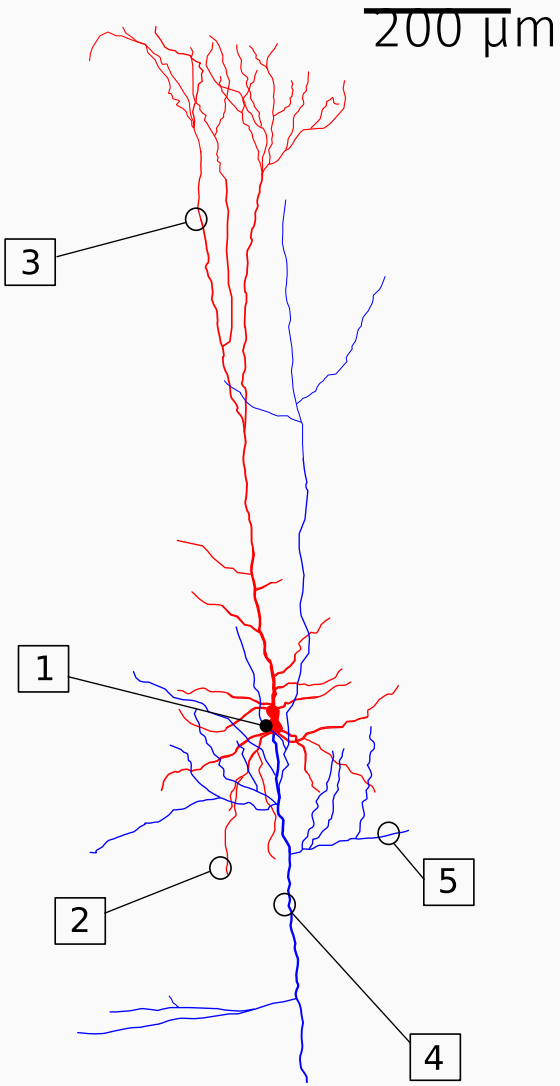
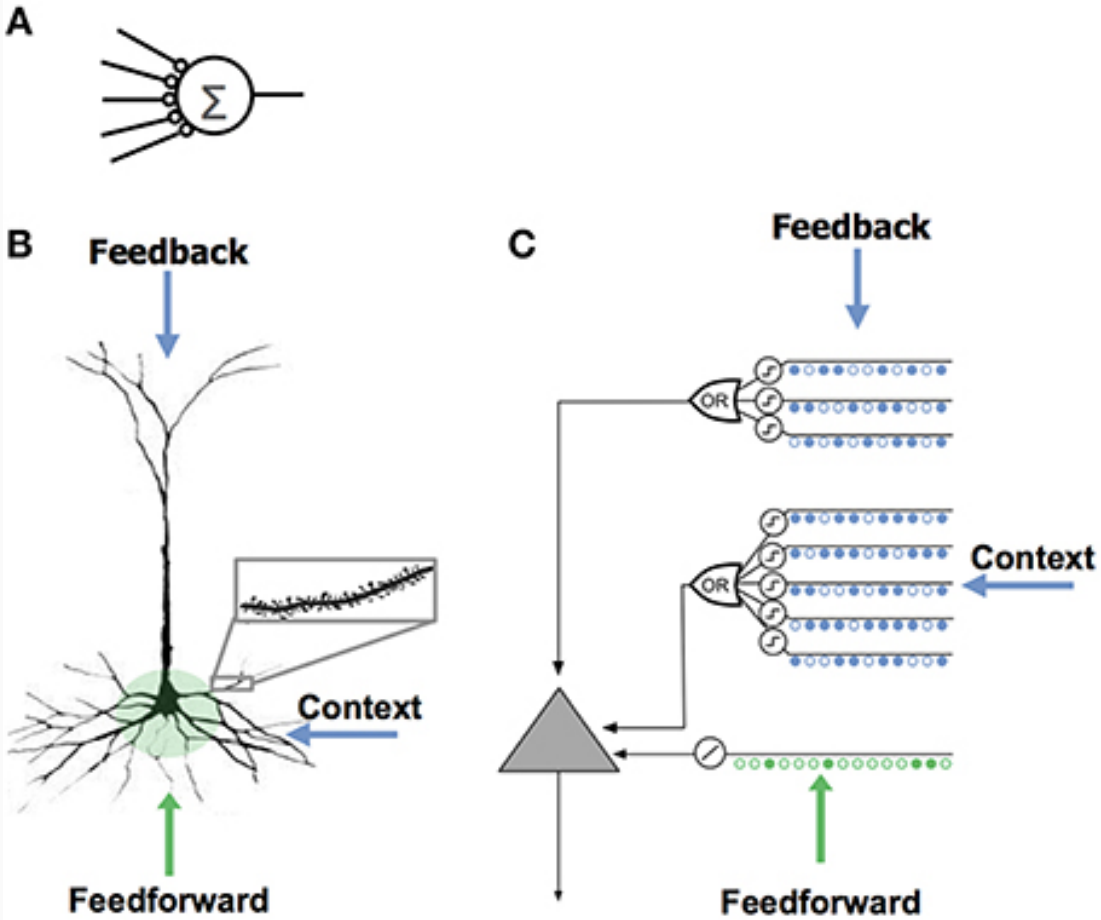
Υψηλού επιπέδου υλοποίηση των αλγορίθμων Hierarchical Temporal Memory σε Julia

