

# Υψηλού επιπέδου υλοποίηση των αλγορίθμων Hierarchical Temporal Memory σε Julia

---

Κωνσταντίνος Σαμαράς-Τσακίρης  
Επιβλέπων καθηγητής: Νίκος Πιτσιάνης  
13 Ιουνίου 2019

# Για τι θα μιλήσουμε;

## Περίληψη της εργασίας

Παρουσιάζεται μια αλγοριθμική θεωρία της νοημοσύνης εν τη γενέσει, η **Hierarchical Temporal Memory (HTM)**. **Βασικοί αλγόριθμοι της θεωρίας υλοποιούνται σε υψηλού επιπέδου γλώσσα με εκφραστική περιεκτικότητα, τη Julia, για την καταπολέμηση της υψηλής γλωσσικής πολυπλοκότητας[9]** (δυσκολίας στην περιγραφή). **Η προγραμματιστική διατύπωση των αλγορίθμων παραμένει κοντά στην πηγαία μαθηματική διατύπωση**, διευκολύνοντας τη μετέπειτα ανάλυση και τον πειραματισμό με νέες αλγοριθμικές ιδέες και προεκτάσεις. Η HTM είναι βιολογικά περιορισμένη θεωρία, που αποσκοπεί καταρχήν στην εξήγηση της λειτουργίας του νεοφλοιού (εγκεφαλική δομή) και μόνο κατ'επέκτασιν σε εφαρμογές τεχνητής νοημοσύνης. Οι υλοποιήσεις σε Julia περιγράφονται αναλυτικά. Επαληθεύονται με τις υλοποιήσεις αυτές βασικές ιδιότητες των αλγορίθμων, εν είδει ελέγχου ορθότητας. Εν τέλει, προτείνονται κατευθύνσεις έρευνας στην HTM που διευκολύνονται από την παρούσα εργασία. Κορωνίδα του έργου είναι η **δημοσίευση πακέτου ανοιχτού λογισμικού** που υλοποιεί τους βασικούς αλγορίθμους HTM σε Julia.

# Για τι θα μιλήσουμε;

## Περίληψη της εργασίας

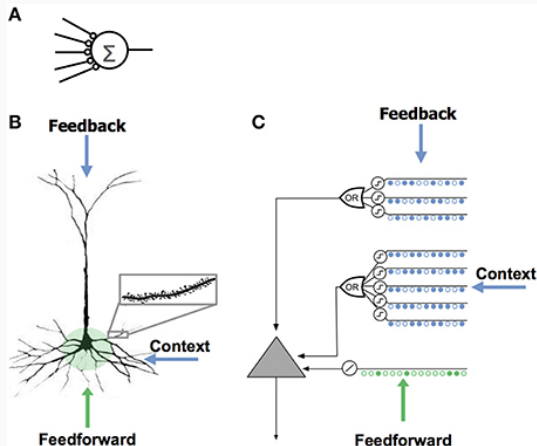
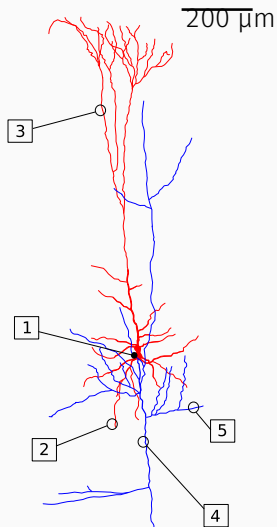
Παρουσιάζεται μια αλγοριθμική θεωρία της νοημοσύνης εν τη γενέσει, η **Hierarchical Temporal Memory (HTM)**. **Βασικοί αλγόριθμοι της θεωρίας υλοποιούνται σε υψηλού επιπέδου γλώσσα με εκφραστική περιεκτικότητα, τη Julia, για την καταπολέμηση της υψηλής γλωσσικής πολυπλοκότητας[9]** (δυσκολίας στην περιγραφή). **Η προγραμματιστική διατύπωση των αλγορίθμων παραμένει κοντά στην πηγαία μαθηματική διατύπωση**, διευκολύνοντας τη μετέπειτα ανάλυση και τον πειραματισμό με νέες αλγοριθμικές ιδέες και προεκτάσεις. Η HTM είναι βιολογικά περιορισμένη θεωρία, που αποσκοπεί καταρχήν στην εξήγηση της λειτουργίας του νεοφλοιού (εγκεφαλική δομή) και μόνο κατ'επέκτασιν σε εφαρμογές τεχνητής νοημοσύνης. Οι υλοποιήσεις σε Julia περιγράφονται αναλυτικά. Επαληθεύονται με τις υλοποιήσεις αυτές βασικές ιδιότητες των αλγορίθμων, εν είδει ελέγχου ορθότητας. Εν τέλει, προτείνονται κατευθύνσεις έρευνας στην HTM που διευκολύνονται από την παρούσα εργασία. Κορωνίδα του έργου είναι η **δημοσίευση πακέτου ανοιχτού λογισμικού** που υλοποιεί τους βασικούς αλγορίθμους HTM σε Julia.

