Computing with High-Dimensional Vectors

Module 10

Relations to neural networks*

Denis Kleyko

---

*Lots of images from Internet were used to prepare this presentation

Outline

The only rule is, there are no rules:

- HD vectors as **input to neural networks**
- Neural networks for **producing HD vectors**
- HD Computing/VSA **connections** to **randomized** neural networks
- Use of HD Computing/VSA **primitives** in neural networks **design**
- **Explaining neural networks** with HD Computing/VSA

HD vectors as input to neural networks

Types of input

- HD vectors as a way to represent input to a network
  - Natural, as neural networks are also working with distributed representations

- Data to be fed to a neural network are high-dimensional and sparse
  - HD vectors can form more compact representation
- Input of varying size
  - Composite data structures
  - HD vectors are fixed size input
- Natural language processing
  - A lot of structure in language which can be potentially represented in HD vectors

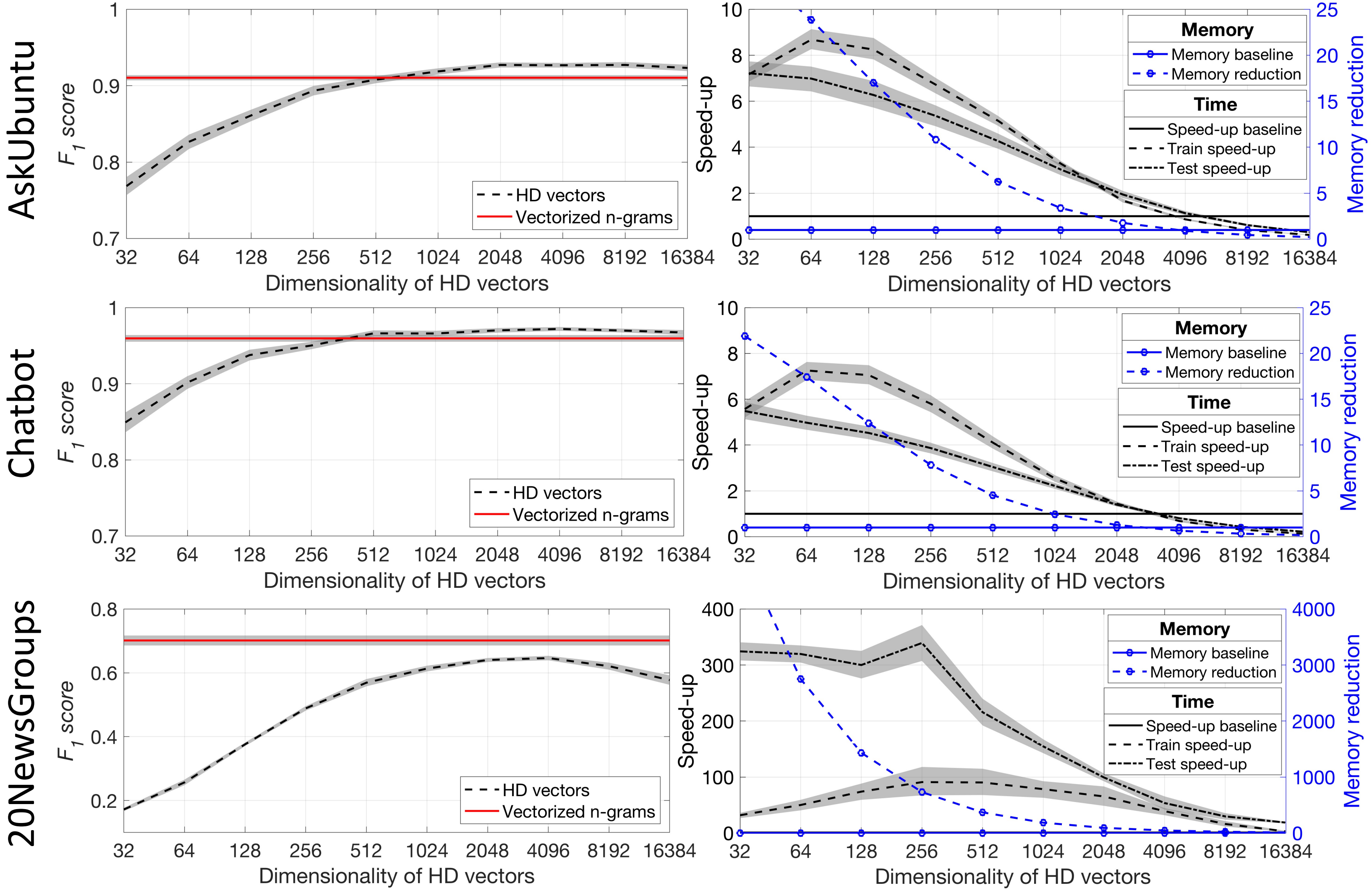
- Expansion of the applicability of neural networks
  - Relieves the pressure of forming the task
    - With fixed size input
    - A sequence suitable for recurrent neural networks

Embedding *n*-gram statistics: Text classification

- Representation of *n*-gram statistics as before

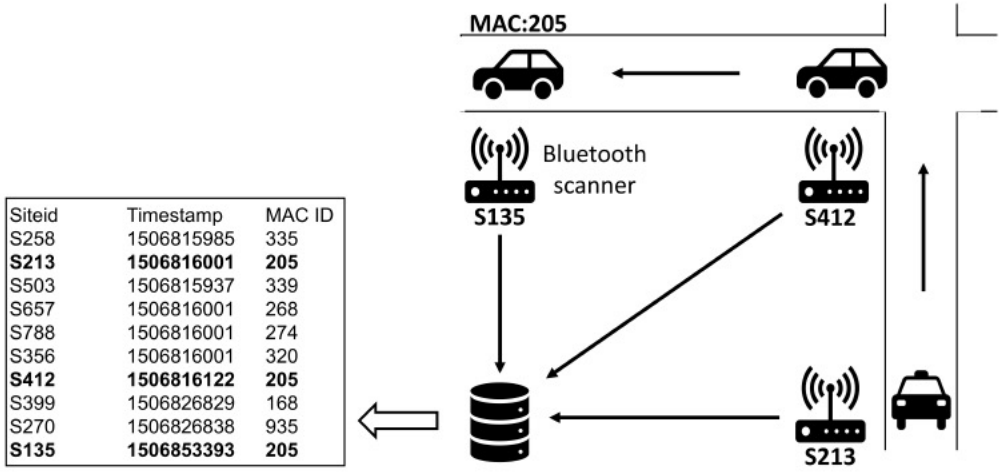
- Text classification
  - 4 datasets
  - 9 ML algorithms