Κωνσταντίνος Σαμαράς-Τσακίρης
Επιβλέπων καθηγητής: Νίκος Πιτσιάνης

13 Ιουνίου 2019

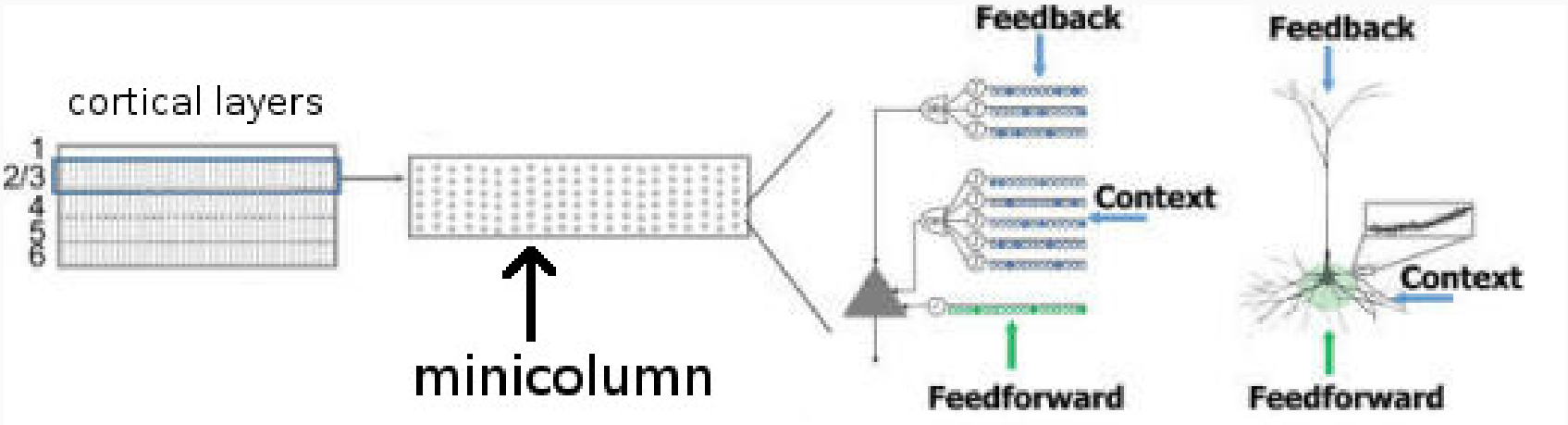
Νευρώνας





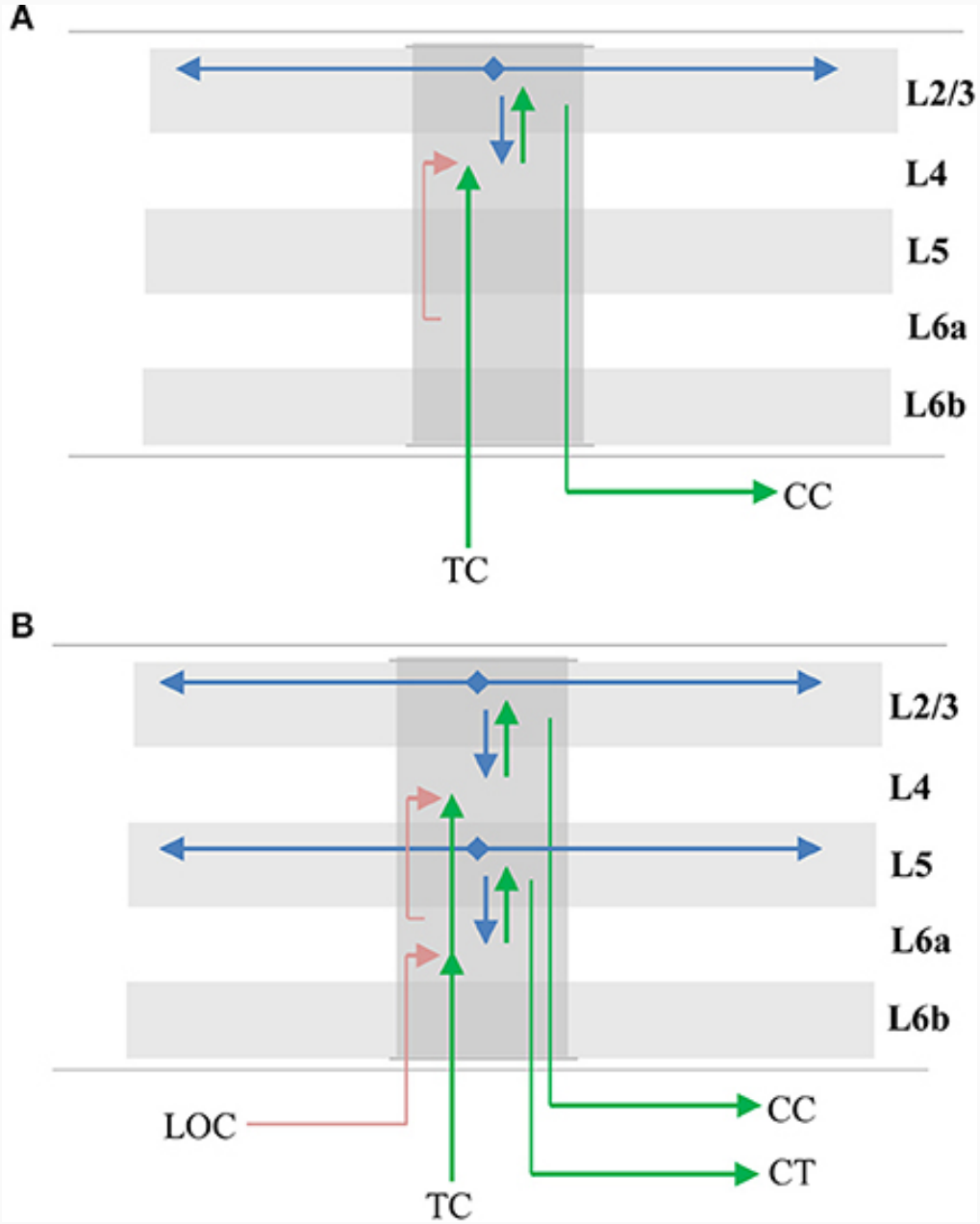
Πηγή: [3], [6]

