02 - Modelo Matemático de una Neurona Artificial - REDES NEURONALES - (Parte 2)

Transcript

En este vídeo vamos a mostrar por qué decimos que una neurona artificial está inspirada en una neurona biológica y asimismo vamos a mostrar la forma de modelar matemáticamente una neurona artificial.

Con estos conocimientos vamos a construir nuestro primer modelo de red neuronal artificial basado en un clasificador lineal.

Empezamos.

Y así en este modelo de red neuronal artificial cada uno de estos círculos de estos elementos representa una neurona y estas neuronas artificiales también están inspiradas en las neuronas biológicas y en este sentido las neuronas biológicas tienen esta forma donde esta parte es la parte del cuerpo y la parte del cuerpo tiene estas terminales de entrada que se llaman dendritas y también tiene una terminal de salida que se llama axon.

Lo que sucede es a grandes rasgos y de manera muy muy resumida es que una señal eléctrica es transmitida a través de esta señal de salida la cual llega a estos puntos de contacto y estos puntos de contacto que es la sinapsis, la terminal sináptica, entra en contacto con dendritas de otra neurona y así esto hace que se libere un neurotransmisor los cuales entran en contacto con un receptor de la neurona que está recibiendo este neurotransmisor provocando que hay una depolarización del nivel de voltaje al cual está esta neurona.

