
Computación Bioinspirada - Práctica N° 6

PROFESOR DEL CURSO: Dennis Barrios Aranibar

FECHA: 24 de Setiembre del 2018

ASISTENTE DEL CURSO: Kevin Christian Rodríguez Siu

Objetivos de la Sesión

- Analizar como una Red Neuronal Perceptrón Multi Capa aproxima una función dada.

Contexto

La Red Neuronal Perceptrón Multi Capa es la evolución natural de la Red Perceptrón. Al haberse demostrado que esta última no podía separar problemas no lineales, cayó en desuso hasta 1986, donde la aplicación del algoritmo BackPropagation para hacer el entrenamiento de múltiples capas de pesos logró convertir este clasificador en una herramienta capaz de separar problemas no lineales.

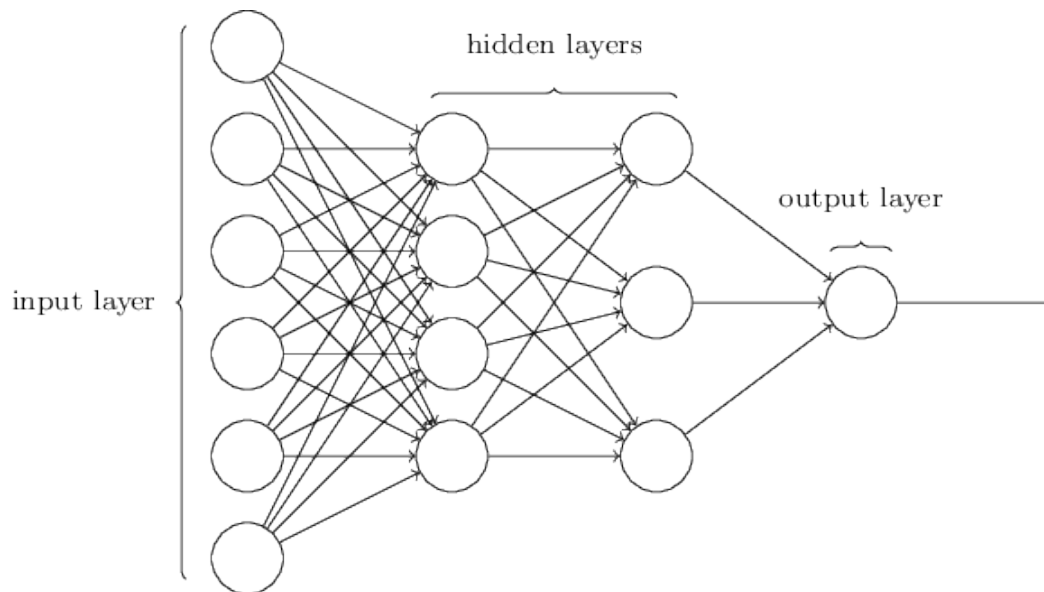


Figura 1: Esquema de una Red Perceptrón Multi Capa

Aunque hay varias formas de representación de este algoritmo, en esta ocasión, pensaremos en todos los elementos como si fueran matrices. Entonces, para una red con solo una capa oculta, requerimos:

- m , n y o , que son respectivamente la cantidad de neuronas en la entrada, la capa oculta y la salida.
- La matriz $W_{EO}(m + 1, n)$, que son los pesos que conectan la capa de entrada con la capa oculta.
- La matriz $W_{OS}(n + 1, o)$, que son los pesos que conectan la capa oculta con la capa de salida.
- La matriz $W_B(2, 1)$, que son los bias, uno por cada capa que tenemos

- La función de activación f , que suele ser una de las funciones mostradas en la figura 2. Las más usuales son la función sigmoide y la función tangente hiperbólica.
- La derivada de la función de activación f' . La derivada de la función sigmoide es: $f'(x) = \frac{e^{-x}}{(e^x - e^{-x})^2}$. La derivada de la función tangente hiperbólica es: $f'(x) = \frac{1}{1+e^{-x}}$.
- Un ratio de aprendizaje α , que suele estar entre 0 y 1.

