Brain-inspired cognitive computing

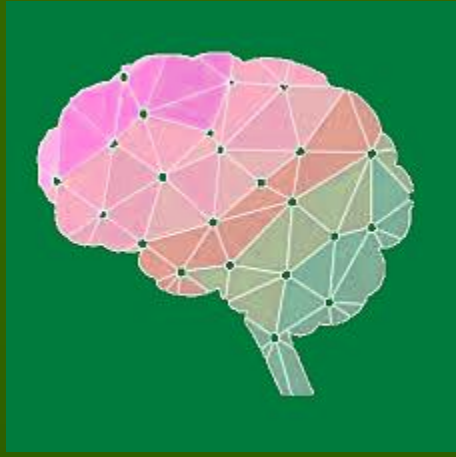
Włodzisław Duch

NeuroCognitive Laboratory, Centre for Modern Interdisciplinary Technologies & Neuroinformatics and Artificial Intelligence University Centre of Excellence in Dynamics, Mathematical Analysis and Artificial Intelligence.

Google: Wlodzislaw Duch

Seminarium KIiA PAN-UMK-UTP 27/11/2020

CD DAMSI



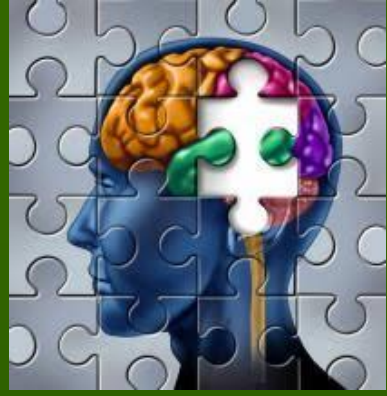
University Centre of Excellence (2020) in the research area "Dynamics, mathematical analysis and artificial intelligence".

1. Dynamics and ergodic theory (Math)
2. Computer science – formal languages and concurrency (Theoretical CS)
3. Entangled states and dynamics of open quantum systems (Math Physics)
4. Neuroinformatics and artificial intelligence (Neuroinformatics)

Neuroinformatics is a combination of two important disciplines on the science front: brain research and artificial intelligence. By using machine learning and signal processing methods, new theories and algorithms for brain signal analysis are developed, verifying hypotheses through experiments.

Our group: inspirations from brain research for better neural algorithms?

NeuroCog projects



Neurocognitive Informatics: understanding complex cognition => creating algorithms that work in similar way.

- Analysis/understanding of brain signals/processes (EEG/MEG/fMRI).
- Understanding neurodynamics, creating better neurofeedback.
- Geometric theory of brain-mind processes.
- Computational intelligence, machine learning inspirations …
- Neurocognitive approach to language, word games.
- Computational creativity, insight, intuition, imagery.
- Imagery agnosia (aphantasia), amusia, musical talent.
- Comprehensive theory of autism, ADHD, phenomics, neuroeducation.
- Brain stem models & consciousness in artificial systems.
- Infants: observation, perception, WM training/development.
- Neural determinism, free will & social consequences.

Publications 2020

- Duch. W. (2020) IDyOT architecture – is this how minds operate? **Physics of Life Reviews** (IF 13.8)
- Rykaczewski, K, Nikadon, J, Duch, W, Piotrowski, T. (2020). SupFunSim: spatial filtering toolbox for EEG. **Neuroinformatics** (IF 5.1).
- Finc, K, Bonna, K, He, X, Lydon-Staley, D.M, Kühn, S, Duch, W, & Bassett, D. S. (2020). Dynamic reconfiguration of functional brain networks during working memory training. **Nature Communications** 11, 2435 (IF 11.8)
- Dreszer J, Grochowski M, Lewandowska M, Nikadon J, Gorgol J, Bałaj B, Finc K, Duch W, Kałamała P, Chuderski A, Piotrowski T. (2020). Spatiotemporal Complexity Patterns of Resting-state Bioelectrical Activity Explain Fluid Intelligence: Sex Matters. **Human Brain Mapping** (IF 4.5)
- Bonna, K, Finc, K, … Duch, W, Marchewka, A, Jednoróg, K, Szwed, M. (2020). Early deafness leads to re-shaping of global functional connectivity beyond the auditory cortex. **Brain Imaging and Behaviour**, 1-14 (IF 3.6).
- Duch W, Mikołajewski D. (2020) Modelling effects of consciousness disorders in brainstem computational model – Preliminary findings. **Bio-Algorithms and Med-Systems** 16(2).

In search of the sources of brain's cognitive activity

Project „Symfonia”, NCN, Kraków, 18.07.2016

FACULTY OF PHYSICS, ASTRONOMY AND INFORMATICS

CENTRE FOR MODERN INTERDISCIPLINARY TECHNOLOGIES

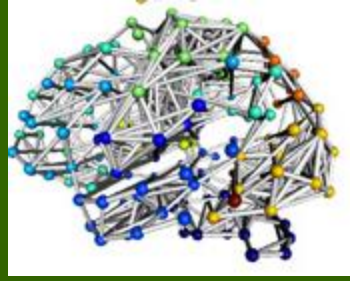
INSTITUTE OF PHYSIOLOGY AND PATHOLOGY OF HEARING

nencki institute of experimental biology

My group of neuro-cog-fanatics

Graduates of: cognitive science, computer science, engineering, mathematics, neuroscience, philosophy, physics, psychology.

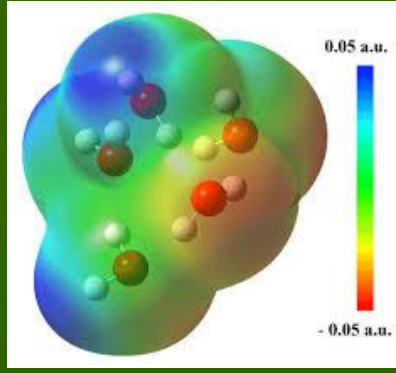
Brain Inspirations



1. Simple neurons, 1 parameter, fixed synaptic connections
   $\Rightarrow$ perceptrons, MLPs.
2. Complex neurons, microcircuits, small neural cortical ensembles with structural connections (fixed, or slowly changing).
3. Complex network states: rich internal knowledge in modules interacting in a flexible way, functional connections activated by priming, working memory control. Attractors of neurodynamics that synchronize many cortical ensembles, solving novel combinatorial problems.
4. Society of minds: for different tasks using flexible arrangement of functional connections between specialized brain regions, a lot of knowledge in such modules, no fixed connections.
5. Society of brains: collaboration between brains on symbolic level.

These inspirations led me to creation of many interesting ideas and algorithms. Some ideas are now verified experimentally using neuroimaging.

Reinventing the wheel



1990-95 - before the Internet and repositories of papers …

RBF rediscovery – probability density in quantum chemistry is described using basis functions, hence probability density estimation idea:

- Duch W (1994) Floating Gaussian Mapping: a new model of adaptive systems. Neural Network World 4:645-654
- Broomhead & Lowe (1988), Multivariable functional interpolation and adaptive networks. Complex Systems 2: 321–355.

But why use only radial functions? Separable functions are better.

MDS and manifolds, visualization of data:

- Duch W (1995) Quantitative measures for the self-organized topographical mapping. Open Systems and Information Dynamics 2:295-302
  Based on a set of 3rd order equations instead of minimization.
- MDS: Torgerson (1958), Sammon 1969.

Discovering the wheel - reverse

Now some people are reinventing my ideas … Ex.

- Duch W. (1996) Computational physics of the mind.
  Computer Physics Communication 97: 136-153 ("unusual contribution").
- Perlovsky, L. I. (2016) Physics of the Mind.
  Frontiers in Systems Neuroscience, 10. fnsys.2016.00084

- Duch W, Diercksen GHF (1995) Feature Space Mapping as a universal adaptive system. Computer Physics Communication 87: 341-371
- Perlovsky LI. Neural networks and intellect: Using model based concepts. New York: Oxford University Press; 2001

Same with meta-learning, similarity-based learning, prototype-based learning, transfer functions, creativity and intuition, and a few other ideas.

- Duch W (2007), Intuition, Insight, Imagination and Creativity. IEEE Computational Intelligence Magazine 2(3), 40-52

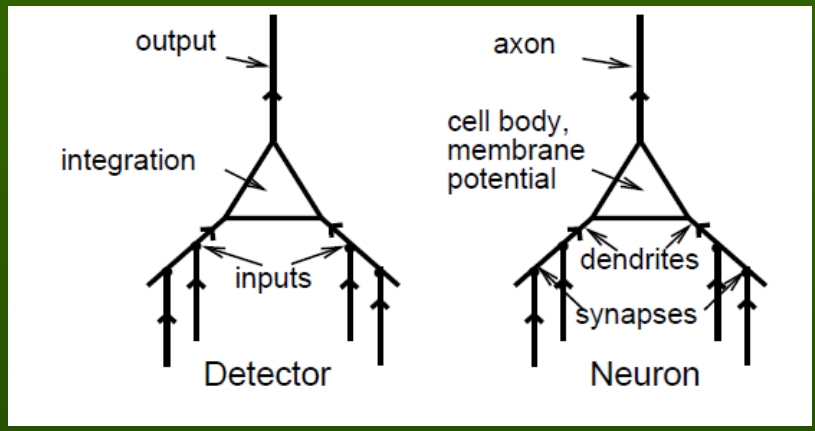
Modeling NN functions

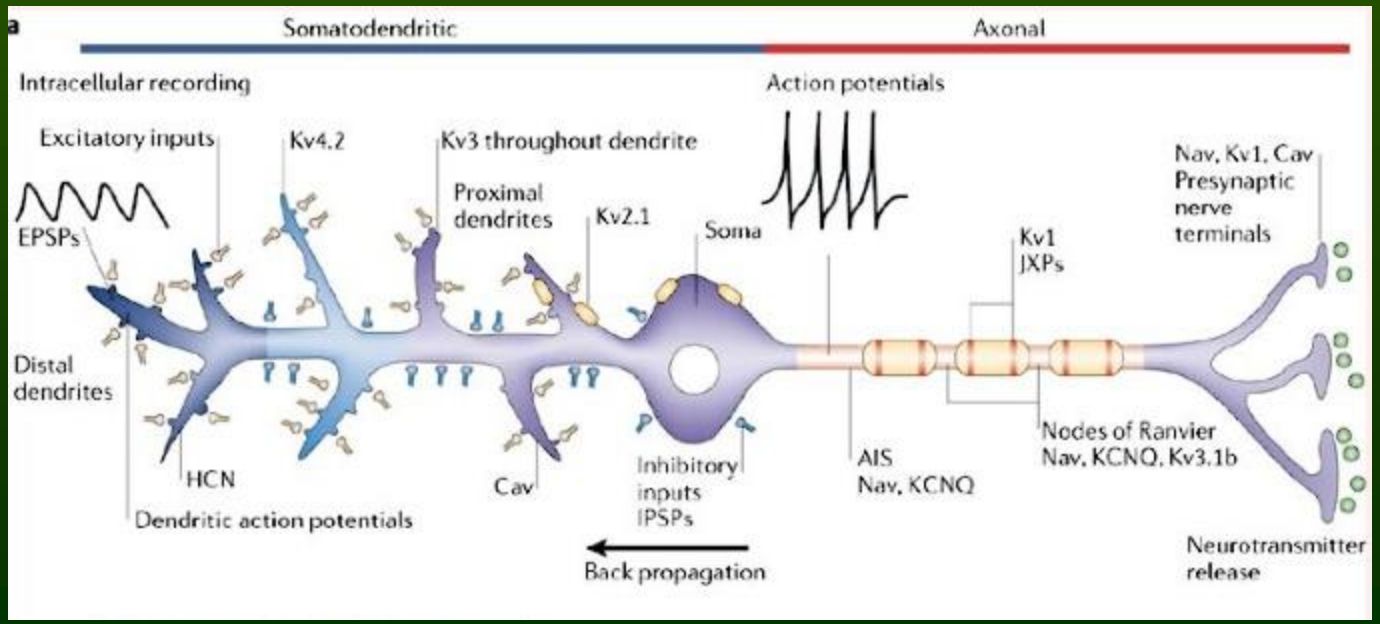
Neurons

Simplest inspiration: neurons (100 G) => perceptron function $\sigma(W*X+\theta)$

Reality: diverse types, ion channels, neurochemistry, complex spatio-temporal integration, >10 K inputs …

Detailed biophysical models of neurons are required for neuropsychiatric disorders, influence of neurotransmitters, drugs, etc.

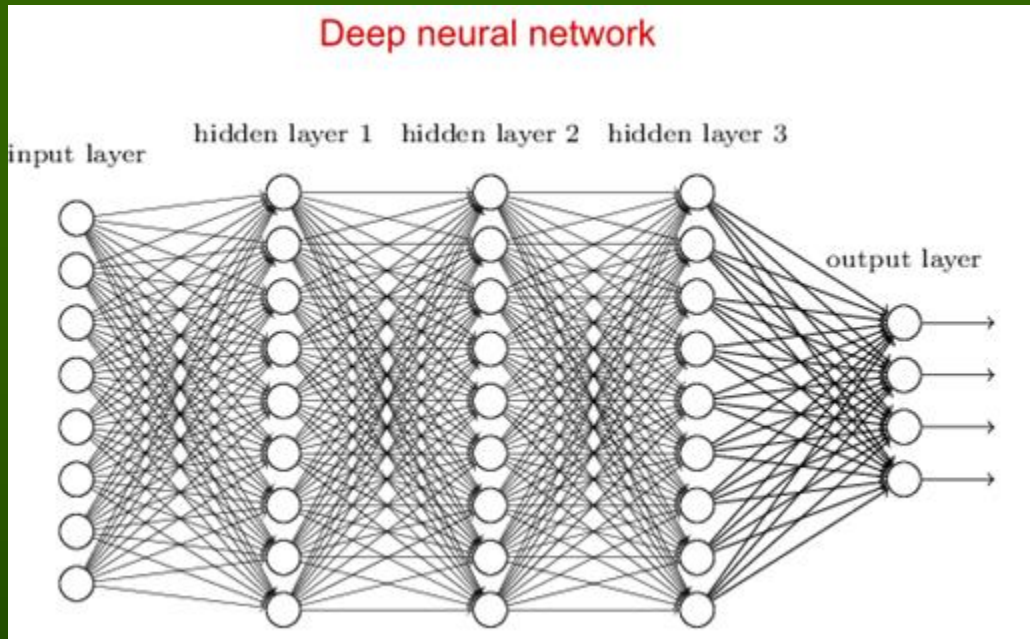


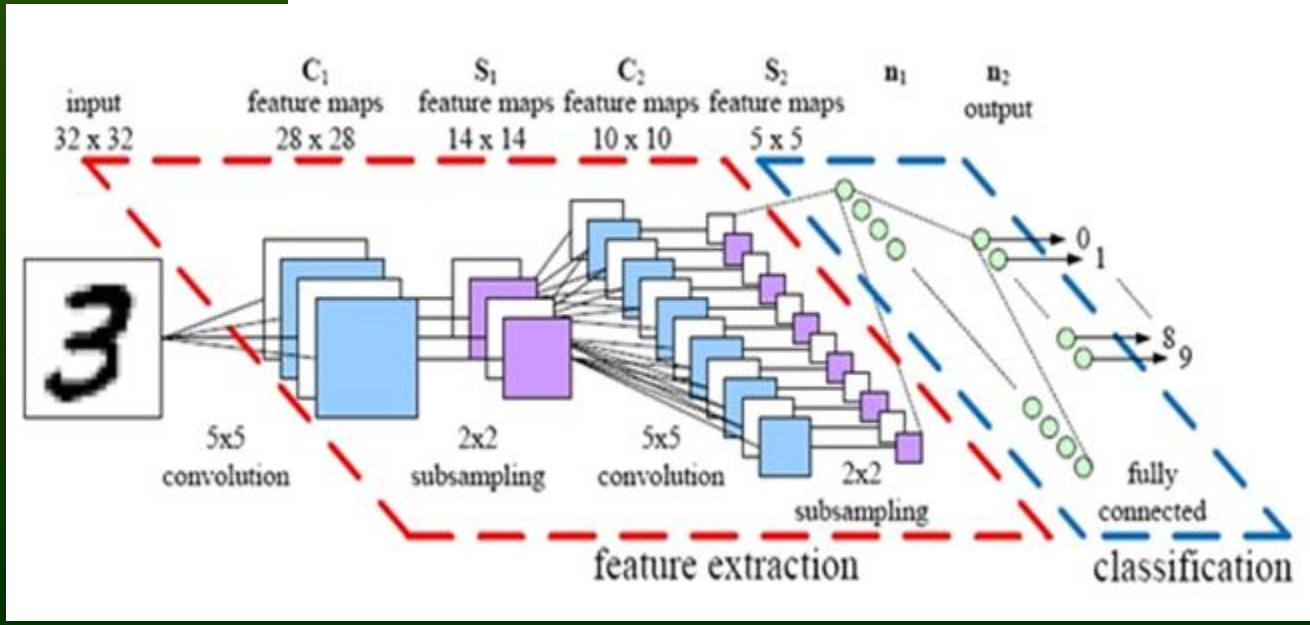


Tensorization of Convolutive Deep Learning NN

Most neural models: networks of simple non-linear neurons (recently ReLu, simplest), exchanging information via fixed connections, adapting simple parameters to learn vector mappings. But backprop like learning has no justification.

Ex: tensor networks
Cichocki Lab, RIKEN BSI



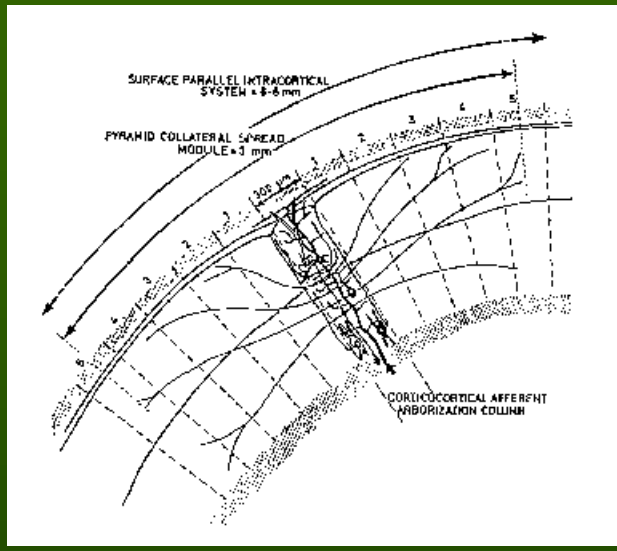


Cortical columns

Cortical columns may learn to respond to stimuli resonating in different way, with complex logic.

Liquid state machine (LSM; Maas, Markram 2004) – large spiking recurrent neural network, randomly connected. Now reservoir computing.

S(t) => LSM (x,t), spatio-temporal pattern of activations, creating separable high dimensional projections that perceptron can handle.



Blue Brain detailed simulation of minicolumn (~10K neurons, 100M synapses) is neither comprehensible nor useful. Simplification for static data:

1) Single hidden layer constructive network based on random projections.

2) Oscillators based on combination of two neurons $\sigma(W\cdot X-b) - \sigma(W\cdot X-b')$ give localized projections ⇔ specific resonant states!
Used in our **MLP2LN architecture** for extraction of logical rules from data.

