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;

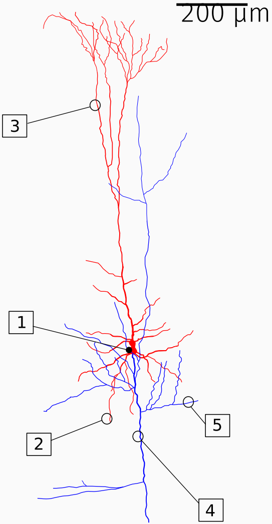
Περίληψη της εργασίας

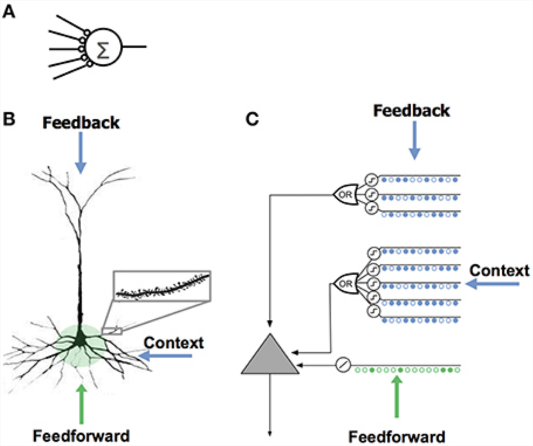
Παρουσιάζεται μια αλγοριθμική θεωρία της νοημοσύνης εν τη γενέσει, η **Hierarchical Temporal Memory** (HTM). Βασικοί αλγόριθμοι της θεωρίας υλοποιούνται σε υψηλού επιπέδου γλώσσα με εκφραστική περιεκτικότητα, τη Julia, για την καταπολέμηση της υψηλής **γλωσσικής πολυπλοκότητας**[9] (δυσκολίας στην περιγραφή). 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

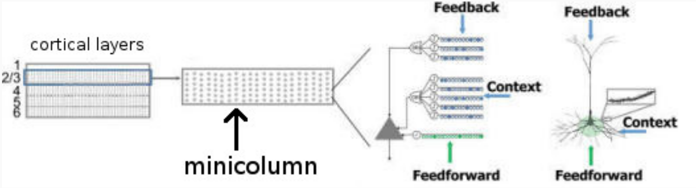
Νευρώνας





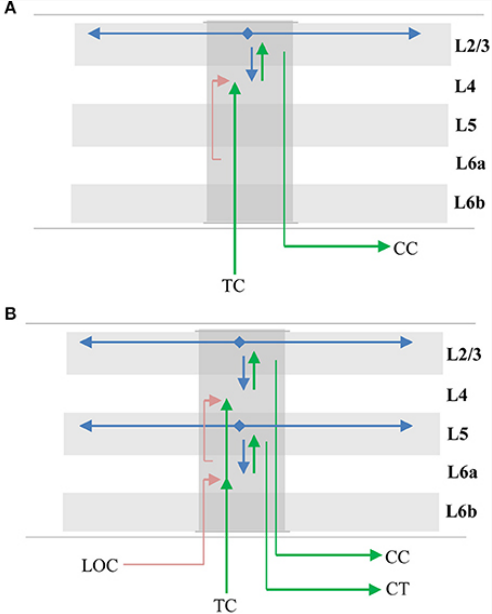
Πηγή: [3], [8]

Μικροστήλες



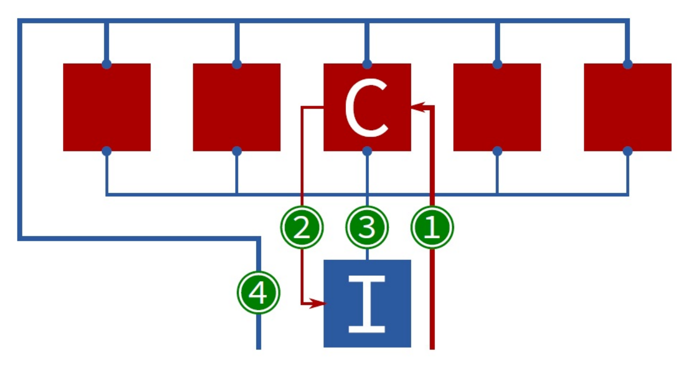
Πηγή: [2]

