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 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. 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.

II. Desarrollo

A. Modelado del problema

Se representó la red neuronal como una **matriz de pesos**. Cada neurona es una columna de pesos, cada capa de neuronas es una matriz de pesos, la red neuronal, por consiguiente, es un vector de matrices (en particular un perceptrón simple es solo una matriz). Lo interesante es que hallar el valor de la red neuronal con un cierto input se reduce a multiplicar el vector entrada por cada una de estas matrices. Cabe destacar que al vector de entrada se le agrega siempre un -1 como último valor para el sesgo. 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**, , la segunda es ***annealed*** que reduce η exponencialmente y por último se tiene un *learning rate* **adaptative** que modifica η en función de los últimos errores obtenidos. El crecimiento es aritmético, con cota en $\eta = 0{,}5$, la reducción exponencial.

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 órden. Por ejemplo la siguiente llamada:

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

Le pasará al perceptrón simple el problema del AND lógico de N bits, con la función de transferencia sigmoidea, $\eta = 0{,}02$, tipo de aprendizaje constante y correrá por 500 épocas.
Cabe destacar que el conjunto de entrenamiento está compuesto por todos los casos posibles (al igual que el de testeo) y en cada época se altera el orden de dicho conjunto.

III. Resultados

A modo de ejemplo, en la sección ***Anexo A*** se puede observar una tabla comparativa entre los distintos tipos de funciones de transferencia para un η pequeño, con tipo de aprendizaje constante. Se decidió duplicar la cantidad de épocas tomadas inicialmente a medida que se incrementaba la cantidad de *inputs*.

Como se puede apreciar en las tablas, las unidades escalón probaron ser las más rápidas en reducir el error.

IV. Conclusión

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

Al oservar los gráficos de error obtenidos al realizar pruebas con funciones sigmoideas y lineales, notamos que el error tiende a cero, pero lo hace más rápido utilizando el *eta* en modo *adaptative* . Este fenómeno puede ser observado en el ***Anexo B*** donde se compara el error obtenido con un η constante versus un η adaptativo.

12 de Abril, 2012.

Anexo A: Tablas

AND

Bits	Eta η	Función de activación	Épocas	Error de aprendizaje
2	0.02	Escalón	50	0
2	0.02	Lineal	50	0.287
2	0.02	Sigmóidea	50	0.571
3	0.02	Escalón	100	0
3	0.02	Lineal	100	0.256
3	0.02	Sigmóidea	100	0.581
4	0.02	Escalón	200	0
4	0.02	Lineal	200	0.181
4	0.02	Sigmóidea	200	0.523
5	0.02	Escalón	400	0
5	0.02	Lineal	400	0.089
5	0.02	Sigmóidea	400	0.421

TABLE I
Comparación de errores de la salida de la red para AND lógico con tipo de aprendizaje constante

OR

Bits	Eta η	Función de activación	Épocas	Error de aprendizaje
2	0.02	Escalón	50	0
2	0.02	Lineal	50	0.310
2	0.02	Sigmóidea	50	0.545
3	0.02	Escalón	100	0
3	0.02	Lineal	100	0.258
3	0.02	Sigmóidea	100	0.568
4	0.02	Escalón	200	0
4	0.02	Lineal	200	0.181
4	0.02	Sigmóidea	200	0.520
5	0.02	Escalón	400	0
5	0.02	Lineal	400	0.089
5	0.02	Sigmóidea	400	0.425

TABLE II
Comparación de errores de la salida de la red para OR lógico con tipo de aprendizaje constante

Anexo B: Gráficos

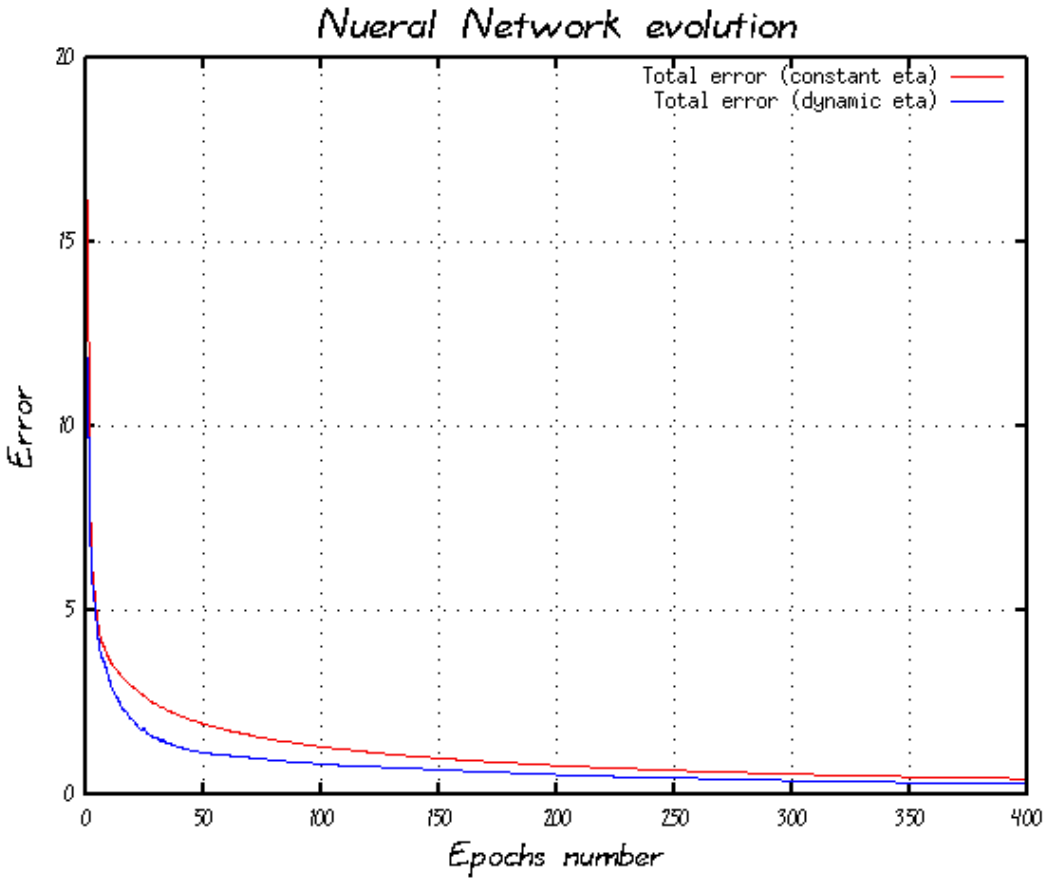

Figura 1: Curva de error con eta constante vs. eta dinámico