

Sistemas de Inteligencia Artificial

ALGORITMOS GENÉTICOS

Trabajo Práctico Especial 4

Grupo 1

Domingues, Matías

Fontanella De Santis, Teresa

Martinez Correa, Facundo

Objetivo

Implementar un motor de algoritmos genéticos,
con capacidad de obtener los pesos
de una red neuronal multicapa,
para resolver la función asignada en el TP 2.

Red Neuronal Utilizada

Una capa oculta de 75 neuronas.

$$\eta = 0.01,$$

$$\varepsilon = 0.001,$$

la función de activación de todas las capas es $\tanh(x)$.

Algoritmo Genético

- 1) Población inicial de N individuos y luego backpropagation con M épocas.
- 2) Calcular el fitness de cada individuo y seleccionar k de ellos.
- 3) Por cada par k_1 y k_2 de los k “padres”, existe una probabilidad p_{cruce} con la que ocurre la cruce, que origina dos hijos. Al final de esta iteración, se obtienen k “hijos”.
- 4) Por cada hijo, existe una posibilidad $p_{mutación}$ de que éste mute. Luego, éste puede aplicar backpropagation con una probabilidad $p_{backpropagation}$.
- 5) Seleccionar N_1 individuos de los padres, y N_2 de los hijos ($N = N_1 + N_2$), y con eso se forma la nueva generación poblacional.
- 6) Si no se cumple la condición de corte, se vuelve al paso 2) .

Representación de los individuos

Un individuo es una lista de matrices, donde cada matriz representa los pesos de conexión entre las neuronas de una capa con la siguiente, considerando también los umbrales.

Cada peso de la matriz en la capa n w_{ij} , es la conexión entre ξ_i^{n+1} y ξ_j^n (donde ξ_i^n es i -ésima neurona de la capa n). Cada uno de los pesos en cuestión es un gen, cuyos valores (o alelos) son números reales ($-\infty < w < \infty, w \in \mathbb{R}$).

Esta estructura es fiel.

Cruce (o crossover)

La estructura se maneja como si fuera un vector formado por la concatenación de las filas de todas las matrices. Esto implica que

$$0 \leq l_1, l_2 (i=0, n-1) \leq r_i c_i - 1$$

siendo l_1 y l_2 los locus para las cruza (dependiendo de cuál sea se usa uno o los dos), r_i y c_i la cantidad de filas y columnas de la matriz de la capa i , respectivamente.

Mutación

Se elige uno o dos locus de la misma manera que con la cruce. Siendo w_{nuevo} el valor del gen mutado, y $w_{anterior}$ su valor antes de la mutación, entonces:

$$w_{nuevo} = w_{anterior} + \text{urand}(-0.5, 0.5).$$

Función de Fitness

- ❖ Favorecer aprendizaje

- ❖ Evitar “sobreentrenamiento”

- ❖ $f(x) = \frac{1}{|E|}$, siendo E el error cuadrático medio al pasar todos los patrones por la red neuronal x .

Ejemplos - Algoritmo 1

Método de selección	Método de reemplazo	Generaciones	Mejor fitness	Fitness promedio	Error final
Elite	Torneo probabilístico	1000	9.7644	1.5537	66.4566
Elite	Torneo probabilístico	5000	4.4766	1.2120	179.172
Ruleta	Boltzmann	1000	216.51	45.758	132.431
Torneo probabilístico	Ruleta	1000	10.349	4.1581	55.0433