Φλοιικές στήλες



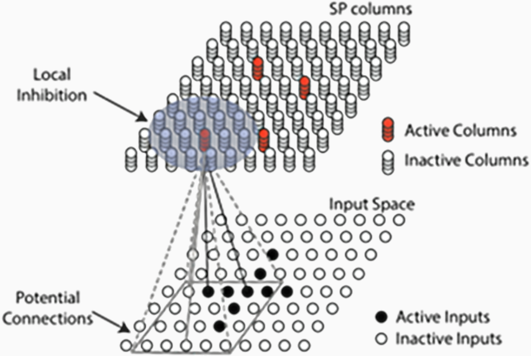
Πηγή: [5]

Αναστολή μεταξύ μικροστηλών



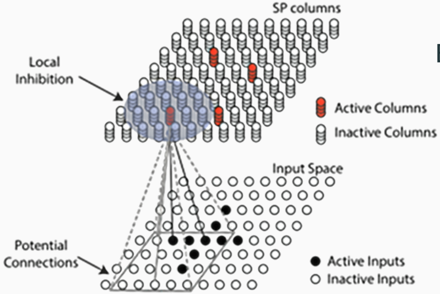
Πηγή: [1]

Χωρικός συγκεντρωτής



Πηγή: [4]

Χωρικός συγκεντρωτής



Μεταβλητές κατάστασης

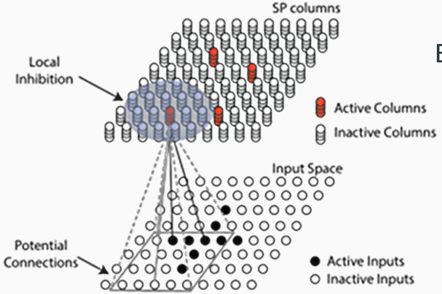
- $\mathbf{D_p}$ $\in \$\mathbb{q}^{N_{in} \times N_{sp}}$ πίνακας συναπτικών μονιμοτήτων
- $\hat{a}_t$ $\in \mathbb{B}^{N_{sp}}$ μεση δραστηριότητα μικροστηλών

Βήματα

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

Μεταβλητές κατάστασης

- $\mathbf{D_p} \in \$q^{N_{in} \times N_{sp}}$ πίνακας συναπτικών μονιμοτήτων
- $\hat{a}_t \in \mathbb{B}^{N_{sp}}$ μεση δραστηριότητα μικροστηλών

Πηγή: [4]

Στοιχεία Χωρικού Συγκεντρωτή

Αντιστοίχιση εισόδου/εξόδου: υπερκύβος

Δείκτης υπερκύβου

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

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

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

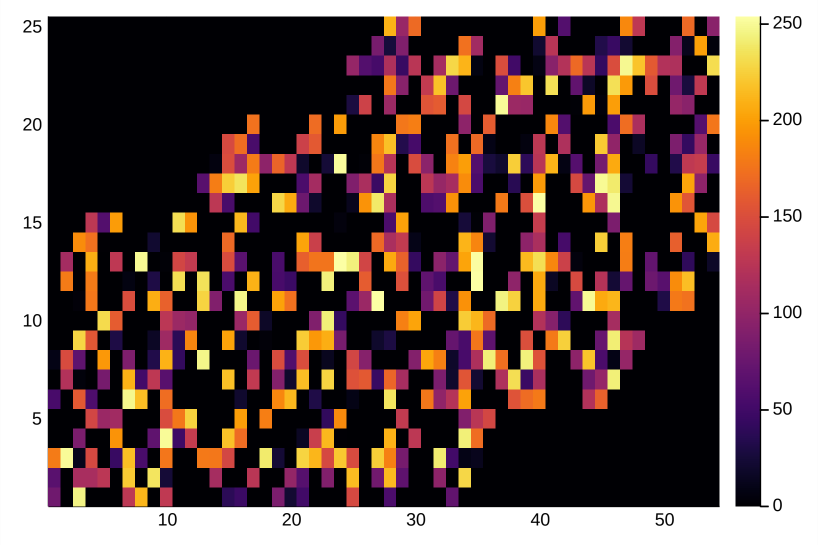
όπου $Z \in U(0,1)$ τυχαίος αριθμός

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

Dₚ= zeros($q, 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

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

Επικάλυψη

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 .≥ θ_permanence
o(z)= @> (b() .* Wₚ()'z) reshape(szₛₚ)
```

Επικάλυψη

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 .≥ θ_permanence
o(z)= @> (b() .* Wₚ()'z) reshape(szₛₚ)
```