- Figures for neural network
  - 3 datasets

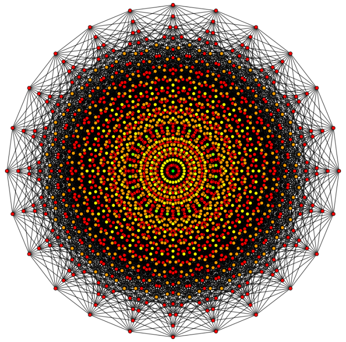


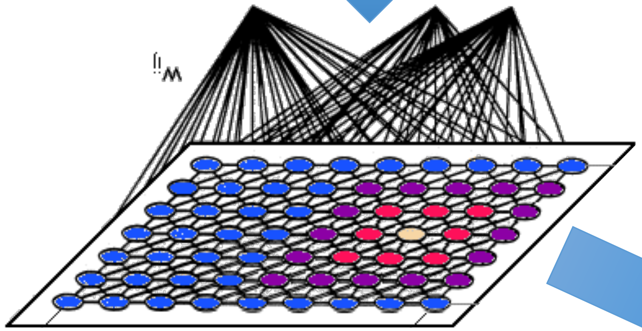
P. Alonso, K. Shridhar, D. Kleyko, E. Osipov, and M. Liwicki, “HyperEmbed: Tradeoffs Between Resources and Performance in NLP Tasks with Hyperdimensional Computing enabled Embedding of n-gram Statistics”, International Joint Conference on Neural Networks (IJCNN). 2021.

Composite data structures: Varying length sequences

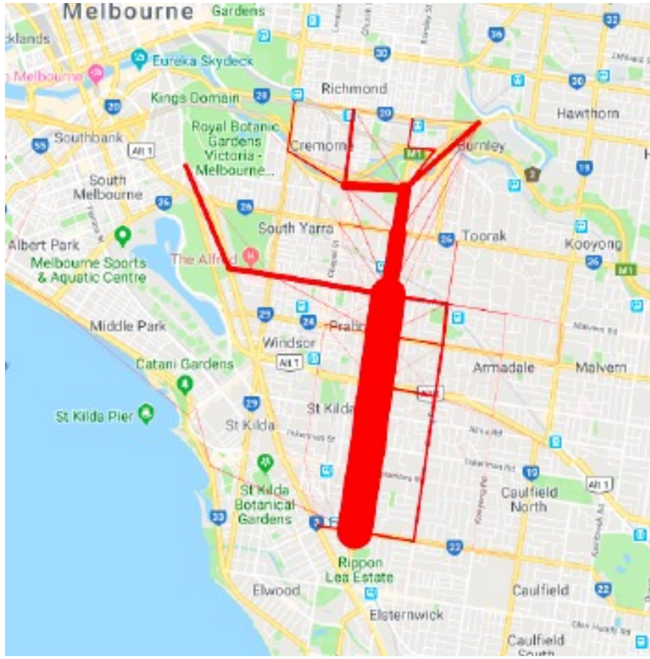


Siteid	Timestamp	MAC ID
S258	1506815985	335
S213	**1506816001**	**205**
S503	1506815937	339
S657	1506816001	268
S788	1506816001	274
S356	1506816001	320
S412	**1506816122**	**205**
S399	1506826829	168
S270	1506826838	935
S135	**1506853393**	**205**





Self-Organizing Map



- Trajectory is considered as a bag of n-grams
  - Trajectories are of variable length
- A tri-gram of locations $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\}$ is represented as HD vector:
  - $\rho^2(\mathbf{l}_1) \oplus \rho^1(\mathbf{l}_2) \oplus \rho^0(\mathbf{l}_3)$

T. Bandaragoda, D. De Silva, D. Kleyko, E. Osipov, U. Wiklund, and D. Alahakoon, “Trajectory Clustering of Road Traffic in Urban Environments using Incremental Machine Learning in Combination with Hyperdimensional Computing”, IEEE Intelligent Transportation Systems Conference (ITSC), 2019.

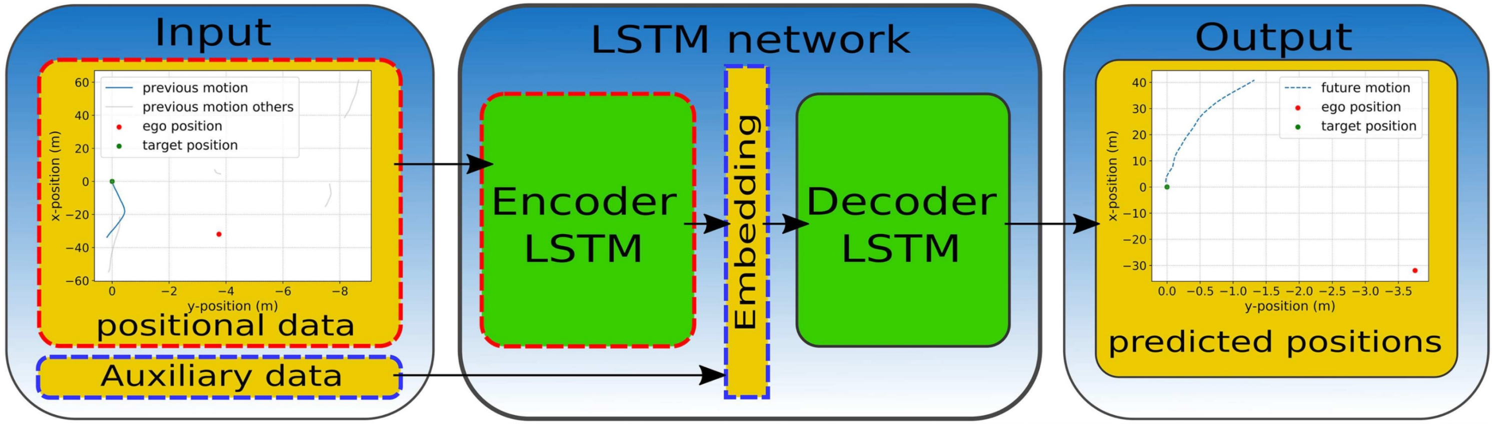
Composite data structures: Vehicle Behavior Prediction