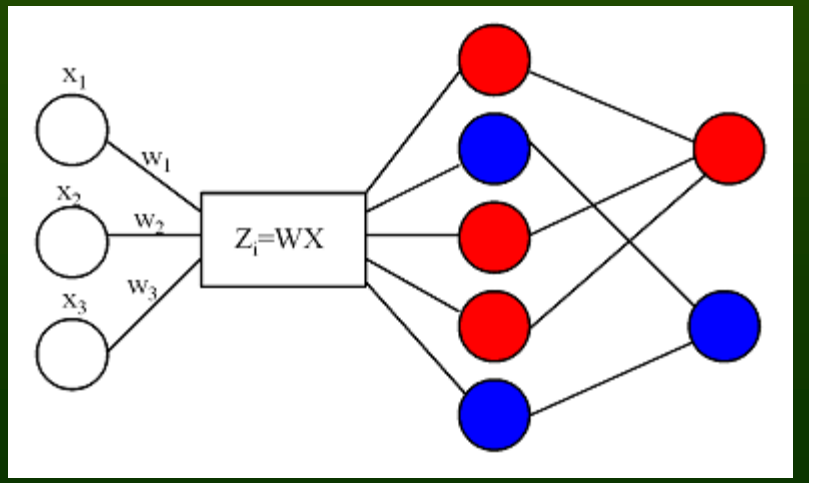
aRPM

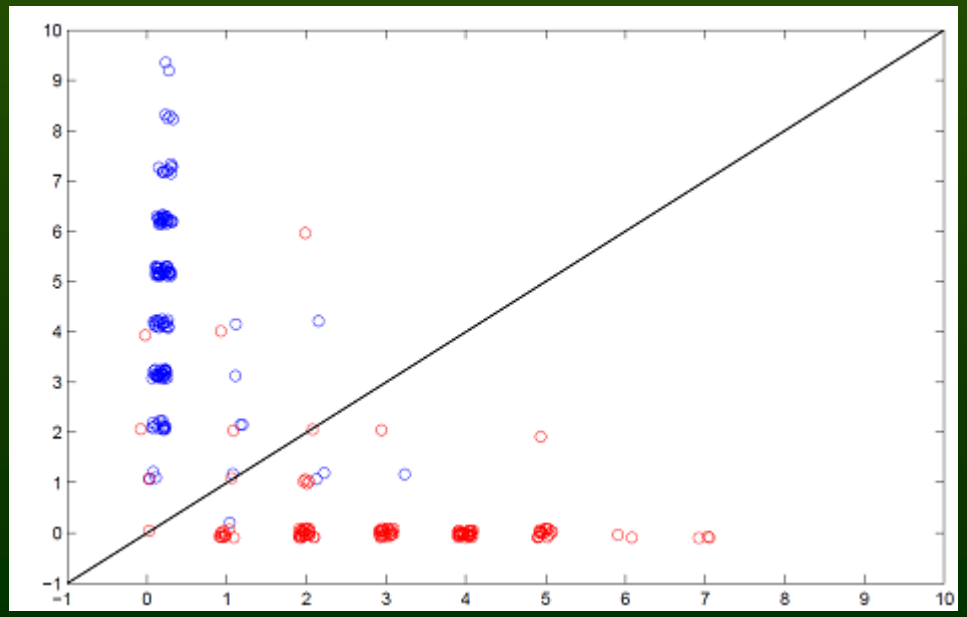
Brain has ~3 mln cortical columns, initially not tuned to input signals.

aRMP, almost Random Projection Machine (with implicit Hebbian learning):

- generate random combinations of inputs (line projection) $z(X)=W \cdot X$,
- find and isolate pure cluster $h(X)=G(z(X))$; use localized kernel on projections, estimate relevance of $h(X)$, ex. $MI(h(X),C)$,
- accept only relevant nodes, other contribute only to noise;
- continue until each input vector activates minimum k hidden nodes.

Count how many nodes vote for each class and plot: one-shot learning!
Trust those far from diagonal!

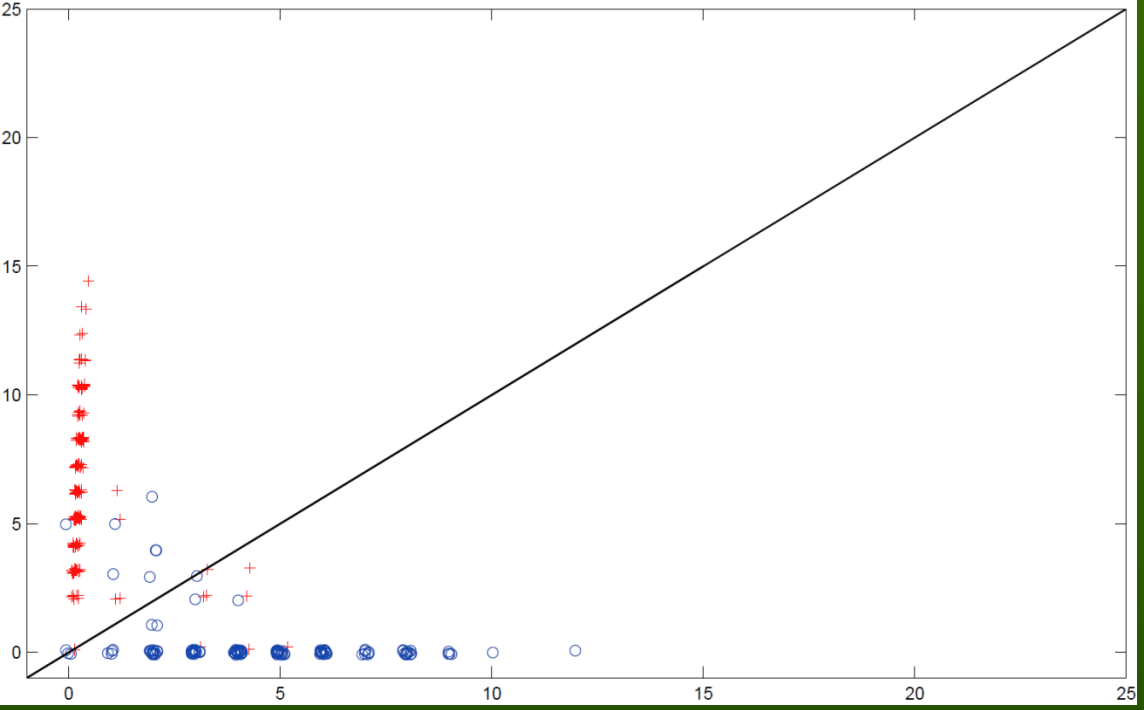




aRPM with margin optimization

Biologically plausible, many improvements are possible:

- Flexibility with kernel choices – models of specific cortical microcircuits that process signals of different modalities in the brain.
- Include competitions among minicolumns, inhibiting those that are too close to the margin, add new node only if it increases discrimination confidence, ex: using function: $F(X)=1/(1+\exp(-(A(C|X)-A(\neg C|X))))$ Total confidence in the model = sum of $F(X)$ over all vectors X.
- Use localized kernels – distribution of projected patterns along random direction W may form pure clusters with many patterns.
- Adapt local kernels if more vectors fall into accepted network nodes to cover larger clusters of inputs .
- Good candidate projection should cover some minimal number η of training vectors, so use pruning for small clusters.
- Linear optimization (LDA, SVM-L) may be used on output layer.



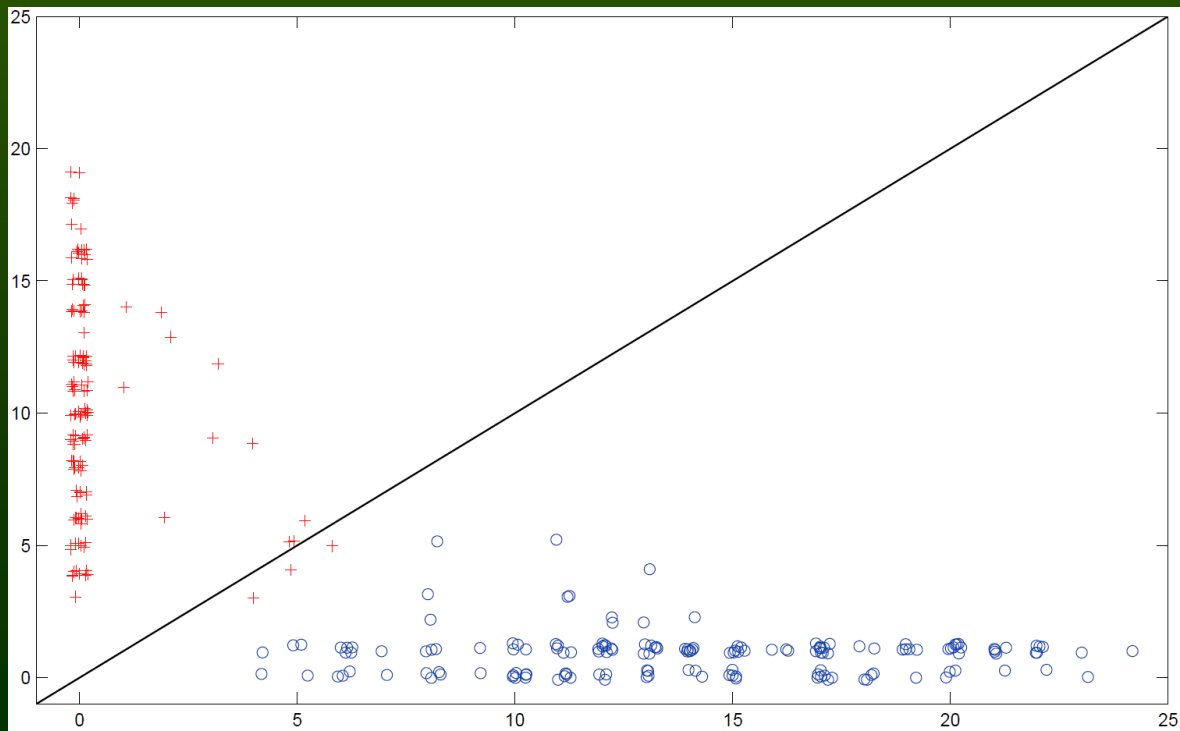
Heart Statlog data

no margin maximization

Axes: number of voting nodes for each vector;
blue o, red + = class

with margin maximization

More vectors activate many nodes from correct class, few vectors are close to the decision border (majority voting).



aRMP with localized kernels and WTA or LDA.

Simplest version of aRMP is frequently better than SVM with Gaussian or linear kernels.

Best results 15x of 27, 10 within error bounds, only 2 statistically worse (arrhytmia, car).

Why insist on backprop?

Dataset	SVML	SVMG	LOKWTA	LOKLDA
arrhythmia	**50.92±17.31**	43.36±21.47	42.00±24.19	39.10±12.98
autos	54.48±13.75	74.29±12.58	58.69±11.03	**74.36±10.40**
balance-scale	84.47±3.17	89.83±2.09	90.71±2.38	**96.46±2.62**
breast-cancer	73.27±6.10	75.67±5.35	**76.58±6.37**	75.09±1.99
breast-w	96.60±2.07	96.77±1.84	96.93±1.62	**97.21±2.13**
car	67.99±2.61	**98.90±0.90**	84.72±3.44	93.57±1.81
cmc	19.14±2.14	34.09±3.67	48.54±2.52	**51.06±4.30**
credit-a	**86.36±2.86**	86.21±2.90	82.67±4.01	84.70±4.91
credit-g	73.95±4.69	**74.72±4.03**	73.10±2.38	72.70±3.86
cylinder-bands	74.58±5.23	76.89±7.57	74.32±6.41	**80.11±7.53**
dermatology	94.01±3.54	94.49±3.88	87.97±5.64	**94.71±3.02**
diabetes	76.88±4.94	76.41±4.22	74.88±3.88	**76.95±4.47**
ecoli	78.48±5.90	84.17±5.82	82.47±3.66	**85.66±5.40**
glass	42.61±10.05	62.43±8.70	64.96±7.72	**71.08±8.13**
haberman	72.54±1.96	72.91±5.93	**76.46±4.34**	73.53±0.72
heart-c	**82.62±6.36**	80.67±7.96	81.07±7.56	81.04±5.17
heart-statlog	**83.48±7.17**	83.40±6.56	81.48±8.73	83.33±7.46
hepatitis	83.25±11.54	84.87±11.98	**89.88±10.14**	84.05±4.40
ionosphere	87.72±4.63	94.61±3.68	85.18±6.28	**95.16±2.72**
iris	72.20±7.59	**94.86±5.75**	94.67±6.89	93.33±5.46
kr-vs-kp	96.03±0.86	**99.35±0.42**	83.73±2.58	98.25±0.45
liver-disorders	68.46±7.36	**70.30±7.90**	57.40±5.72	69.72±6.57
lymph	81.26±9.79	**83.61±9.82**	76.96±13.07	80.52±7.91
sonar	73.71±9.62	86.42±7.65	**86.57±7.01**	86.52±8.39
vote	96.12±3.85	**96.89±3.11**	92.57±7.52	93.95±4.18
vowel	23.73±3.13	**98.05±1.90**	92.49±3.37	97.58±1.52
zoo	91.61±6.67	93.27±7.53	88.47±5.35	**94.07±6.97**

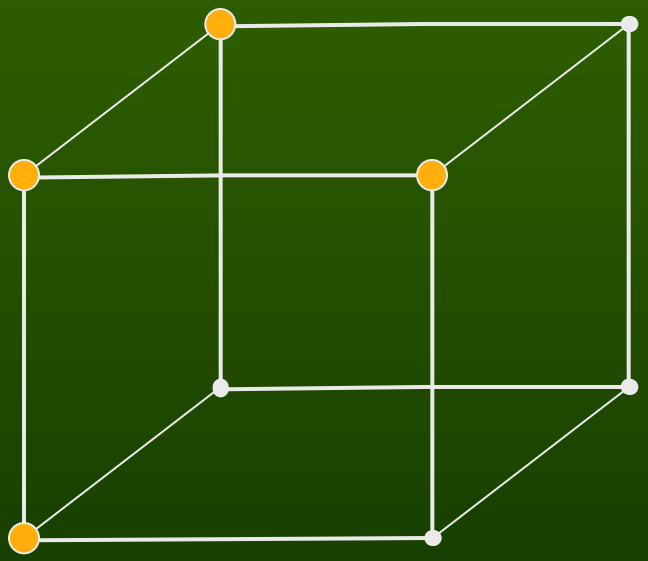
aRMP results

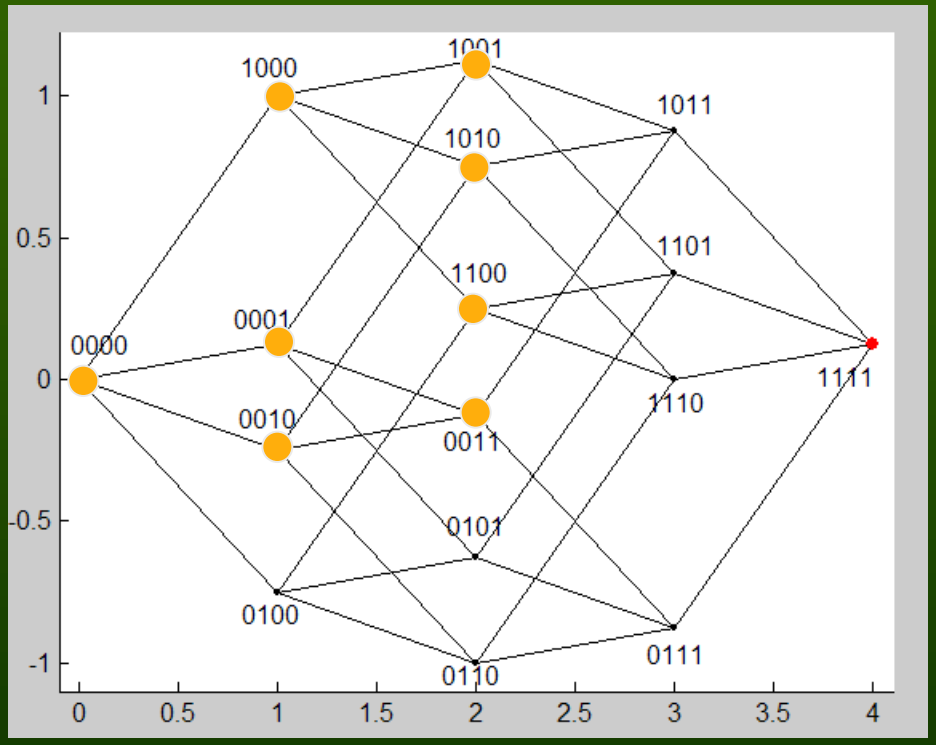
Simplest method that solves highly-non separable problems like parity!

Dataset	Method							
	NB	kNN	SSV	SVM(L)	SVM(G)	aRPM-no	aRPM (WTA)	aRPM(LDA)
Append.	83.1 ± 10.2	87.0 ± 10.6	87.9 ± 7.4	85.1 ± 6.0	85.9 ± 6.4	82.6 ± 9.3	87.7 ± 8.1	88.0 ± 6.7
Diabetes	68.1 ± 2.3	75.2 ± 4.1	73.7 ± 3.8	76.4 ±4.7	75.7 ± 5.9	67.7 ± 4.2	61.2 ± 5.7	76.7 ± 4.4
Glass	68.6 ± 9.0	69.7 ± 7.4	69.7 ± 9.4	40.2 ±9.6	63.2 ± 7.7	65.0 ± 9.9	60.3 ± 8.5	68.9 ± 8.3
Heart	76.5 ± 8.6	82.8 ± 6.7	74.7 ± 8.7	83.2 ±6.2	83.5 ± 5.3	78.3 ± 4.2	80.1 ± 7.5	83.1 ± 4.7
Liver	58.6 ± 3.8	62.6 ± 8.5	68.9 ± 9.7	68.4 ±5.9	69.0 ± 8.4	61.1 ± 5.1	67.5 ± 5.5	72.7 ± 7.9
Wine	98.3 ± 2.6	94.9 ± 4.1	89.4 ± 8.8	96.0 ± 5.9	97.8 ± 3.9	68.6 ± 7.8	94.3 ± 5.8	97.7 ± 4.0
Parity8	28.9 ± 4.6	100 ± 0	49.2 ± 1.0	34.1 ±11.7	15.6 ± 22.7	99.2 ± 1.6	100 ± 0	34.7 ± 3.8
Parity10	38.1 ± 3.3	100 ± 0	49.8 ± 0.3	44.1 ±5.0	45.6 ± 4.3	99.5 ± 0.9	100 ± 0	40.3 ± 2.7

What can be learned?

Linearly separable or almost separable problems are relatively simple.
Make it cheap: learning with O(*nd*) complexity (2012), 2/3 of UCI is easy.
Deform planes or add extra dimensions to make data separable.



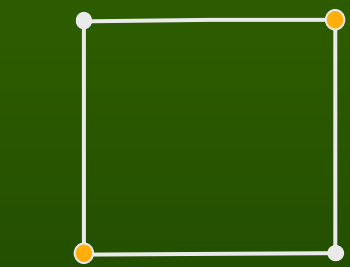


How to define “slightly non-separable”, or relatively easy to learn?
Now we have only separable problems and one vast realm of the rest.

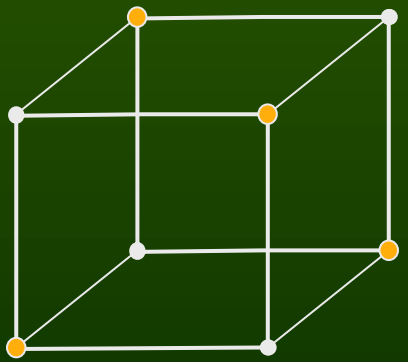
Neurons learning complex logic

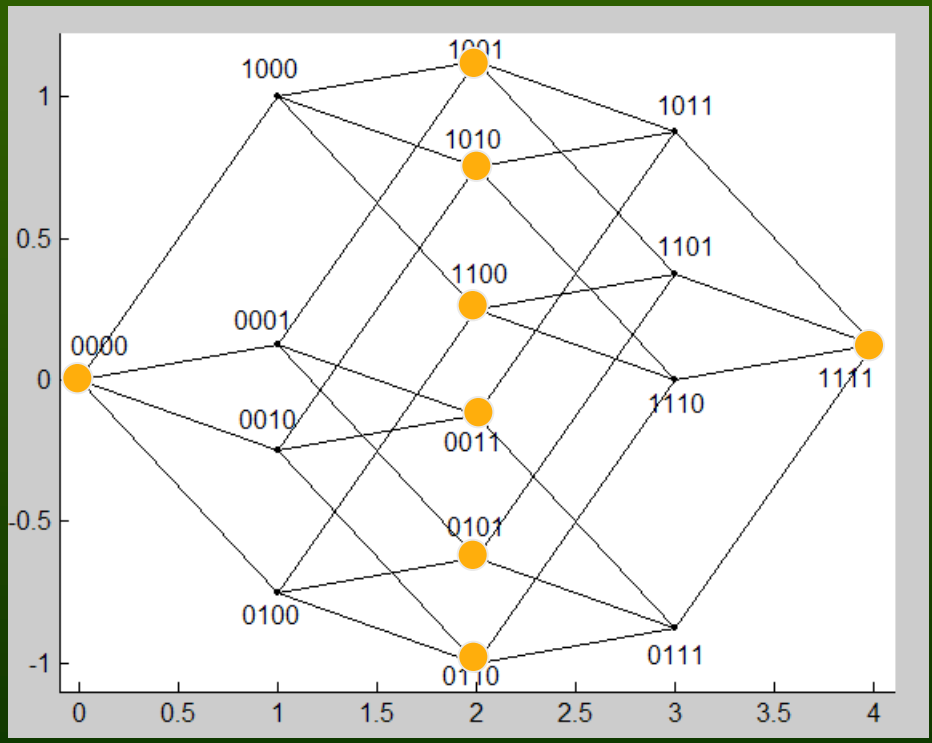
Boole'an functions are difficult to learn, n bits but 2^n nodes => combinatorial complexity; similarity is not useful, for parity all neighbors are from the wrong class. MLP networks have difficulty to learn functions that are highly non-separable.

Ex. of 2-4D parity problems.