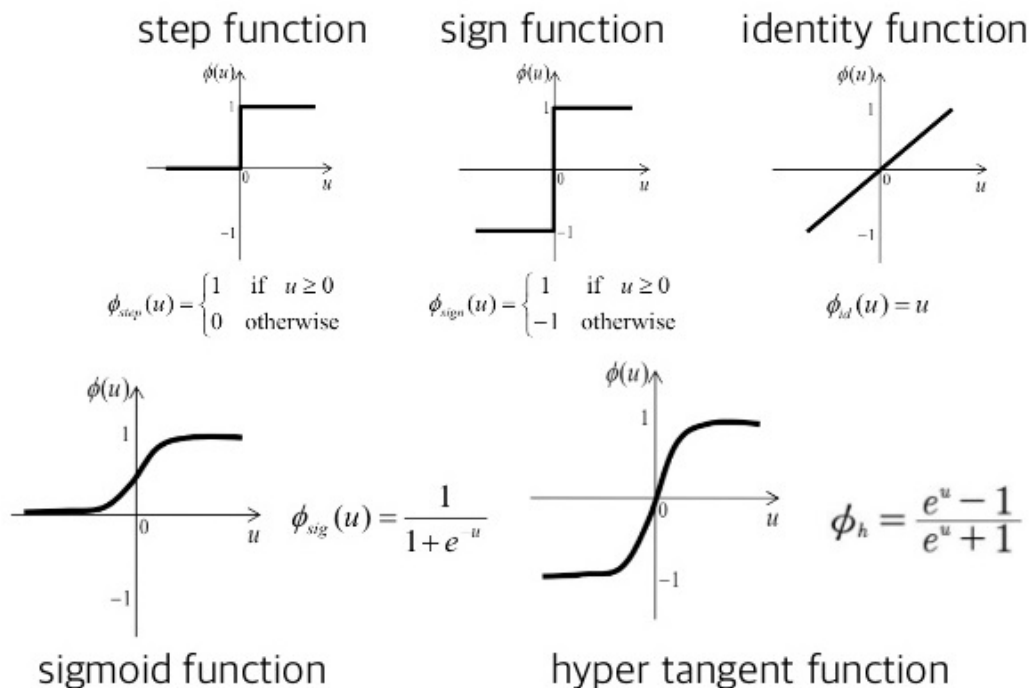


Figura 2: Funciones de Activación Populares

Y para utilizarlo, se debe hacer el siguiente proceso:

1. Primero, definir las entradas y salidas esperadas, que tendrán la forma: $X(m, 1)$ y $Y(o, 1)$.
2. Luego, generar los pesos y los bias W_{EO} , W_{OS} y W_B aleatoriamente.
3. Luego viene el **Paso Hacia Adelante**:
 - Dada una entrada X , se calcula la matriz X_{EXT} , que es meramente una extensión de la entrada con el primer elemento de la matriz W_B .
 - Se calcula la matriz $V_{EO} = X_{EXT} \times W_{EO}$.
 - Se calcula la matriz $Y_{EO} = f(V_{EO})$
 - Luego, se calcula la matriz Y_{EXT} , que es la extensión de la matriz Y_{EO} con el segundo elemento de la matriz W_B
 - Se calcula la matriz $V_{OS} = Y_{EXT} \times W_{OS}$.
 - Se calcula la matriz $Y_{OS} = f(V_{OS})$.
4. Después, viene la **Propagación del Error hacia atrás**:
 - Calcular la matriz del error de la capa oculta hacia la capa de salida $E_{OS} = Y - Y_{OS}$
 - Calcular la matriz $Y_{PS} = f'(V_{OS})$

- Actualizar la matriz W_{OS} de la siguiente forma: $W_{OS} = W_{OS} + \alpha \times (Y_{EXT}^T \times (E_{OS} \times Y_{PS}))$
 - Calcular la matriz del error de la capa de entrada hacia la capa oculta $E_{EO} = W_{OS} \times (Y_{PS}^T \times E)$.
 - Calcular la matriz $Y_{PO} = f'(V_{EO})$
 - Actualizar la matriz W_{EO} de la siguiente forma: $W_{EO} = W_{EO} + \alpha \times (X_{EXT} \times (E_{EO} \times Y_{PO}))$
5. Para mejores resultados, mezclar las entradas cuando se haya pasado por todas.
 6. Continuar hasta que se hayan hecho un cierto número de iteraciones o el error E haya alcanzado un número muy cercano a cero (0).
 7. Para utilizarla luego del entrenamiento, sólo se realiza el paso hacia adelante y se toma como resultado la matriz Y_{OS} .