Μικροστήλες



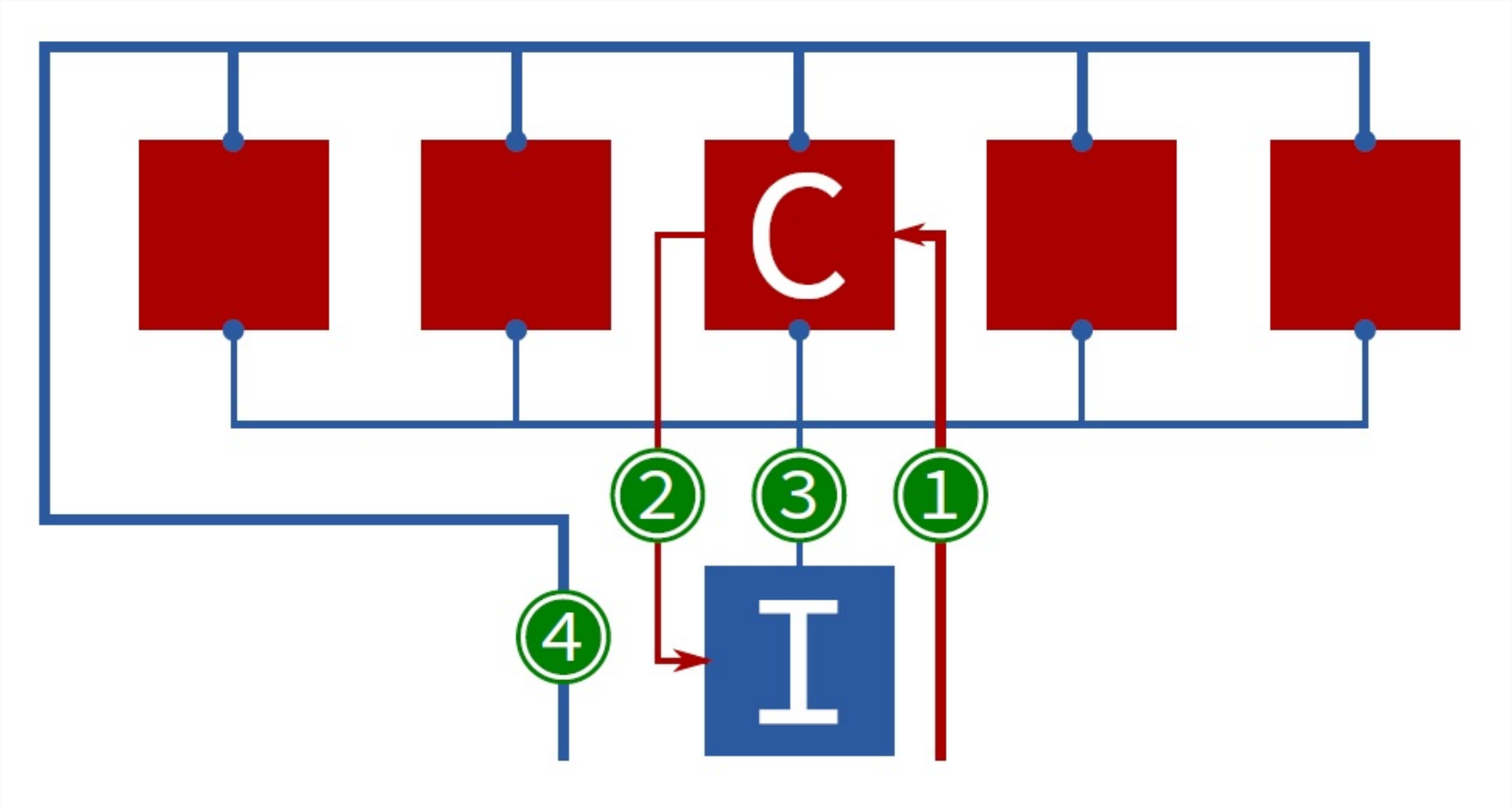
Πηγή: [2]

Φλοιικές στήλες



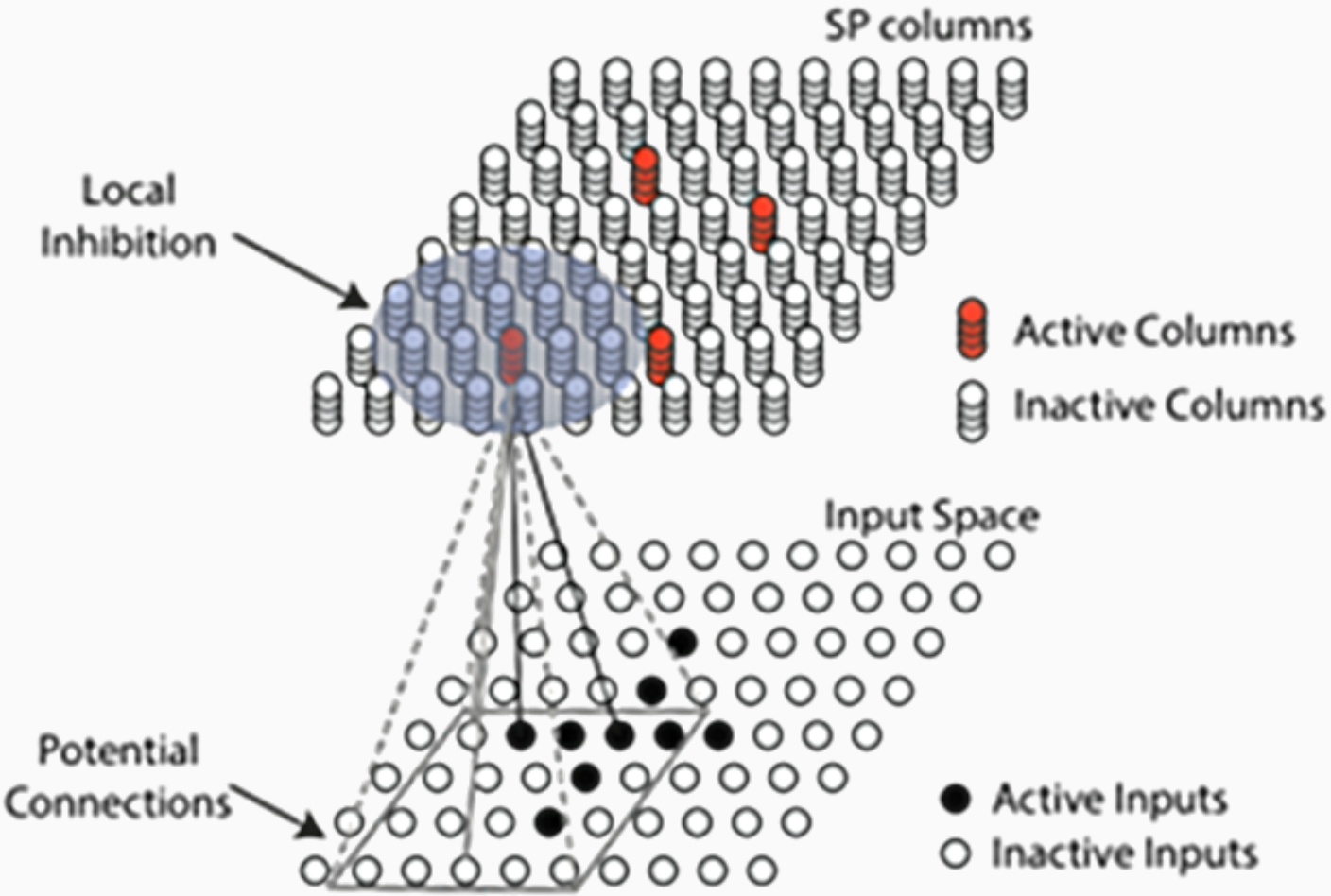
Πηγή: [5]