- Πώς η πρότυπη υλοποίηση  
1450 γραμμών κώδικα → 340 γραμμές υψηλού επιπέδου

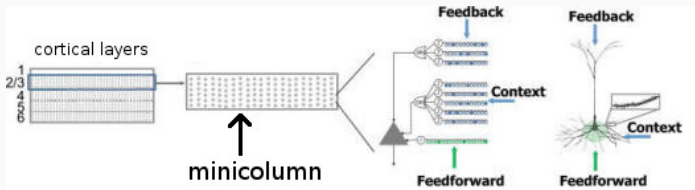
# Hierarchical Temporal Memory

---

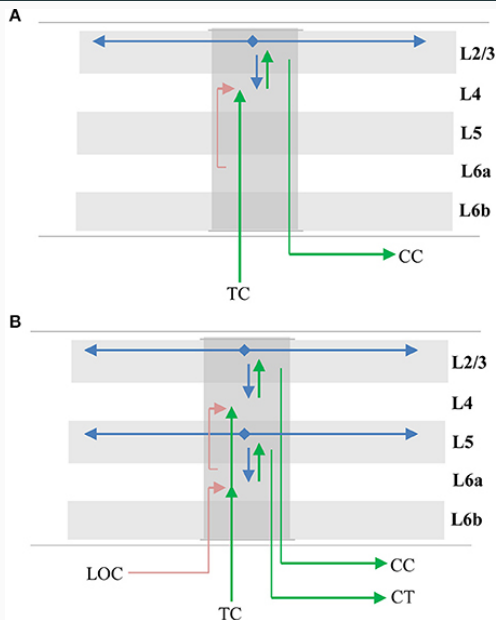
# Νευρώνας



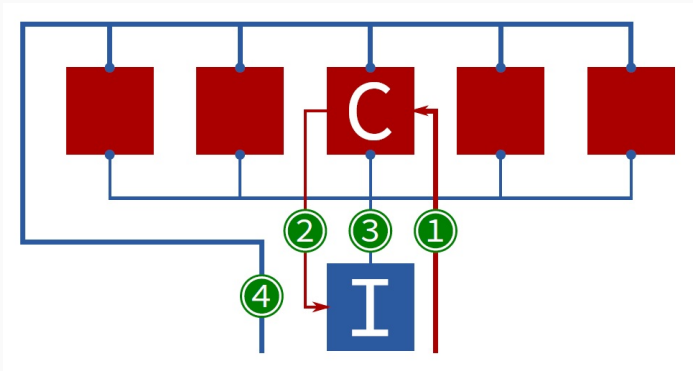
# Μικροστήλες



# Φλοιικές στήλες

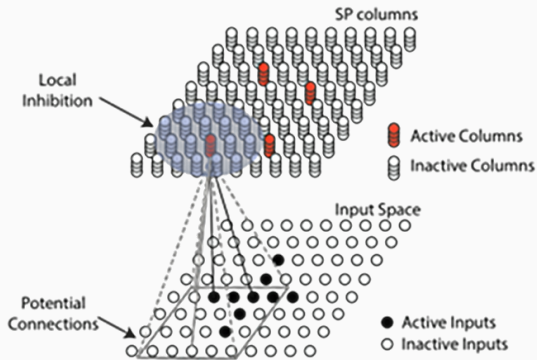


# Αναστολή μεταξύ μικροστηλών

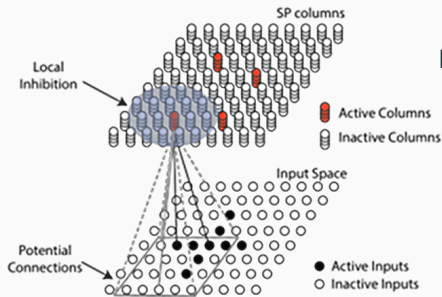




# Χωρικός συγκεντρωτής



# Χωρικός συγκεντρωτής



## Μεταβλητές κατάστασης

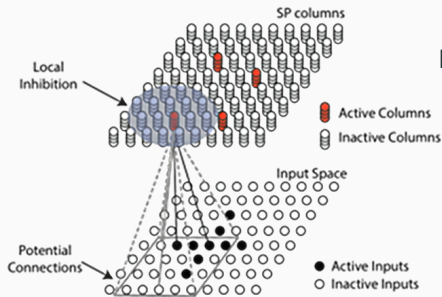
$\mathbf{D_p} \in \mathbb{S}^{N_{in} \times N_{sp}}$  πίνακας συναπτικών μονιμοτήτων

$\hat{a}_t \in \mathbb{B}^{N_{sp}}$  μεση δραστηριότητα μικροστηλών

## 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

# 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

## Μεταβλητές κατάστασης

$\mathbf{D_p} \in \mathbb{S}q^{N_{in} \times N_{sp}}$  πίνακας συναπτικών μονιμοτήτων