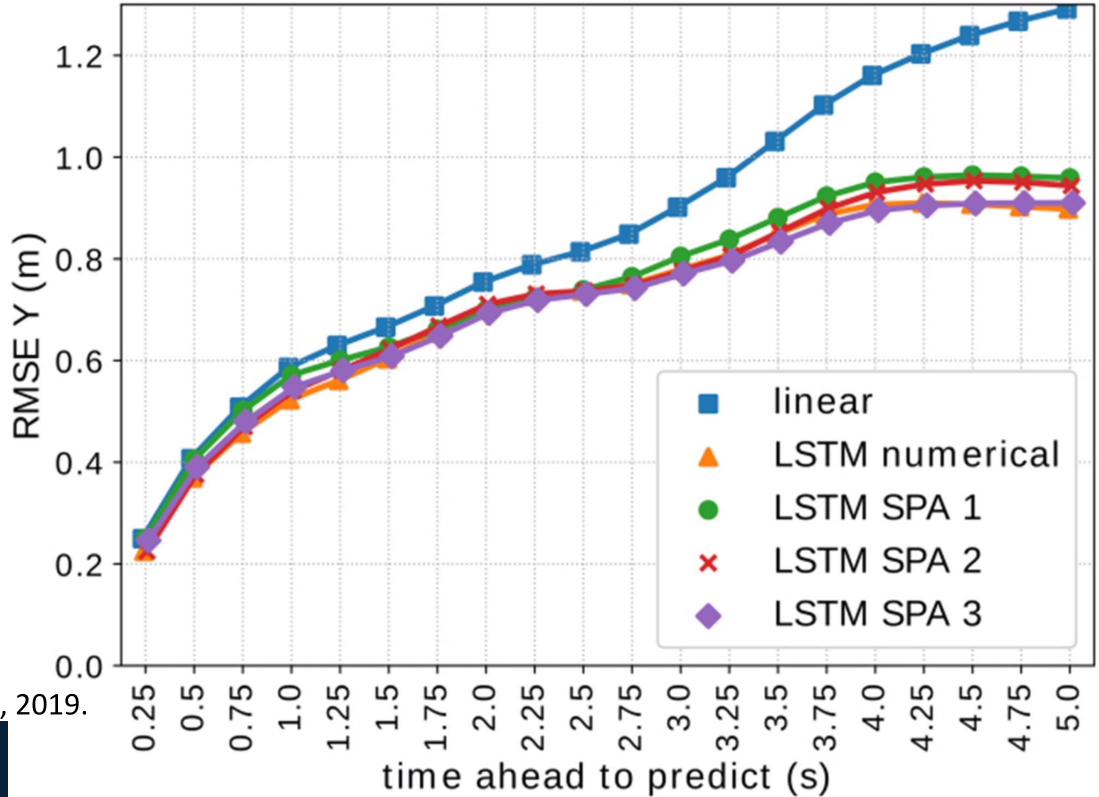
- HD vectors to encapsulate spatial information of multiple objects using the binding operation
  - Number of objects is a variable
- HD vectors as input to a LSTM for seq -to-seq prediction of vehicle positions
  - 5s into the future

$$\mathbf{S}_t = \underbrace{\mathbf{TARGET} \circledast \mathbf{TYPE}_{target} \circledast \mathbf{X}^{x_t} \circledast \mathbf{Y}^{y_t}}_{\text{target-vehicle}}$$

$$\underbrace{\oplus \sum_{obj} \mathbf{TYPE}_{obj} \circledast \mathbf{X}^{x_{obj,t}} \circledast \mathbf{Y}^{y_{obj,t}}}_{\text{other objects}}$$