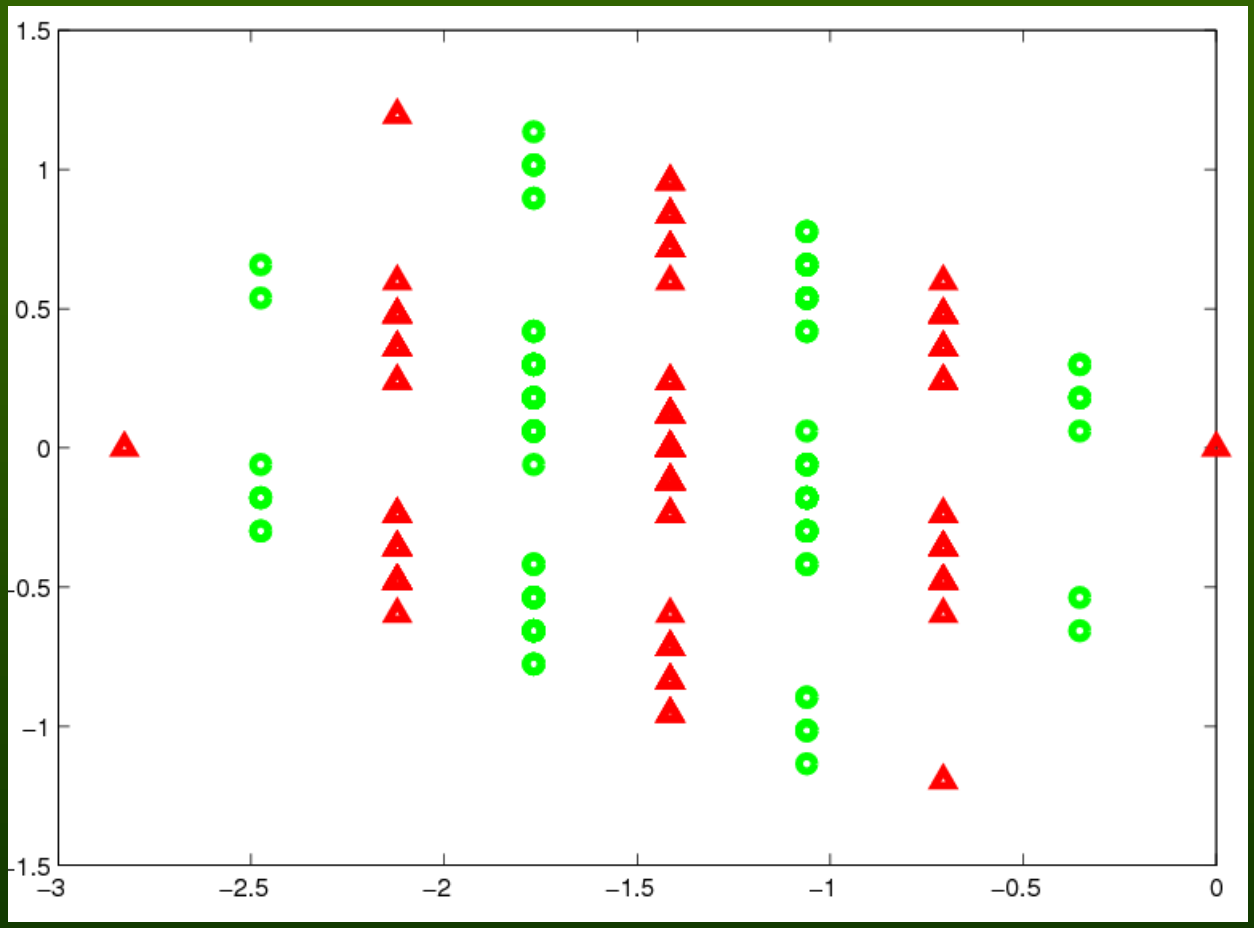
Neural logic can solve it without counting; find a good point of view.





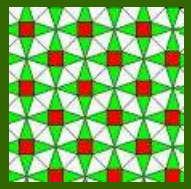
Projection on W=(111 ... 111) gives clusters with 0, 1, 2 ... n bits; solution requires abstract imagination + easy categorization.

Interval transformation



8-bit parity data: 9-separability is much easier to achieve than full linear separability; almost impossible to train MLP on such data.

Boolean functions



n=2, 16 functions, 12 separable, 4 not separable.

n=3, 256 f, 104 separable (41%), 152 not separable.

n=4, 64K=65536, only 1880 separable (3%)

n=5, 4G, but << 1% separable ... bad news!

n=8, 10^{77} functions! How many separable or nearly separable?
Bioinformatics or neuroimaging data may require n >100.

Existing methods may learn some non-separable functions,
but most functions cannot be learned !

Example: *n*-bit parity problem; many papers in top journals.

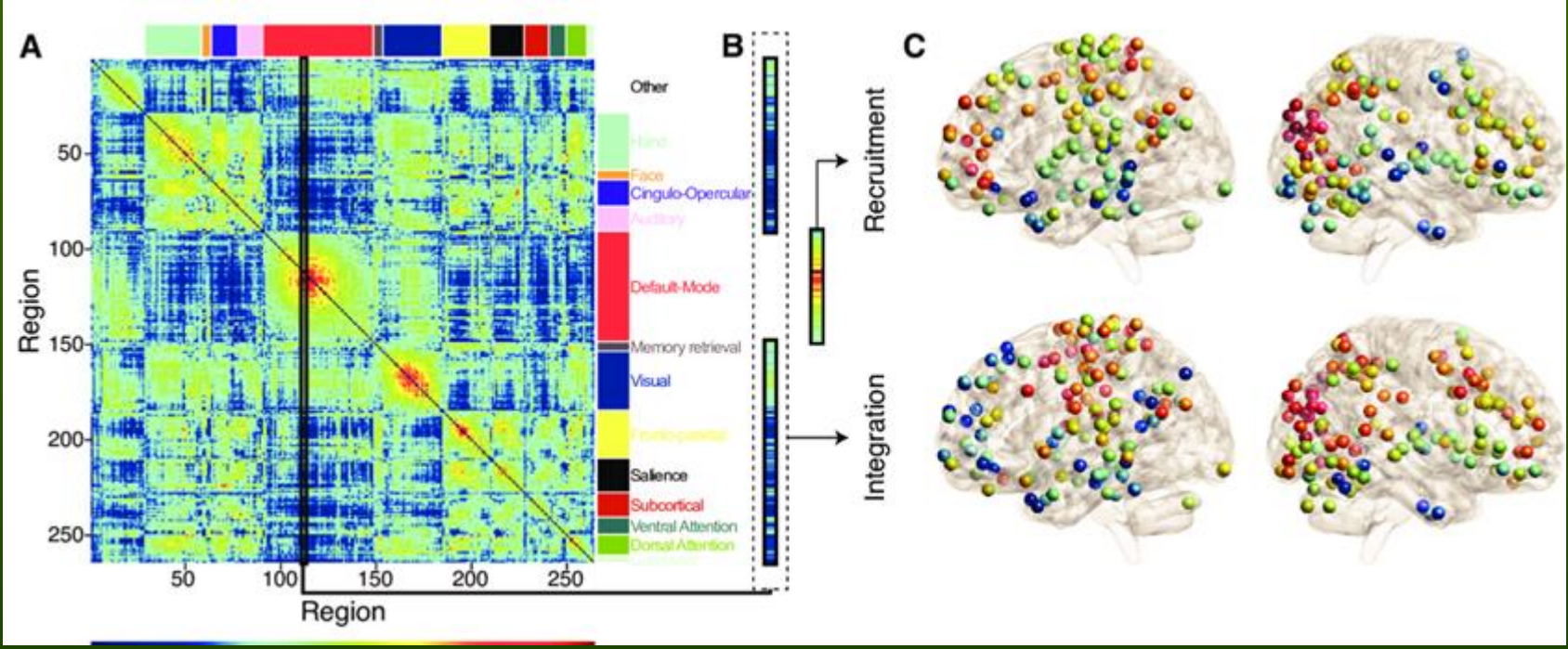
No off-the-shelf systems are able to solve such problems.

For all parity problems SVM with typical kernels fails completely!

Such problems are solved only by special neural architectures or special classifiers – if the type of function is known.

But parity is still trivial ... solved by $y = \cos\left(\omega \sum_{i=1}^{n} b_i\right)$

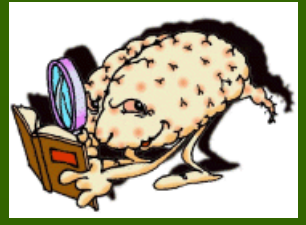
Much more complex logic …



Different activations => same cognitive functions in different context?
Complex logic may be needed to learn from this type of data.

Mattar, M. G, Cole, M. W, Thompson-Schill, S. L, & Bassett, D. S. (2015).
A **Functional Cartography of Cognitive Systems**.
PLOS Computational Biology, *11*(12), e1004533.

Goal of learning



If simple topological deformation of decision borders is sufficient linear separation is achieved in high dimensional spaces, "flattening" non-linear decision borders; this is frequently the case in pattern recognition problems. RBF/MLP networks with one hidden layer solve the problem.

For complex logic this is not sufficient; networks with localized functions need exponentially large number of nodes.

Such situations arise in AI reasoning problems, real perception, object recognition, text analysis, bioinformatics ...

Linear separation is too difficult, set an easier goal.
Linear separation: projection on 2 half-lines in the kernel space:
line $y=WX$, with $y<0$ for class – and $y>0$ for class +.

1) Simplest extension: **separation into k-intervals, or k-separability**.
For parity: find direction W with minimum # of intervals, $y=W\cdot X$
2) Or: try to transform data to a distribution that can be easily analyzed.

k-sep learning

Try to find lowest k with good solution:

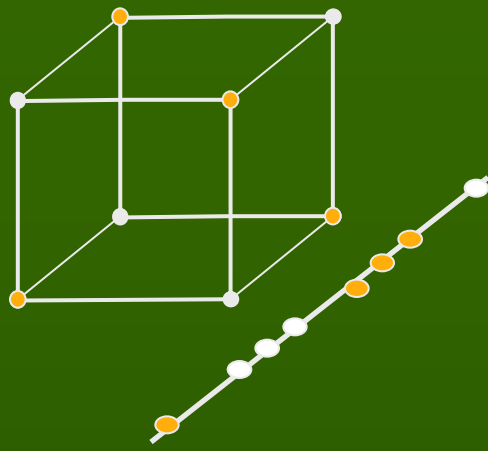
- Assume k=2 (linear separability), try to find a good solution; MSE error criterion
- $$E(\mathbf{W},\theta)=\sum_{\mathbf{X}}\left(y(\mathbf{X};\mathbf{W})-C(\mathbf{X})\right)^2$$
- if k=2 is not sufficient, try k=3; two possibilities are C_+,C_-,C_+ and C_-, C_+, C_- this requires only one interval for the middle class;
- if k<4 is not sufficient, try k=4; two possibilities are C_+, C_-, C_+, C_- and C_-, C_+, C_-, C_+ this requires one closed and one open interval.

Network solution ⇔ to minimization of specific cost function.

$$E(\mathbf{W},\lambda_1,\lambda_2)=\sum_{\mathbf{X}}\left(y(\mathbf{X};\mathbf{W})-C(\mathbf{X})\right)^2+\lambda_1\sum_{\mathbf{X}}\left(1-C(\mathbf{X})\right)y(\mathbf{X};\mathbf{W})$$
$$-\lambda_2\sum_{\mathbf{X}}C(\mathbf{X})y(\mathbf{X};\mathbf{W})$$

First term = MSE, second penalty for “impure” clusters, third term = reward for the large clusters.

3D case



3-bit functions: X=[$b_1 b_2 b_3$], from [0,0,0] to [1,1,1]

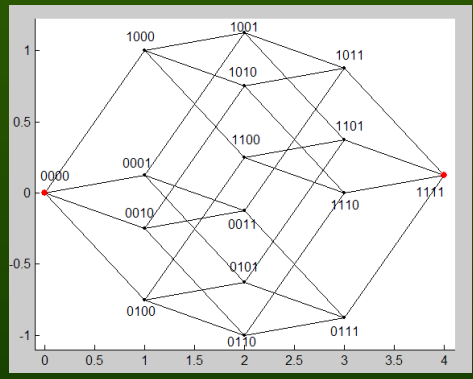
f(b_1,b_2,b_3) and ¬f(b_1,b_2,b_3) are symmetric (color change)

8 cube vertices, 2^8=256 Boolean functions.

0 to 8 red vertices: 1, 8, 28, 56, 70, 56, 28, 8, 1 functions.

For optimized direction **W** on all 2^8 functions index projection W·X gives:

k=1 in 2 cases, all 8 vectors in 1 cluster (8 black or white)
k=2 in 14 cases, 8 vectors in 2 clusters (linearly separable)
k=3 in 42 cases, clusters E O E or O E O Even, Odd
k=4 in 70 cases, clusters E O E O or O E O E
Symmetrically, k=5-8 for 70, 42, 14, 2 cases.



Most logical functions have 4 or 5-separable projections.

4-bit functions: 16 cube vertices, 2^{16}=65636=64K functions. $2^{2\wedge N}$

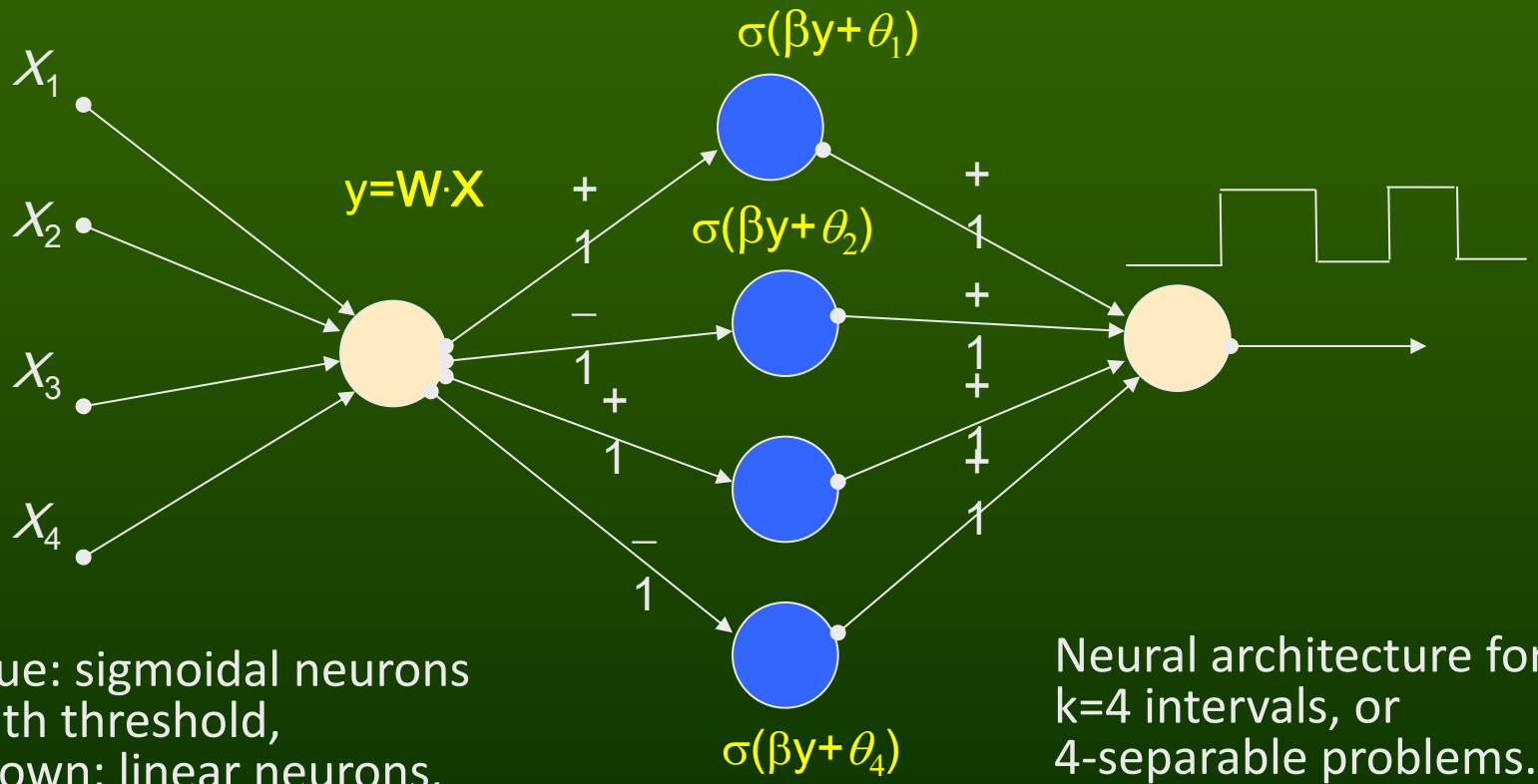
Learning = find best set of projections, all vectors falling into large clusters.

Enforcing separability on k-separable data is hard for k>2,
but achieving k-separability as a goal makes learning easier.

k-separability

Can one learn all Boolean functions?

Problems may be classified as 2-separable (linear separability); non separable problems may be broken into k-separable, k>2.



Blue: sigmoidal neurons with threshold, brown: linear neurons.

Neural architecture for k=4 intervals, or 4-separable problems.

QPC, Projection Pursuit

What is needed to learn data with complex logic?

- cluster non-local areas in the X space, use W·X
- capture local clusters after transformation, use G(W·X-θ)

SVMs fail because the number of directions W that should be considered grows exponentially with the size of the problem *n*.

What will solve it? Projected clusters (M. Grochowski PhD)!

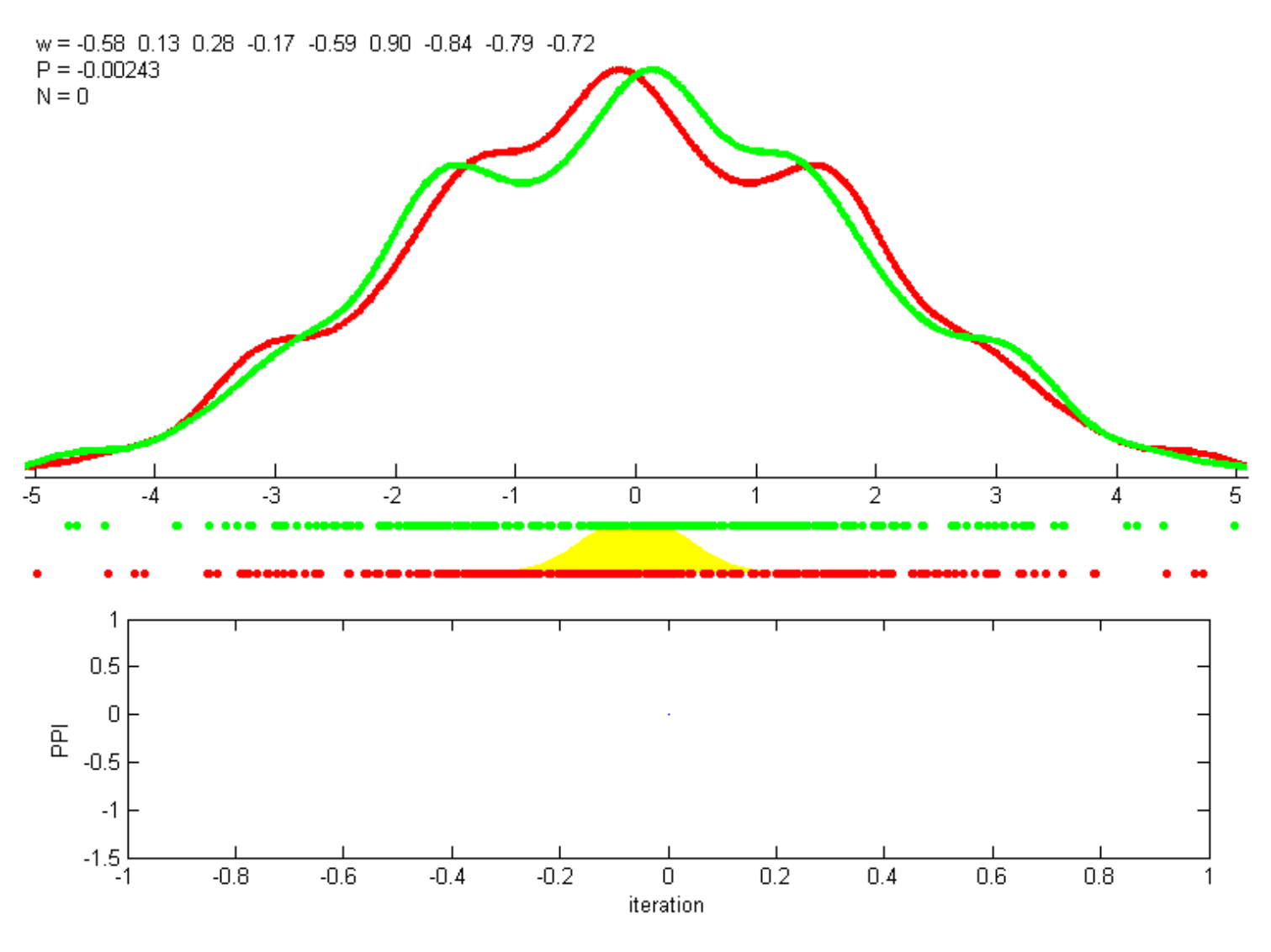
1. A class of constructive neural network solution with G(W·X-θ) functions combining non-local/local projections, with special training algorithms.
2. Maximize the leave-one-out error after projection: take some localized function *G*, count in a soft way cases from the same class as X_k.

$$Q(\mathbf{W}) = \sum_{\mathbf{X}} \left[A^{+} \sum_{\mathbf{X}_k \in C} G\left(\mathbf{W}\cdot\left(\mathbf{X}-\mathbf{X}_k\right)\right) - A^{-} \sum_{\mathbf{X}_k \notin C} G\left(\mathbf{W}\cdot\left(\mathbf{X}-\mathbf{X}_k\right)\right) \right]$$

Grouping and separation; projection may be done directly to 1 or 2D for visualization, or higher D for dimensionality reduction, if W has *d* columns.