Ejercicios

1. Programar una Red Neuronal Perceptrón Multi Capa que sea capaz de recibir dos entradas y dar una salida. Debe contener una capa oculta.
2. Luego, entrenarla para que pueda reconocer las siguientes funciones. Debe mostrar en tiempo real como se va acomodando para hacer la separación de las clases dadas.
 - La función lógica AND (Y) con las siguientes entradas y salidas esperadas:

Entrada 1	Entrada 2	Salida Esperada
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1

- La función lógica OR (O Inclusivo) con las siguientes entradas y salidas esperadas:

Entrada 1	Entrada 2	Salida Esperada
-1	-1	-1
-1	1	1
1	-1	1
1	1	1

- La función lógica XOR (O Exclusivo) con las siguientes entradas y salidas esperadas:

Entrada 1	Entrada 2	Salida Esperada
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1

Es necesaria una interfaz gráfica que muestre como la red va entrenando, incluyendo las entradas y una distinción de cuales son de cada clase (positivas y negativas), incluyendo el error actual. Puedes usar el lenguaje de programación de tu preferencia.

Actividades

1. En el caso de las funciones AND y de OR, analiza: ¿que otras entradas nos darían un resultado positivo y cuales un resultado negativo?
2. En el caso de la función XOR, ¿se cumplió el objetivo de separar ambas clases? ¿Es posible aproximar esta función con una red perceptrón multicapa?
3. Describe que cambios deberías hacer en tu código en caso quisieras entrenar una red con dos capas ocultas que debe separar elementos en 3 clases distintas. Señala las partes importantes.

Desarrollo y Entrega

- El trabajo debe ser desarrollado en la sesión de laboratorio.
- Se debe entregar digitalmente (en un PDF vía email de preferencia) un informe conteniendo el desarrollo de todas las actividades, imágenes de los gráficos y los códigos implementados.
- Plazo de entrega del informe: 24 de Setiembre del 2018.

Cuadro 1: Rúbrica de Evaluación - Práctica 6

Criterio	Deficiente (25%)	Regular (50%)	Bueno (75%)	Excelente (100%)	Total de Puntos
Modelado de Solución y Generación de Estado Inicial	No existe un modelado de la técnica o solución programada, o el estado inicial de la Red Neuronal está definido de forma clara.	Se han definido los elementos principales de la Red Neuronal y del proceso a ejecutar, pero no son completamente funcionales. El Estado Inicial no siempre se genera correctamente.	Se han definido los elementos de la Red Neuronal, y estos tienen relación a la resolución del problema que se va a realizar. El Estado Inicial se genera correctamente.	Se han definido todos los elementos de la Red Neuronal y se ha hecho una relación exacta con la resolución del problema a resolver. Cada componente este correctamente definido y el Estado Inicial de la Red se genera correctamente siempre.	5.5
Ejecución de la Técnica y Código Fuente	No existe código fuente, no es ejecutable o no se relaciona con el problema o la solución propuestos.	Existe código fuente ejecutable que tiene algunas nociones de los requerimientos del problema, pero no llega a resolver del todo el problema propuesto.	Existe código fuente ejecutable que cubre los requerimientos del problema, ejecuta la técnica pedida y muestra algún tipo de resultados.	Existe código fuente ejecutable y fácilmente legible que cubre los requerimientos del problema, ejecuta la técnica pedida y muestra resultados de acuerdo a lo solicitado en la práctica.	4.5
Obtención de Resultados y Visualización (Act. 1)	No hay resultados visibles, o sólo se ha mostrado el proceso de ejecución y no los resultados obtenidos.	Hay muestra del proceso de ejecución y de los resultados obtenidos, pero estos no son claros o su visualización no se relaciona a lo pedido en las actividades.	Hay muestra del proceso de ejecución y de los resultados según el formato solicitado.	Hay muestra del proceso de ejecución y de los resultados según el formato solicitado, existiendo además una breve discusión sobre los mismos.	4
Análisis Comparativo de Resultados (Act. 2 y 3)	No existe un análisis de los resultados obtenidos, o este no está documentado apropiadamente.	Existe un registro de los resultados obtenidos y una comparación entre los mismos, pero no se hace un análisis con mayor profundidad.	Existe un registro de los resultados obtenidos y un análisis entre los mismos, indicando similitudes y diferencias.	Existe un registro de los resultados obtenidos, y un análisis entre los mismos que indica similitudes, diferencias y el porqué de los resultados obtenidos, indicando también posibilidades de mejora.	6