Αναστολή μεταξύ μικροστηλών



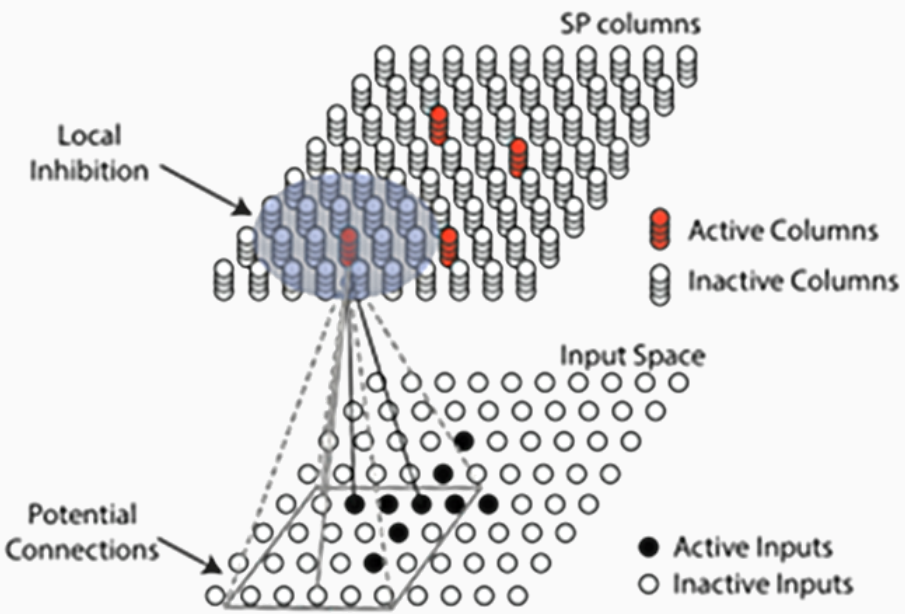
Πηγή: [1]

Χωρικός συγκεντρωτής



Πηγή: [4]

Χωρικός συγκεντρωτής

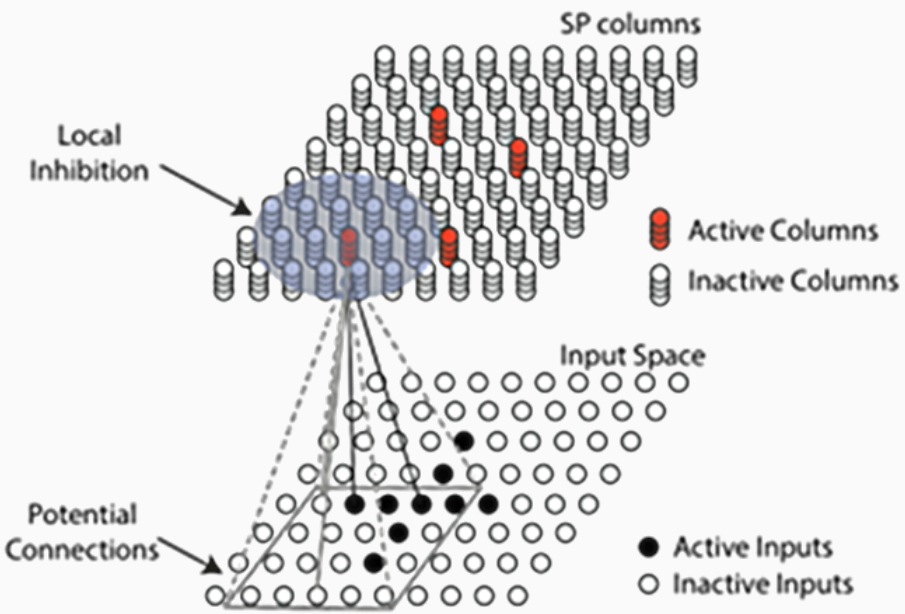


Βήματα

0. Αντιστοίχιση χώρων εισόδου/εξόδου, αρχικοποίηση εγγύς συνάψεων
1. Επικάλυψη μικροστηλών με συνδεδεμένες εξόδους
2. Παρώθηση
3. Τοπική αναστολή
4. Ενεργοποίηση μικροστηλών που νίκησαν
5. Εκμάθηση συνάψεων ενεργών μικροστηλών

Πηγή: [4]

Χωρικός συγκεντρωτής



Βήματα

0. Αντιστοίχιση χώρων εισόδου/εξόδου, αρχικοποίηση εγγύς συνάψεων
1. Επικάλυψη μικροστηλών με συνδεδεμένες εξόδους
2. Παρώθηση
3. Τοπική αναστολή
4. Ενεργοποίηση μικροστηλών που νίκησαν
5. Εκμάθηση συνάψεων ενεργών μικροστηλών

Πηγή: [4]

Στοιχεία Χωρικού Συγκεντρωτή

Αντιστοίχιση εισόδου/εξόδου: υπερκύβος

Δείκτης υπερκύβου

$$I(x_j; x_i^c, \gamma) = \textit{true} \iff x_j \in \text{hypercube}$$

x^c κέντρο υπερκύβου

γ ακτίνα υπερκύβου

Αντιστοίχιση εισόδου/εξόδου: υπερκύβος

Δείκτης υπερκύβου

$$I(x_j; x_i^c, \gamma) = \textit{true} \iff x_j \in \text{hypercube}$$

x^c κέντρο υπερκύβου

γ ακτίνα υπερκύβου

```
struct Hypercube{N}
  xᶜ::NTuple{N,Int}
  γ::Int
  sz::NTuple{N,Int}
  indices::CartesianIndices{N}
end
Hypercube(xᶜ,γ,sz)= Hypercube(xᶜ,γ,sz, start(xᶜ,γ,sz))
start(xᶜ,γ,sz)= CartesianIndices(map( (a,b)-> a:b,
                    max.(xᶜ .- γ, 1), min.(xᶜ .+ γ, sz) ))
Base.collect(hc::Hypercube)= map(c->c.I, collect(hc.indices))

jl> collect(Hypercube((1,1),1,(5,5)))
2×2 Array{Tuple{Int64,Int64},2}:
 (1, 1)  (1, 2)
 (2, 1)  (2, 2)
```