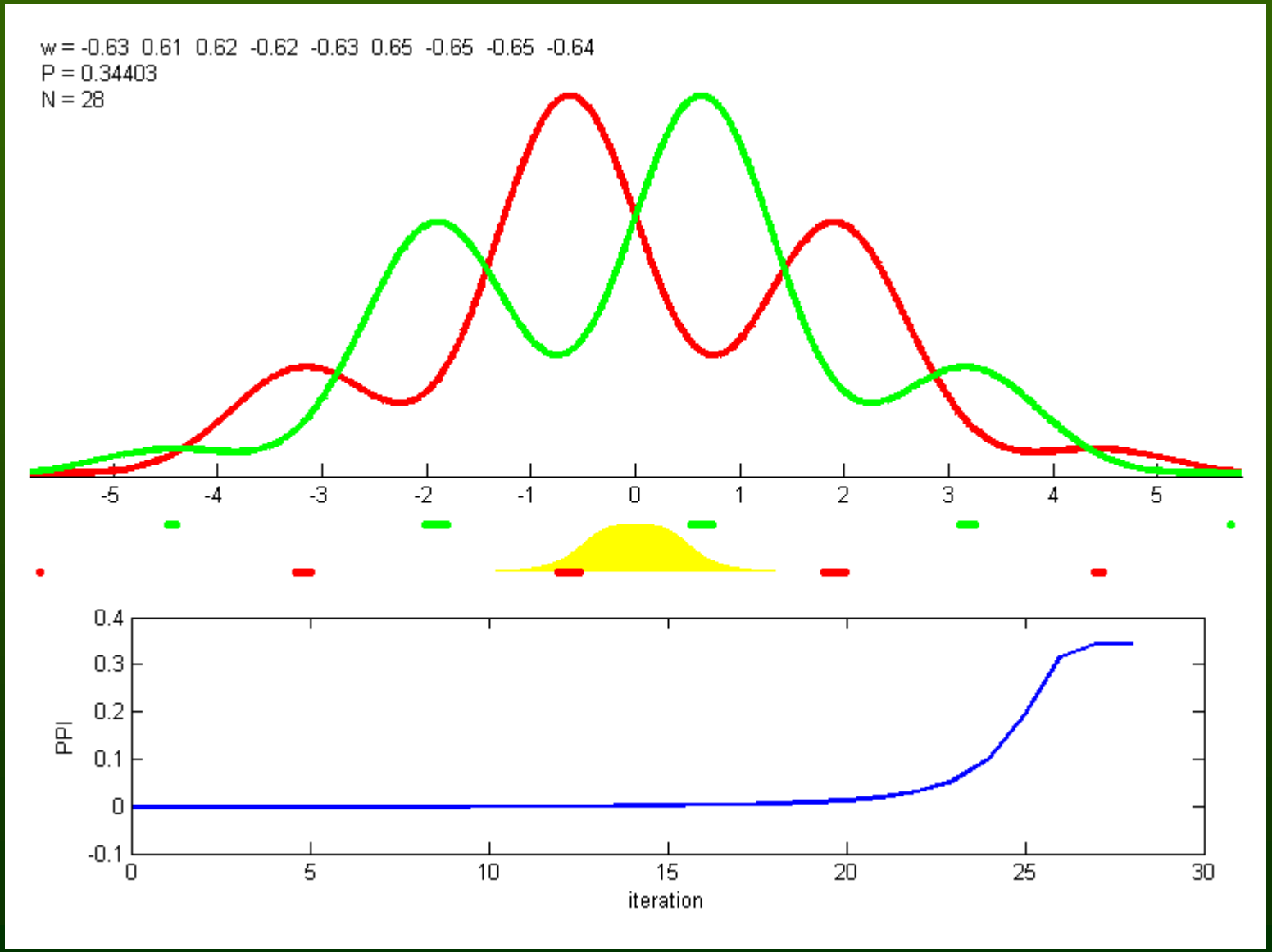
Parity n=9

Simple gradient learning; QPC index shown below.

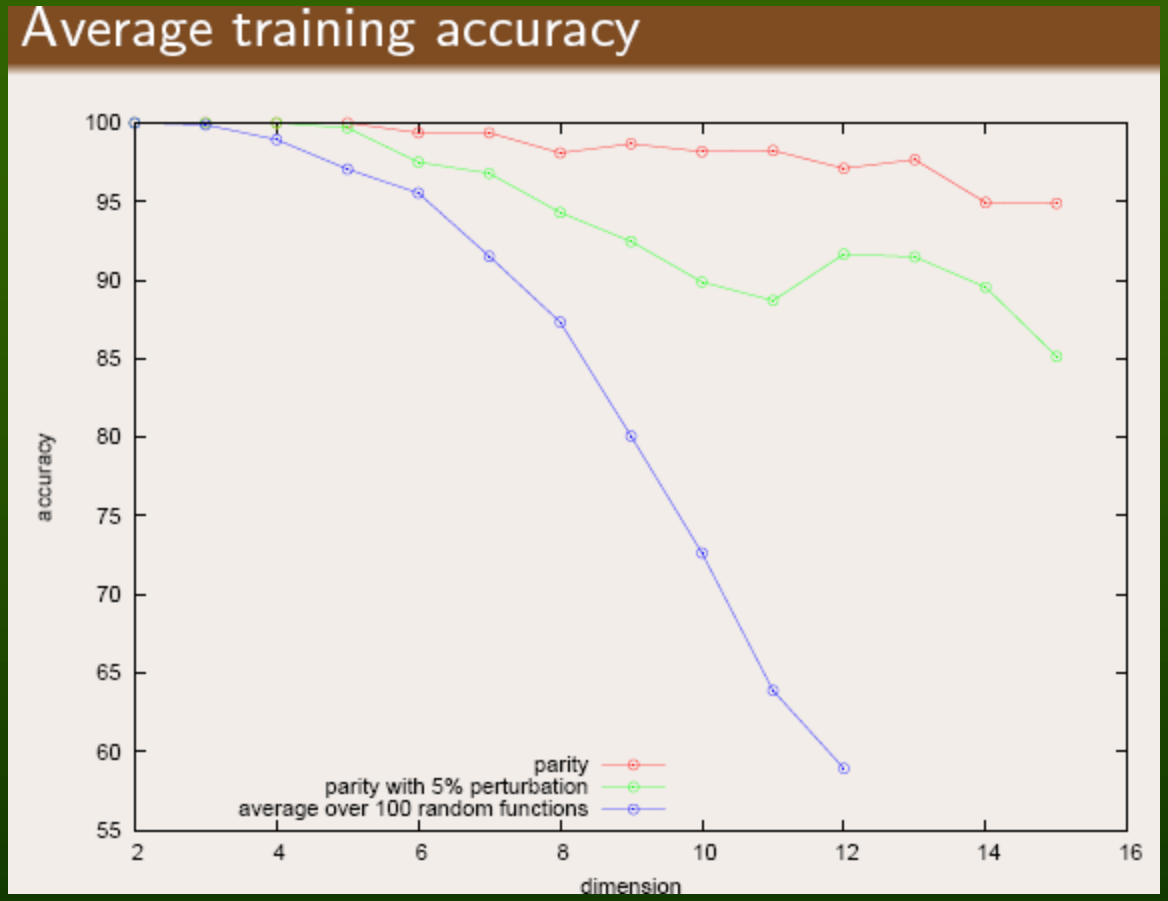


Parity n=9

Simple gradient learning; QPC index shown below.

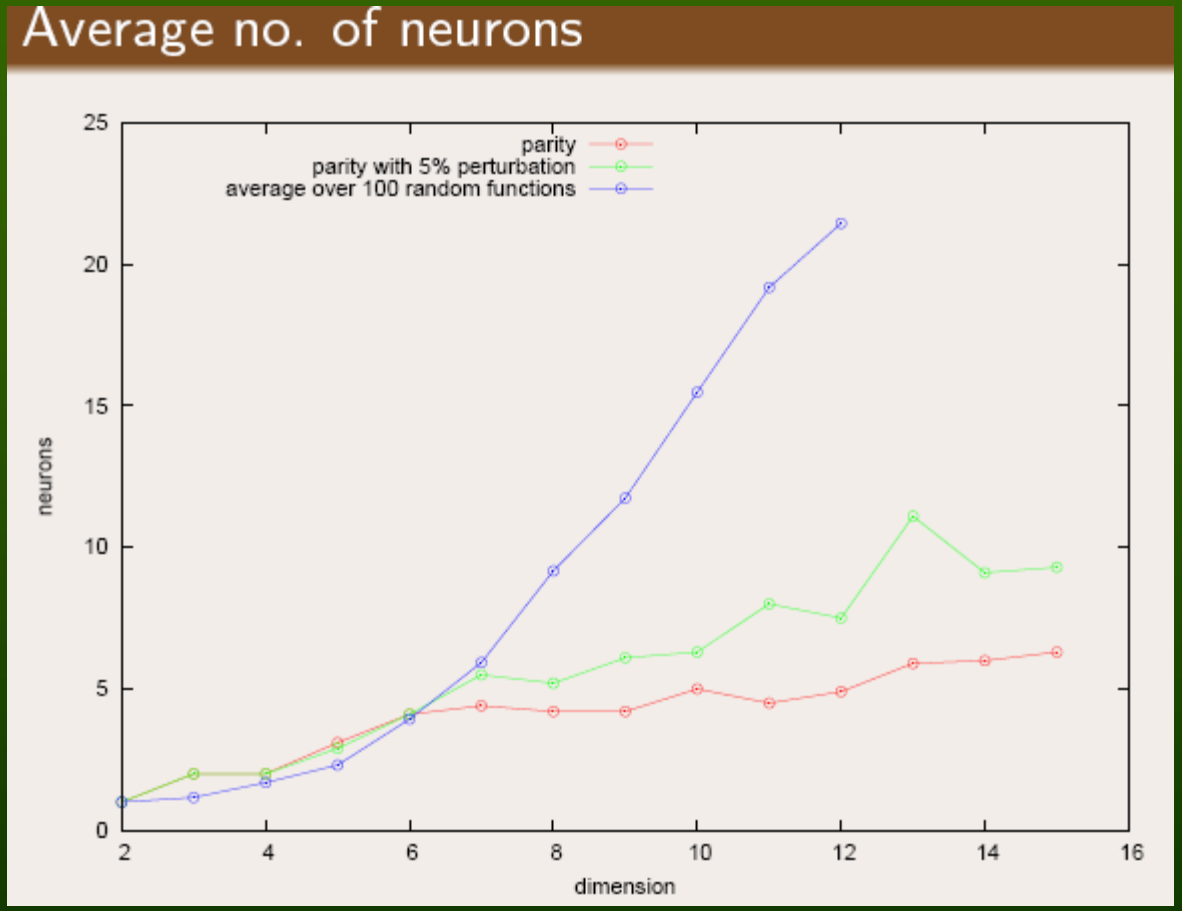


Learning hard functions



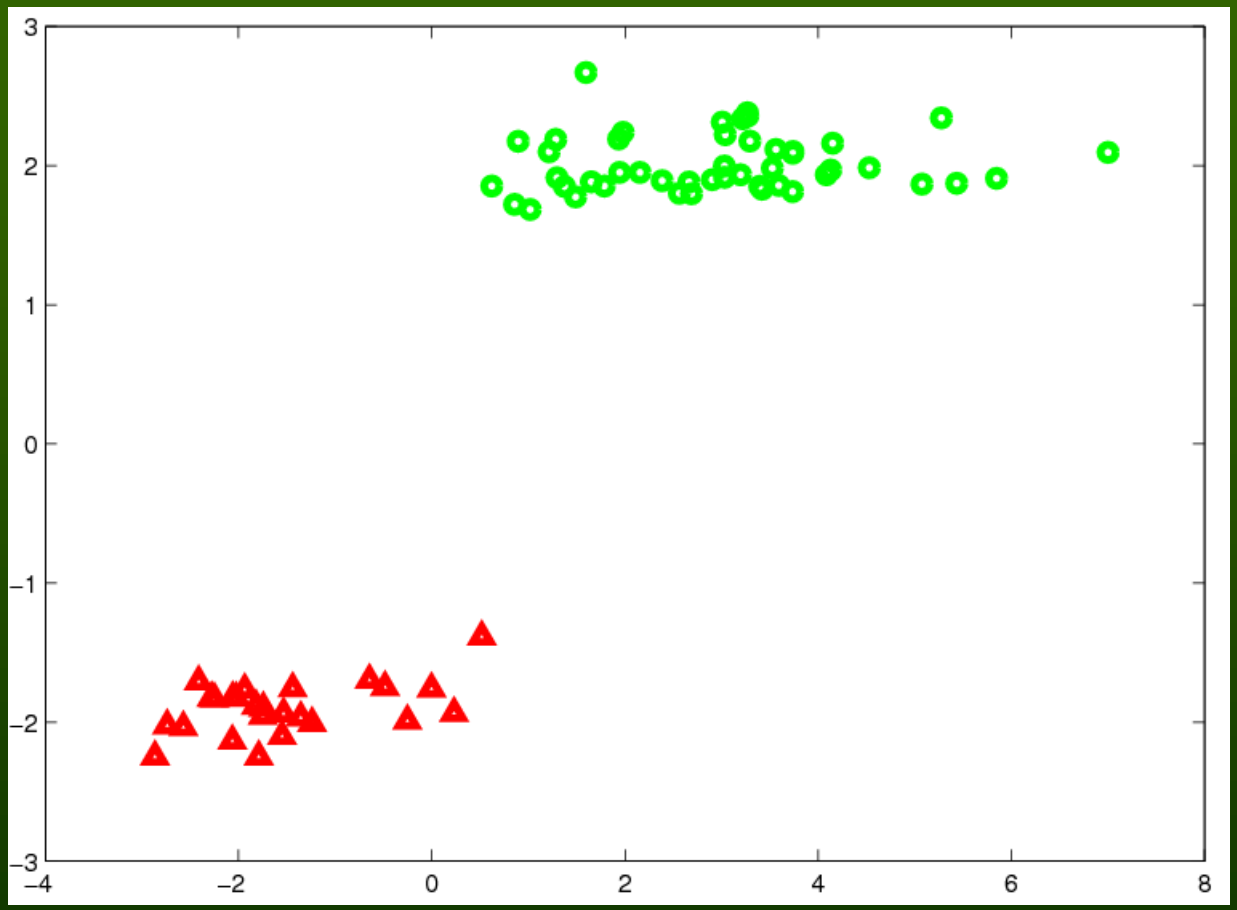
Training almost perfect for parity, with linear growth in the number of vectors for k-sep. solution created by the constructive neural algorithm.

Learning hard functions



Training almost perfect for parity, with linear growth in the number of vectors for k-sep. solution created by the constructive neural algorithm.

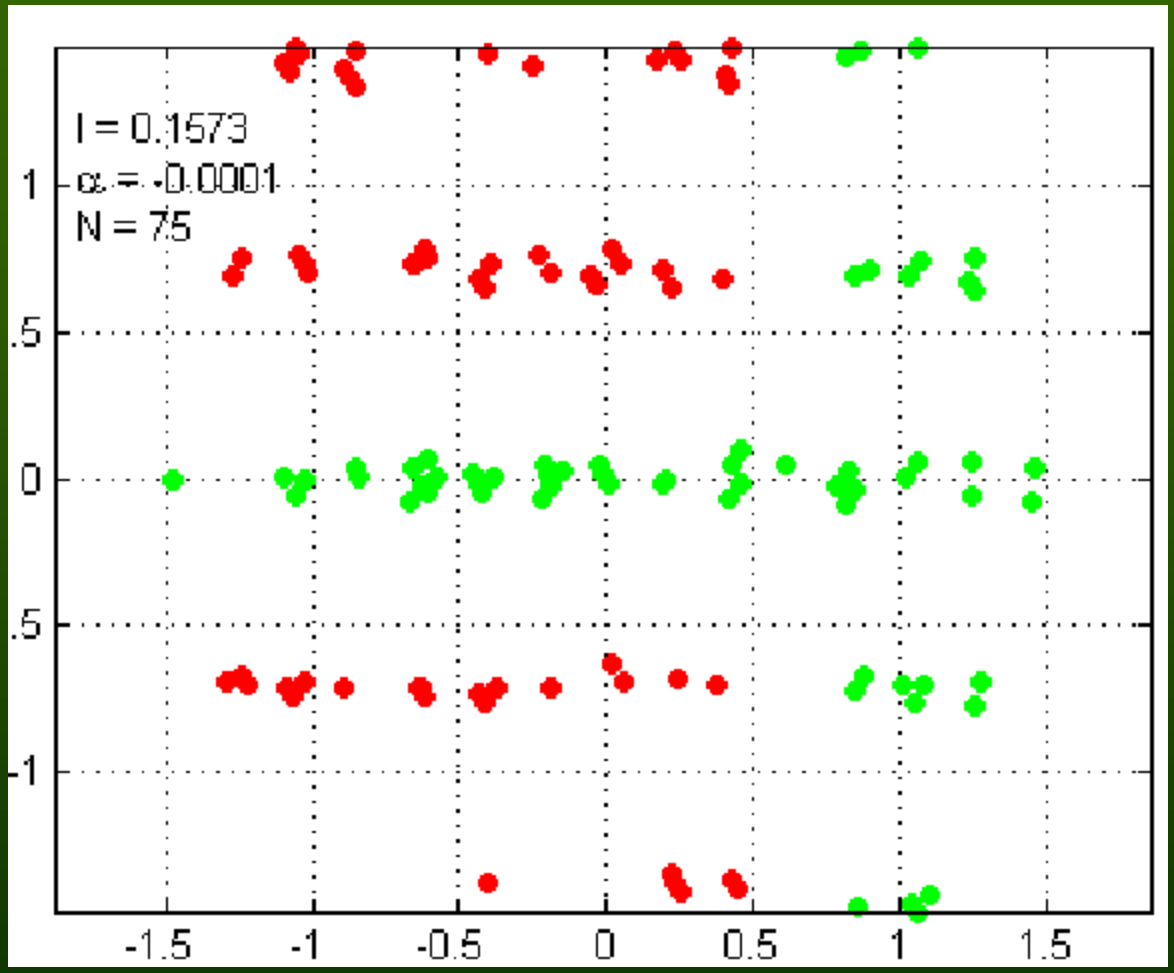
Linear separability



SVM visualization of Leukemia microarray data,

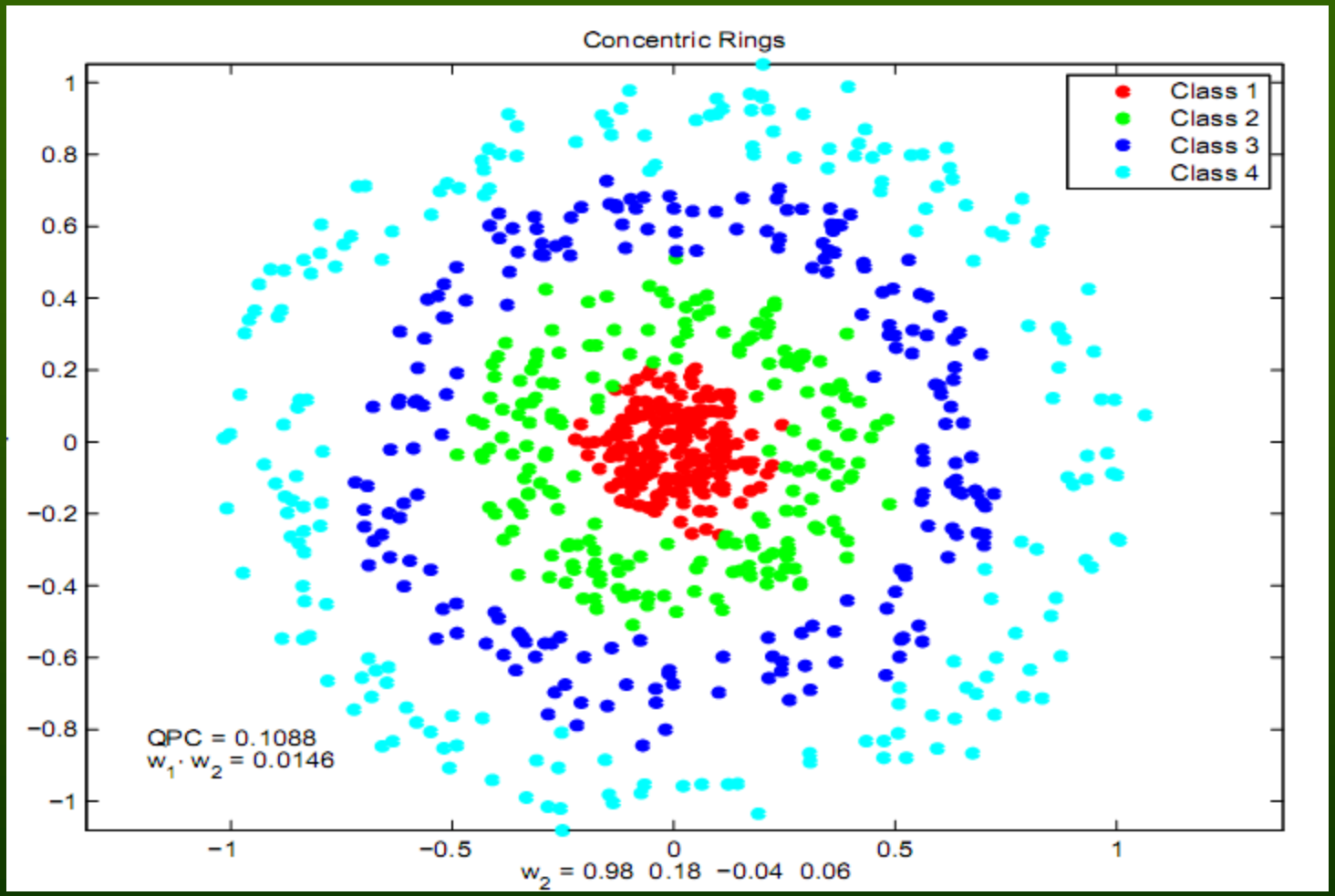
Horizontal axis x=WX, vertical - orthogonal projection.