$\hat{a}_t \in \mathbb{B}^{N_{sp}}$  μεση δραστηριότητα μικροστηλών

# Στοιχεία Χωρικού Συγκεντρωτή

---

# Αντιστοίχιση εισόδου/εξόδου: υπερκύβος

Δείκτης υπερκύβου

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

$x^c$  κέντρο υπερκύβου

$\gamma$  ακτίνα υπερκύβου

# Αντιστοίχιση εισόδου/εξόδου: υπερκύβος

## Δείκτης υπερκύβου

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

$x^c$  κέντρο υπερκύβου

$\gamma$  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

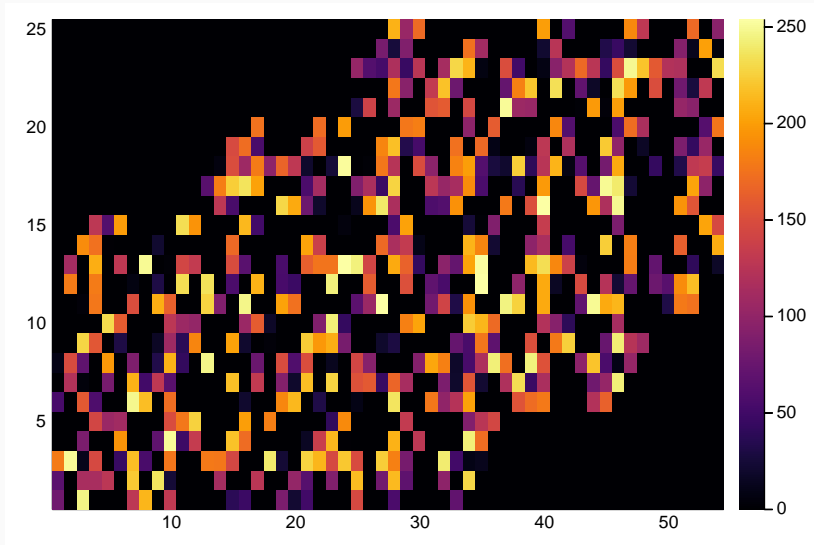
```
c2lin = LinearIndices(szin)
c2lsp = LinearIndices(szsp)

Dp = zeros($\mathbb{N}$, prod(szin), prod(szsp))
foreach(out_lattice()) do yi
    # Linear indices from hypercube
    x = @>> yi xc xi collect map(x->c2lin[x...])
    Dp[x, c2lsp[yi...]] = permanences(@> yi xc xi)
end

@> yi xc xi === xi(xc(yi))
```