Αρχικοποίηση συνάψεων

Εν δυνάμει συνδέσεις

$$\Pi_i = \{j \mid I(x_j; x_i^c, \gamma) \wedge Z_{ij} < p\}$$

όπου $Z \in U(0, 1)$ τυχαίος αριθμός

Αρχικοποίηση συνάψεων

Εν δυνάμει συνδέσεις

$$\Pi_i = \{j \mid I(x_j; x_i^c, \gamma) \wedge Z_{ij} < p\}$$

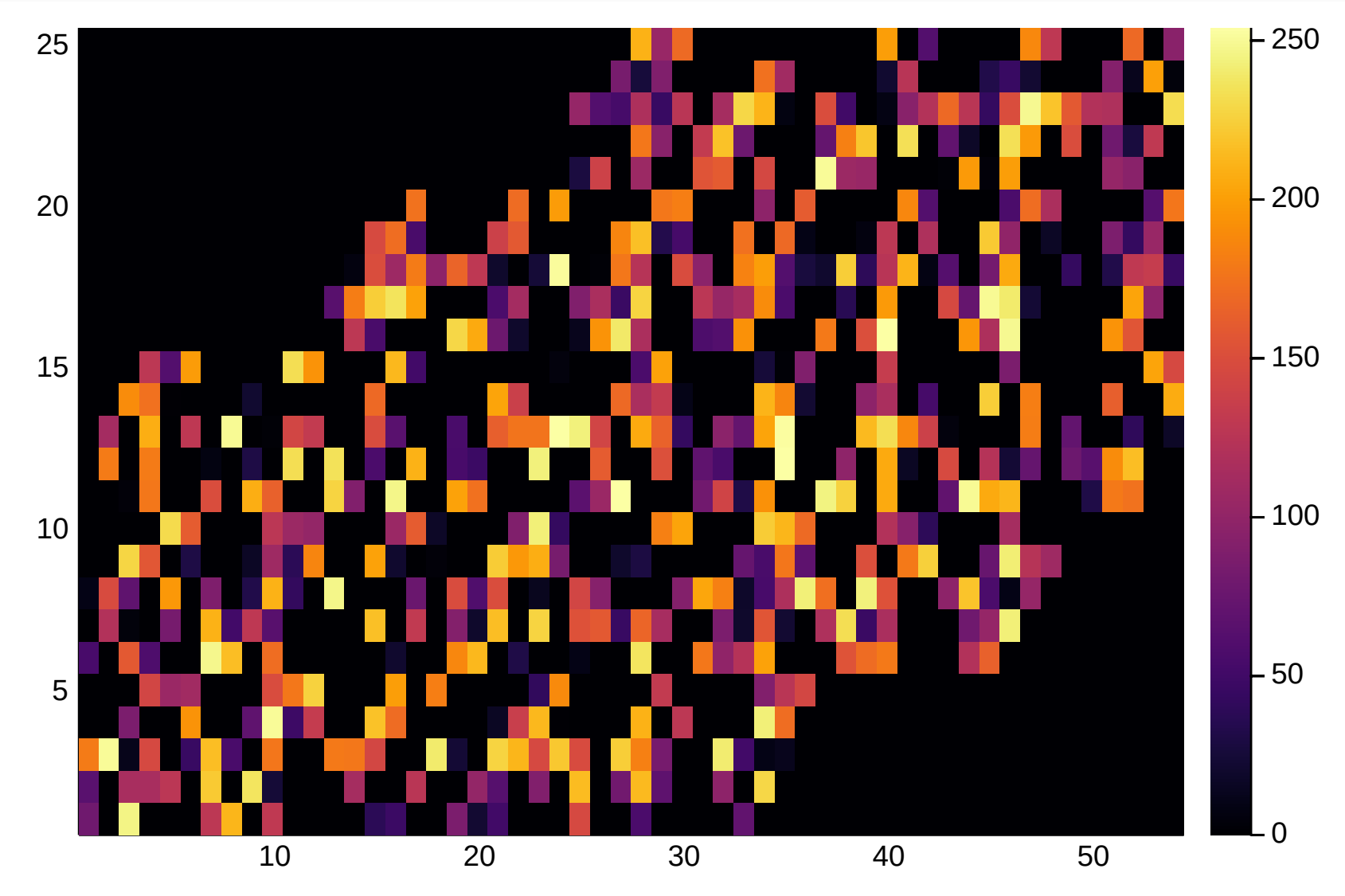
όπου $Z \in U(0,1)$ τυχαίος αριθμός

```
c2l_in= LinearIndices(sz_in)
c2l_sp= LinearIndices(sz_sp)

D_p= zeros($q, prod(sz_in),prod(sz_sp))
foreach(out_lattice()) do y_i
  # Linear indices from hypercube
  x= @>> y_i x^c x_i collect map(x->c2l_in[x...])
  D_p[x, c2l_sp[y_i...]]= permanences(@> y_i x^c x_i)
end

@> y_i x^c x_i == x_i(x^c(y_i))
```

Αρχικοποίηση συνάψεων



Επικάλυψη

Επικάλυψη μικροστηλών με συνδεδεμένες εισόδους

$$\mathbf{W} = \mathbf{D} \geq \theta_c$$

$$o = b\mathbf{W}\,z$$

$\mathbf{W}$ $[\ell_{in} \times \ell_{sp}]$ συνδεδεμένες συνάψεις

z $[\ell_{in}]$ είσοδος

b $[\ell_{sp}]$ παρώθηση

Επικάλυψη

Επικάλυψη μικροστηλών με συνδεδεμένες εισόδους

$$\mathbf{W} = \mathbf{D} \geq \theta_c$$

$$o = b\mathbf{W}\,z$$

$\mathbf{W}$ $[\ell_{in} \times \ell_{sp}]$ συνδεδεμένες συνάψεις

z $[\ell_{in}]$ είσοδος

b $[\ell_{sp}]$ παρώθηση

```
W_p()= D_p .≥ θ_permanence
o(z)= @> (b() .* W_p()'z) reshape(sz_sp)
```

Επικάλυψη

Επικάλυψη μικροστηλών με συνδεδεμένες εισόδους

$$\mathbf{W} = \mathbf{D} \geq \theta_c$$

$$o = b\mathbf{W}\,z$$

$\mathbf{W}$ $[\ell_{in} \times \ell_{sp}]$ συνδεδεμένες συνάψεις

z $[\ell_{in}]$ είσοδος

b $[\ell_{sp}]$ παρώθηση

```
W_p()= D_p .≥ θ_permanence
o(z)= @> (b() .* W_p()'z) reshape(sz_sp)
```

broadcasting

```
f(x::Int)= x+1;
jl> f.([1; 10; 100])
3-element Array{Int64,1}:
   2
  11
 101
```

Εκμάθηση συνάψεων

Κανόνας πλαστικότητας

$$\mathbf{\Delta D} = p^{+}(z \circ \mathbf{D} \circ c) - p^{-}(\neg z \circ \mathbf{D} \circ c$$

c ενεργοποίηση χωρικού συγκεντρωτή

Εκμάθηση συνάψεων

Κανόνας πλαστικότητας

$$\mathbf{\Delta D} = p^{+}(z \circ \mathbf{D} \circ c) - p^{-}(\neg z \circ \mathbf{D} \circ c$$

c ενεργοποίηση χωρικού συγκεντρωτή

Απλούστερος τρόπος

```
learn!(Dₚ,z,a)= begin
  Dₚ[z,a]  .= (Dₚ[z,a].>0) .* (Dₚ[z,a]   .⊕ p⁺)
  Dₚ[.!z,a].= (Dₚ[z,a].>0) .* (Dₚ[.!z,a] .⊖ p⁻)
end
```