- Best result in crowded and potentially dangerous driving situations

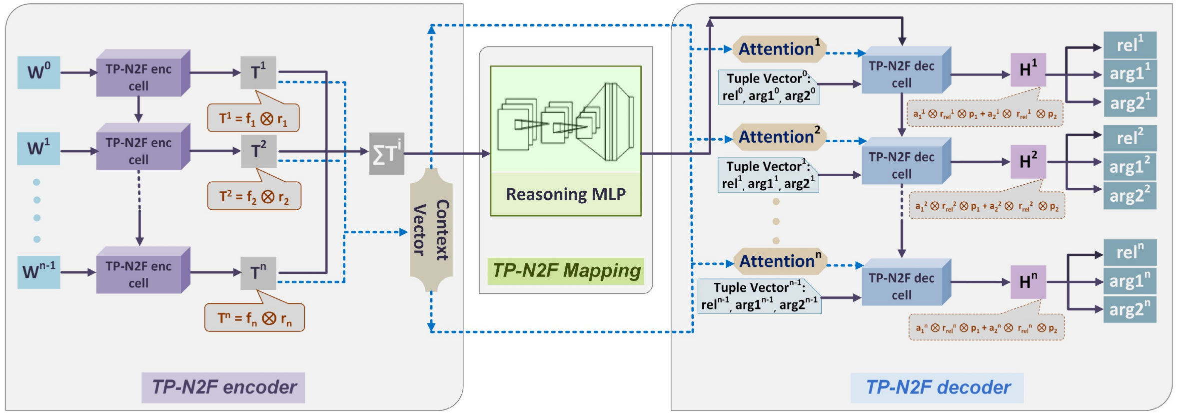


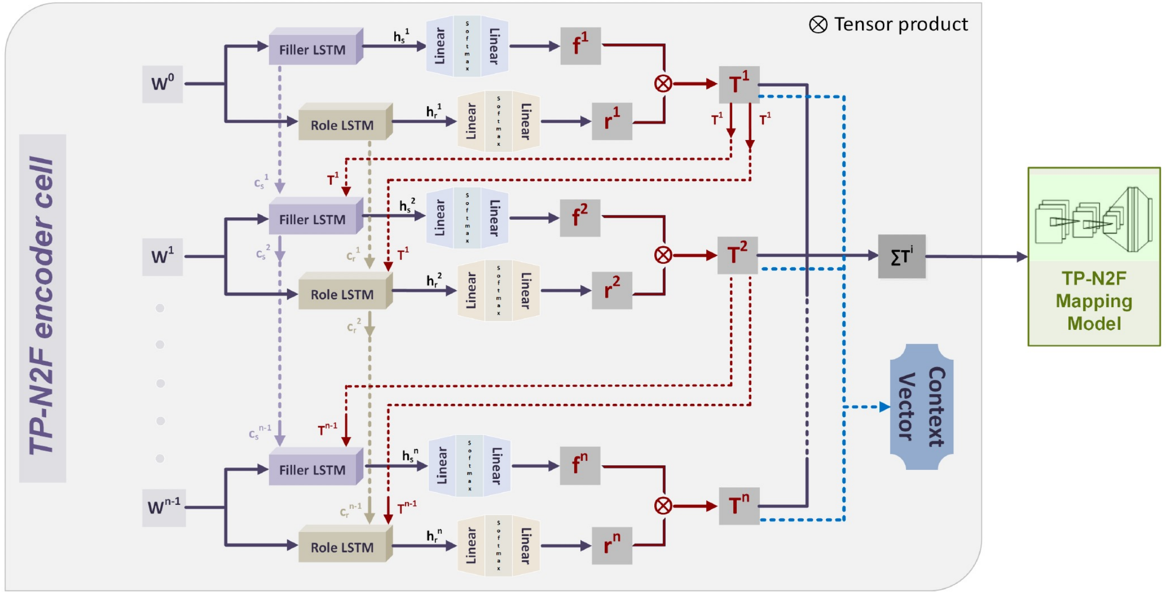


F. Mirus, P. Blouw, T. C. Stewart, J. Conradt, "An Investigation of Vehicle Behavior Prediction Using a Vector Power Representation to Encode Spatial Positions of Multiple Objects and Neural Networks", Frontiers in Neurorobotics, 2019.

Composite data structures: Natural-to Formal-Language Generation

- Tensor Product Representations-based binding
    - Claim: the use of TPRs allows explicit capturing of relational structure to support reasoning
- Represent input data as superposition of tensors representing role-filler pairs
    - Structured representations of inputs are mapped to the structured representations of outputs
- Represent output data as tensor
- The model is not straightforward
    - But demonstrated to obtain good results on 2 datasets
        - MathQA
        - AlgoLisp



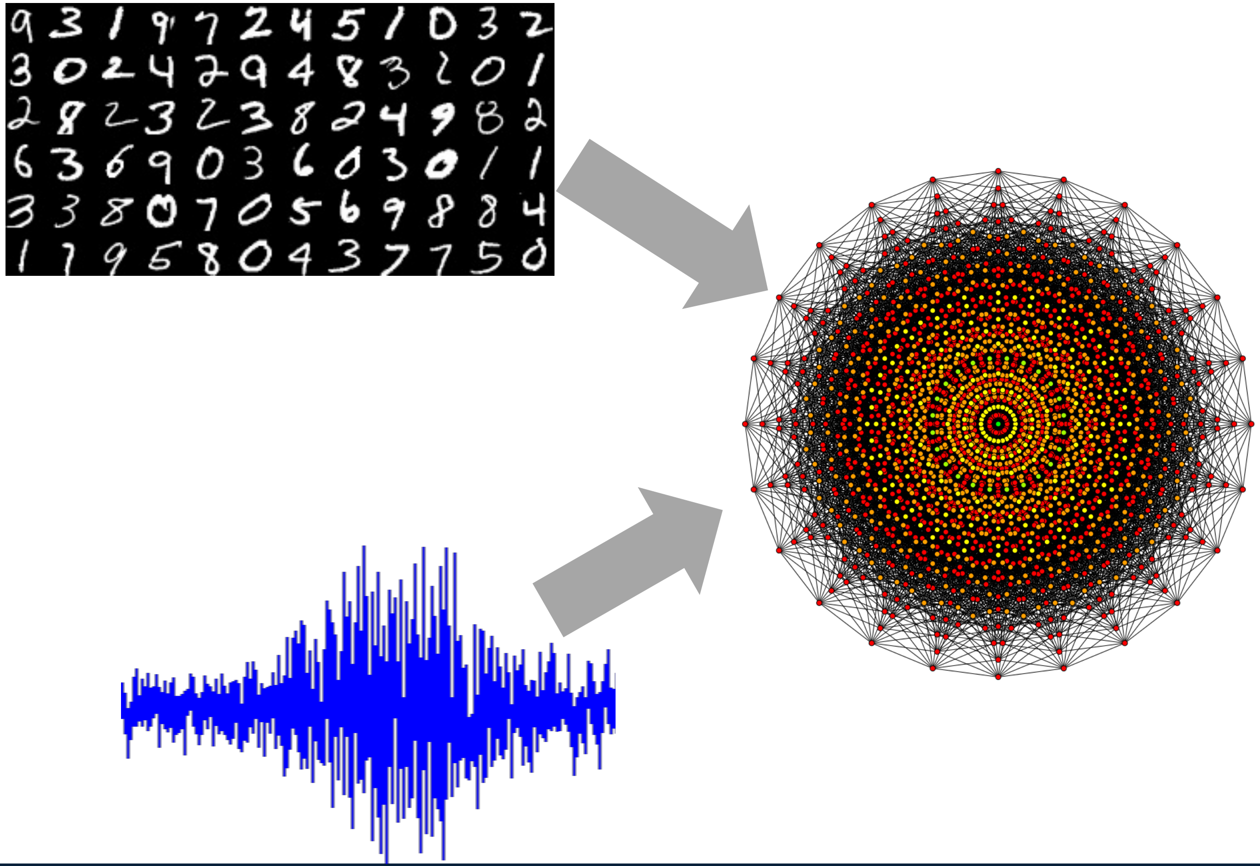


K. Chen, Q. Huang, H. Palangi, P. Smolensky, K. D. Forbus, J. Gao, “Mapping Natural-language Problems to Formal-language Solutions using Structured Neural Representations”, International Conference on Machine Learning (ICML), 2020.