# Αρχικοποίηση συνάψεων



Επικάλυψη μικροστηλών με συνδεδεμένες εισόδους

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

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

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

$z$   $[\ell_{in}]$  είσοδος

$b$   $[\ell_{sp}]$  παρώθηση

Επικάλυψη μικροστηλών με συνδεδεμένες εισόδους

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

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

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

$z$  [ $\ell_{in}$ ] είσοδος

$b$  [ $\ell_{sp}$ ] παρώθηση

```
Wp()= Dp .>= theta_permanence  
o(z)= @(z) (b() .* Wp()'z) reshape(sz_sp)
```

Επικάλυψη μικροστηλών με συνδεδεμένες εισόδους

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

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

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

$z$  [ $\ell_{in}$ ] είσοδος

$b$  [ $\ell_{sp}$ ] παρώθηση

```
Wp()= Dp .>= theta_permanence  
o(z)= @> (b() .* Wp()'z) reshape(sz_sp)
```

broadcasting

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

Κανόνας πλαστικότητας τύπου Hebbian (STDP, spike-timing-dependent-plasticity)

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

όπου  $c$  αποτέλεσμα του  $z$ .

$c$  ενεργοποίηση χωρικού συγκεντρωτή

Κανόνας πλαστικότητας τύπου Hebbian (STDP, spike-timing-dependent-plasticity)

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

όπου  $c$  αποτέλεσμα του  $z$ .

$c$  ενεργοποίηση χωρικού συγκεντρωτή

Απλούστερος τρόπος

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

# Εκμάθηση συνάψεων

Κανόνας πλαστικότητας τύπου Hebbian (STDP, spike-timing-dependent-plasticity)

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

όπου  $c$  αποτέλεσμα του  $z$ .

$c$  ενεργοποίηση χωρικού συγκεντρωτή

Καλύτερα

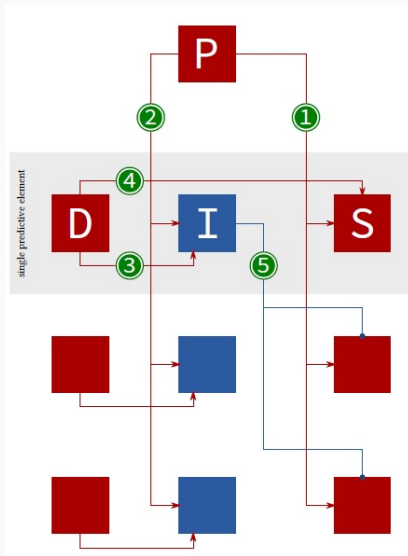
```
learn!(D_p,z,a)= begin
  D_pactive= @view D_p[:,a] # the only elements we touch
  activeConn= (D_pactive .> 0) .& z
  inactiveConn= (D_pactive .> 0) .& !z
  D_pactive.= activeConn .* (D_pactive .⊕ p⁺) .+
               inactiveConn .* (D_pactive .⊖ p⁻)
end
```