Εκμάθηση συνάψεων

Κανόνας πλαστικότητας

$$\Delta \mathbf{D} = p^{+}(z \circ \mathbf{D} \circ c) - p^{-}(\neg z \circ \mathbf{D} \circ c$$

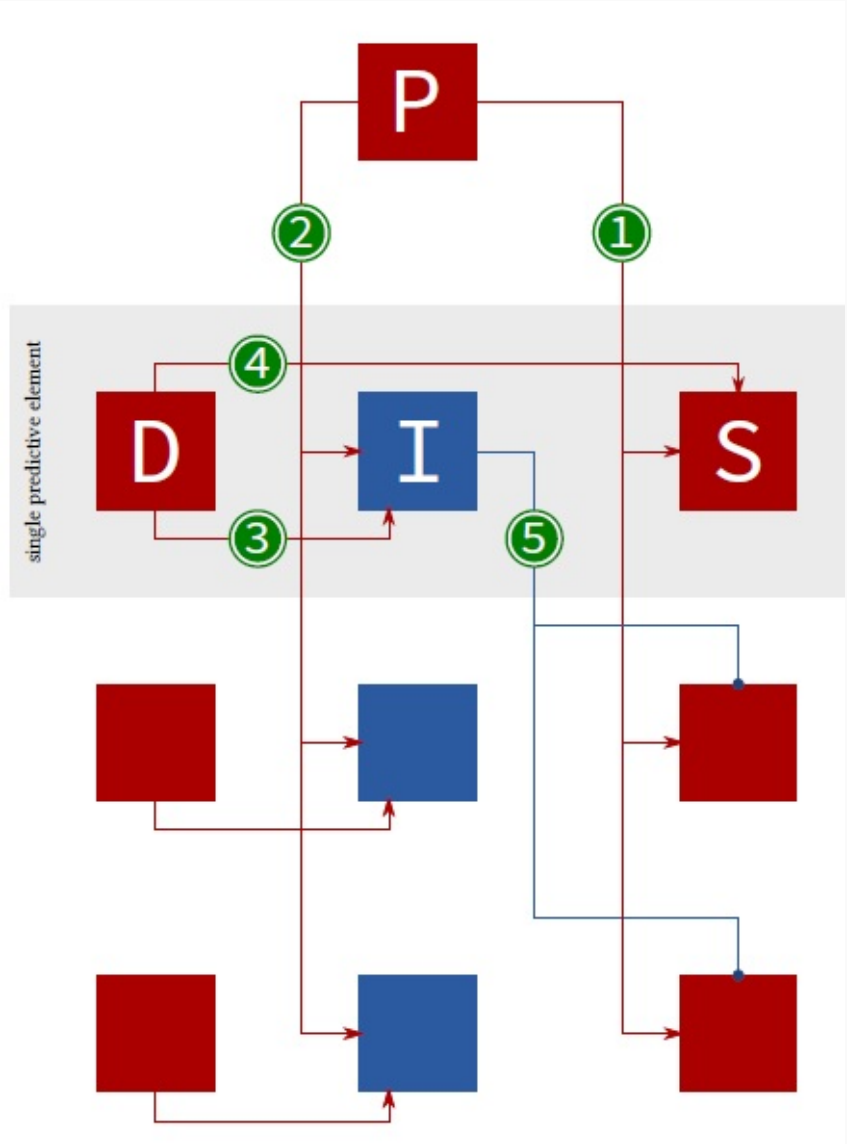
c ενεργοποίηση χωρικού συγκεντρωτή

Καλύτερα

```
learn!(Dₚ,z,a)= begin
  Dₚactive= @view Dₚ[:,a]   # the only elements we touch
  activeConn=   (Dₚactive .> 0) .&   z
  inactiveConn= (Dₚactive .> 0) .& .!z
  Dₚactive.= activeConn   .* (Dₚactive .⊕ p⁺) .+
             inactiveConn .* (Dₚactive .⊖ p⁻)
end
```