Neural networks for producing HD vectors

Types of output

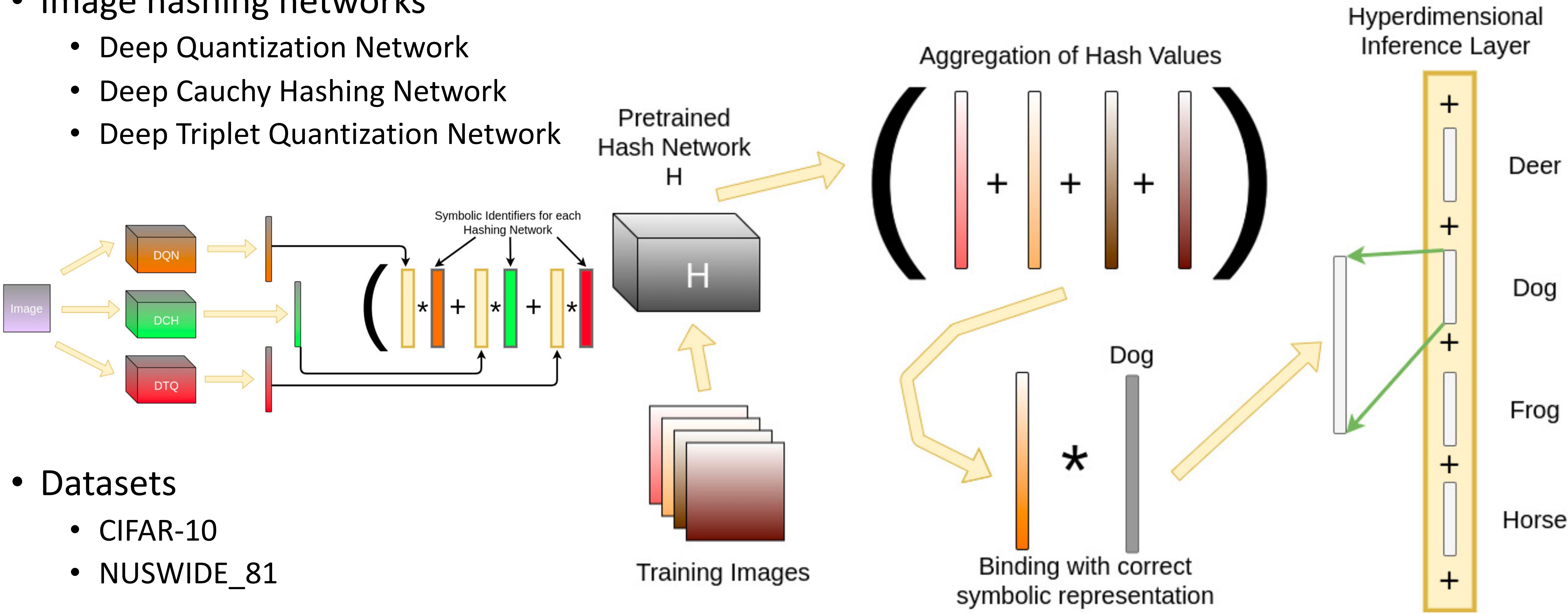
- Transforming data to HD vectors might be a non-trivial task
  - Unstructured and of non-symbolic nature: images
  - Stimulates the interface between neural networks and HD computing/VSA in the other direction



- Transform activations of neural network layer(s) to HD vectors
  - Pre-trained convolutional neural networks
    - Increase the dimensionality
    - Change the format of representations
  - Purposefully train a network
    - Define cost function

Binary HD vectors from images

- Image hashing networks
  - Deep Quantization Network
  - Deep Cauchy Hashing Network
  - Deep Triplet Quantization Network

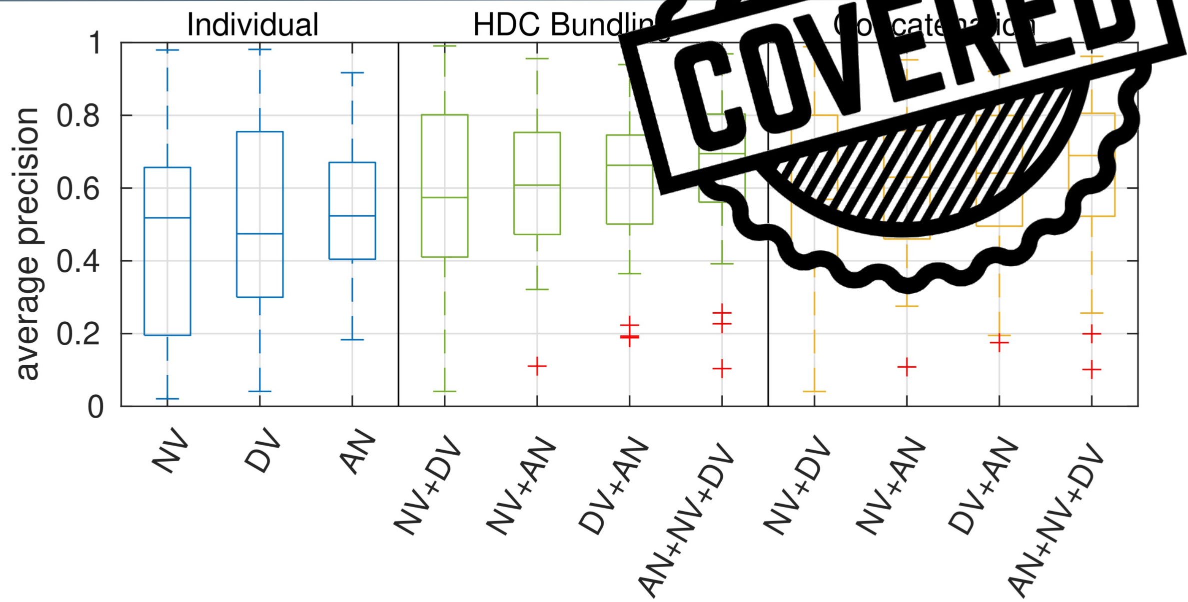


- Datasets
  - CIFAR-10
  - NUSWIDE_81

A. Mitrokhin, P. Sutor, D. Summers-Stay, C. Fermuller, Y. Aloimonos, “Symbolic Representation and Learning with Hyperdimensional Computing”, Frontiers in Robotics and AI, 2020.