## Χρονική μνήμη

---



# Αναστολή εντός μικροστηλών

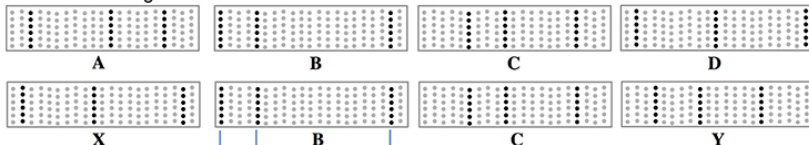


# Μνήμη ακολουθιών

## A Cellular layers learn sequences

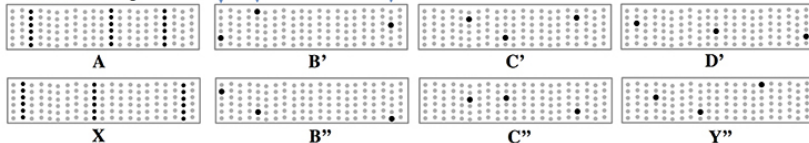


## B Before learning



Same columns,  
but only one cell active per column.

## C After learning



# Μνήμη ακολουθιών

## A Prediction of next input



**A input**



**B' predicted**

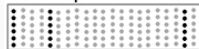


**B input**



**C' predicted**

## B Multiple simultaneous predictions



**B input**



**C' and C'' predicted**



**C input**



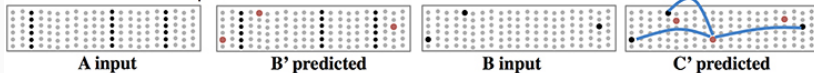
**D' and Y'' predicted**

## Βήματα

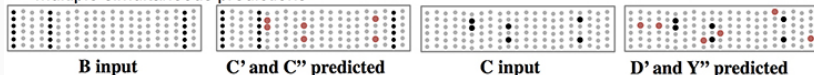
1. Ενεργοποίηση
2. Προσδοκία/πρόβλεψη
3. Υπολογισμός νικητών νευρώνων & δενδριτών
4. Εκμάθηση συνάψεων ενεργών νευρώνων

# Μνήμη ακολουθιών

## A Prediction of next input



## B Multiple simultaneous predictions



## Βήματα

1. Ενεργοποίηση
2. Προσδοκία/πρόβλεψη
3. Υπολογισμός νικητών νευρώνων & δενδριτών
4. Εκμάθηση συνάψεων ενεργών νευρώνων

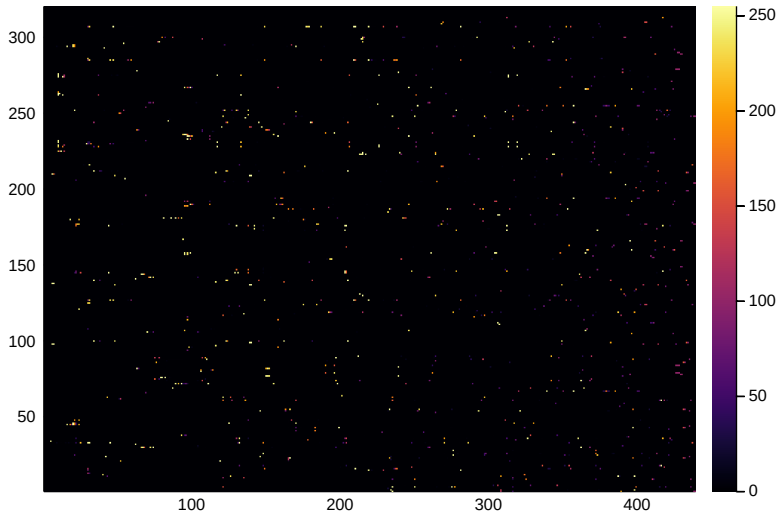
## Μεταβλητές κατάστασης

$\mathbf{D}_d \in \mathbb{S}^{N_n \times N_s}$  αραιός πίνακας συναπτικών μονιμοτήτων προσυναπτικοί νευρώνες  $\times$  μετασυναπτικοί δενδρίτες

