Digit Recognition by The Random Neural Network using Supervised Learning

M. Mokhtari

E.H.E.I., Université René Descartes, 45 rue des Saints Pères, 75006 Paris, France

Abstract

The Random Neural Network model with positive and negative neurons is studied as an autoassociative memory for pattern recognition. A neuron can excite its neighbours if it emits positive signals or can inhibit them by emission of negative signals. There is accumulation of positive signals at positive neurons and negative signals at negative neurons. Let $k(i)$ be the vector of neuron potentials at time t . If all the steady state excitation probabilities q_i are such that $0 < q_i < 1$, the stationary probability distribution of the network's state is expressed by :

$$p(k) = \prod_{i=1}^n (1 - q_i) q_i^{k_i} \quad \text{where} \quad q_i = \gamma^+(i) / [\gamma(i) + \gamma^+(i)] \text{ if } i \in P, \quad q_i = \gamma(i) / [\gamma(i) + \gamma^+(i)] \text{ if } i \in N \quad (1)$$

$$\gamma^+(i) = \Lambda(i) + \sum_{j \in P} q_j r(j) p^+(j, i) + \sum_{j \in N} q_j r(j) p^-(j, i) \quad (2)$$

$$\gamma(i) = \lambda(i) + \sum_{j \in P} q_j r(j) p^-(j, i) + \sum_{j \in N} q_j r(j) p^+(j, i) \quad (3)$$

with

P (respectively N) : set of positive (respectively negative) neurons

$\Lambda(i)$ (respectively $\lambda(i)$) : external arrival rate of positive (respectively negative) signals to neuron i

$r(i)$: emission rate of signals by neuron i

$p^+(i, j)$ (respectively $p^-(i, j)$): probability that i emits a positive (respectively negative) signal to j , if $i \in P$

$p^+(i, j)$ (respectively $p^-(i, j)$): probability that i emits a negative (respectively positive) signal to j , if $i \in N$.

Let us write: $w^+(i, j) = r(i) p^+(i, j) \geq 0$, $w^-(i, j) = r(i) p^-(i, j) \geq 0$, $r(i) = \sum_j [w^+(i, j) + w^-(i, j)]$. E. Gelenbe has presented a learning method based on gradient descent for the recurrent (i.e. general) random network model. The algorithm lets the network learn both n by n matrices $W^{+k} = \{w^{+k}(i, j)\}$ and $W^{-k} = \{w^{-k}(i, j)\}$ by computing for each k th input, a new value W^{+k} and W^{-k} which minimizes the cost function $E^k = 1/2 \sum_{i=1}^n a_i (q_i^k - d_i^k)^2$

where a_i is the contribution of neuron and $d_i^k \in [0, 1]$ correspond to its desired value. The rule for weight update is (4) : $w^k(u, v) = w^{k-1}(u, v) - \eta \sum_{i=1}^n a_i (q_i^k - d_i^k) [\partial q_i / \partial w(u, v)]^k$ where η is a constant and $w(u, v)$ is either $w^-(u, v)$ or $w^+(u, v)$.

We wanted to realize an auto-associative memory, which learns a set of K patterns and recognizes them even if they are noisy. We have taken 2 layers of n neurons. The output layer is a duplication of the input layer. When a pattern x^k is applied as input, it characterizes network neurons as positive or negative :

if $x_i^k = -1$ then $i \in N$, $\lambda(i) = \lambda^k(i)$, $\Lambda(i) = 0$; if $x_i^k = 1$ then $i \in P$, $\Lambda(i) = \Lambda^k(i)$, $\lambda(i) = 0$ where $i \in \text{Layer 1}$.

Let us denote by α_0 the storage threshold once the learning stage is finished: $\alpha_0 = \min_{1 \leq k \leq K} (\min_{1 \leq i \leq n} q_i^k)$.

The recognition of a pattern x is defined as:

(1) for $i \in \text{Layer 1}$, if $x_i = 1$ then $i \in P$ else $i \in N$;

(2) for $i \in \text{Layer 2}$, compute its q_i as a positive neuron (q_i^+) and a negative neuron (q_i^-) :

(3) compute the output y

if $q_i^+ > q_i^-$ then $i \in P \Rightarrow y_i = 1$ and $q_i = q_i^+$

if $q_i^- > q_i^+$ then $i \in N \Rightarrow y_i = -1$ and $q_i = q_i^-$

if $q_i^+ = q_i^-$ then $y_i = x_i$ and $q_i = q_i^+ = q_i^-$

(4) if $q_i \geq \alpha_0$ for $1 \leq i \leq n$, the recognition is finished meaning that $y (=x)$ is a stored pattern.

(5) else, x is still noisy. Reinject this output y to the system which then becomes a new input x . Go to (1)

(6) we stop x injection when the reinjection number is larger than n by the maximal supposed noise rate, because at worst there is only 1 correction per pass.

The recognition has been applied to the 10 decimal numerals coded on 64 components. We have corrupted the input patterns by a noise rate which reach 20%. The recognition performance is successful for 97%.