Aggregation of Image Descriptors

- Bunch of image descriptors
  - DELF
  - NetVLAD (NV)
  - AlexNet (AN)
  - DenseVLAD (DV)
- Place recognition datasets from mobile robotics
- Random projection controls dimensionality

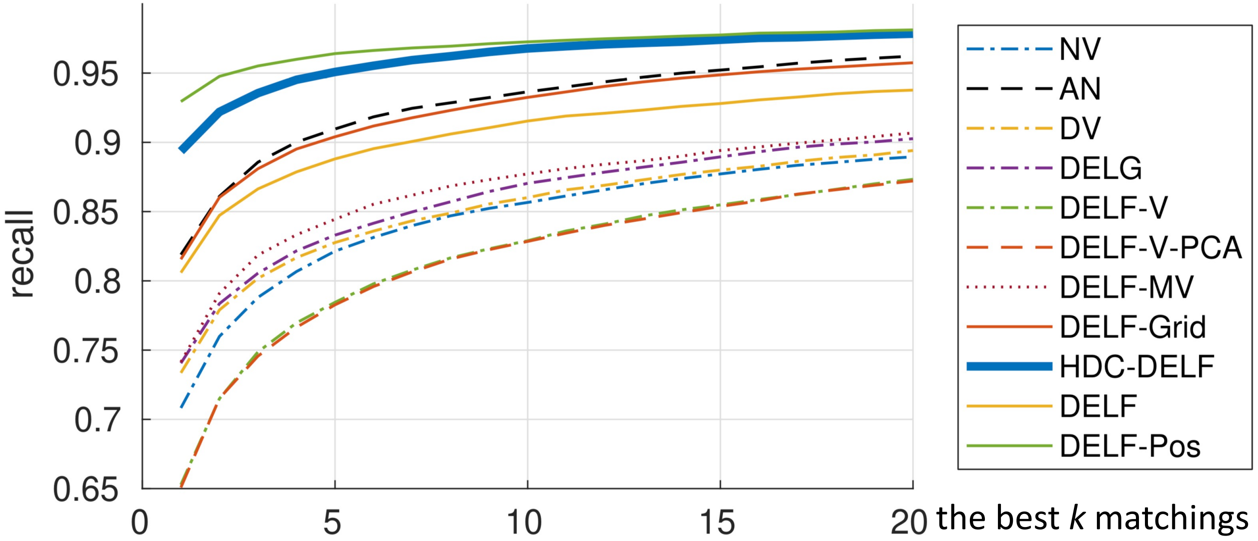


- Form HD vector from holistic image descriptors:

$$H = \bigoplus_{i=1}^{k} H_i = \sum_{i=1}^{k} H_i$$

- Form HD vector from local image descriptors:

