

# RNA-TRAB11

Cecília Carneiro e Silva

23/05/2018

## 1 Redes concorrentes

O objetivo deste exemplo é construir uma **Rede Neural Competitiva** para formar uma **Look-up table** para a função:  $y = 1/x$ . A quantidade de cluster será variada dos exemplos. A entrada é uma lista, em ordem randômica, de pares '(x y)' que obedecem a função acima.

- Entrada: (range 1 10 0.001), 9001 entradas  
'((1 1.0) ... (9.999999999999897 0.100000000000000103))

### 1.1 Considerações

- Pesos iniciais gerados aleatoriamente.
- Atualização dos pesos:

#### - Atualização dos pesos

$$W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t))$$

Taxa de  
Aprendizagem:

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \quad t = 1, 2, 3, \dots$$

Taxa de  
Aprendizagem  
pela distância:

$$\Theta(t) = \exp\left(-\frac{dist^2}{2\sigma^2(t)}\right) \quad t = 1, 2, 3, \dots$$

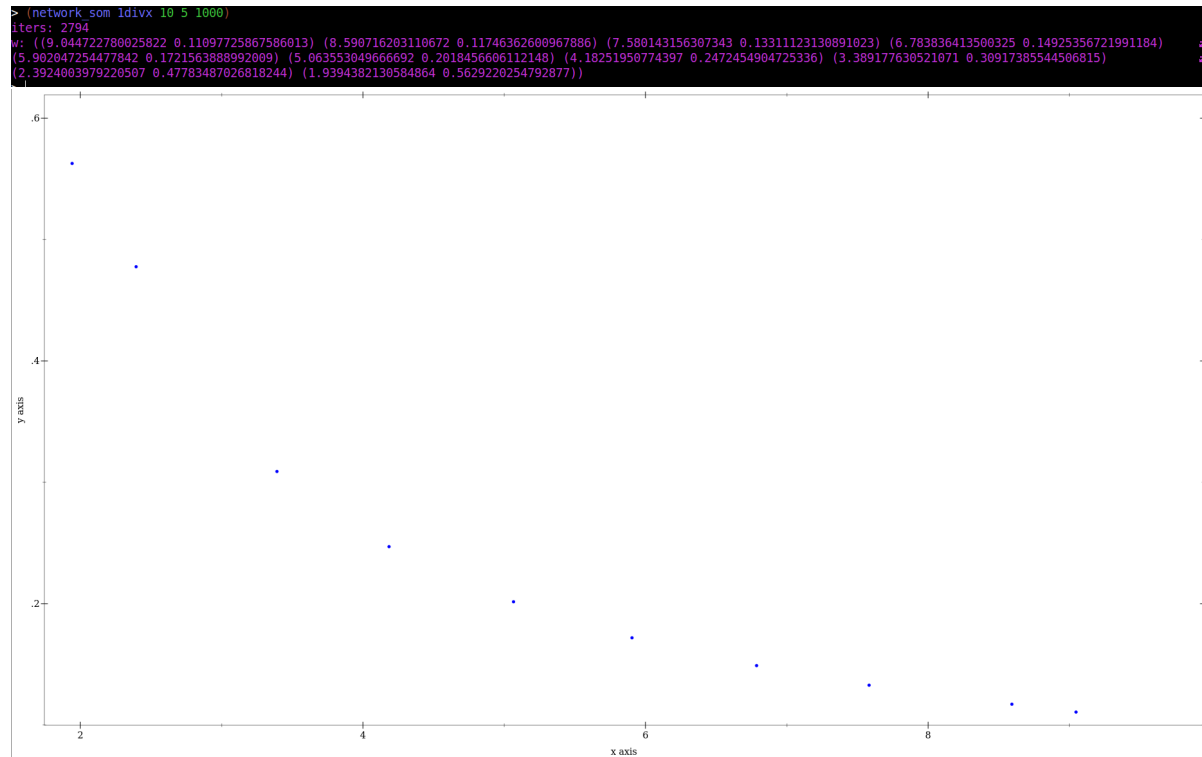
- Parada: quando  $\Delta w$  for melhor do que a precisão ( $1e-10$ ).

### 1.1.1 Testes

```
;;(define ldivx (shuffle (ldivx_fun 10)))  
> (network_som input_list n_output_x radius_max n_iterations_max [n_output_y 1])
```

### 1.2 Teste 1

- Cluster: 10
- Raio máximo: 5
- Iterações-cálculos: 1000



### 1.3 Teste 2

- Cluster: 12
- Raio máximo: 5
- Iterações-cálculos: 1000

```
> (network_som ldivx 12 5 1000)  
iters: 2784  
w: ((1.7873808947355532 0.6006899082191296) (2.168278379010437 0.5156162652969469) (3.003902538478849 0.34710609700570577) (3.671525456872644 0.280865331085508) 2  
(4.410153484269562 0.23138435374442312) (5.121591546651027 0.19822411468986254) (5.846541814044075 0.17302635216591386) (6.557656440184226 0.15387540860906487) 2  
(7.296935824719853 0.13808319017351278) (7.967180566775154 0.12623271122531074) (8.815802957538647 0.1141212025743838) (9.197195619003294 0.10900723414845048))
```