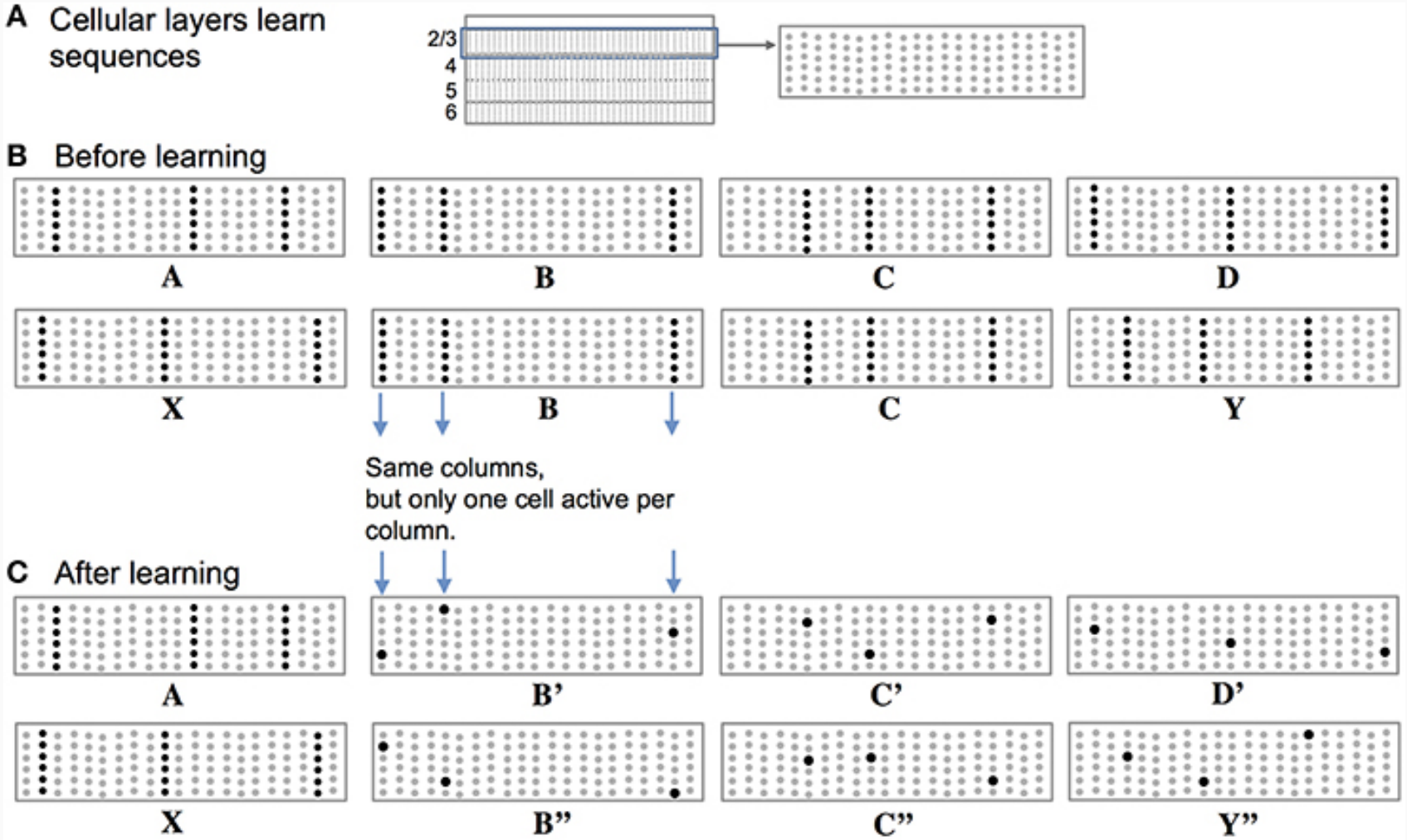
Χρονική μνήμη

Αναστολή εντός μικροστηλών



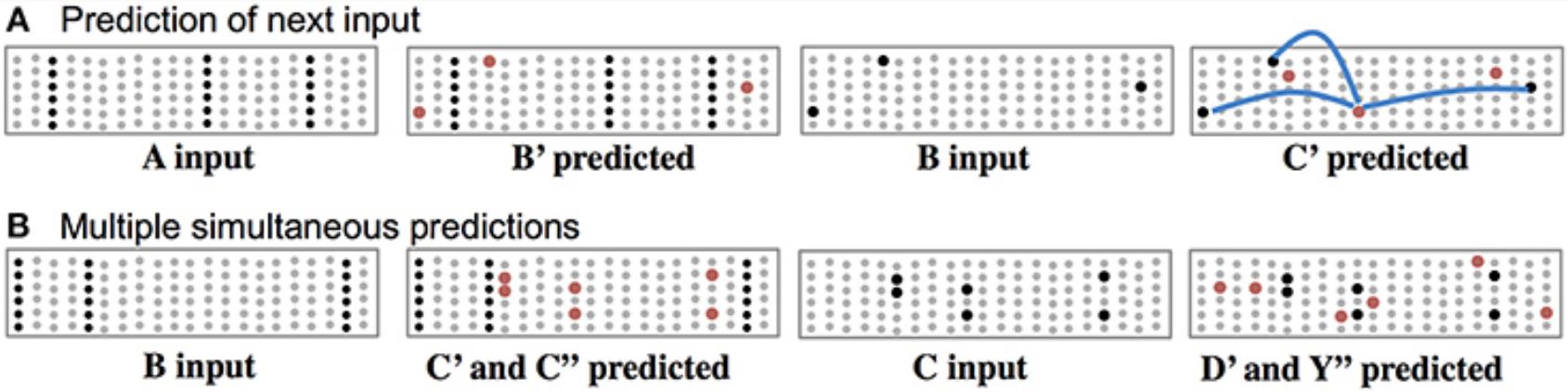
Πηγή: [1]

Μνήμη ακολουθιών



Πηγή: [3]

Μνήμη ακολουθιών



Βήματα

1. Ενεργοποίηση
2. Προσδοκία/πρόβλεψη
3. Εκμάθηση συνάψεων ενεργών νευρώνων

Μεταβλητές κατάστασης

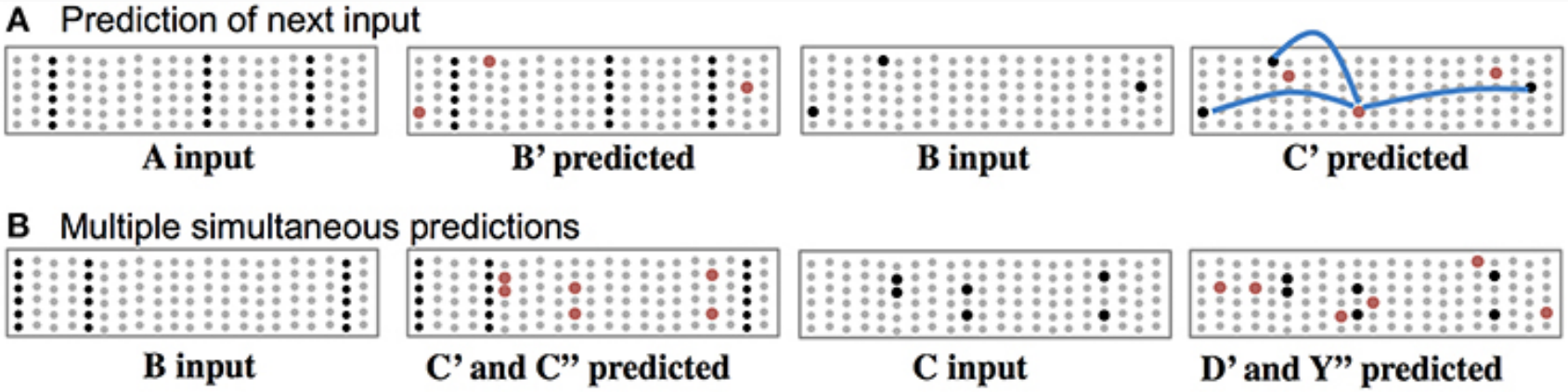
$D_d \in \mathbb{Sq}^{N_n \times N_s}$: πίνακας συναπτικών μονιμοτήτων

$NS \in \mathbb{B}^{N_n \times N_s}$: πίνακας γειτνίασης νευρώνων - δενδριτών

$SC \in \mathbb{B}^{N_s \times Nc}$: πίνακας γειτνίασης δενδριτών - μικροστηλών

Πηγή: [3]

Μνήμη ακολουθιών



Βήματα

1. Ενεργοποίηση
2. Προσδοκία/πρόβλεψη
3. Εκμάθηση συνάψεων ενεργών νευρώνων

Μεταβλητές κατάστασης

$D_d \in \mathbb{Sq}^{N_n \times N_s}$: πίνακας συναπτικών μονιμοτήτων

$NS \in \mathbb{B}^{N_n \times N_s}$: πίνακας γειτνίασης νευρώνων - δενδριτών

$SC \in \mathbb{B}^{N_s \times Nc}$: πίνακας γειτνίασης δενδριτών - μικροστηλών

Πηγή: [3]