broadcasting

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

Εκμάθηση συνάψεων

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

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

όπου c αποτέλεσμα του z.

- c ενεργοποίηση χωρικού συγκεντρωτή

Εκμάθηση συνάψεων

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

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

όπου c αποτέλεσμα του z.

- c ενεργοποίηση χωρικού συγκεντρωτή

Απλούστερος τρόπος

```
learn!(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 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.

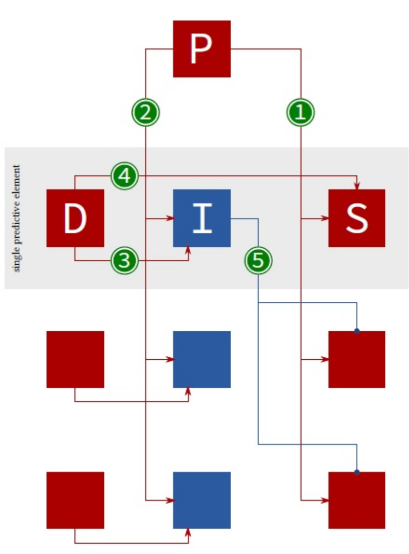
c ενεργοποίηση χωρικού συγκεντρωτή

Καλύτερα

```
learn!(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
```

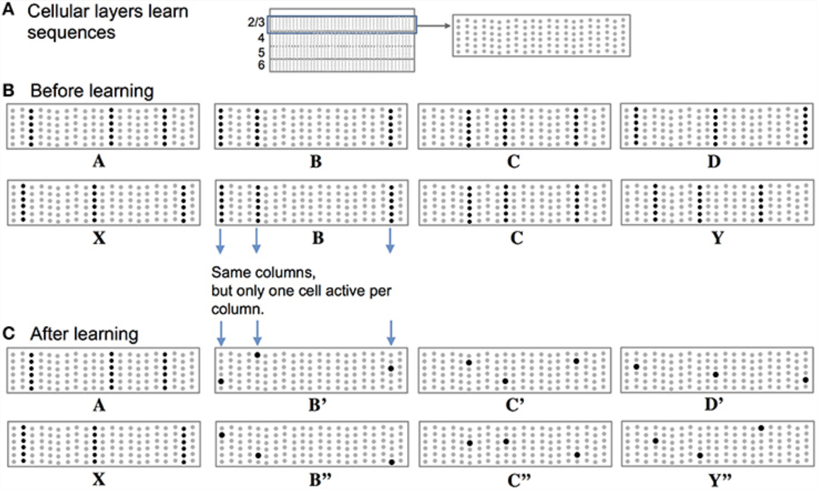
Χρονική μνήμη

Αναστολή εντός μικροστηλών



Πηγή: [1]

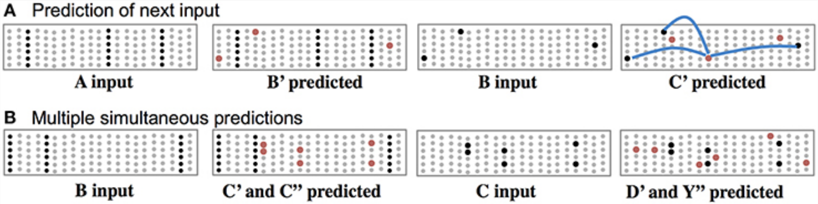
Μνήμη ακολουθιών



Μνήμη ανώτερης τάξης

Πηγή: [3]

Μνήμη ακολουθιών

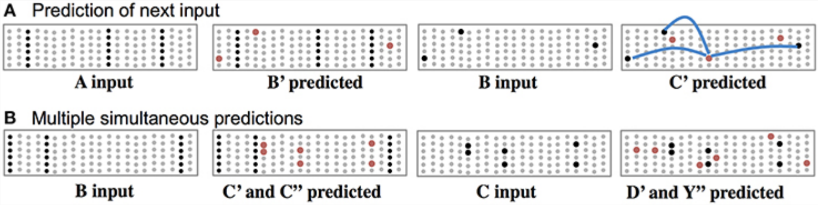


Αναπαράσταση αμφιβολίας

Βήματα

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