$$L = \bigoplus_{i=1}^{k} L_i \otimes P_i$$

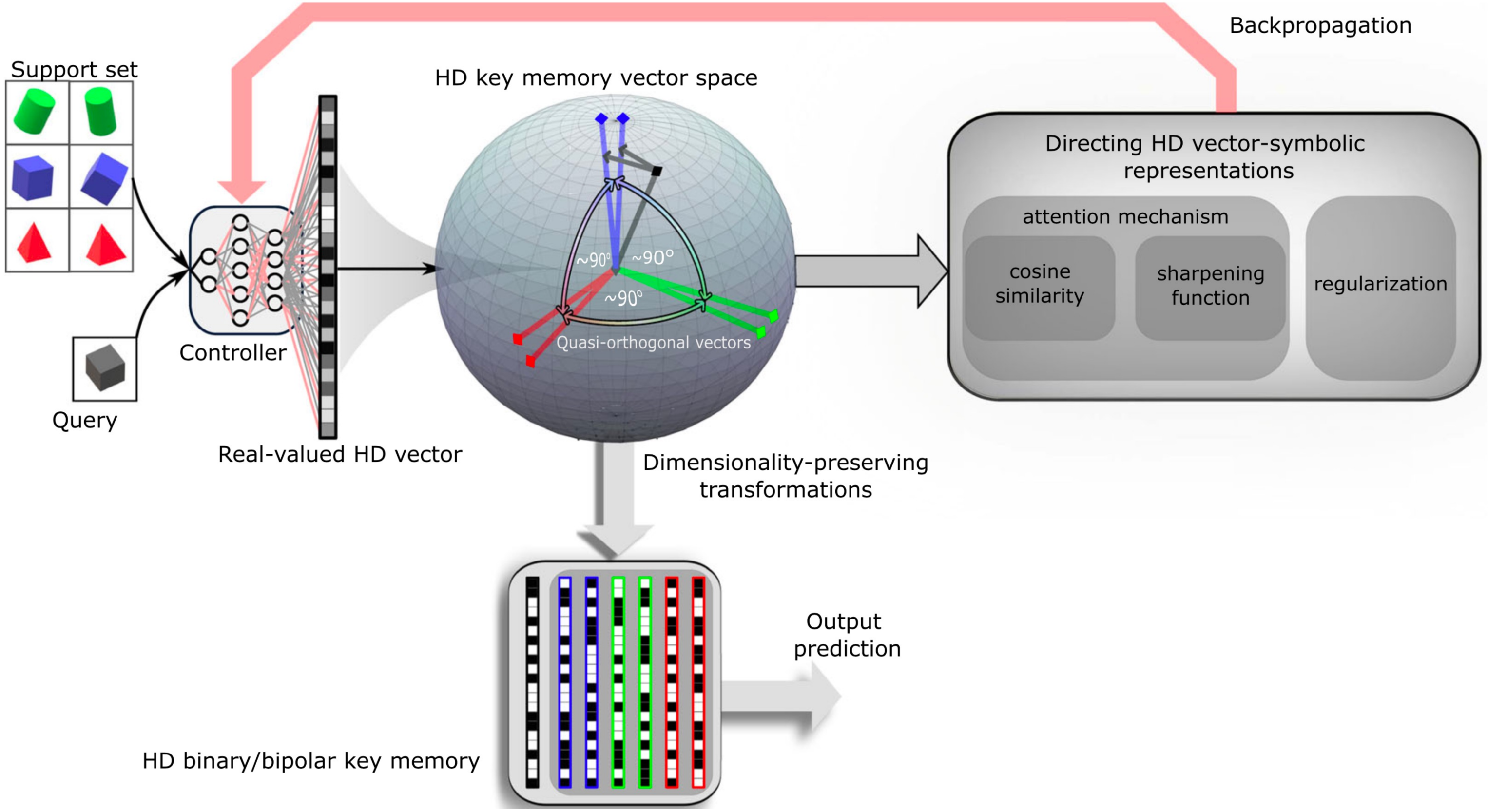


P. Neubert, S. Schubert, “Hyperdimensional Computing as a Framework for Systematic Aggregation of Image Descriptors,” Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

Binary HD vectors from images

- Memory-augmented networks
  - Controller – Convolutional network
  - Key-value memory
    - Content addressable memory
    - Explicit memory
  - Few-shot learning

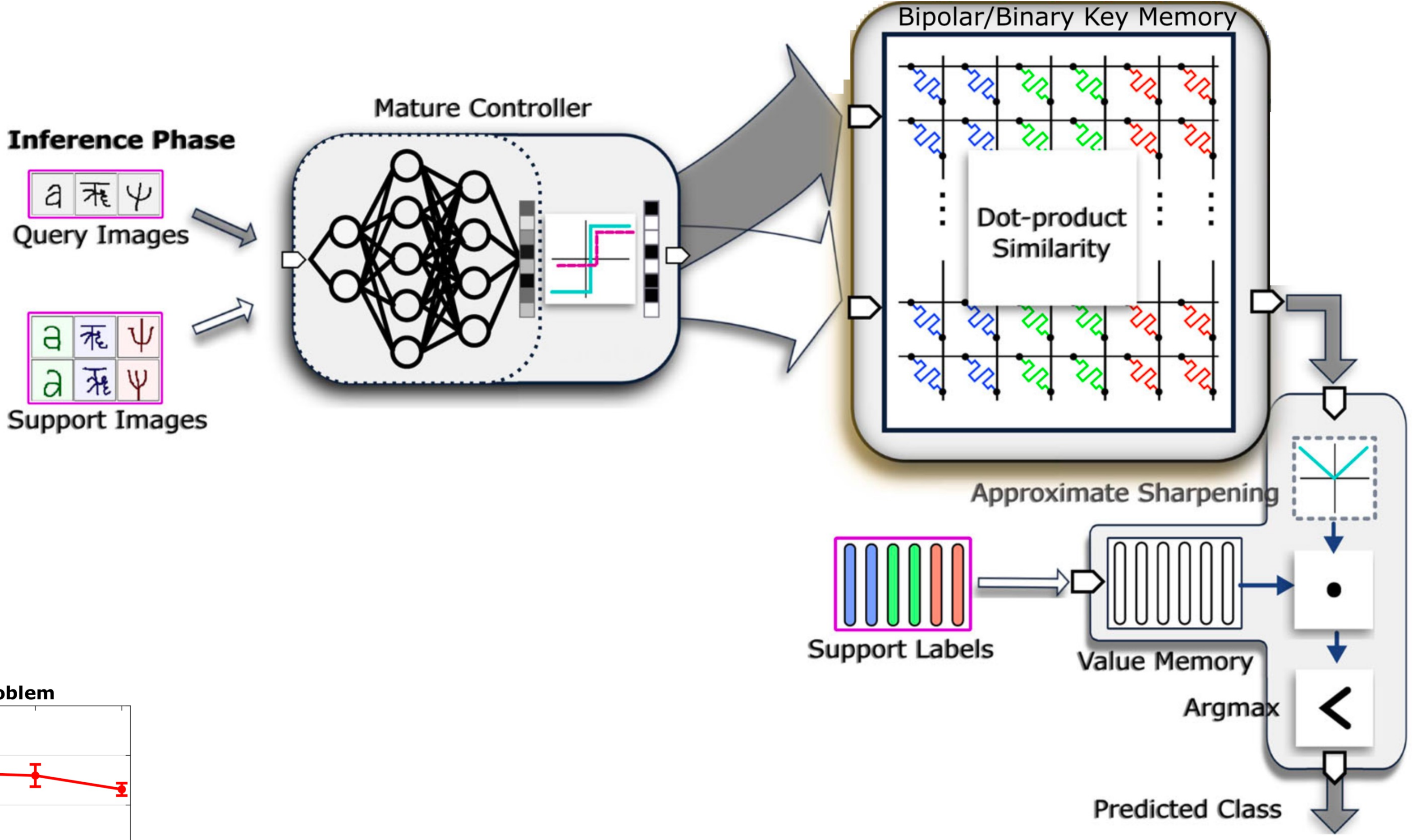
- Evaluation
  - Omniglot dataset
  - Phase-change memory devices



