

Redes Neuronales

Gonzalo V. Castiglione, Alan E. Karpovsky, Martín Sturla
Estudiantes Instituto Tecnológico de Buenos Aires (ITBA)

12 de Abril de 2012

Entrega Preliminar 2 - Informe

Resumen—El presente informe busca analizar y comparar distintas implementaciones y arquitecturas de redes neuronales con aprendizaje supervisado que resuelvan el problema del *AND* y *OR* lógico para N bits con $2 \leq N \leq 5$ a través del uso de tres variantes de funciones de transferencia.

Palabras clave—perceptrón, función de transferencia, red neuronal, aprendizaje supervisado, conjunto de entrenamiento, conjunto de testeo

I. INTRODUCCIÓN

LAS redes neuronales son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales y los seres humanos. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Cabe destacarse que las redes neuronales no dejan de ser un modelo y es debido a esto que no tienen en cuenta muchas de las funciones y cualidades del cerebro humano.

Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por dos funciones:

- **Función de excitación:** Por lo general consiste en la sumatoria de cada entrada multiplicada por el peso de su interconexión (*suma pesada de las entradas*). Si el peso de una conexión es positivo, la misma se denomina excitatoria; si es negativo, se denomina inhibitoria.
- **Función de transferencia:** Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas son la *función sigmoidea* (para obtener valores en el intervalo $[0, 1]$) y la *tangente hiperbólica* (para obtener valores en el intervalo $[-1, 1]$).

II. DESARROLLO

A. Modelado del problema

Se representó la red neuronal como una **matriz de pesos**. Cada neurona es una fila de pesos, cada capa de neuronas es una matriz de pesos, la red neuronal, por

consiguiente, es un vector de matrices. Lo interesante es que hallar el valor de la red neuronal con un cierto input se reduce a multiplicar el vector input por cada una de estas matrices. A su vez esta implementación facilita el manejo de errores.

Para las unidades escalón el error está definido como la **Distancia Hamming Levenshtein**, para los otros tipos de unidades se decidió utilizar el **error cuadrático medio**.

B. Features

Se implementaron tres tipos de funciones para modificar la variable η (*learn rate*): La primera de ellas es a valor **constante**, es decir que η nunca es modificada, la segunda es **annealed** que reduce η exponencialmente y por último se tiene un *learning rate* **dinámico** que lo que hace es ver si el error se reduce consistentemente y, en tal caso, incrementa el η sin superar un valor máximo preestablecido (0,5), caso contrario, η es reducida exponencialmente.

C. Ejecución

La implementación considera la llamada a una única función *main* a la que se le pasan los siguientes parámetros de configuración: operación lógica (AND / OR), cantidad de bits (entre 2 y 5), cantidad de épocas, eta (η), tipo de aprendizaje *constant*, *annealed*, *dynamic* en ese orden. Por ejemplo la siguiente llamada:

```
main('AND', N, 500, 'SIGMOID', 0.02, 'COSNTANT')
```

Le pasará al perceptrón simple el problema del AND lógico de 3 bits, con la función de transferencia simoidea, $\eta = 0,02$, tipo de aprendizaje constante y correrá por 500 épocas.

Es de interés destacar que, entre época y época, se decidió mezclar los patrones del conjunto de entrenamiento.

III. RESULTADOS

A. AND lógico

Como se puede apreciar en las tablas de la sección **Anexo A**, utilizando unidades escalón, el AND lógico tarda muy pocas épocas en presentar un error menor que

10^{-3} . Es interesante compararlo con los resultados obtenidos al utilizar las funciones de transferencia lineales y sigmóideas: fijando el η y el tipo de aprendizaje, se observa que para estas últimas dos funciones de transferencia se requieren, en el caso de un *AND* 3-ario, más de 300 o 400 épocas para alcanzar dicho cometido ($E < 10^{-3}$).

IV. CONCLUSIÓN

LUEGO de realizar numerosas pruebas variando los parámetros de configuración de la red neuronal, podemos concluir que hemos tenido mejores resultados mejores resultados utilizando $\beta = 1$ y seteando η con valores altos. Esto puede deberse posiblemente a la naturaleza planar del problema.

ANEXO A: TABLAS

Bits	Eta η	Función de activación	Épocas	Error de aprendizaje
2	a	Escalón	a	d
2	b	Lineal	b	e
2	c	Sigmóidea	c	f
3	a	Escalón	a	d
3	b	Lineal	b	e
3	c	Sigmóidea	c	f
4	a	Escalón	a	d
4	b	Lineal	b	e
4	c	Sigmóidea	c	f
5	a	Escalón	a	d
5	b	Lineal	b	e
5	c	Sigmóidea	c	f

TABLE I

COMPARACIÓN DE RED PARA AND LÓGICO CON TIPO DE APRENDIZAJE CONSTANTE

ANEXO B: GRÁFICOS