Rules



QPC visualization of Monks dataset with simple logical structure, two logical rules are needed, or combination of two projections.

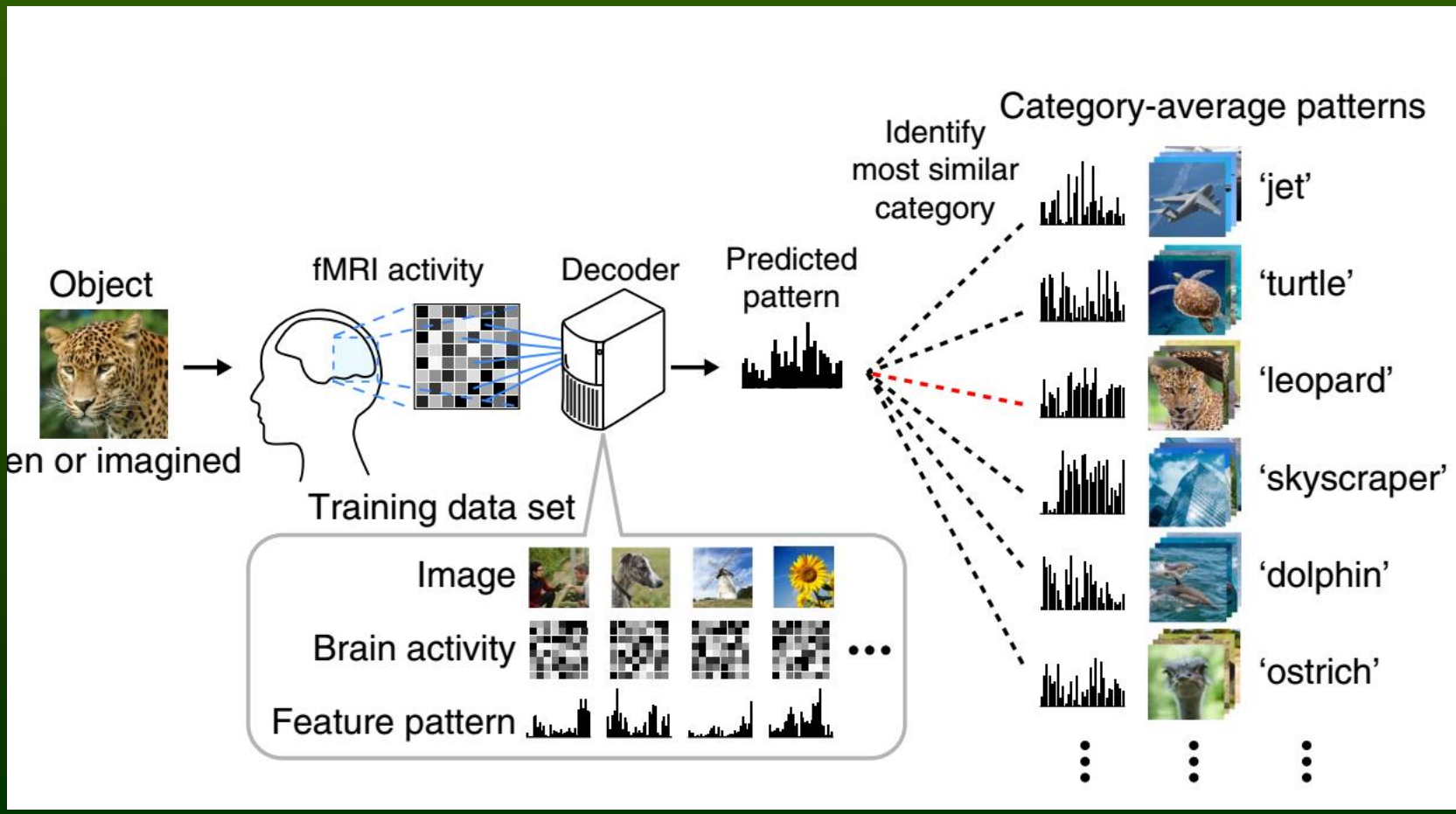
Circular distribution



QPC visualization of concentric rings in 2D with strong noise in remaining 2D; transform: nearest neighbor solutions, combinations of ellipsoidal densities.

Brain activity ⇔ Mental image

fMRI voxel activity (2^3 mm) can be correlated with deep CNN network features; using these features closest image from large database is selected.
Horikawa, Kamitani, Generic decoding of seen and imagined objects using hierarchical visual features. Nature Communications 2017.



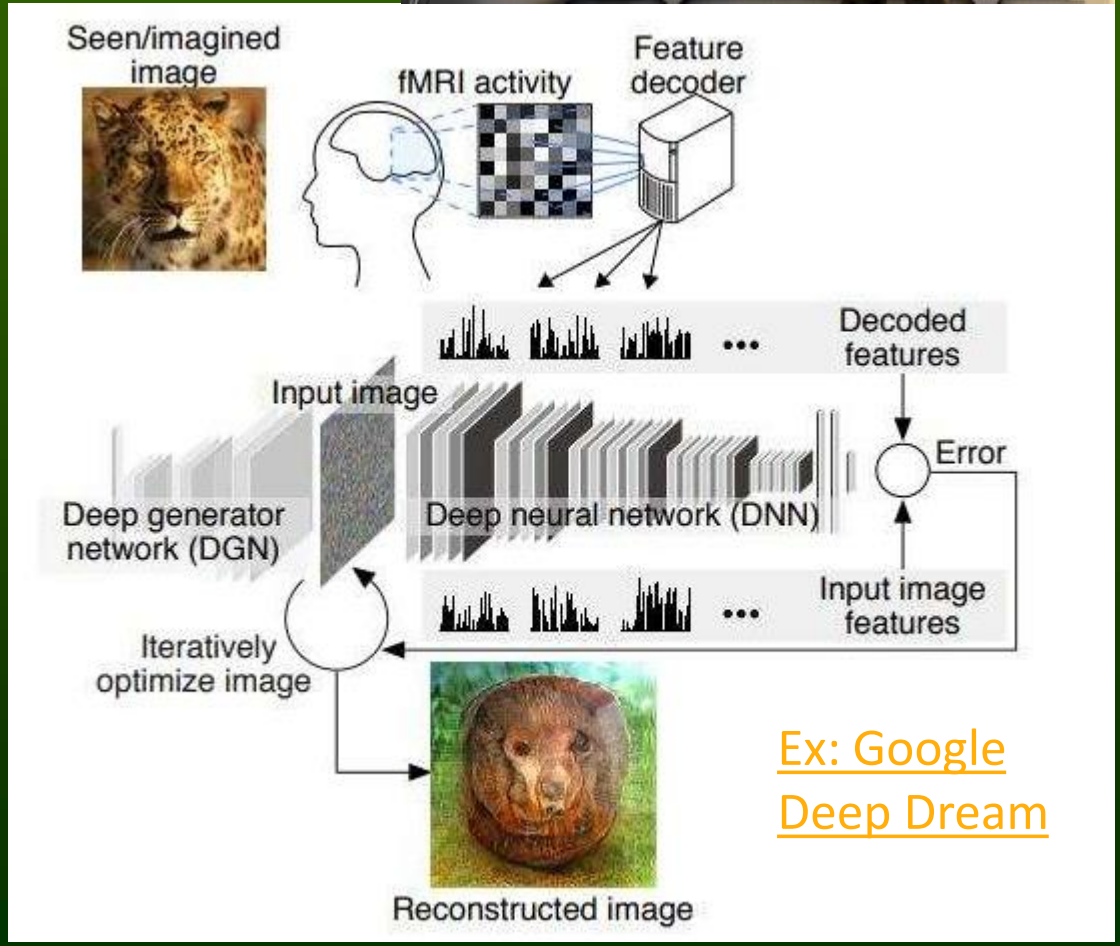
fMRI ⇔ CNN

Convert activity of the brain into the mental images that we are conscious of.

Try to estimate features from patterns at different brain areas/cortical layers.

Ex: 8-layer convolution network, ~60 mln parameters, feature vectors from randomly selected 1000 units in each layer are used to represent images at different level of processing.

Output: DGN creates vector of features that is used to reconstruct image.



Ex: Google Deep Dream

Support Feature Machines

General principle: complementarity of information processed by parallel interacting streams with hierarchical organization (Grossberg, 2000).

Cortical minicolumns provide various features for higher processes.

Create information that is easily used by various ML algorithms: explicitly build enhanced space adding more transformations.

- X , original features
- Z=WX, random linear projections, other projections (PCA< ICA, PP)
- Q = optimized Z using Quality of Projected Clusters or other PP techniques.
- H=[Z_1,Z_2], intervals containing pure clusters on projections.
- K=K(X,X_i), kernel features.
- HK=[K_1,K_2], intervals on kernel features

Kernel-based SVM ⇔ linear SVM in the explicitly constructed kernel space, enhancing this space leads to improvement of results.
LDA is one option, but many other algorithms benefit from information in enhanced feature spaces; best results in various combination X+Z+Q+H+K+HK.

SFM vs SVM

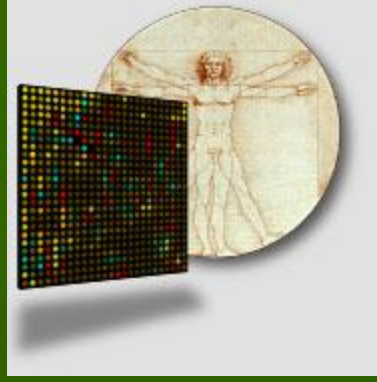
SFM generalize SVM approach by explicitly building feature space: enhance your input space adding kernel features $z_i(X)=K(X;SV_i)$

+ any other useful types of features. SFM advantages comparing to SVM:

- Kernel-based SVML allows for various feature selection methods, local kernel optimization, use of arbitrary classifiers – construct new feature spaces!
- Linear discrimination on explicit representation of features
  = easy interpretation of SFM functions as combination of local similarity evaluations (biologically plausible).

Dataset	K	H	K+H	Z+H	K+H+Z
Appendicitis	86.8±11	89.8±7.9	89.8±7.9	89.8±7.9	89.8±7.9
Diabetes	77.6±3.1	76.7±4.3	79.7±4.3	79.2±4.5	77.9±3.3
Heart	81.2±5.2	84.8±5.1	80.6±6.8	83.8±6.6	78.9±6.7
Hepatitis	82.7±6.6	83.9±5.3	83.9±5.3	83.9±5.3	83.9±5.3
Ionosphere	94.6±4.5	93.1±6.8	94.6±4.5	93.0±3.4	94.6±4.5
Parity8	11±4.3	99.2±1.6	97.6±2.0	99.2±2.5	96.5±3.4
Sonar	83.6±12.6	66.8±9.2	82.3±5.4	73.1±11	87.5±7.6

Universal Learning Machines



ULM is composed from two main modules:

- feature constructors,
- simple classifiers.

In machine learning features are used to calculate:

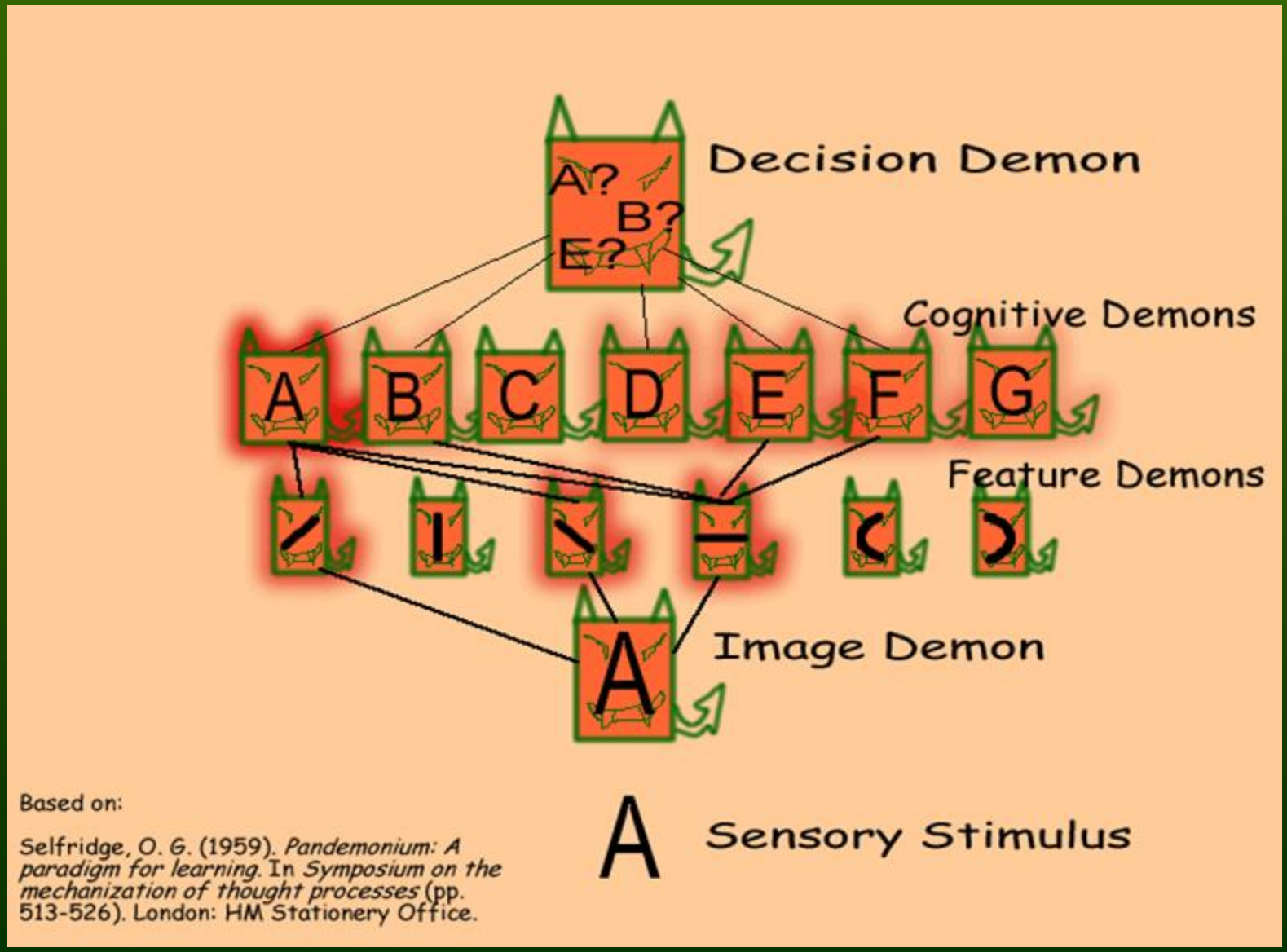
- linear combinations of feature values,
- calculate distances (dissimilarites), scaled (includes selection)

Is this sufficient?

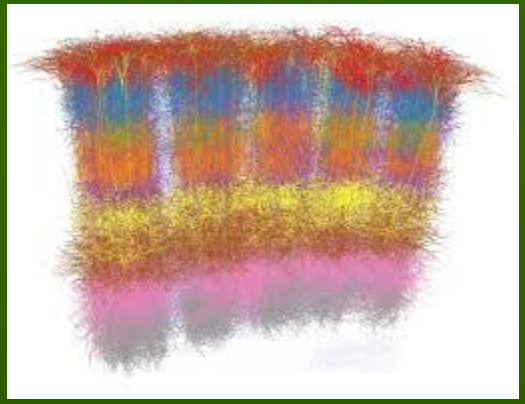
- No, non-linear functions of features carry information that cannot be easily recovered by CI methods.
- Kernel approaches: linear solutions in the kernel space, implicitly add new features based on similarity $K(X,S_V)$.
- ULM idea: create potentially useful, redundant set of futures.
  How? Learn what other models do well! Implement transfer learning.
- Learn from others, not only on your own errors! Cf. pre-reining in GPT-3.

From simple neurons to neural ensembles

Selfridge's Model (1959)

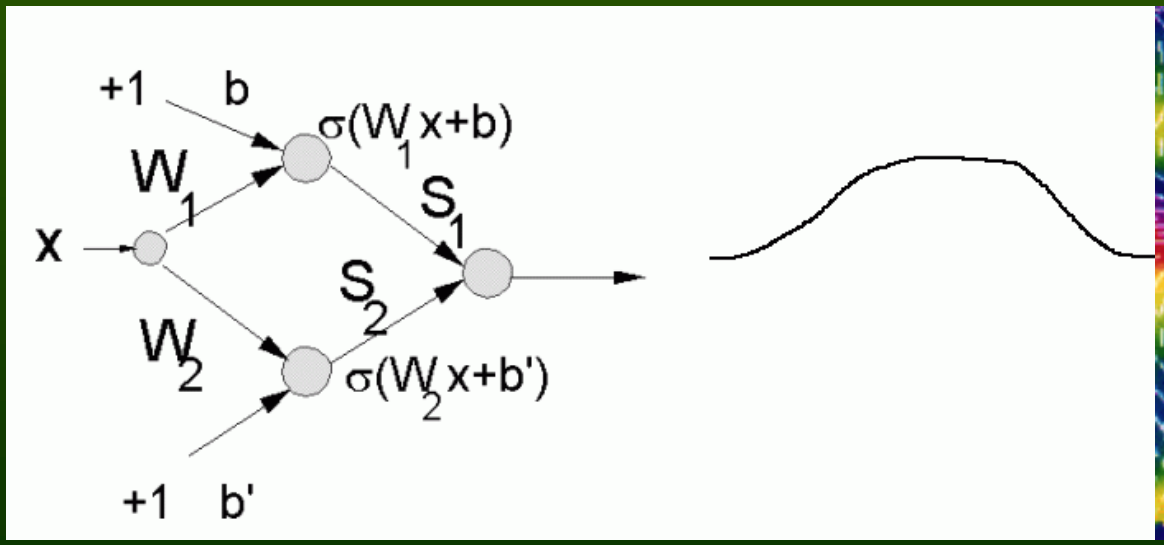


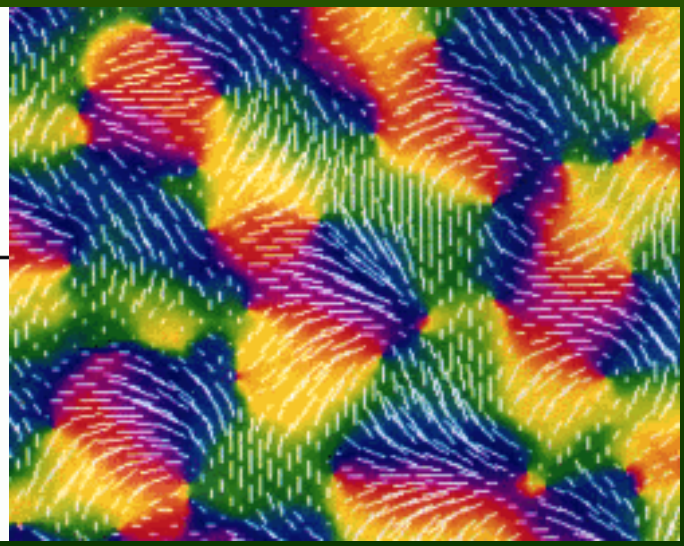
Brain Networks



Neural Cell Ensembles (NCE) or neuronal ensembles, introduced by D. Hebb, 1949.