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

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

# Μνήμη ακολουθιών



# 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

$\pi_{ij}$  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 συνάψεων

Κανόνας πλαστικότητας τύπου Hebbian (αντίστοιχος με ΧΣ)

$$\Delta \mathbf{D}_{\mathbf{d}} = p^+(\alpha^{t-1} \circ \mathbf{D}_{\mathbf{d}} \circ \mathbf{WS}) - p^-(\neg \alpha^{t-1} \circ \mathbf{D}_{\mathbf{d}} \circ \mathbf{WS})$$

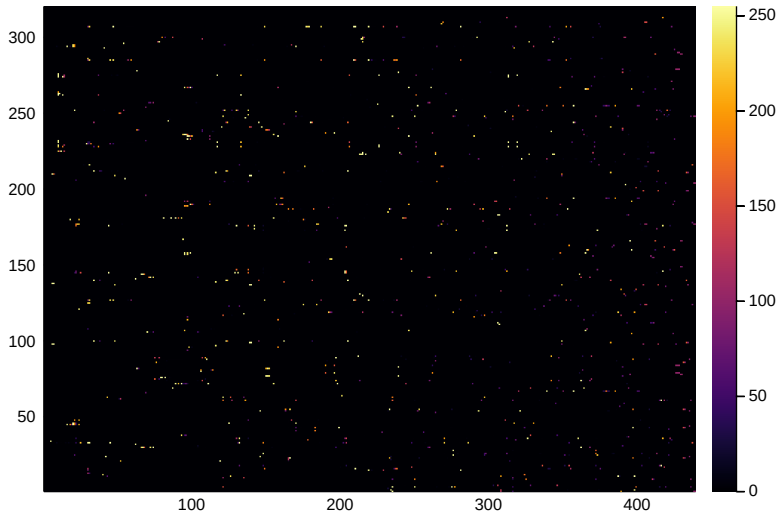
$\alpha$  ενεργοί νευρώνες

$\mathbf{WS}$  «νικητές» δένδριτες:  $\in \pi^{t-1} \cap \alpha^t$

Διατρέχοντας τον αραιό (CSC)  $\mathbf{D}_{\mathbf{d}}$  κατά στήλες:

```
function learn_sparsesynapses(synapses_activeCol,input_i,z,p+,p-)
    z_i= z[input_i]
    synapses_activeCol.= z_i .* (synapses_activeCol .⊕ p+) .+
                           .!z_i .* (synapses_activeCol .⊖ p-)
end
sparse_foreach((scol,cell_i)->
    learn_sparsesynapses(scol,cell_i, pa, p+,p-),
    D_d, WS)
```