Ejemplos- Algoritmo 1

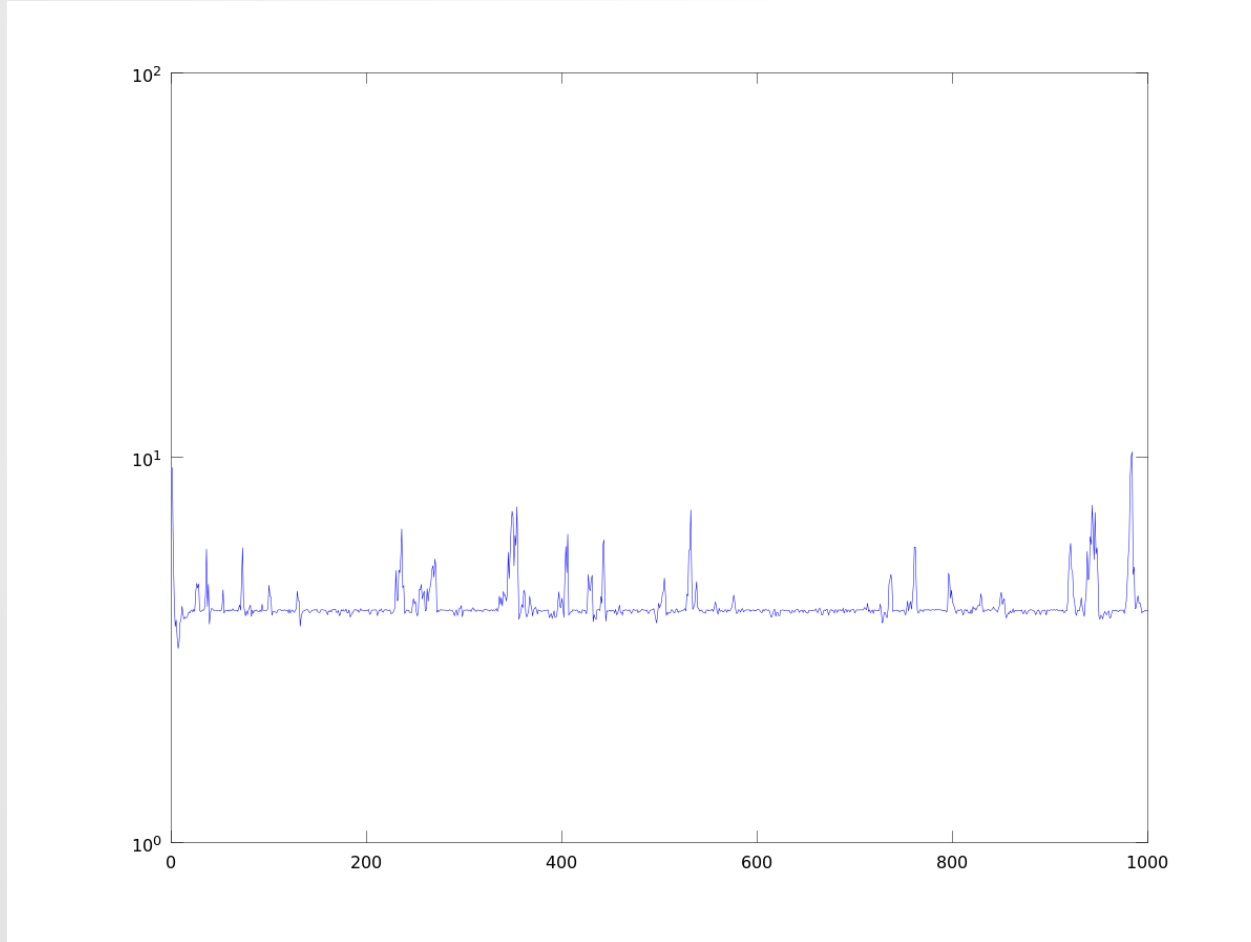


Gráfico 1 - Fitness de los individuos con torneo probabilístico-ruleta (reemplazo 1)

Conclusiones

- ❖ Combinaciones que más consideran al azar son las que obtienen mejores resultados.
- ❖ El hecho de utilizar el operador backpropagation con una probabilidad bastante alta, hizo que el algoritmo genético funcione mejor que el algoritmo del trabajo predecesor. Esto puede atribuirse al hecho de que dicho operador refina los pesos de los individuos, logrando “desarrollarlos” para que, aquellos que podrían ser más aptos en un futuro, se desarrollen y puedan ser elegidos con mayor probabilidad.