Two neurons create soft trapezoidal function, bicentral transfer function. Ensembles of bicentral neurons that implement localized functions G(W*X), model resonances in cortical columns. Ex: primary visual cortex has columns reacting to orientation and ocular dominance (Hubel and Wiesel, Nobel 1981).





Basis: modules, not neurons, but still fixed connections.

Taxonomy - TF

Bicentral (2Slope, Rot2Slope, ...) (29, 30, [12])
Act: $A2$–$A4$, Out: $\prod(Ai^-, Ai^+, \sigma)$

G-Conic (27)
Act: $I + D^b$, Out: σ

G-Ridella (28)
Act: $I^+ + D^+$, Out: σ

Bicentral (25,26)
Act: $A1, A3$, Out: $\prod(Ai^-, Ai^+, \sigma)$

Conic (22)
Act: $I + D$, Out: σ

Ridella (21)
Act: $I^+ + D^+$, Out: σ

C_{GL1} (23)
Act: $I + D$, Out: $\frac{1}{1+A}$

C_{GL1} (23)
Act: $I + D$, Out: $\frac{1}{1+A}$

Multivariate Gaussian (13)
Act: D^b, Out: G

Multivariate Sigmoid (14)
Act: D^b, Out: σ

$\bar{G}_2$ (15)
Act: D_i, Out: $\prod \frac{1}{1+A}$

$\bar{G}_3$ (16)
Act: D_i, Out: $\frac{1}{1+\sum A}$

Gaussian-bar (17)
Act: D^b, Out: $\sum G$

Sigmoidal-bar (18)
Act: D^b, Out: $\sum \sigma$

Lorentzian (19)
Act: I, Out: $\frac{1}{1+\sum A}$

Window (20)
Act: I, Out: G

Gaussian (11)
Act: D, Out: G

Radial coordinate (8)
Act: D, Out: A

Multiquadratics (9)
Act: D, Out: $(b^2 + D^2)^\alpha$

Thin-plate spline (10)
Act: D, Out: $(bD)^2 \ln(bD)$

Gaussian Approximations (12)
Act: D, Out: $G_1 = 2 - 2\sigma(r^2)$, $G_2 = \tanh(r^2)$, $G_{2n} = \frac{1}{1+r^{2n}}$, splines approx. [12]

Logistic (5)
Act: I, Out: σ

Other Sigmoids
Act: I, Out: tanh, arctan

Sigmoids Approximations ($s2, s3$) (6–7)
Act: I, Out: $\Theta(I)\frac{I}{1+s} - \Theta(-I)\frac{I}{1-s}$, $\frac{sI}{1+\sqrt{1+s^2I^2}}$, $\frac{sI}{1+|sI|}$, $\frac{sI}{\sqrt{1+s^2I^2}}$

Heaviside (2)
Act: I, Out: $\Theta(I;\theta)$

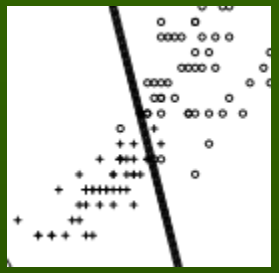
Multistep (3)
Act: I, Out: $\varsigma(I)$

Semi-linear (4)
Act: I, Out: $s_l(I;\theta_1,\theta_2)$

HAS decision trees

Decision trees select the best feature/threshold value for univariate and multivariate trees:

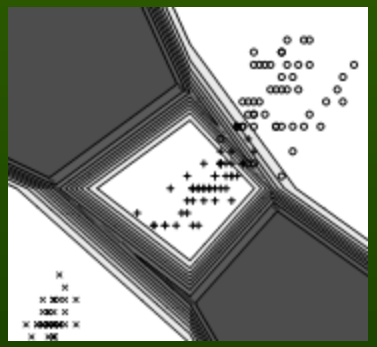
$$X_i < \theta_k \text{ or } T(\mathbf{X};\mathbf{W},\theta_k) = \sum_i W_i X_i < \theta_k$$



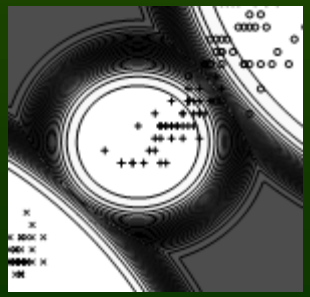
Decision borders: hyperplanes.

Introducing tests based on L_α Minkovsky metric.

$$T(\mathbf{X};\mathbf{R},\theta_R) = \|\mathbf{X}-\mathbf{R}\|_\alpha = \sum_i |X_i - R_i|^{1/\alpha} < \theta_R$$



Such DT use radial kernel features!



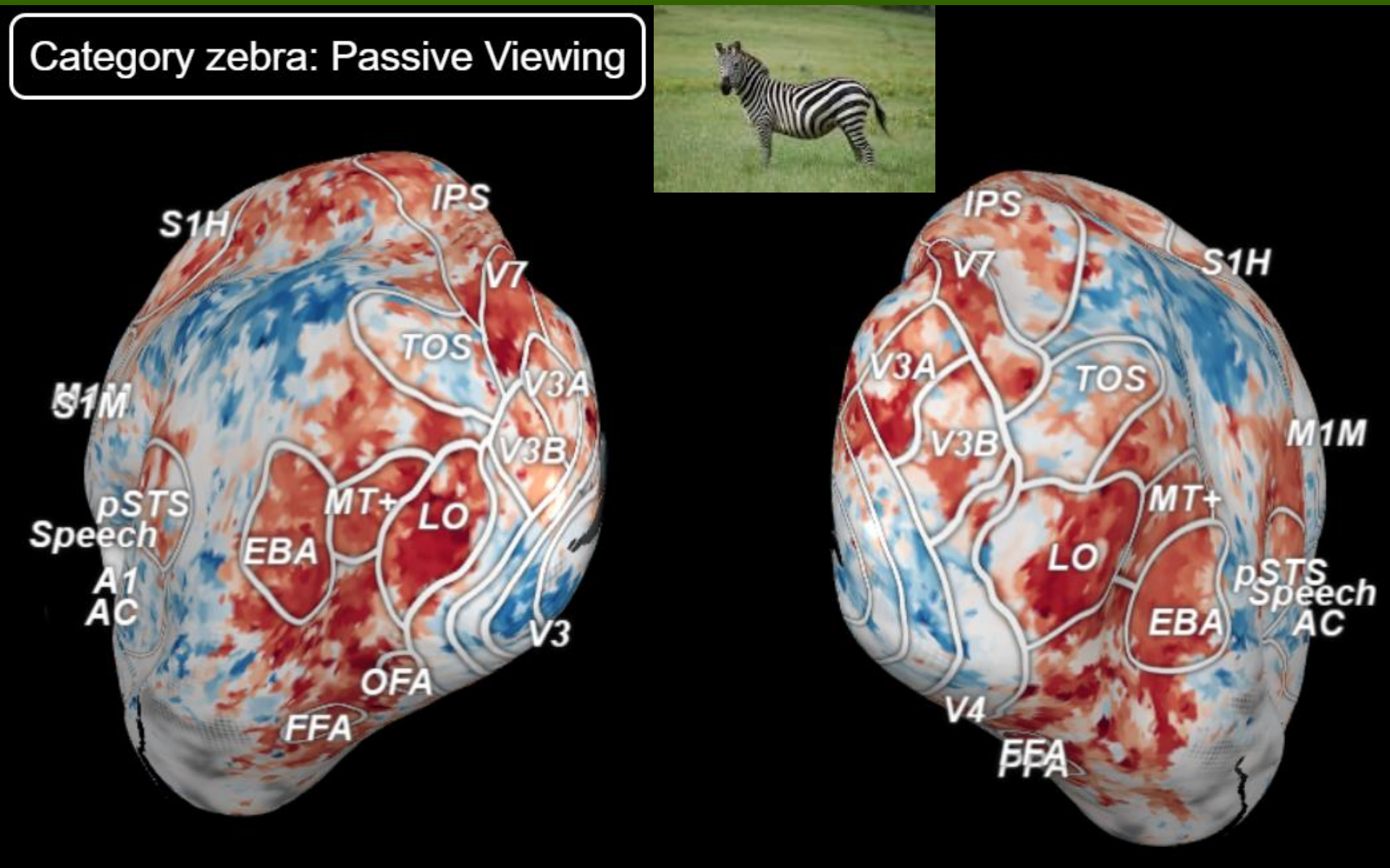
For L_2 spherical decision border are produced.

For L_∞ rectangular border are produced.

For large databases first clusterize data to get candidate references R.

Beyond perception

Brain activity beyond visual cortex

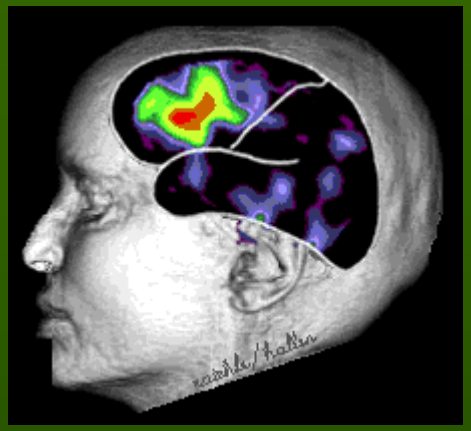


Geometric model of mind

Brain ⇔ Psyche

Objective ⇔ Subjective

Neurodynamics: bioelectrical activity of the brain, neural activity measured using EEG, MEG, NIRS-OT, PET, fMRI, other techniques.

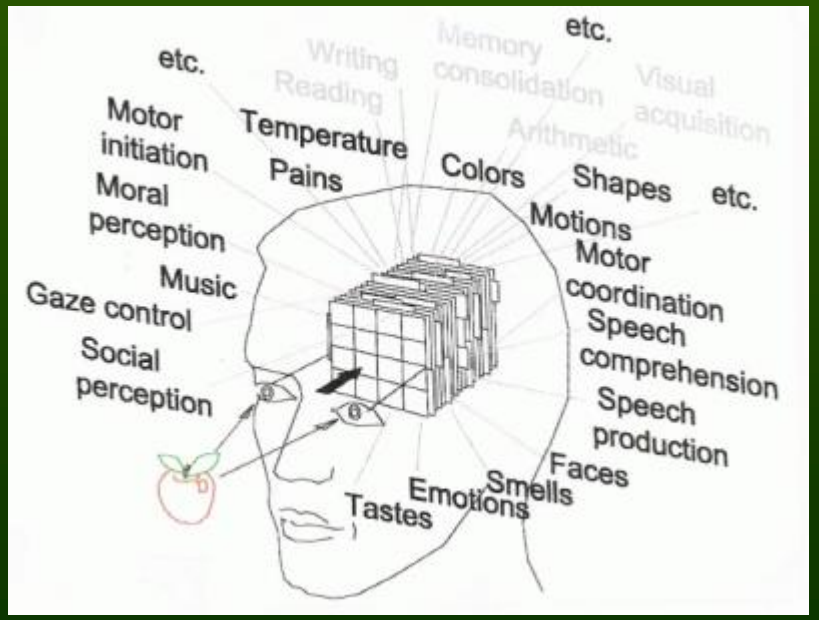


Mapping S(M)⇔S(B) but how to describe the state of mind? Brain fingerprints?

Verbal description is not sufficient.
A space with dimensions that measure different aspects of experience is needed.

Mental states, movement of thoughts ⇔ trajectories in psychological spaces.

Problem: good phenomenology. We are not able to describe our mental states.



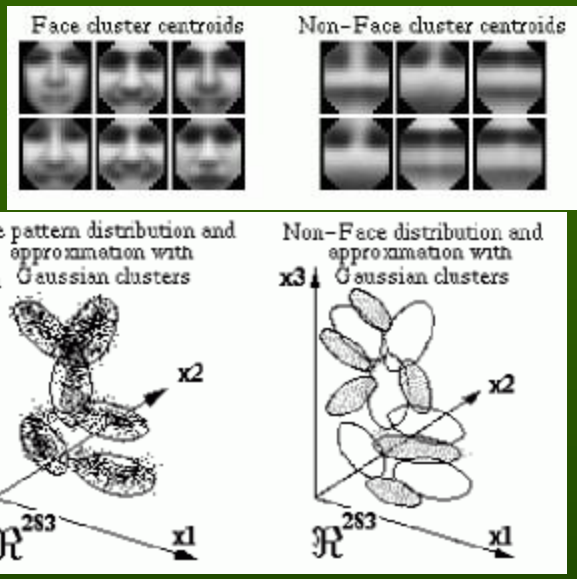
Hurlburt & Schwitzgabel, Describing Inner Experience? MIT Press 2007

FSM - neurofuzzy systems

Feature Space Mapping (FSM) constructive neurofuzzy system. Neural adaptation, estimation of probability density distribution (PDF) using single hidden layer network (RBF-like), with nodes realizing separable basis functions (SBF networks):

$$RBF(X;P) = \sum_i W_i \|X_i - P_i\|$$

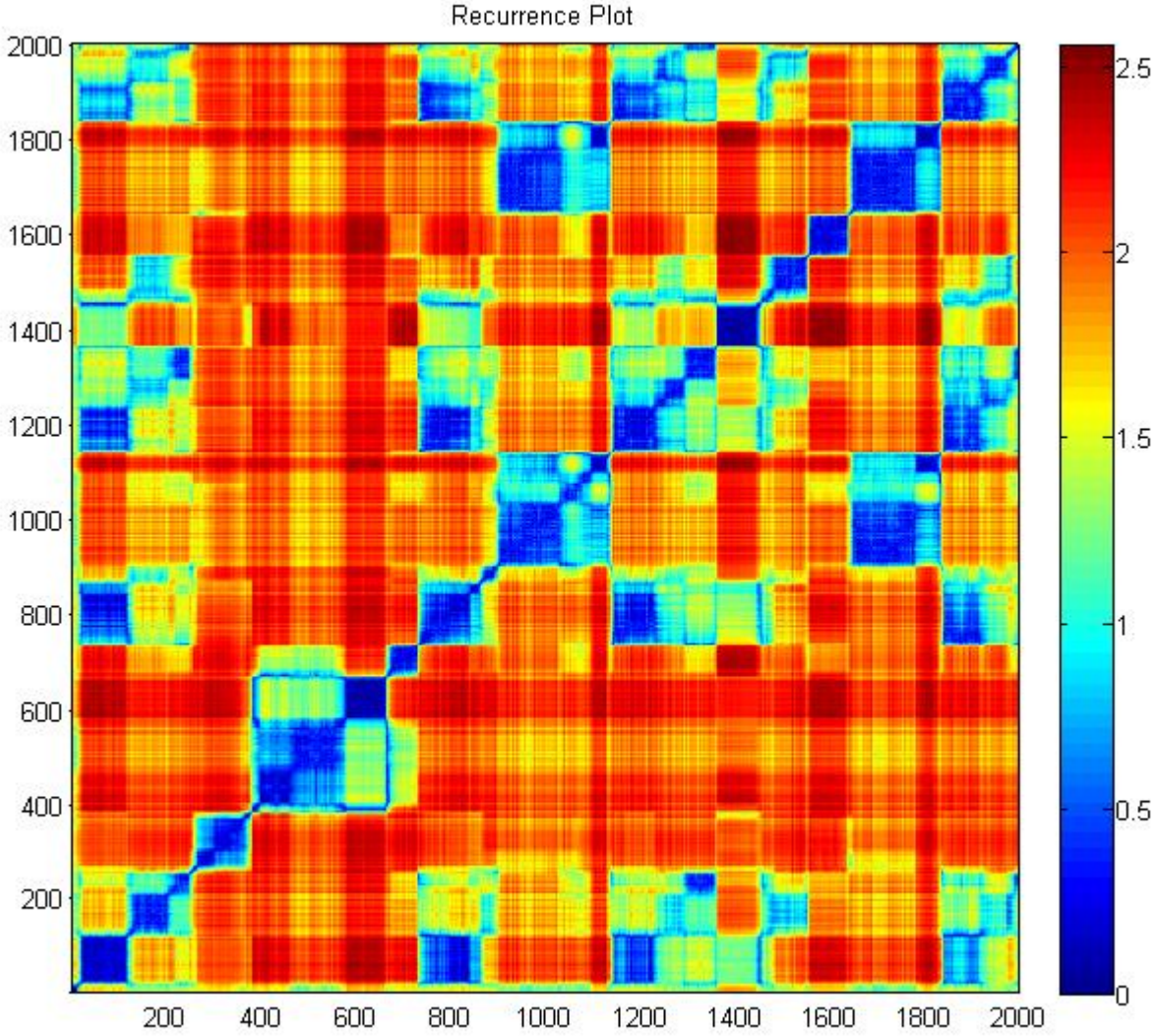
$$FSM(X;P) = \sum_i W_i \prod_{j=1} G_{ij}(X_{ij} - P_{ij})$$



Model of mental processes–FSM nodes representing attractors, mental events.

Duch W, Diercksen GHF (1995) Feature Space Mapping as a universal adaptive system. Computer Physics Communications 87: 341-371

Duch W (1997) Platonic model of mind as an approximation to neurodynamics. In: Brain-like computing and intelligent information systems, ed. S-i. Amari, N. Kasabov (Springer, Singapore 1997), chap. 20



Activation of 140 semantic layer units starting from the word „gain”: rapid transitions between a sequence of related concepts is seen. Real EEG is coming.

Prototype-based rules

C-rules (Crisp), are a special case of F-rules (fuzzy rules).

F-rules (fuzzy rules) are a special case of P-rules (Prototype)! But not vv.

P-rules may use probabilistic metrics, have the form:

IF $P = \arg\min_R D(X,R)$ THAN Class(X)=Class(P)

D(X,R) = dissimilarity (distance) function, determining local decision borders.

P-rules are easy to interpret and offer better description than F-rules!

IF X=You are most similar to the P=Superman
THAN You are in the Super-league.

"Similar" may involve different features or D(X,P). What is similar for the brain? Kernel features in SVM are particular example of similarity functions.

- Duch W, Adamczak R, Grąbczewski K, A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. IEEE Trans. on Neural Networks
- Duch W, Setiono R, Zurada J.M, Computational intelligence methods for understanding of data. Proc. of the IEEE 92(5) (2004) 771- 805

Similarity-based framework

Search for good models requires a frameworks to build and evaluate them.
$p(C_i|X;M)$ posterior classification probability or $y(X;M)$ approximators,
models M are parameterized in increasingly sophisticated way.
Similarity-Based Learning (SBL) or S-B Methods provide such framework.

(Dis)similarity:

- more general than feature-based description,
- no need for vector spaces (structured objects),
- more general than fuzzy approach (F-rules are reduced to P-rules),
- includes nearest neighbor algorithms, MLPs, RBFs, separable function networks, SVMs, kernel methods, specialized kernels, and many others!

A systematic search (greedy, beam), or evolutionary search in the space of all SBL models is used to select optimal combination of parameters & procedures, opening different types of optimization channels,
trying to discover appropriate bias for a given problem.

Result: several candidate models are created, already first very limited version gave best results in 7 out of 12 Stalog problems.

SBM framework components

- Pre-processing: objects O => features X, or (diss)similarities D(O,O').
- Calculation of similarity between features $d(x_i,y_i)$ and objects D(X,Y).
- Reference (or prototype) vector R selection/creation/optimization.
- Weighted influence of reference vectors $G(D(R_i,X))$, i=1..k.
- Functions/procedures to estimate p(C|X;M) or approximator y(X;M).
- Cost functions $E[D_T;M]$, various model selection/validation procedures.
- Optimization procedures for the whole model M_a.
- Search control procedures to create more complex models M_{a+1}.
- Creation of ensembles of (global, local, competent) models.

- M={X(O), d(··), D(··), k, G(D), {R}, $\{p_i(R)\}$, E[·], K(·), S(··)}, where:
- $S(C_i,C_j)$ is a matrix evaluating similarity of the classes;
  a vector of observed probabilities $p_i(X)$ instead of hard labels.

The kNN model $p(Ci|X;kNN) = p(C_i|X;k,D(\cdot),\{D_T\})$;

the RBF model: $p(Ci|X;RBF) = p(Ci|X;D(\cdot),G(D),\{R\})$,

MLP, SVM and many other models may all be "re-discovered" as a part of SBL.