Πηγή: [3]



Αναπαράσταση αμφιβολίας

Βήματα

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

Μεταβλητές κατάστασης

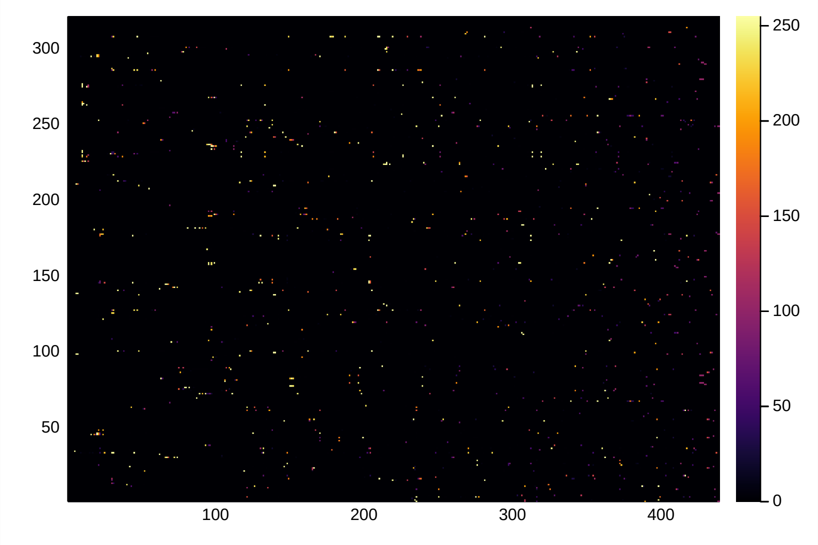
$\mathbf{D_d} \in \$\mathrm{q}^{N_n \times N_s}$ αραιός πίνακας συναπτικών μονιμοτήτων προσυναπτικοί νευρώνες × μετασυναπτικοί δενδρίτες

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

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

Πηγή: [3]

Μνήμη ακολουθιών



Ενεργοποίηση χρονικής μνήμης

Ενεργοποίηση

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

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 συνάψεων

Κανόνας πλαστικότητας τύπου Hebbian (αντίστοιχος με ΧΣ)

$$\mathbf{\Delta D_d} = p^{+}(\alpha^{t-1} \circ \mathbf{D_d} \circ WS) \; - \; p^{-}(\neg\alpha^{t-1} \circ \mathbf{D_d} \circ WS)$$

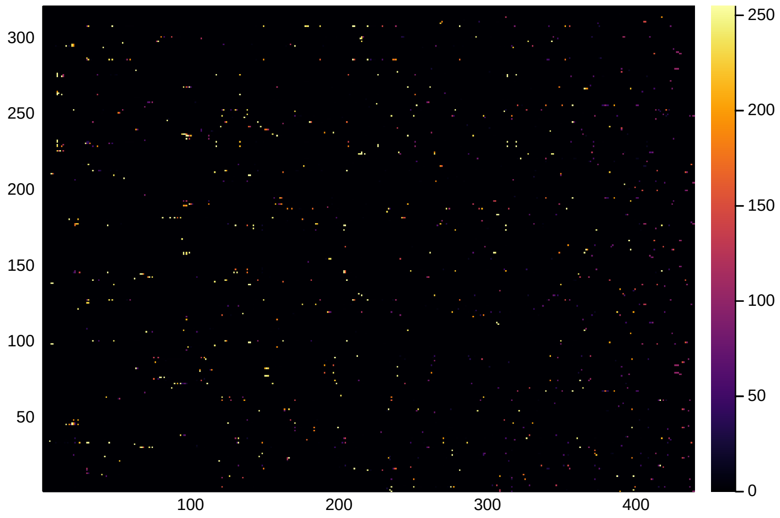
α ενεργοί νευρώνες

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

Διατρέχοντας τον αραιό (CSC) $\mathbf{D_d}$ 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, pα, p⁺,p⁻),
               D_d, WS)
```

