Santiago Cruz Todd

→ 10/11/24

Resumenes de Redes Neuronales:

¿Qué son las Redes Neuronales?

= LA NEURONA =

Es un modelo computacional que ha sufrido mejoras en base a las últimas innovaciones en tecnología. Es una familia de algoritmos que modela comportamientos diversos.

El comportamiento que toma este modelo se basa en la interacción de partes mas simples trabajando conjuntamente, la neurona. Es la unidad básica del procesamiento; éstas reciben valores externos, que poseen unidades o conexiones de entra, y estos 'estímulos' externos poseen información. Todo este procesamiento generará un valor de salida; este proceso siendo análogo al de una función matemática Internamente, utiliza los valores de entrada para realizar una suma ponderada de ellos, la ponderación viene dado por el peso asignado a las valores de entrada, cada conexión tiene un valor, "una palanca"; los pesos son los parámetros ajustables de nuestro modelo. Similar al comportamiento de una regresión lineal.

Utilizando un comportamiento de Variable binarias, la neurona puede modelar un comportamiento lógico, las entradas (var. binarias) representan condiciones "como tener los lentes de VR. o tener nachos" y dado hay valores presentes asignados a cada variable 1 y 0. Y la misma salida que tendrán los soluciones también sería binaria.

Los ajustes de parámetros también se llaman BIAS, y los modelos lógicos pueden tener una puerta AND (si las salidas sería 1, ambas entradas 1)

También este la situación donde hay un incumplimiento parcial de las condiciones y las entradas no son linealmente separables (XOR) y en este caso una sola neurona no es suficiente.

Estas son algunas de las utilidades de las Redes neuronales:

...

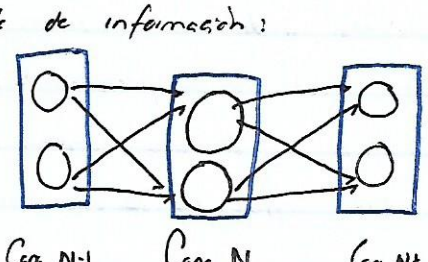
machine learning

Reconocimiento de caracteres, de imágenes, voz, predicción bursátil, generación de texto, traducción, prevención de fraude, análisis genético...

II - "La Red"

¿Cómo se organizan las neuronas?

Los pares de neuronas se ordenan en forma de capas que realizan el recibimiento y salida de información:



"aprendizaje jerarquizado"

Cada capa procesa la información de manera jerárquica.

Las primeras capas extraen características básicas y las posteriores elaboran conceptos más abstractos. Esto es clave para tareas más complejas como clasificación de imágenes o predicciones.

Más capas significa más complejo y esto resulta en el Deep Learning.

Pero...

Esta conexión secuencial de neuronas "lineales" equivale a haber sido solo una. Significando que el proceso iterativo ha sido fútil... y colapse.

Todas las líneas deben tener alguna manipulación — por medio de las funciones de activación; estas transformaciones no lineales permiten calcular cosas

Algunas funciones comunes son:

- El Sigmoide: produce salidas 0 y 1, para probabilidades

$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$$

* Satura valores; muy grandes se satren a 1, los bajos en 0.

- Tangente Hiperbólica: Con rango variable de -1 a 1, útil para valores negativos

$$f(x) = \tanh(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

- ReLu "Rectificado Lineal", que simplifica los cálculos manteniendo valores positivos y anulando los negativos

Estas deformaciones permiten activar las neuronas.

Parte III "Redes Neuronales"

Backprop

Parte del propósito del "Machine Learning" es que ajustando el modelo este sea capaz de modificarse a sí mismo a partir de los datos. Aprendizaje Automático.
El perceptrón, considerado que trabaja con una única neurona es un precursor para un modelo mucho más complejo; estos modelos tienen sus restricciones pero los modelos de aprendizaje aplicables al perceptrón. La "primavera" de la IA vino después de la publicación del texto "back-propagation", y esto implicaba la auto-ajustación que impedía a los modelos automatizar su aprendizaje.
Los cálculos que se realizan es analizar el error de la salida hacia las capas iniciales de la red. Esto permite una retropropagación del error.
La retropropagación permite calcular todos los gradientes necesarios con un solo

paso hacia atrás. La retropropagación funciona como una cadena de responsabilidades, donde cada nodo de la red (neurona), representa una etapa que contribuye a un resultado final. Retrospectivamente se puede determinar qué pesos tuvieron mayor influencia en el error final y ajusta sus parámetros en consecuencia.

Part 4: "Las matemáticas del Back-Prop"

La retropropagación calcula las derivadas parciales de los parámetros de la red, los pesos w y sesgos b, con respecto al costo. Esto es fundamental para minimizar el error durante el entrenamiento usando el descenso de gradiente. La diferencia clave de este método es que el backprop identifica las responsabilidades "gradientes" en el error; el descenso del gradiente usa estos gradientes para ajustar los parámetros y mejorar en todo caso el modelo.

Una parte interesante del video es la "posible" aplicación del algoritmo para medir el rendimiento de varias áreas de una empresa, en donde cada sector / empleado funciona como una neurona y en base a la retrospectiva mencionada anteriormente se puede evaluar la funcionabilidad de cada una.

Fórmulas vistas

↳ ¿Cómo varía el costo ante un cambio del parámetro w?

$$\frac{\partial C}{\partial w} \rightarrow \text{weigh}$$

$$\frac{\partial C}{\partial b} \rightarrow \text{bias}$$

Fórmulas:

C coste

↑

w^L (valor del parámetro) b^L → (Z^L) a

a: función de activación

Suma ponderada

$w^L x + b^L \longrightarrow Z^L$ resultado de la suma ponderada

$a(Z_L)$ → aplicada la func. de activación, en la última capa

$c(a(Z^L))$ = función de Coste

Composición de funciones

Todo esto siguiendo una Chain Rule.

$\frac{\partial C}{\partial w^L}$

$\frac{\partial C}{\partial b^L}$

$Z^L = w^L a^{L-1} + b^L$ $\quad c(a^L(Z^L))$

Se necesitan calcular todas las derivadas intermedias

$$\frac{\partial C}{\partial w^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial w^L}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial b^L}$$

$$c(a_j^L) = \frac{1}{2} \sum_j (y_j - a_j^L)^2$$