Prototypes for images

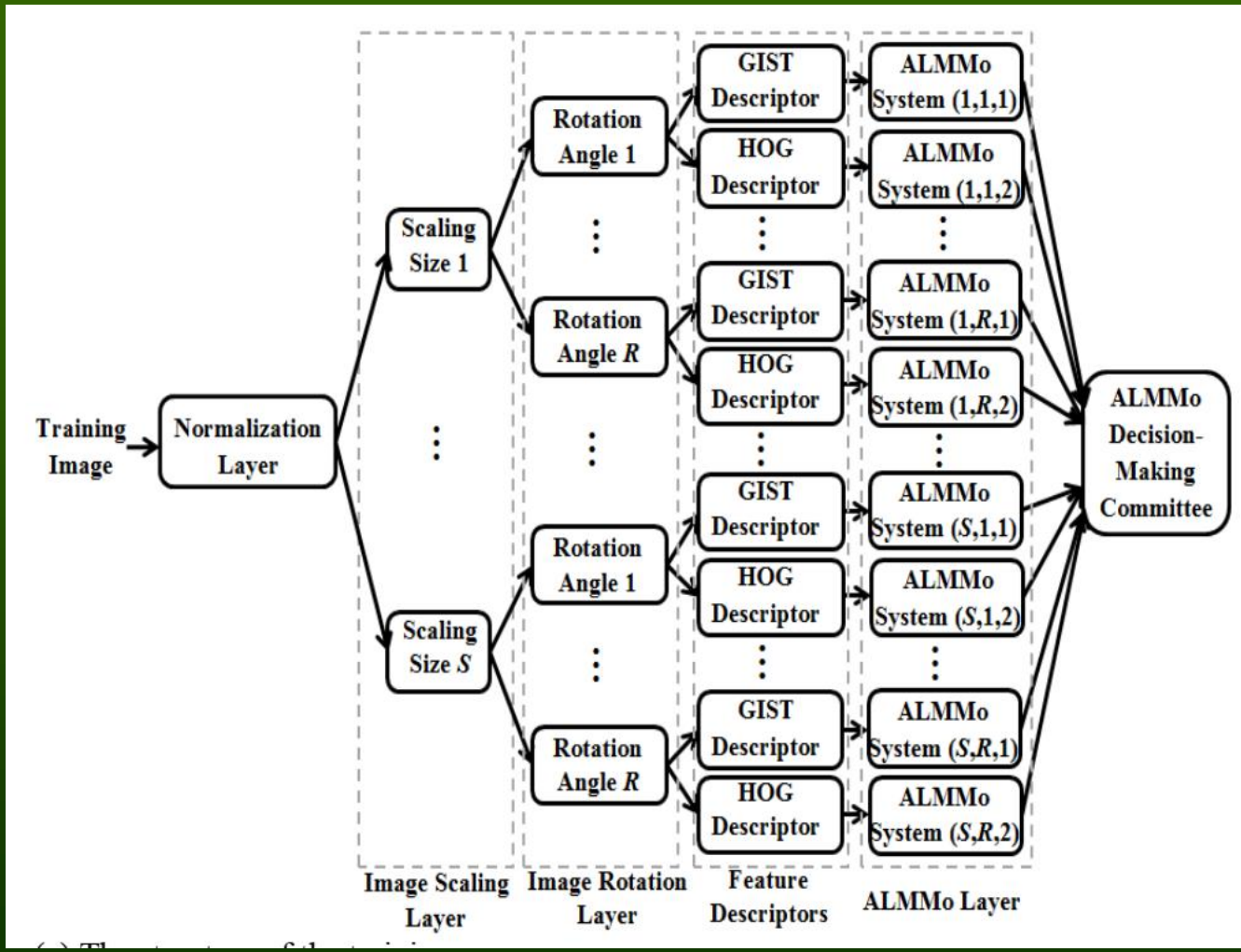
Stable and transparent interpretation, based on similarity.

Lazy learning.

Almost as good as deep learning on hand written digits (NIPS data).

~ Pandemonium architecture, Selfridge 1959!

P. Angelov, X. Gu, MICE: **Multi-layer Multi-model Images Classifier. Ensemble**, CYBCONF 2017



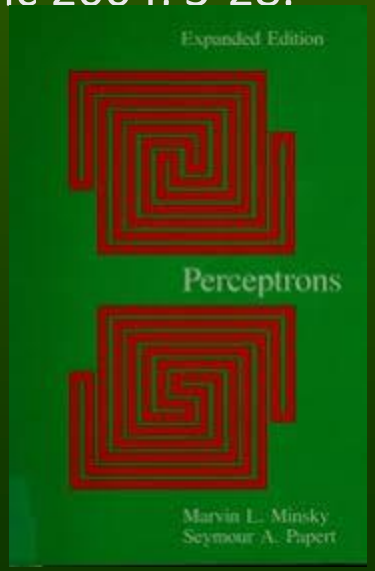
Functional brain networks

The Society of Mind

NN people ignored AI people (and vv). Will more sophisticated version of pandemonium – based on deep learning – lead to AGI?

The Society of Mind (Minsky 1986) presents theory of natural intelligence based on interactions of mindless agents constituting a "society of mind", or multi-agent model, but at the symbolic level.

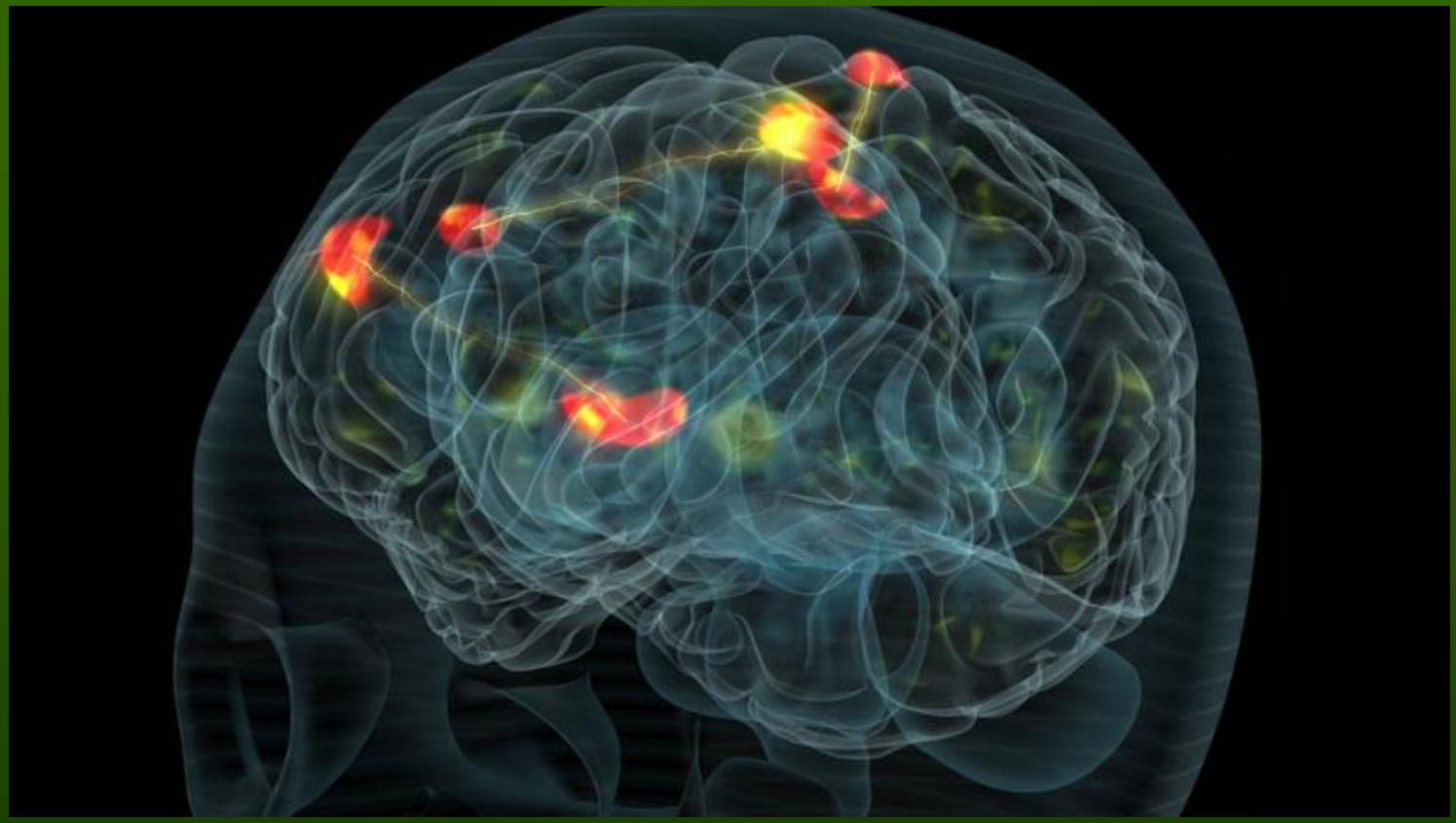
Duch W, Mandziuk J, *Quo Vadis Computational Intelligence?* Advances in Fuzzy Systems - Applications and Theory Vol. 21, World Scientific 2004. 3-28.



Minsky+Papert – MLPs is universal approximator but cannot solve connectedness problem.

$\Rightarrow$ NCE, modules, more internal knowledge, adding phase solves it, opens new complexity problem class.
This shows importance of various transfer functions and led to the FSM model with separable transfer functions.

Mental state: strong coherent activation

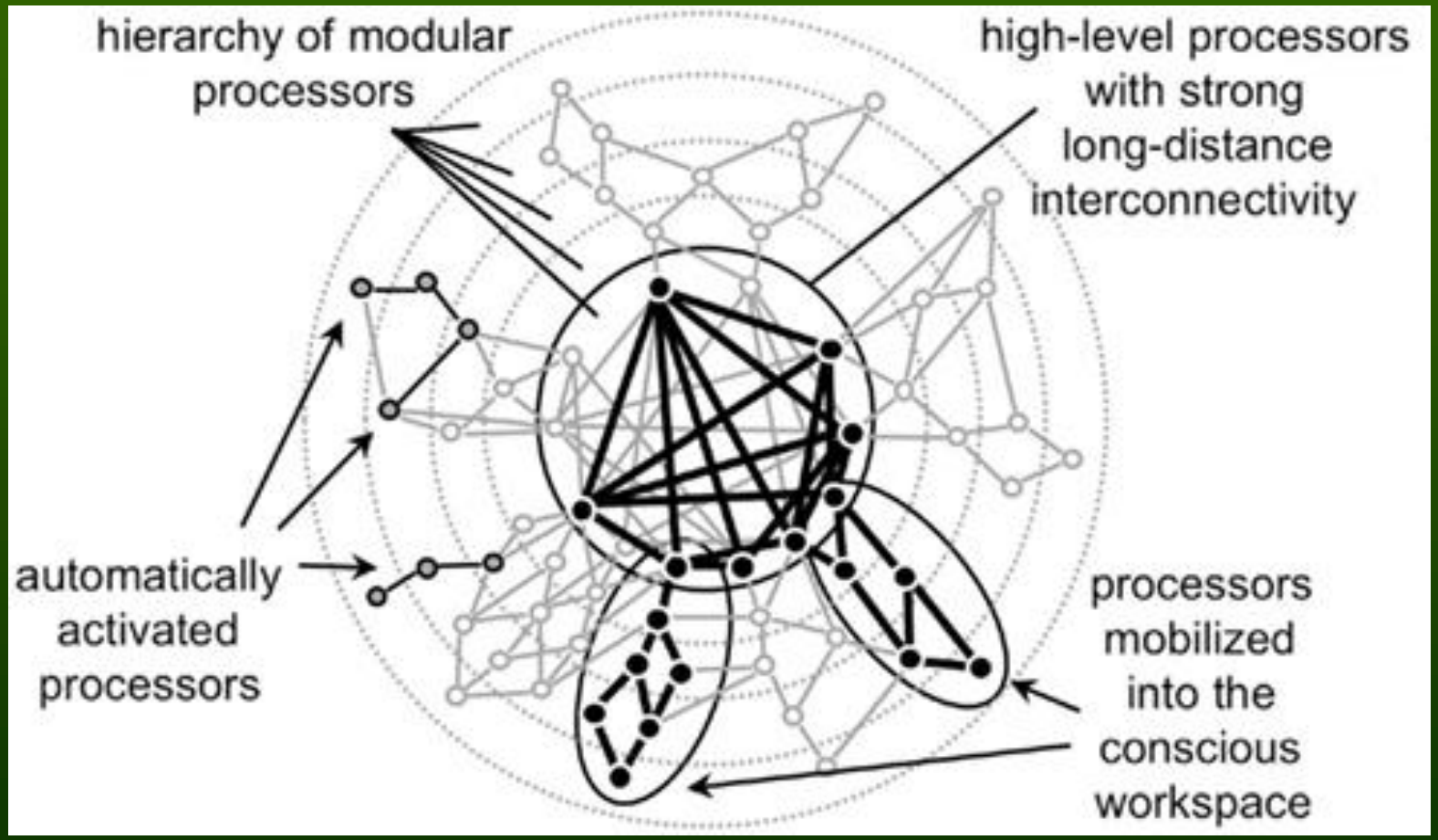


Many processes go on in parallel, controlling homeostasis and behavior. Most are automatic, hidden from our Self. What goes on in my head?

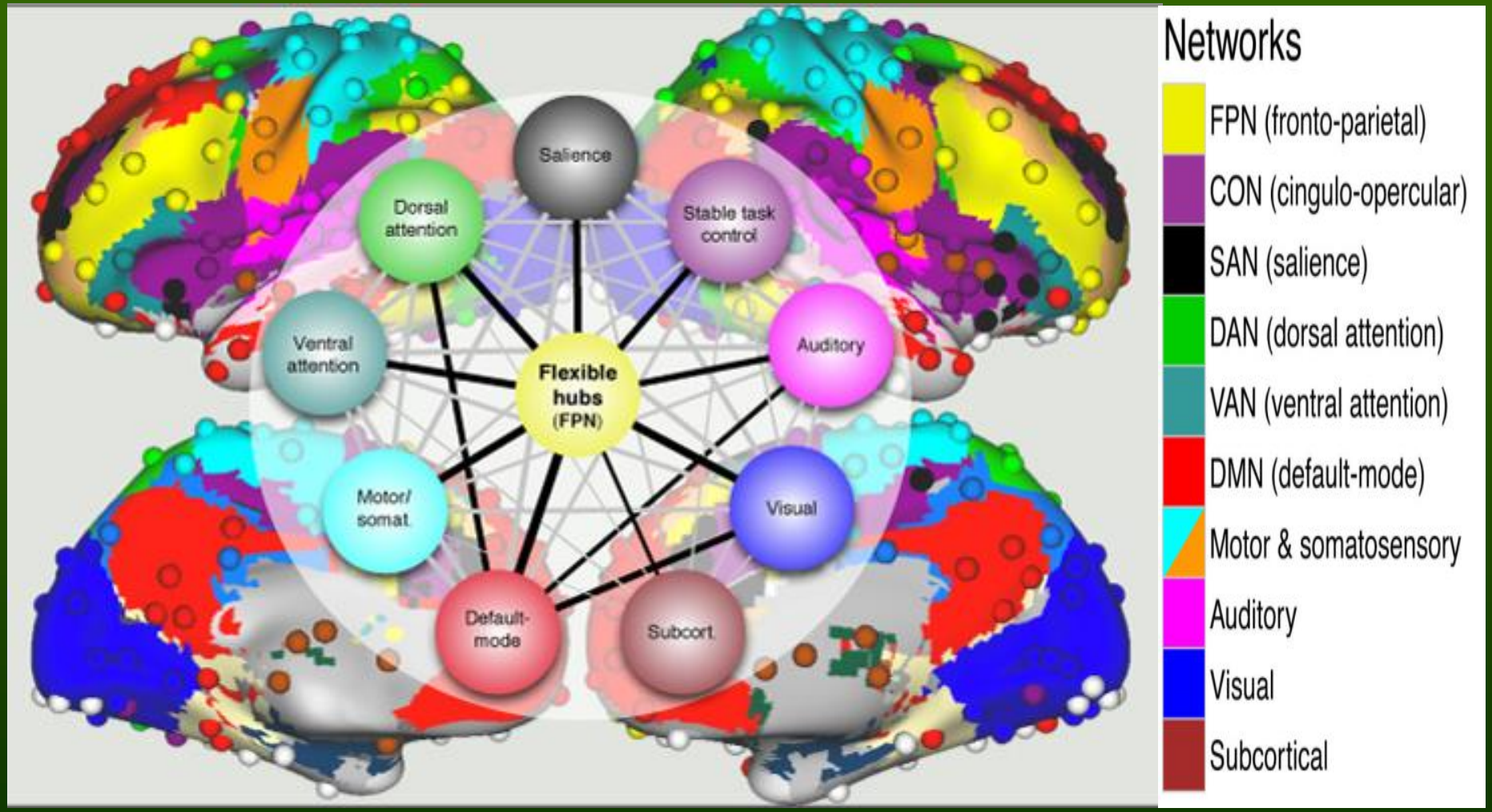
Various subnetworks compete for access to the highest level of control - consciousness, the winner-takes-most mechanism leaves only the strongest. How to extract stable intentions from such chaos? BCI is never easy.

GNWT

Global Neuronal Workspace Theory (Dehaene et al. 1998)



Neurocognitive Basis of Cognitive Control



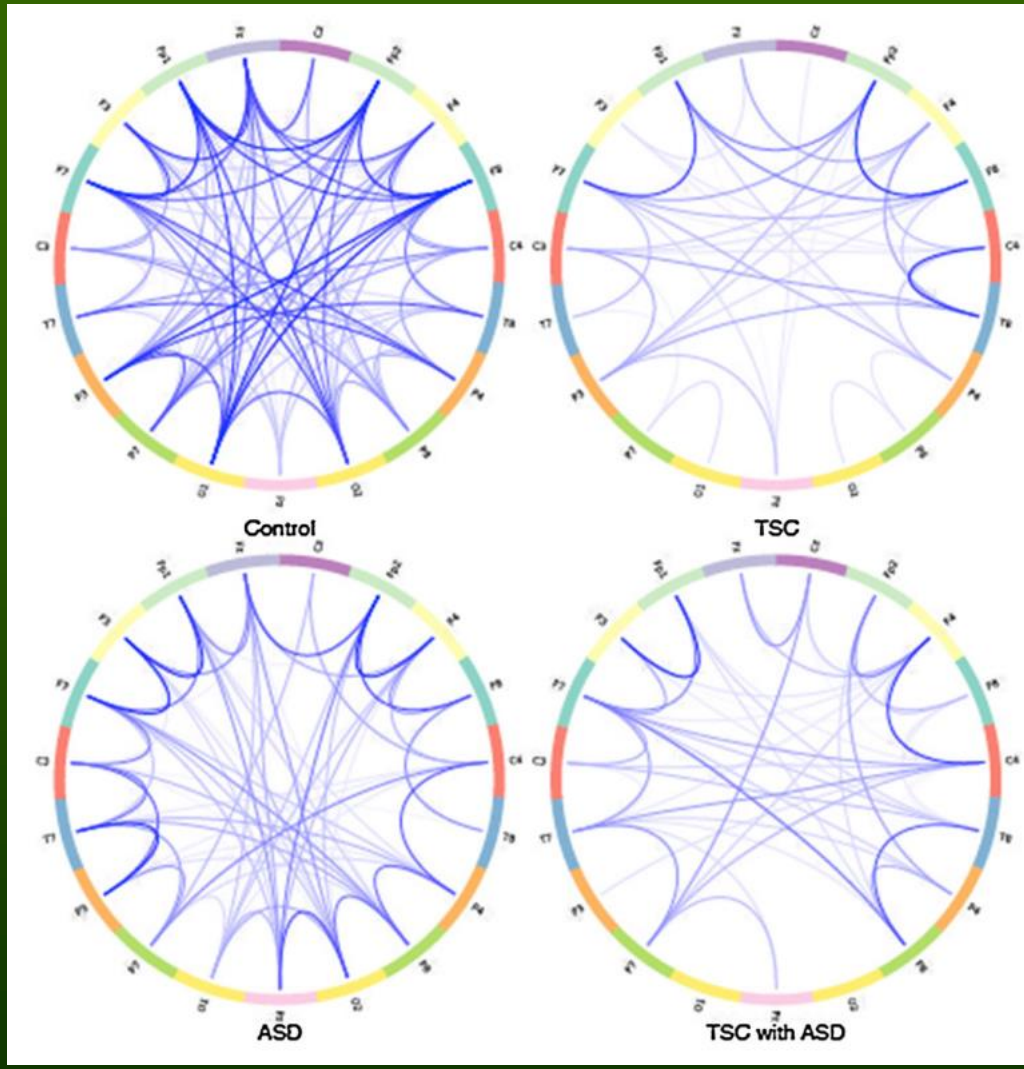
Central role of fronto-parietal (FPN) flexible hubs in cognitive control and adaptive implementation of task demands (black lines=correlations significantly above network average). Cole et al. (2013).

Pathological functional connections

Comparison of connections for patients with ASD (autism spectrum), TSC (Tuberous Scelrosis), and ASD+TSC.

Weak or missing connections between distant regions prevent ASD/TSC patients from solving more demanding cognitive tasks.

Network analysis becomes very useful for diagnosis of changes due to the disease and learning.