Recordemos que esta es una sola conexión pero hay otras conexiones y hay conexiones en todas estas terminales terminales sinápticas con las dendritas por lo que todo esta suma todo este valor agregado de depolarización es que hay dependiendo de la interacción que haya entre neurotransmisor y receptor pueden ocasionar que haya una acción excitatoria que haga que se propague una señal eléctrica o que haya una acción de inhibición y que no se propague dicha señal pero suponiendo que que esta señal es propagada esta sería el resultado entonces de todas estas interacciones entre las diferentes sinapsis que hay de diferentes neuronas conectadas a la dendrita de esta neurona e inspirado en esto con esta inspiración surge el concepto de una neurona artificial la cual no es otra cosa que una sobre simplificación de las neuronas biológicas porque la comunicación sináptica y la comunicación entre neuronas es un proceso muy complejo que todavía está siendo estudiado por lo tanto es muy importante notar que no debemos decir que las neuronas la inteligencia artificial o las redes neuronales funcionan como el cerebro simplemente tienen inspiración en él y de ahí hacen una operación que es muy muy muy sencilla la forma en la que se sinula es que tenemos este esta neurona artificial que sería este círculo aquí esta es una neurona artificial y lo que tenemos es pesos asociados a cada una de estas entradas donde podríamos decir que bueno esta entrada corresponde a las dendritas y aquí tenemos un peso asociado vamos a llevarlo w1 w2 w3 esto porque es la variable más común para representar los pesos de las neuronas w4 y lo que tendríamos acá son una conexión sináptica donde lo que llegamos acá lo que llega aquí es la salida de otra neurona y lo que tendríamos es esta conexión aquí supongamos que este es otra neurona cuya cuya entrada primera entrada está conectada a la salida de esta neurona y la forma de modelar este tipo de comunicación de forma matemática es que la salida que tenemos acá de esta neurona es el resultado vamos a decir y va a ser el resultado de la multiplicación de cada uno de estos pesos por cada una de sus entradas supongamos que esta es la entrada x1 esta es la entrada x2 esta es la entrada x3 y esta es la entrada x4 lo que tendríamos entonces que la salida lleva a ser la suma de w1 por x1 más w2 por x2 más w3 por x3 más w4 por x4 y podemos observar que esta operación no es otra cosa más que un producto punto por lo cual lo podemos representar en forma vectorial como w multiplicado por el vector x donde w y x en este caso son vector es un producto punto y esta es la forma en la que se simula la operación de las neuronas artificiales es una operación lineal que nos va a dar un valor numérico que puede ser prácticamente cualquier número y muchas veces lo que queremos es que el valor de activación de las neuronas tenga un rango específico por ejemplo en algún momento se pensó que este valor debería tener un valor entre 0 y 1 por lo cual esta salida se utiliza como la entrada a otra función no lineal además el hecho de que se necesita una función no lineal las salidas es muy importante porque si nada más tenemos aquí una función lineal y esto lo conectamos a otra neurona que realiza otra función lineal la salida que tenemos acá y dos va a ser una función lineal finalmente por lo que es muy importante que más de ser esta operación tengamos una una no linealidad a grandes rasgos este es la forma en la que las neuronas artificiales fueron inspiradas de las neuronas biológicas vamos a hablar de esto con más detalles a continuación vamos a regresar al ejemplo de nuestro gato y vamos a suponer que este gato esta imagen del gato la dividimos en cuatro píxeles y de nueva cuenta esto sabemos que aquí hay muchos más píxeles en cada una de estas pero para simplificarnos lo que estamos haciendo lo que estamos suponiendo es que esta imagen del gato la convertimos a un vector es decir la desenrollamos si fuera si fuera esta imagen cuadrada convertimos a un vector y podemos decir que vamos a le estoy dando aquí valores arbitrarios entonces cada uno de estos cuadritos cada uno de estos bueno son rectángulos que tenemos acá representan el valor correspondiente a cada uno de los cuatro píxeles que estoy suponiendo tiene esta imagen del gato el primer pixel este pixelote voy a suponer que tiene un valor de punto 12 el segundo pixel tiene un punto 257 y el tercero de punto 25 y el cuarto de punto 34 se realiza la multiplicación de cada elemento de cada pixel este sería el elemento x1 este sería el elemento x2 este sería el elemento x3 este sería el elemento x4 esta sería nuestra imagen de entrada nuestro dato de entrada x y cada uno se multiplica por su por su correspondiente peso sináptico w1 w2 w3 y w4 de tal forma que al hacer esta operación en un tipo de producto punto es decir cada elemento elemento elemento y luego la suma de esos productos tenemos un valor de 1.09 este valor va a ser el resultado de 0.12 por 0.17 más 0.57 por 0.32 y así sucesivamente algo que puede pasar es que querramos que esta neurona solamente se active cuando tenga valores mayores a 2 por decir algo y para garantizar esto lo que podemos hacer es aquí sumarle un valor al que le vamos a llamar un bias un tipo offset que tiene un valor de 2 de tal forma que el resultado de esta operación se puede sintetizar como nuestras w's ya en un vector de w's vamos a poner la transpuesta por nuestro vector de x más este valor de offset para garantizar que la salida en este caso corresponda al valor que resultante del producto punto pero que tenga un valor más grande que determinado nivel en este caso sería 3.09 y esta es la salida de esta neurona ahora como mencioné brevemente anteriormente si esta neurona la pensamos conectar con otra neurona para hacer un procesamiento jerárquico en cascada como vimos en esta en estas redes neuronales más complejas donde cada unidad está conectada a otras 12 neuronas para que el poder de abstracción el poder de representación de cada una de estas neuronas sea mayor o sea sea más poderoso que el de una función lineal es necesario que la salida que tenemos acá de cada neurona sea el resultado de una función no lineal por lo tanto después de haber hecho la operación w's la transpuesta de w por x más b se le aplica a este resultado este resultado va a ser un número va a ser un valor numérico un un escalar en este caso este valor mostrado acá a esta a este valor se le va a