# Εκμάθηση συνάψεων

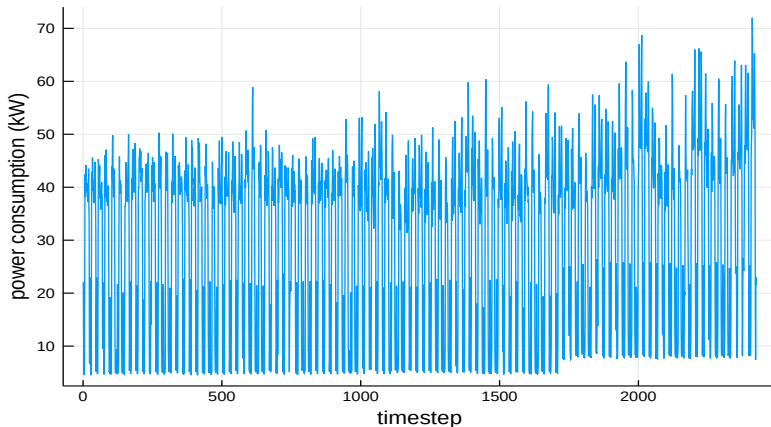




## Πείραμα πρόβλεψης χρονοσειράς

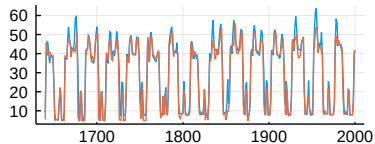
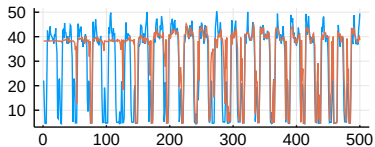
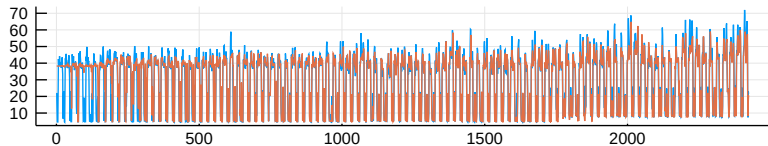
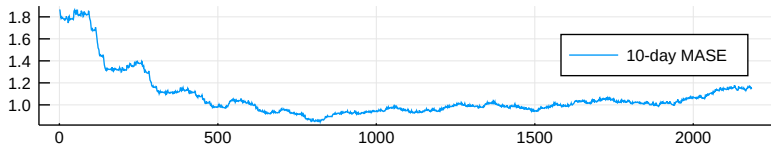
---





Σχήμα 2: Ωριαία κατανάλωση ισχύος σε γυμναστήριο

# Πρόβλεψη 1 στιγμή μπροστά



Χρονική συγκέντρωση. Πόλωση από την προσδοκώμενη ακολουθία [5]

Συνένωση πολλών περιοχών σε ιεραρχικό μοντέλο [6]

Μελέτη κανόνων εκμάθησης από οπτική θεωρίας γράφων [7]

# Βιβλιογραφία

---



S. Billaudelle και S. Ahmad, «Porting HTM Models to the Heidelberg Neuromorphic Computing Platform», 8 Μάι. 2015. arXiv: 1505.02142 [cs, q-bio]. διεύθυν.: <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.



J. Hawkins, M. Lewis, M. Klukas, S. Purdy και S. Ahmad, «A Framework for Intelligence and Cortical Function Based on Grid Cells in the Neocortex,» **Frontiers in Neural Circuits**, τόμ. 12, 2019, ISSN: 1662-5110. DOI: 10.3389/fncir.2018.00121.



E. Kipouridis και K. Tsihclas, «On the Convergence of Network Systems,», 11 Φεβ. 2019. arXiv: 1902.04121 [cs, math]. διεύθυν.: <http://arxiv.org/abs/1902.04121> (επίσκεψη 10/06/2019).



Fabuio. (2017-07-06), Pyramidal neuron, διεύθυν.: [https://upload.wikimedia.org/wikipedia/commons/c/c1/Pyramidal\\_cell.svg](https://upload.wikimedia.org/wikipedia/commons/c/c1/Pyramidal_cell.svg) (επίσκεψη 31/05/2019).



B. Chazelle. (), Natural Algorithms and Influence Systems, διεύθυν.: <https://cacm.acm.org/magazines/2012/12/157889-natural-algorithms-and-influence-systems/abstract> (επίσκεψη 28/05/2019).

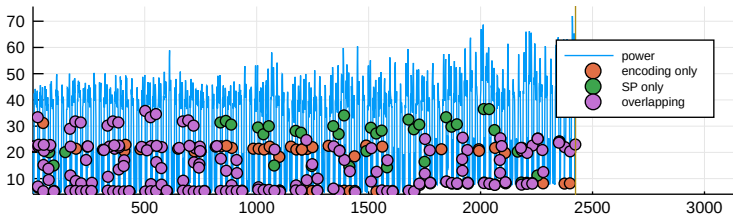
# Παράρτημα

---



# Διατήρηση ομοιότητας από χωρικό συγκεντρωτή

Spatial Pooler mapping property evaluation



Percentage of overlapping encoder and SP SDRs