J.F. Glazebrook, R. Wallace, Pathologies in functional connectivity, feedback control and robustness. Cogn Process (2015) 16:1–16

Effect of cognitive load

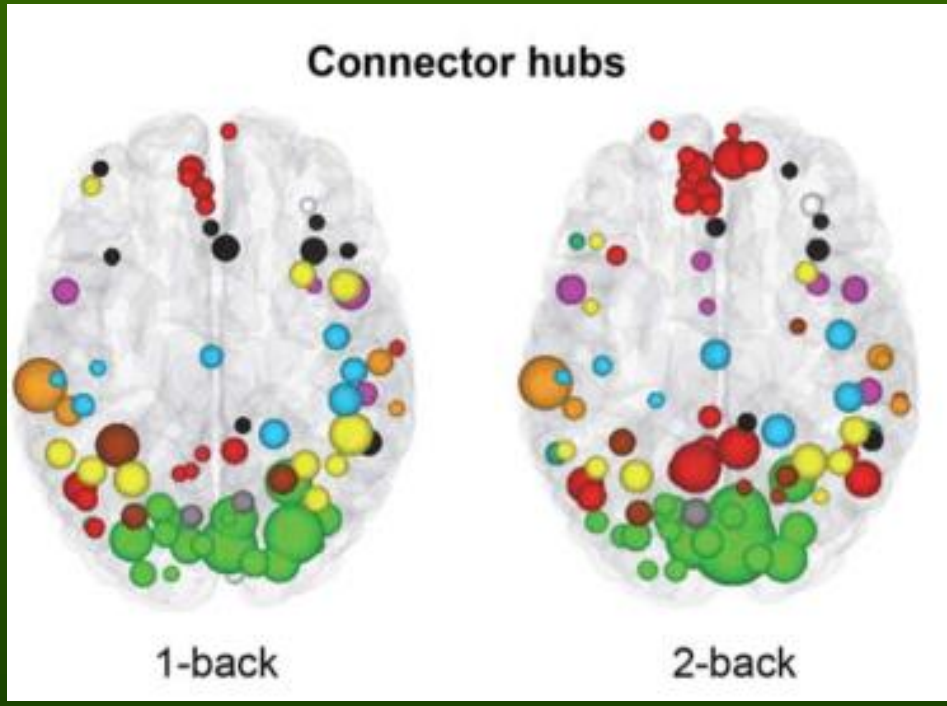
Simple and more difficult tasks, requiring the whole-brain network reorganization.

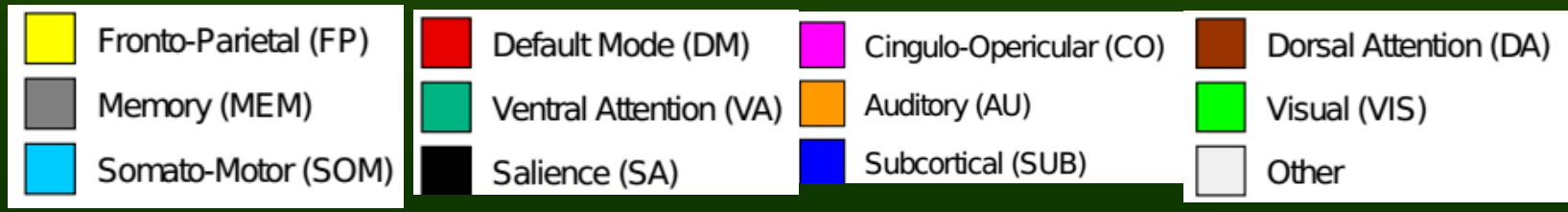
Left: 1-back Top: connector hubs

Right: 2-back Bottom: local hubs

Average over 35 *participants.*

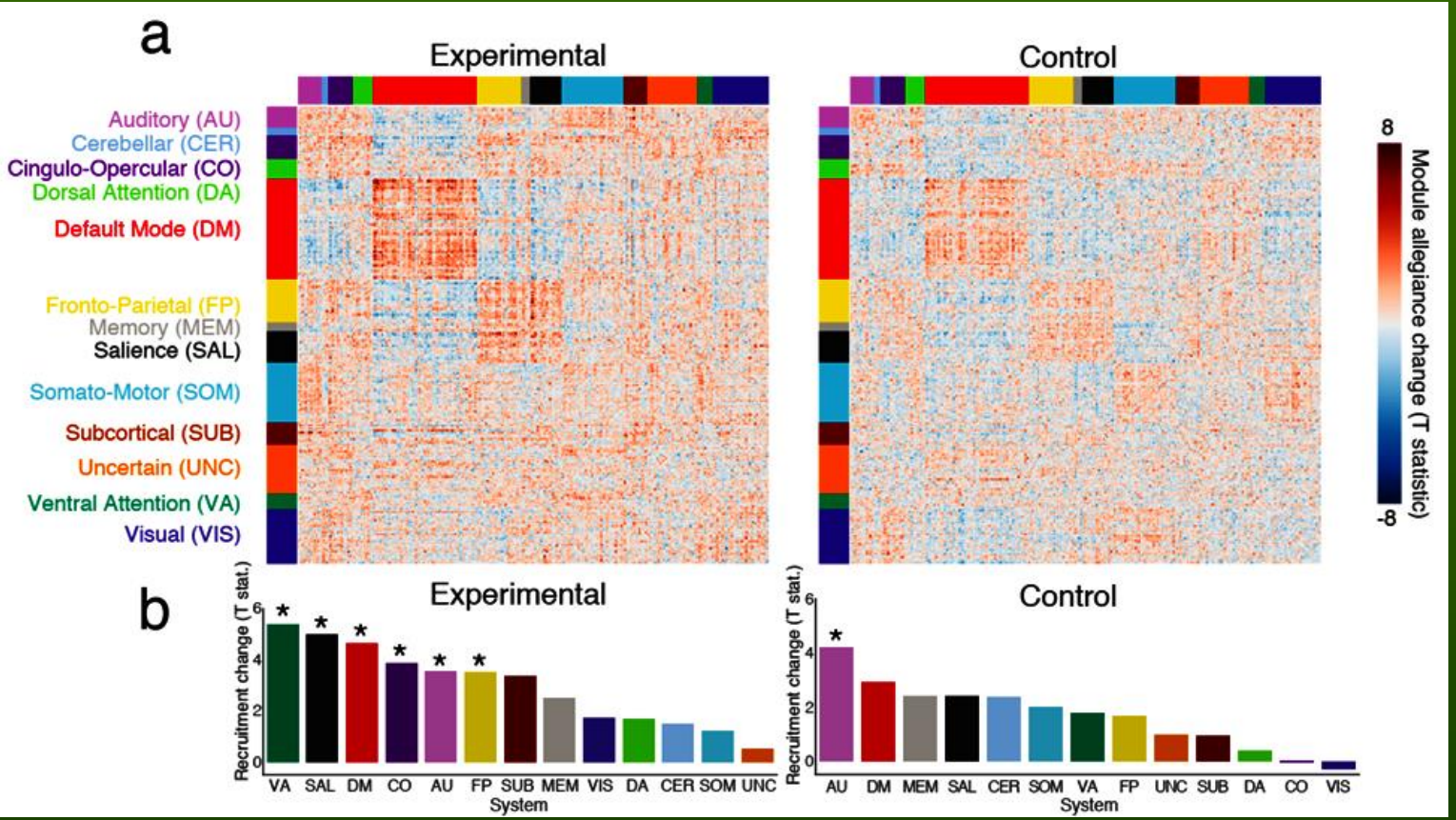
Dynamical change of the landscape of attractors, depending on the cognitive load. DMN areas engaged in global binding!





K. Finc et al, HBM (2017).

Working memory training



Whole-brain changes in module allegiance between the 'Naive' and 'Late' 6-week working memory training stages.

(**a**) Changes in node allegiance as reflected in the two-tailed *t*-test.

(**b**) Significant increase * in the default mode, fronto-parietal ventral attention, salience, cingulo-opercular, and auditory systems recruitment.

K. Finc et al, Nature Communications (2020).

Transformation-based framework

Find simplest model that is suitable for a given data, creating non-sep. that is easy to handle: simpler models generalize better, interpretation.

Compose transformations (neural layers), for example:

- Matching pursuit network for signal decomposition, QPC index.
- PCA network, with each node computing principal component.
- LDA nets, each node computes LDA direction (including FDA).
- ICA network, nodes computing independent components.
- KL, or Kullback-Leibler network with orthogonal or non-orthogonal components; max. of mutual information is a special case.
- c^2 and other statistical tests for dependency to aggregate features.
- Factor analysis network, computing common and unique factors.

Evolving Transformation Systems (Goldfarb 1990-2008), giving unified paradigm for inductive learning, structural processes as representations.

T-based meta-learning

To create successful meta-learning through search in the model space fine granulation of methods is needed, extracting info using support features, learning from others, knowledge transfer and deep learning.

Learn to compose, using complexity guided search, various transformations (neural or processing layers), for example:

- Creation of new support features: linear, radial, cylindrical, restricted localized projections, binarized … feature selection or weighting.
- Specialized transformations in a given field: text, bio, signal analysis, ….
- Matching pursuit networks for signal decomposition, QPC index, PCA or ICA components, LDA, FDA, max. of mutual information etc.
- Transfer learning, granular computing, learning from successes: discovering interesting higher-order patterns created by initial models of the data.
- Stacked models: learning from the failures of other methods.
- Schemes constraining search, learning from the history of previous runs at the meta-level.

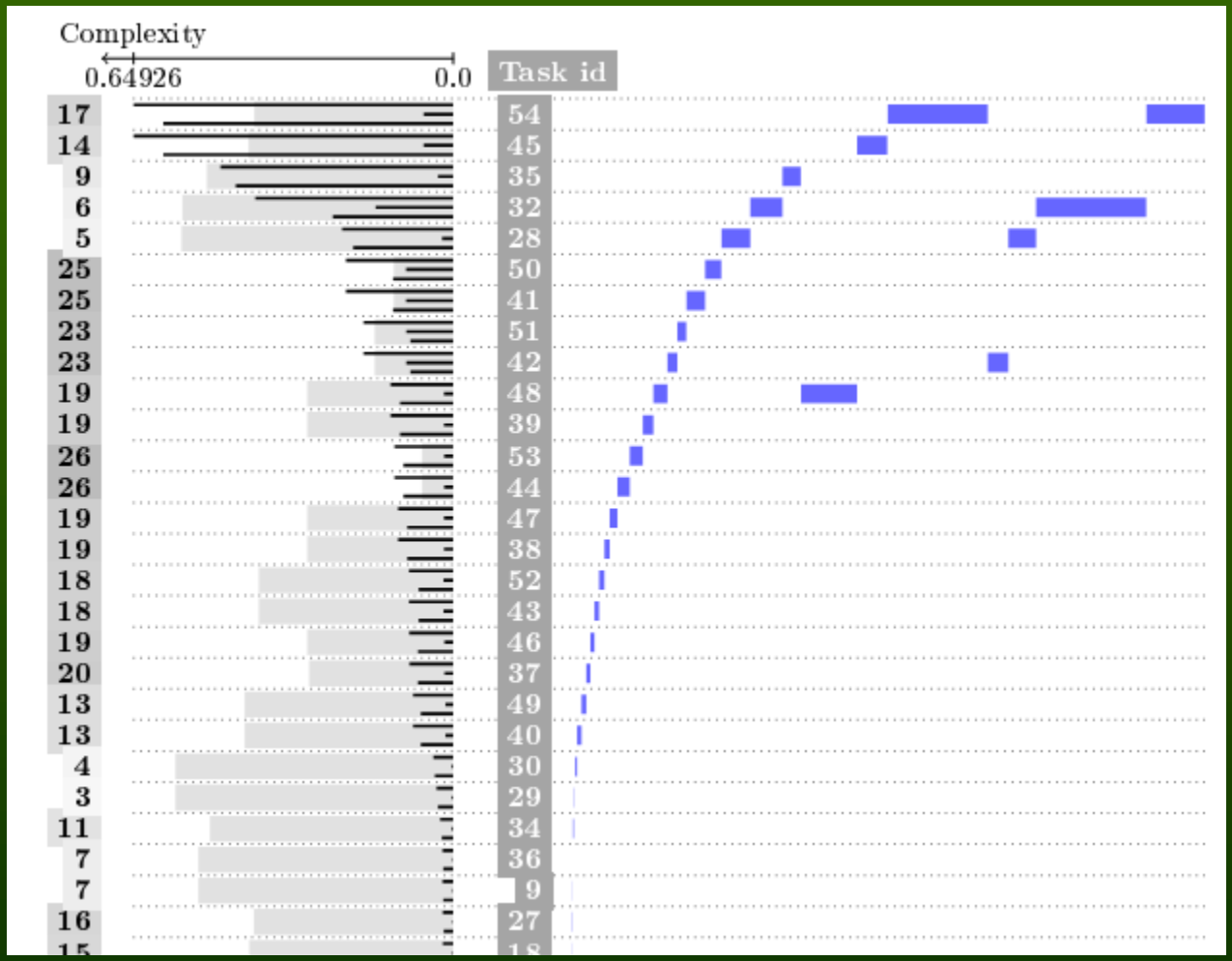
Complex machines on vowel data

Number on far left = final ranking.

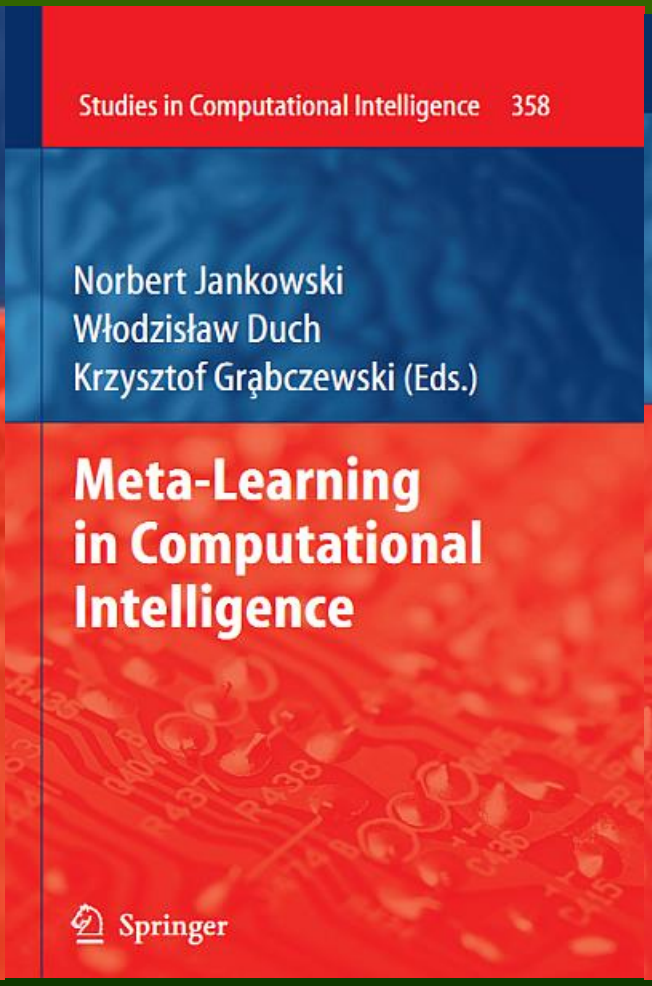
Gray bar = accuracy

Small bars (up-down) show estimation of: total complexity, time, memory.

Numbers in the middle = process id (refer to models in the previous table).




Krzysztof Grąbczewski
Meta-Learning in Decision Tree Induction
Springer


Studies in Computational Intelligence 358
Norbert Jankowski
Włodzisław Duch
Krzysztof Grąbczewski (Eds.)
Meta-Learning in Computational Intelligence
Springer


Studies in Computational Intelligence 63
Włodzisław Duch
Jacek Mańdziuk (Eds.)
Challenges for Computational Intelligence
Springer

Neuroscience => AI

Hassabis, D., Kumaran, D., Summerfield, C., Botvinick, M. (2017). **Neuroscience-Inspired Artificial Intelligence**. *Neuron*, *95*(2), 245–258.

Affiliations: **Google DeepMind**, Gatsby, ICN, UCL, Oxford.

Bengio, Y. (2017). The **Consciousness Prior**. *ArXiv:1709.08568.*

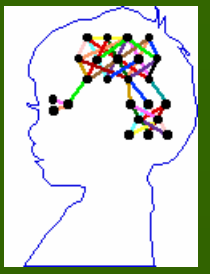
Amoset al. (2018). **Learning Awareness Models**. ICRL, *ArXiv:1804.06318*.

AI Systems inspired by Neural Models of Behavior:
(A) **Visual attention**, foveal locations for multiresolution ‘‘retinal’’ representation, prediction of next location to attend to.
(B) **Complementary learning systems** and episodic control: fast learning hippocampal system and parametric slow-learning neocortical system.
(C) Models of **working memory** and the Neural Turing Machine.

(D) Neurobiological models of **synaptic consolidation**.

SANO, new Centre for Individualized Computational Medicine in Kraków (EU Team project, with Sheffield Uni, Fraunhofer Society, Research Centre Juelich.

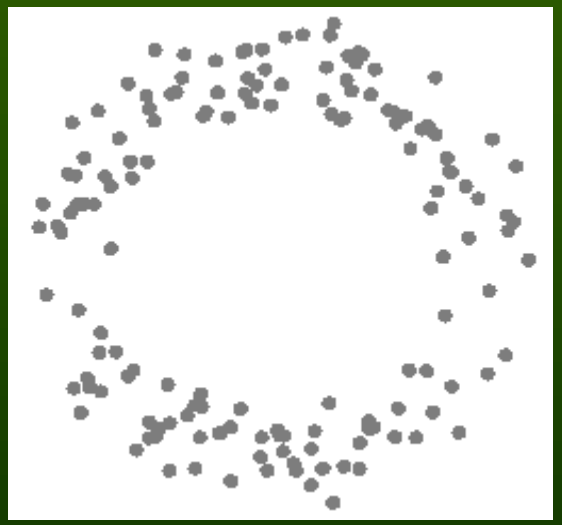
Summary

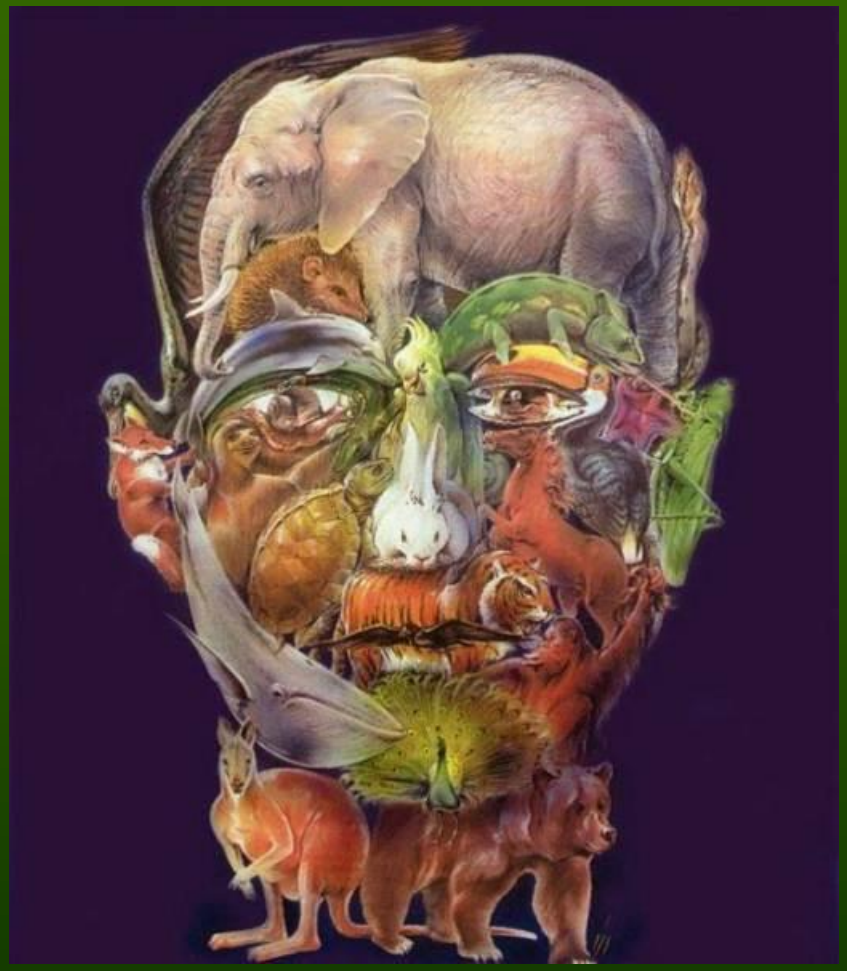


1. Challenging data cannot be handled with existing DM tools.
2. Visualization of hidden neuron's shows that frequently perfect but non-separable solutions are found despite base-rate outputs.
3. Linear separability is not the best goal of learning, other targets that allow for easy handling of final non-linearities may work better.
4. *k*-separability defines complexity classes for non-separable data.
5. Similarity-based framework enables meta-learning as search in the model space, heterogeneous systems add fine granularity.
6. Transformation-based learning shows the need for component-based approach to DM, discovery of simplest models and support features.
7. SFM and ULM may do more than SVMs.
8. Meta-learning should replace experts in automatically creating new optimal learning methods on demand.

Many things to finish. But everyone works now on deep learning …

Thank for synchronization of your neurons





Google: W. Duch
=> talks, papers, projects, lectures …