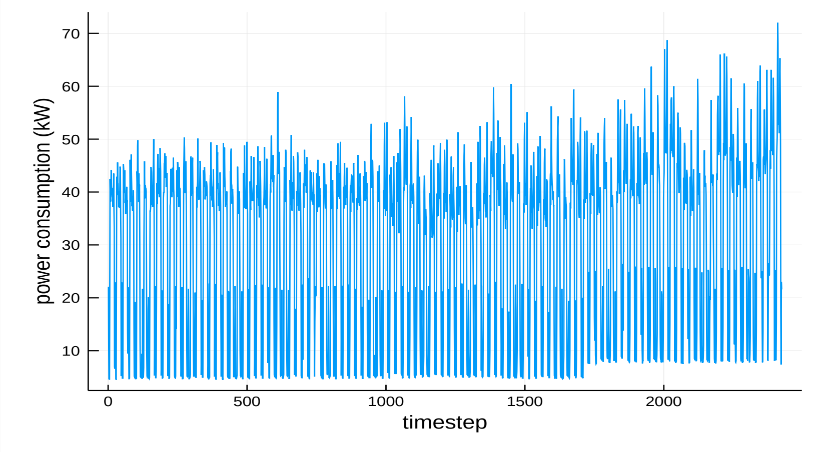
Εκμάθηση συνάψεων



Πείραμα πρόβλεψης χρονοσειράς


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

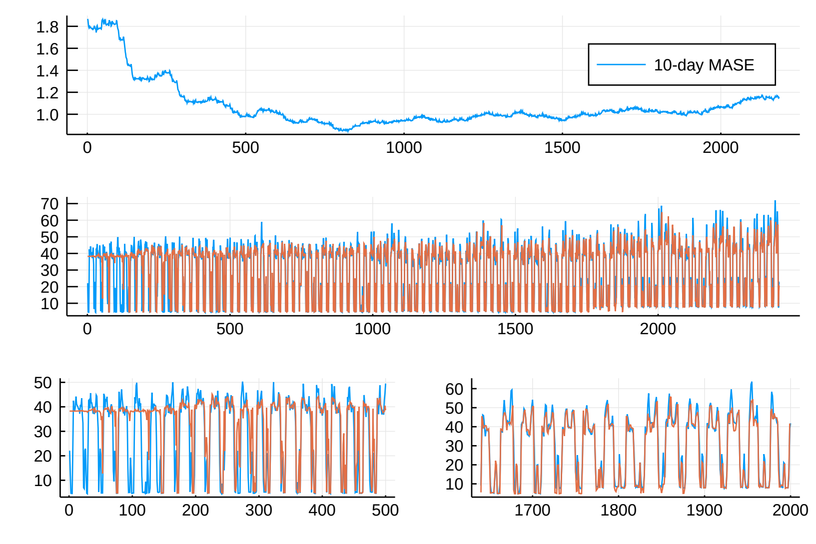
Πηγή: [4]



Σχήμα 2: Ωριαία κατανάλωση ισχύος σε γυμναστήριο

Πηγή: [4]

Πρόβλεψη 1 στιγμή μπροστά



Προτάσεις για μελέτη στην HTM

Χρονική συγκέντρωση. Πόλωση από την προσδοκώμενη ακολουθία [5]

Συνένωση πολλών περιοχών σε ιεραρχικό μοντέλο [6]

Μελέτη κανόνων εκμάθησης από οπτικη 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]. 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.

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. Tsichlas, «On the Convergence of Network Systems,», 11 Φεβ. 2019. arXiv: 1902.04121 [cs, math]. διεύθν.: http://arxiv.org/abs/1902.04121 (επίσκεψη 10/06/2019).

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

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

Παράρτημα

Διατήρηση ομοιότητας από χωρικό συγκεντρωτή