la cual se active de forma similar a la Neurona de McCulloch y Pitts, supongamos que tengo un problema el cual es el siguiente. Problema: Puedo salir de fiesta el Sábado? En base al problema tengo varios datos, los cuales podrían ser:

1. Tengo Tareas? Si o No
2. Mis amigos quieren salir? Si o No
3. La discoteca está abierta el Sábado? Si o No
4. Está haciendo buen clima? Si o No

Y estos datos los podríamos representar con un 1 o un 0 y en base a esos resultados la neurona diseñada ya previamente me dirá si puedo salir o no a fiesta el sábado. La sumatoria o agregación de estos valores se podría representar de la siguiente fórmula.

$$\sum_{i=1}^n X_i = X_1 + X_2 + X_3 + X_4 \quad (1)$$

Donde x_1 a x_4 es cada una de los valores arrojados por nosotros a cada una de las preguntas planteadas, entonces por ejemplo X_1 , sería la pregunta Tengo Tareas?, hipotéticamente dire que no entonces tenemos un 0, y así sucesivamente con cada pregunta. Hasta llegar al final y sumar los resultados de cada valor, para realizar una sumatoria y tomar una decisión. Ahora entramos a un problema, el cual se genera principalmente por la pregunta X_3 , la cual dice La discoteca está abierta el Sábado?, si este input se convierte en 0 es lógico que los demás inputs da igual que valores tengan pero todo el problema será 0. Para este problema McCulloch y Pitts, definen los inputs como inhibidores y excitadores. Neuronalmente una neurona la cual sea excitadora, produce una sinapsis más fuerte entre neuronas, libera neurotransmisores, mientras que las neuronas inhibidoras, liberan neurotransmisores inhibidores los cuales hacen que la neurona se inhiba. En este caso X_1, X_2 y X_4 , son neuronas excitadoras. En el caso de las inhibidoras, generalmente cuando un inhibidor se activa la neurona general no se activa, automáticamente la neurona va a arrojar 0, independientemente del resto de neuronas.

Obviamente esta neurona tiene muchas limitaciones, ya que es solo un ejemplo, el principal es el de recibir valores binarios, el día de hoy necesitamos

resolver problemas los cuales la mayoría no arroja valores binarios si no rangos de numeros o numéricos. Tambien con este algoritmo de McCulloch Y Pitts, no puedo poner neuronas mas pesadas que otras, osease en pocas palabras poner unos eventos mas importantes que otros. Por ejemplo en el problema dado Hacer Tareas es mas importante que si mis amigos quieren salir.

2 Perceptron

El Perceptron es propuesto npor Frank Rosenblatt en 1958, esta propuesta de Frank Rosenblatt es mejorada y refinada por Marvin Minsky y Seymour Papert, básicamente el Perceptrón es una Neurona de McCulloch y Pitss, añadiendo el concepto del "peso", planteando un mecanismo para ajustar tal peso. A diferencia de la neurona de McCullochh y Pitts, esta neurona permite recibir valores reales, en vez de solo poder recibir entradas binarias. Se basa en un tipo de neurona artificial conocida como Threshold Logic Unit.

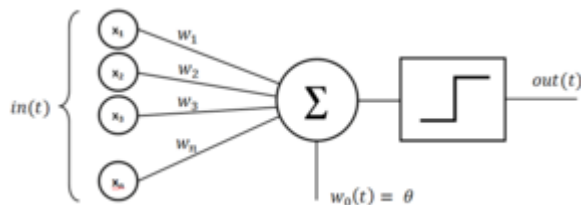
El perceptron tiene un $X_{sub 0}$,y $w_{sub 0}$ que basicamente es un BIAS ya puesto, para poder saber el threshold inicial el cual se encuentra en la sumatoria. Básicamente TLU, es una neurona la cual se activa despues de realizar la sumatoria de cada una de los valores, y se activa si cumple con el requisito del threshold. Ya que en el TLU o en el Perceptrón se computa tambien el peso, su formula de la sumatoria tambien cambia quedando de la siguiente forma.

$$\sum_{i=1}^n Z = X_1 * W_1 + X_2 * W_2 + X_3 * W_3 + X_4 * W_4 + b \quad (2)$$

Donde X es cada una de las variables con las que se evaluán el problema, y la w el peso de cada de estas variables, entre mas grande sea w mas importancia se la va a dar al valor de esa evaluación.

