



SISTEMAS DE INTELIGENCIA ARTIFICIAL

INFORME PRELIMINAR N°3

Red neuronal con aprendizaje supervisado

Autores:

Pablo Ballesty - 49359

Nicolás Magni - 48008

Guillermo Liss - 49282

18 de abril de 2012

Resumen

El objetivo del presente documento es detallar el diseño e implementación de una red neuronal multicapa utilizando aprendizaje supervisado para resolver las operaciones lógicas de *paridad* y *simetría* para N bits de entrada con $2 \leq N \leq 5$.

1. Desarrollo

1.1. Decisiones de implementación

A diferencia de la entrega anterior, ahora se crean N redes con $2 \leq N \leq 5$ para cada método. Luego de correr los entrenamientos para las redes, los pesos aprendidos quedan persistidos en archivos de texto, que luego se utilizarán para evaluar cada método con la entrada que ingrese el usuario.

1.2. Funciones de transferencia

Se utilizan 2 funciones de transferencia. A continuación se definen las mismas

- **sigmoid**

$$g(x) = \tanh(x) \quad (1)$$

$$g'(x) = 1 - \tanh(x)^2 \quad (2)$$

- **lineal**

$$g(x) = x/4 \quad (3)$$

$$g'(x) = 1/4 \quad (4)$$

1.3. Entrenamiento

Para entrenar los pesos de las distintas redes, se utiliza *backpropagation* con condición de corte por mínimo error cuadrático aceptable. Al algoritmo de aprendizaje se le realizaron las siguientes mejoras

- **η adaptativo:** Dependiendo de la diferencia entre el error cuadrático de la última época corrida, con el error cuadrático de la anterior, se realizan pequeños cambios sobre η . Si esta diferencia es positiva se disminuye el valor de η en un porcentaje b , y si ocurren diez diferencias consecutivas negativas se aumenta el valor de η en un valor constante a . En este caso, $b = 0.1$ y $a = 0.01$. El fin de estos cambios es que cuando el error aumenta, realizar cambios pequeños sobre los pesos tratando de salir del aumento, mientras que cuando el error va descendiendo realizar cambios más grandes con el objetivo de acelerar el aprendizaje.
- **Recomienzo:** el entrenamiento permite fijar una cantidad máxima de épocas a correr, y en caso de no llegar a un error cuadrático menor al esperado, se comienza nuevamente, tomando una nueva inicialización aleatoria. El objetivo de esta mejora es evitar que el algoritmo se quede siempre en un mínimo local.

2. Resultados

Para probar el comportamiento del algoritmo de aprendizaje se entrenaron distintas arquitecturas hasta cumplir un error cuadrático medio menor a $err = 0.01$, para las operaciones lógicas de paridad y simetría, con entradas de $N = 5$ bits. Se variaron, para la arquitectura de cada operación lógica, el valor $\alpha = \text{cantidad de neuronas en la capa oculta}$ y η el *learning rate*.

Se puede ver en el apéndice, en los cuadros A.1 y A.2 para las operaciones lógicas de simetría y paridad respectivamente, las comparaciones entre las distintas arquitecturas de redes neuronales.

3. Conclusiones

A. Tabla de resultados y gráficos

A.1. Simetría

α	η	# épocas
4	0.5	114
	0.25	87
	0.1	159
3	0.5	16
	0.25	3456
	0.1	106
2	0.5	61
	0.25	4628
	0.1	∞

A.2. Paridad

α	η	# épocas
5	0.5	6729
	0.25	4966
	0.1	655
6	0.5	66
	0.25	8120
	0.1	481
7	0.5	3222
	0.25	18045
	0.1	516



