

¿QUÉ ES UNA RED NEURONAL? PARTE 1 LA NEURONA

Las redes neuronales son una familia de algoritmos potentes que ayudan a modelar comportamientos inteligentes.

Conceptos básicos

- Neurona: Unidad básica de procesamiento, tiene conexiones de entrada que recibe valores de entrada para estimularlo y obtener un valor de salida, hace una suma ponderada de cada conexión de entrada a la cual tendrá asociado un valor que identificará que tanto afectará a esa conexión y será la que podremos modificar. Las neuronas son una variante de una regresión lineal, ya que también contamos con un término independiente que nos dará control para mover a la función lo que nos dará sesgo, y para tener control sobre nuestra función manipularemos siempre la variable de sesgo.
- Modelo de una puerta lógica AND u OR: Frontera entre las dos clases a clasificar.
- Modelo de una puerta lógica XOR: No tiene solución ya que depende de una sola neurona, y solo se puede usar agregando otra.

¿QUÉ ES UNA RED NEURONAL? PARTE 2 LA RED

Con una sola neurona no se pueden separar linealmente a una nube de puntos distribuidos. A través de capas (niveles) se van agrupando las neuronas, todas aquellas que se encuentren en la misma capa reciben la misma información de las neuronas de la capa anterior. Se busca que la red pueda aprender conocimiento jerarquizado.

Composición de una red neuronal:

- Capa de entrada: Primera. Recibe las variables que afectan.
- Capas ocultas: Las combinaciones de acuerdo los valores recibidos.
- Capa de salida: Última. Aporta información más elaborada y compleja.

Funciones:

- Función de activación: Es una función escalonada que dice que para un valor de entrada mayor al umbral el input es uno y para una menor al umbral será cero, para producir inmediatamente el cambio. A grandes rasgos sirve para distorsionar el plano generado por la neurona.
- Función sigmoide: permite variar la forma u orientación de acuerdo a cambios en parámetros de la red y es útil a partir de 4 variables.

JUGANDO CON REDES NEURONALES. PARTE 2.5

Una manera de experimentar con redes neuronales la podemos encontrar en la página de internet "Play ground tensor flow", que permite jugar con redes neuronales modificando variables y función de activación para ver los resultados que esos cambios generan.

Elementos

- Datos: Nubes de puntos, regresión, encontrar la función ajustable a las variables, clasificación, separación de puntos.
- Variables de entrada: Gradiente, eje X valores positivos y negativos; Eje Y.
- Variables de salida: Input space (espacio de entrada), colocar en los ejes las entradas y visualizar los valores, solo se puede en dos dimensiones aunque puede hacer más. Para mejor visualización de resultados se pueden agregar varias capas de entrada u ocultas y más neuronas, lo que generaría patrones de entradas.
- Capas ocultas: Podemos separar nubes de puntos y encontrar fronteras. Discretizar para agregar funciones de activación (añadir una no linealidad) y donde podemos separar los puntos. Añadir capa oculta para encontrar soluciones según las manipulaciones. Cambiar el tipo de funciones de activación: rectilínea uniforme, tangente hiperbólica, sigmoide y lineal.

¿QUÉ ES UNA RED NEURONAL? PARTE 3 BACKPROPAGATION

Redes neuronales: Son capaces de modelar información compleja, pueden usarse para resolver tareas de regresión o clasificación, ayudan a ganar más intuición. Se busca que la red aprenda por si sola a partir de los datos, aprendizaje automático.

Algoritmos

- Persceptron: Concepto parecido a redes neuronales pero más ambiguo, este era en los años 50.
- Backpropagation: Usando un nuevo algoritmo de aprendizaje se podría conseguir que una red neuronal autoajustara sus parámetros para así aprender una representación interna de información que esté procesando.

Descenso del gradiente: Conseguimos la estrategia perfecta para ajustar los parámetros de una regresión lineal reduciendo el error.

¿Cómo varía el coste ante un cambio del parámetro W ? El gradiente influye a todas las conexiones por lo tanto no es tan fácil como una regresión lineal, por lo tanto combinamos el descenso del gradiente con el algoritmo backpropagation.

Retropropagación de errores: Analizar la cadena de responsabilidades hacia atrás desde el error, siendo nuestra primera parada la última capa en función de cuánto se halla implicado cada neurona en general y seguimos retrocediendo capa por

capa para encontrar cuál fue el principal error para cada neurona y sus parámetros, solamente propagando una única vez el error hacia atrás.

Conclusión: Los errores encontrados con el algoritmos serán los usados para calcular las derivadas parciales de cada parámetro de la red conformando así el vector gradiente, que es lo que se ocupa para el descenso del gradiente y así minimizar el error y entrenar a la red.