

1.- Explica las partes de una neurona artificial

En general, está compuesta por 4 secciones:

1: Entradas

Aquí están incluidos los datos de entrada, generalmente variables de entrada a través de las cuales se pretende que la red neuronal artificial "aprenda". También, en esta sección se tienen los pesos sinápticos (o pesos solamente), estos son cifras que pueden tomar valores positivos, de 0 o negativos, mediante la manipulación de estos pesos se le da "más importancia" o "menos importancia" a cada dato de entrada en el proceso de aprendizaje. Un valor positivo en un peso para una entrada actuará como amplificador de la importancia de tal dato para el aprendizaje y ajuste a la salida, e inversamente, un valor negativo a un peso tendrá un efecto de atenuador de tal dato.

2: Regla de propagación (Sigma)

Básicamente es la relación que se le dará a cada neurona con las demás, la regla mayormente utilizada es una suma ponderada, es decir, consiste en sumar las entradas multiplicando cada una por un peso.

3: Función de activación

Con base al valor resultante de la etapa anterior, se tiene una función de activación que se encarga de devolver una salida a partir de una entrada o conjunto de entradas, la entrada es el valor generado por la etapa anterior, y la salida se produce al transformar tales entradas mediante una función matemática, las más común es la sigmoideal, pero también está la de arco tangente pero es computacionalmente más costosa que la sigmoideal, por el efecto de la derivada de la función.

4: Salidas

Esta proporciona el valor o los valores de salida con base a la activación de las neuronas.

2.- ¿Por qué consideramos al Perceptrón una red neuronal offline?

Por que el perceptrón clasifica cada dato al multiplicar esa entrada por el vector de pesos, necesita "ajustar" o "aprender" primero todos esos pesos y lograr un ajuste a los datos de salida provistos. Es decir, se considera offline por que necesita primero "aprender" de todos los datos que tenga disponibles y hacer esto en un solo "batch" o una sola secuencia de repeticiones del proceso de aprendizaje.

Esto debido a que el propósito de un proceso offline es que el modelo aprenda la dinámica o estructura del fenómeno a través de todos los datos con los cuales se le entrena para que lo logre. De tal manera que en un aprendizaje offline, los elementos de entrada, su orden y su identidad (salida que generan) es conocida por el proceso que lo estará aprendiendo

3.- ¿Qué es alfa en el perceptrón y cómo afecta su funcionamiento?

Podría decirse que es la "tasa de aprendizaje", esta afecta directamente las proporciones de cambio en los pesos durante el proceso de entrenamiento, un alfa cercana a 0 provoca en cambios menos significativos (o conservadores), mientras que un alfa cercana

a 1 hará que las modificaciones de los pesos sea algo más significativas (o radicales).

4.- Explica la etapa de aprendizaje y el método que usa para hacerlo.

Se comienza por la elección de pesos para todas las neuronas, esta elección suele ser aleatoria y sólo existen heurísticos como guía, no hay una regla absoluta sobre el criterio de inicialización de pesos.

Se hace con una técnica llamada propagación hacia atrás, el principio de esto es que se busca ajustar los pesos de las conexiones entre las neuronas de la red en función de las diferencias entre los valores de salida que se tienen (objetivo) y los valores de salida generados por la red (ajustados). Y la regla de retropropagación es simple, para cada neurona en la capa de salida (salida ajustada) se le calcula desviación o diferencia respecto a la salida objetivo como un error delta. y este luego se utiliza para cambiar los pesos hacia la neurona previa (de ahí retropropagación), como un efecto cadena. Uno de los conceptos principales para este proceso es que la propagación hacia atrás se hace mediante las derivadas de la función activación respecto a cada capa y cada neurona de cada capa hacia atrás.

Finalmente, un criterio de paro para la retropropagación es que, dado que los pesos de las conexiones entre neuronas se modifican a cada iteración, se puede afirmar que este proceso ha terminado cuando los valores de los pesos permanecen estables, es decir que las derivadas no están teniendo cambios significativos ya.

5.- Usa un Perceptrón simple con umbral para llenar los valores faltantes de la siguiente tabla.

| x1 | x2 | x3 | x4 | d |
|----|----|----|----|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | ? |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | ? |
| 1 | 0 | 0 | 0 | ? |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | ? |
| 1 | 1 | 1 | 1 | 1 |

Se sabe que es una tabla XOR, entonces se tienen las respuestas válidas para las d=?

Ver script adjunto llamado IDI_II_Tarea_7_Perceptron_JFME.py