2.1 Arquitectura del Perceptrón

El perceptron se corresponde a unica capa de TLUs o Neuronas y permite la clasificación de instancias en diferentes clases binarias simultaneamente.



Donde cada $x_{sub 1}$ se refiere a una TLU en la neurona, y se realiza la sumatoria de cada capa. En una TLU, o perceptrón tendríamos 3 parametros del modelo en la TLU, donde se tiene pesos(w), y BIAS(b), y pueden ser n parametros. También cabe recalcar que la primera parte de un perceptron se llama Input Layer, y las capas de TLu, se llaman Output Layer, donde son

las ultimas de la Red. También podemos separar los valores de cada una de las TLUs, e input layers con digitos sub de la letra seguido de el dub Output Layer, en pocas palabras, el input 1 primero enlazado con el TLU ultimo sería X sub 1 sub n, donde n es la cantidad de TLUs en la neurona, en el primer TLU seria X sub 1 sub 1. Ahora vamos a intentar gráficar las entradas y formulas de un red de 3 TLUs. La primera TLU o entrada de valores al pereceptrón se daría de la siguiente forma.

$$h_w(x)_1 = a_1(x_1w_11 + x_2w_21 + b_1) \quad (3)$$

$$h_w(x)_2 = a_2(x_1w_12 + x_2w_22 + b_2) \quad (4)$$

$$h_w(x)_3 = a_3(x_1w_13 + x_2w_23 + b_13) \quad (5)$$

Donde x sub 1 w sub 12, donde el primer numero enlazado a w es la posición del input layer, y el segundo numero de w la posición del output layer.

Si yo quiero empezar a entrenar una Nuerona tipo Perceprón necesito empezar a decidir que tipos de evaluaciones se le va a hacer al problema, y en base a ello el numero de TLUs que se van a utilizar para la Red de TLUs, y deducir como le pasaré la información a la Neurona.Será un aprendizaje supervisado por lo cuál sabré cuales son las respuestas de cada evaluación que le realizo a la Nuerona. Capaz todas las TLUs, me da una evaluación diferente a la anterior, por lo que la salida o el Output layer será en base a un sistema que decida yo, por ejemplo la cantidad de veces que se repite un Numero.

2.2 Funciones de Activación

El perceptron permite clasificar mediante varias neuronas un problema,sea con una función o sea con varias funciones, algunas de las funciones mas comunes son:

1. Heaviside Function

$$heaviside(z) = \begin{cases} 0 & \text{si } z < 0 \\ 1 & \text{si } z \geq 0 \end{cases} \quad (6)$$

2. Sign Function

$$sgn(z) = \begin{cases} -1 & \text{si } z < 0 \\ 0 & \text{si } z = 0 \\ 1 & \text{si } z > 0 \end{cases} \quad (7)$$

Aquí una gráfica de la función Heaviside: Básicamente en la TLU, tiene un función de agregación z, y una función de activación llamada a.Una de la características principales de una TLU, respecto a una neurona Binaria, esta neurona puede llegar a recibir un valor que no sea binario, por ejemplo 14, 13,-12-890, etc.Es cierto que un TLU, tiene clasificación binaria, pero entonces como podemos clasificar un numero 14 por ejemplo.

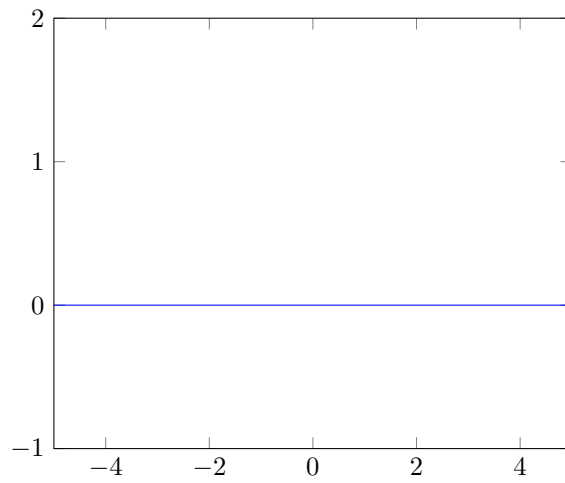


Figure 1: Gráfico de la función por rangos

3 shMessages

3.1 Links Importantes del Proyecto

1. GitHub del proyecto
2. Bases de Datos con las que se entrenó el Algoritmo:
 - (a) Base de datos 1