Ενεργοποίηση χρονικής μνήμης

Ενεργοποίηση

$$\alpha_{ij} = \begin{cases} 1, & j \in c \wedge \pi_{ij}^{t-1} = 1 \text{ (πρόβλεψη)} \\ 1, & j \in c \wedge \sum_i \pi_{ij}^{t-1} = 0 \text{ (έξαρση)} \\ 0, & \text{αλλιώς} \end{cases} \quad (1)$$

c ενεργές μικροστήλες

π_{ij} προβλεπτικοί νευρώνες, j: μικροστήλη, i: νευρώνας στη j

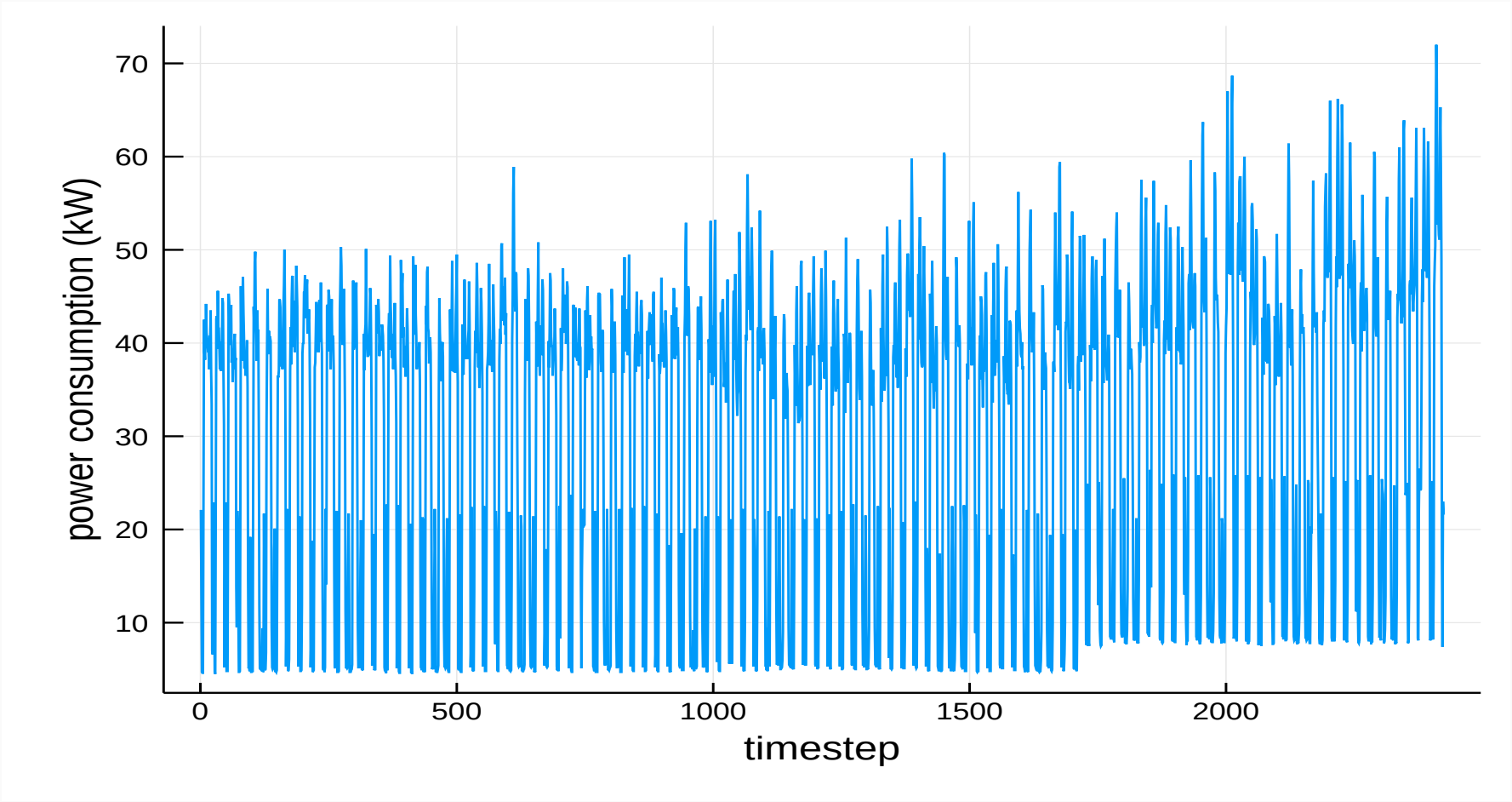
```
burst(c,Π)= c .& .!@percolumn(any,Π, k)
predicted(c,Π)= @percolumn(&,Π,c, k)
activate(c,Π)= (predicted(c,Π) .| burst(c,Π)')|> vec
```

Εκμάθηση

Πείραμα πρόβλεψης χρονοσειράς


Data
Encoder
binary vectors
HTM spatial pooler
SDR
HTM sequence memory
SDR
Classifier
Predictions Classification

Πηγή: [4]



Σχήμα 2: Ωριαία κατανάλωση ισχύος σε γυμναστήριο

Πηγή: [4]

Βιβλιογραφία

Αναφορές

S. Billaudelle και S. Ahmad, «Porting HTM Models to the Heidelberg Neuromorphic Computing Platform,», 8 Μάι. 2015. arXiv: 1505.02142 [cs, q-bio]. διεύθν.: http://arxiv.org/abs/1505.02142 (επίσκεψη 02/06/2019).

Y. Cui, S. Ahmad και J. Hawkins, «Continuous Online Sequence Learning with an Unsupervised Neural Network Model,» **Neural Computation**, τόμ. 28, αρθμ. 11, σσ. 2474–2504, 14 Σεπτ. 2016, ISSN: 0899-7667. DOI: 10.1162/NECO_a_00893.

J. Hawkins και S. Ahmad, «Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex,» **Frontiers in Neural Circuits**, τόμ. 10, 2016, ISSN: 1662-5110. DOI: 10.3389/fncir.2016.00023.

Y. Cui, S. Ahmad και J. Hawkins, «The HTM Spatial Pooler—A Neocortical Algorithm for Online Sparse Distributed Coding,» **Frontiers in Computational Neuroscience**, τόμ. 11, 2017, ISSN: 1662-5188. DOI: 10.3389/fncom.2017.00111.


J. Hawkins, S. Ahmad και Y. Cui, «A Theory of How Columns in the Neocortex Enable Learning the Structure of the World,» **Frontiers in Neural Circuits**, τόμ. 11, 2017, ISSN: 1662-5110. DOI: 10.3389/fncir.2017.00081.


Fabuio. (2017-07-06), Pyramidal neuron, διεύθν.: https://upload.wikimedia.org/wikipedia/commons/c/c1/Piramidal_cell.svg (επίσκεψη 31/05/2019).