aplicar una función de activación no lineal por lo tanto llamamos que la salida de la neurona es una activación y esa activación lo que va a permitir es que al concatenar muchas neuronas formando una red neuronal profunda el poder de representación sea mucho mayor ya que si concatenáramos solamente la salida de una función lineal con otra función lineal el resultado seguiría siendo una función lineal por lo tanto es sumamente importante utilizar una función de activación que introduzca una no linealidad en la salida de cada neurona inicialmente la función de activación más utilizada es la función sigmunt actualmente vale la pena mencionar que ya no es la más utilizada y solamente se utiliza bajo ciertos escenarios por ejemplo como como gates en redes neuronales recurrentes incluso a principios de este siglo los primeros 10 años del 2000 al 2000 días era la función del 2000 al 2012 era la función de activación más utilizada ahora se sabe que no es la más óptima y hay otras que vamos a estudiar en el siguiente vídeo sin embargo para fines prácticos para fines de introducción al tema de redes neuronales es útil considerar esta función y las características de esta función es que va entre 0 y 1 aquí estamos en 1 y conforme vamos viendo vamos viendo que para valores grandes aquí vamos a decir que más o menos por aquí debe ser 5 para valores positivos muy grandes la función converge en 1 y para valores de x negativos muy grandes la función se va a 0 lo que nos garantiza esta función es que la salida vaya en un nivel entre 0 y 1 estas funciones de activación cuando están en la última capa de una neurona por ejemplo supongamos este nivel este este sistema lineal muy sencillo y que lo que quisiéramos hacer es que aprenda a reconocer gatos si tenemos esta activación con este modelo con estos valores obtendríamos que 3.09 sería la entrada nuestra función sigmo y hacer un valor muy cercano a unos de tal forma que nos estaría diciendo que el modelo estaría prediciendo con mucha confianza que nuestro resultado que nuestra imagen pertenece a la imagen de un gato y así vamos a introducir el concepto de un clasificador con una neurona que nos permita decir con cierta probabilidad con cierto nivel de confianza si la imagen analizada corresponde a un gato o no pero por ejemplo qué pasa si queremos hacer un sistema basado en redes normales que determine si una imagen pertenece a la clase gato a la clase perro o a la clase baja y por clase me refiero a qué tipo de imagen es es la palabra que vamos a estar utilizando constantemente supongamos entonces que tenemos este sistema este es nuestra red neuronal completa donde solamente tenemos una capa de salida que ya es la capa de salida que recibe los píxeles de las imágenes de entrada estos cuatro píxeles corresponden a los píxeles de estas imágenes y dependiendo de los pesos sinápticos asociados a cada una de estas neuronas vemos que para la primera neurona tenemos pesos son estas conexiones que están en verde para la segunda neurona son estas conexiones en naranja y para la tercera son estas conexiones en gris y cada salida nos representa una clase nos representa un tipo un tipo de especie de tal forma que esta salida el número que salga de esta neurona que está aquí en color verde va a representar la clase gato la de este naranja va a representar la clase perro y la clase pájaro al final en color gris vamos a hacer un ejemplo de un clasificador lineal donde lo que queremos hacer es reconocer tres clases la clase gato la clase perro y la clase pájaro y cada una de estas neuronas debería reaccionar con mayor nivel cuando la imagen de entrada corresponde a un elemento de su clase es decir en esta imagen de un gato esta neurona debería ser la que tenga un nivel de activación más alto pero esos niveles de activación corresponden con los pesos sinápticos asociados a cada una de las neuronas aquí por ejemplo tenemos en color verde y quiero mencionar que estos valores son totalmente arbitrarios fueron seleccionados por mí a mano no fueron aprendidos no son valores nada más para ilustrar cuál es el proceso de cálculo cuando se hace una clasificación en este caso de tres elementos entonces lo que tenemos es que esta conexión tendría un peso sináptico de 0.17 la segunda de 0.32 esta tercera de 0.12 y finalmente esta de aquí de 2.53 vemos aquí que los colores están indicando cuáles son los pesos sinápticos de cada neurona en el caso de las conexiones en verde están dadas por esta fila las conexiones en naranja por la segunda fila y las conexiones en gris que son las conexiones asociadas a los a los pesos sinápticos de la tercera neurona están en color gris y entonces lo que tenemos es que la operación se puede realizar como un producto de una matriz por un vector donde el vector representa los píxeles de la imagen de entrada y de nuevo a cuenta estoy asumiendo que son píxeles que esta imagen tiene nada más cuatro píxeles no sabemos que tiene mucho más no importa asumimos que son cuatro píxeles 0.2 0.5 17.25 .34 de tal forma que hacemos la multiplicación de esta matriz que representa los pesos sinápticos de las neuronas con el vector que representa la imagen si nos damos cuenta esta matriz tiene una dimensión de tres filas por cuatro columnas y este vector es un vector columna por lo que son cuatro filas y una columna al hacer la multiplicación de una matriz con un vector de este tamaño el resultado nos va a dar un vector con dimensiones de tres filas y una columna y a ese resultado recordemos que le queremos sumar a cada una de estas neuronas le vamos a sumar un parámetro vallas que nos va a permitir o garantizar que las neuronas tengan un nivel mayor a determinado valor esta esta b que en este caso también están dados de forma arbitraria por estos valores 2.23 para la neurona que corresponde a la clase verde a la clase gato 2.75 para la neurona que corresponde a la clase aquí que está en naranja la clase que sería la clase de los perros y 2.57 para la clase de los pájaros de nueva cuenta repito estos valores son arbitrarios nada más estoy señalando que así sería como se hace la operación por lo tanto esta operación como habíamos mencionado anteriormente se daría por la multiplicación matriz por vector más y notemos que el resultado de esta matriz de la multiplicación de la matriz por este vector va a tener las mismas dimensiones que el vector b por lo tanto va a poder va a poder sumarse sin problemas elemento elemento y así tenemos que el resultado de esta de esta toda esta operación va a tener estos valores aquí lo importante es notar que los valores por ejemplo el valor que se está dando para la clase gato es mayor que los valores para la clase perro y para la clase pájaro de esta manera podemos decir que nuestra red está reconociendo correctamente que la clase la imagen de la entrada es una imagen corresponde a un gato porque es al que le está dando mayor valor esta salida todavía es lineal no estamos aplicando la no linealidad que pasa si ahora utilizamos un elemento de entrada que corresponde a la clase perro y ahora los píxeles corresponden a estos valores de nueva cuenta los píxeles son inventados no obstante notemos que el valor de w y el valor del bias del offset no van a cambiar porque esos son valores que ahorita yo yo definí y lo que queremos es ver qué también funcionan estos valores para predecir cualquiera de los elementos que estamos analizando sean gatos perros o pájaros entonces de nueva cuenta hacemos la operación de matriz por vector le sumamos el bias y obtenemos estos resultados y nos damos cuenta que el valor más alto corresponde al valor de la clase pájaro el segundo valor más alto corresponde al valor de la clase gato y el valor más bajo es el valor de la clase perro la cual es la clase correcta por lo que a diferencia del ejemplo anterior donde habíamos visto que el valor más alto efectivamente correspondía a la clase gato o sea nos estaba prediciendo que que la imagen más probable era de un gato en este caso nos está diciendo que esta imagen de entrada lo más probable es que corresponda a un pájaro es el valor más alto aquí es donde entra el proceso que tenemos que definir para calcular estos valores w estos valores de peso sinápticos y estos valores de bias también todos estos son parámetros de nuestra red neuronal que la red neuronal va a aprender por ejemplo suponiendo que la clase de entrada es esta imagen de nueva nueva cuenta los parámetros w y b son los mismos calculamos cambiamos ahora la los píxeles los valores de los píxeles a valores arbitrarios y nos damos cuenta que la salida efectivamente predice bien que el valor más alto que arroja es para la clase que corresponde a los pájaros entonces está prediciendo bien este valor no obstante el valor predicho para la clase para esta imagen de perro es incorrecto cómo hacemos que la predicción sea la correcta para esto se necesita definir de forma matemática qué tan buenos son estos valores w qué tan buenos son estos valores b estos valores son los que la red neuronal va a aprender estos son los valores que van a permitir que dependiendo de una entrada la red neuronal evalúe una salida que depende de la entrada x y los parámetros w y bias para esto necesitamos saber de manera cuantitativa qué tan buenos son los valores que se tienen actualmente y esto se hace a través de una función de pérdida una los function que dependiendo de la clase que está predicha y de la clase correcta va a evaluar qué tan bueno qué tan aproximado el valor es de la clase real por ejemplo en este caso dado que el valor más alto pertenece a la clase pájaro la función la pérdida va a ser menor que en el ejemplo anterior donde el valor de la clase correcta es el más bajo aquí tiene que haber una función que castigue más estos parámetros de tal forma que los cambie más así lo que se va a buscar es que esa función de pérdida se minimiza y ese va a ser el proceso de aprendizaje de la red neuronal sin embargo eso lo vamos a ver en el siguiente vídeo para terminar si quisiéramos hacer esta implementación de la predicción para cada uno de estos elementos lo que haríamos entonces sería una multiplicación de matrices donde lo que tenemos es por un lado la matriz con nuestros pesos nuestra matriz w y por otro lado tenemos una matriz donde ponemos en cada una de las columnas cada uno de los ejemplos por ejemplo aquí la primera columna es el gato la segunda columna es el perro y la tercera columna es el pájaro y así podemos realizar esta operación esta multiplicación de matrices en un solo paso que los softwares como como diferentes librerías de python o incluso matlab realizan de forma muy eficiente por lo tanto esta operación se va a realizar como multiplicación de matrices y de ahí solamente le sumamos los valores de los valles y de en un solo paso prácticamente tenemos los resultados de las predicciones para cada una de las imágenes de entrada donde lo que estamos viendo en la primera columna es que en el caso de esta imagen de entrada que corresponde a un gato la clase gato predijo el valor más alto lo cual es correcto la clase perro el valor más bajo y la clase pájaro el siguiente valor para la entrada de perro la clase que tuvo la predicción con el valor más alto fue la clase pájaro lo cual es incorrecto porque sabemos que debería haber sido el valor con la clase perro por lo cual quiere decir que estos parámetros que tenemos acá no son los óptimos no son ideales en el último ejemplo la clase pájaro efectivamente el valor de la clase pájaro fue el valor más alto para esta imagen de entrada de tipo pájaro pero no obstante estos valores tienen que ser ajustados para que prediga de manera correcta la mayor cantidad de datos cómo aprende una red normal y cómo se hacen los cálculos de los valores de los pesos sinápticos y el vallas son los temas que vamos a abordar para entender cómo funciona la red normal y de ahí nos vamos a enfocar a cómo programarla vamos a utilizar primero python para programarla desde cero y después nos vamos a seguir usando python pero ya vamos a utilizar frameworks de deep learning que nos facilitan mucho el trabajo en este vídeo presentamos nuestro primer modelo de red neuronal artificial la cual solamente tuvo una capa que nos permitió clasificar tres posibles tipos de imágenes de entrada sin embargo los resultados presentados son solamente el resultado de una operación lineal de una transformación lineal por lo tanto en el siguiente vídeo vamos a presentar una función de activación que es la función de activación más utilizada en problemas de clasificación multiclase es decir de más de una clase y lo que nos va a permitir esta función es representar estos scores o marcadores que obtuvimos mediante nuestra función lineal a probabilidades las cuales van a representar qué tan posible que tan probable es que una determinada imagen pertenezca a una clase en particular vamos a continuar [SILENCIO] [MÚSICA]