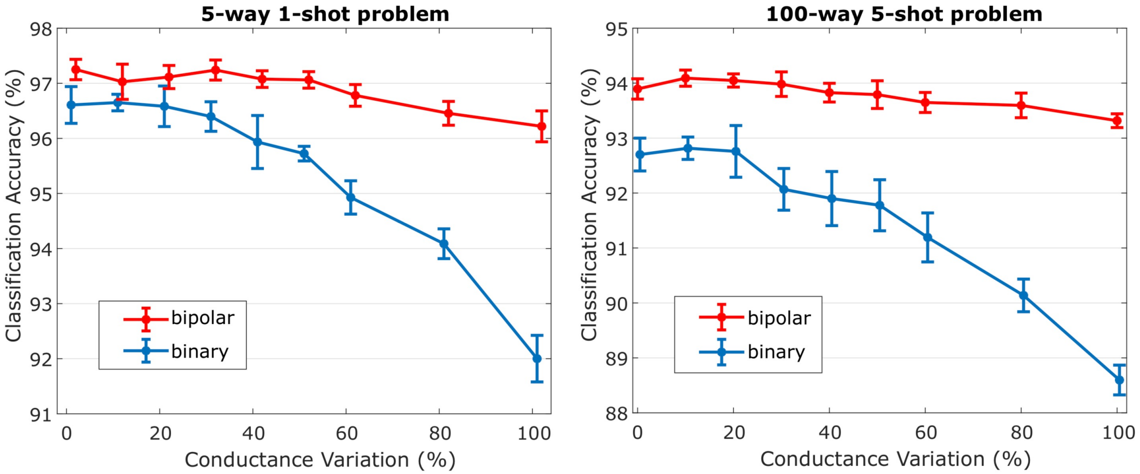
G. Karunaratne, M. Schmuck, M. Le Gallo, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi, "Robust High-dimensional Memory-augmented Neural Networks", Nature Communications, 2021.

Binary HD vectors from images

- Memory-augmented networks
  - Controller – Convolutional network
  - Key-value memory
    - Content addressable memory
    - Explicit memory
  - Few-shot learning



- Evaluation
  - Omniglot dataset
  - Phase-change memory devices

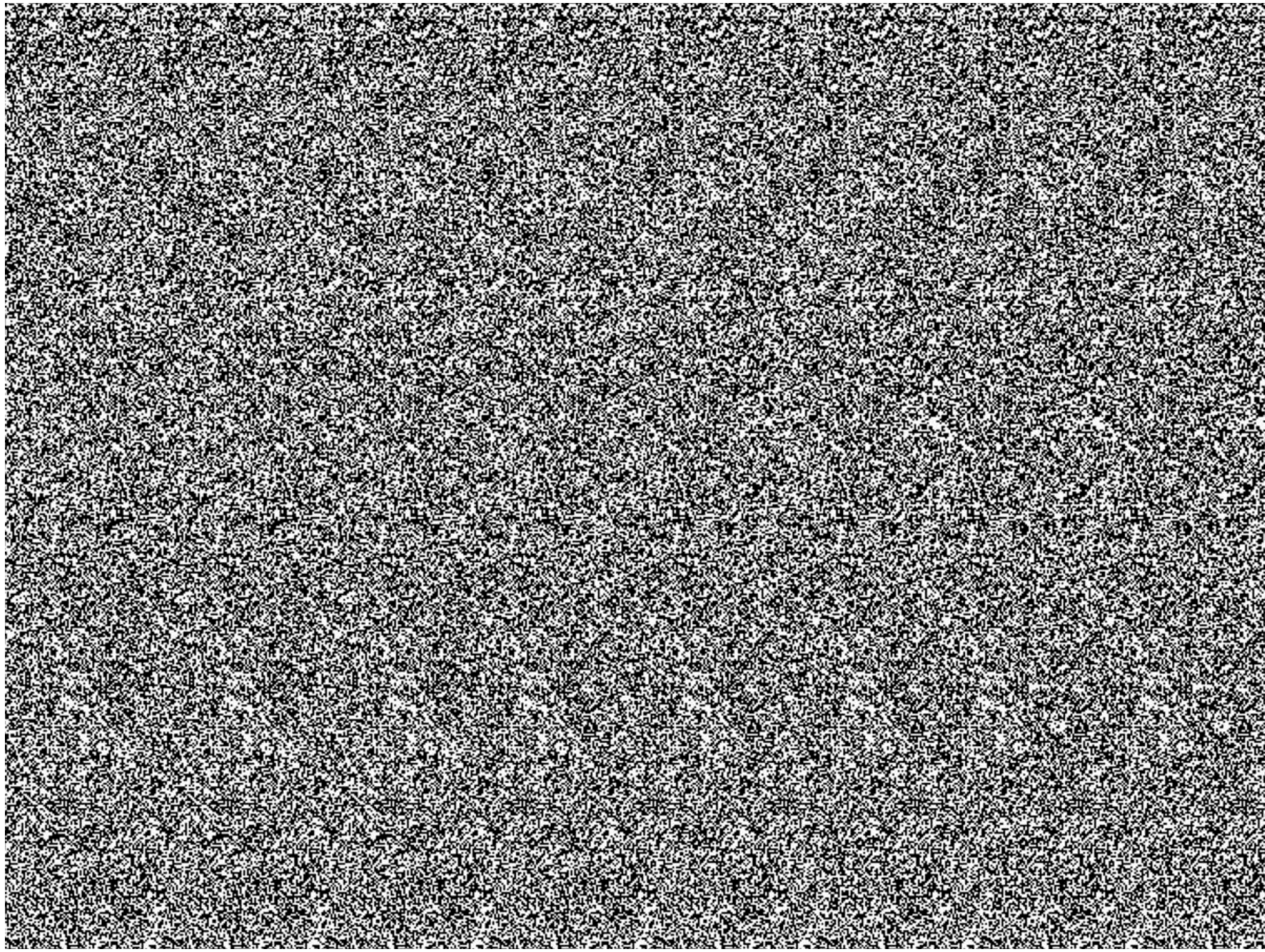


G. Karunaratne, M. Schmuck, M. Le Gallo, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi, "Robust High-dimensional Memory-augmented Neural Networks", Nature Communications, 2021.

HD Computing/VSA connections to randomized neural networks

Randomness in neural networks

- Stochastic assignment of a subset of the networks' weights
    - Simpler (often linear) optimization problem



- Several broad families of models:
    - Randomized kernel Approximations
        - Chris's Lecture for Module 8
    - **Randomized feed-forward networks**
    - **Randomized connected recurrent networks**

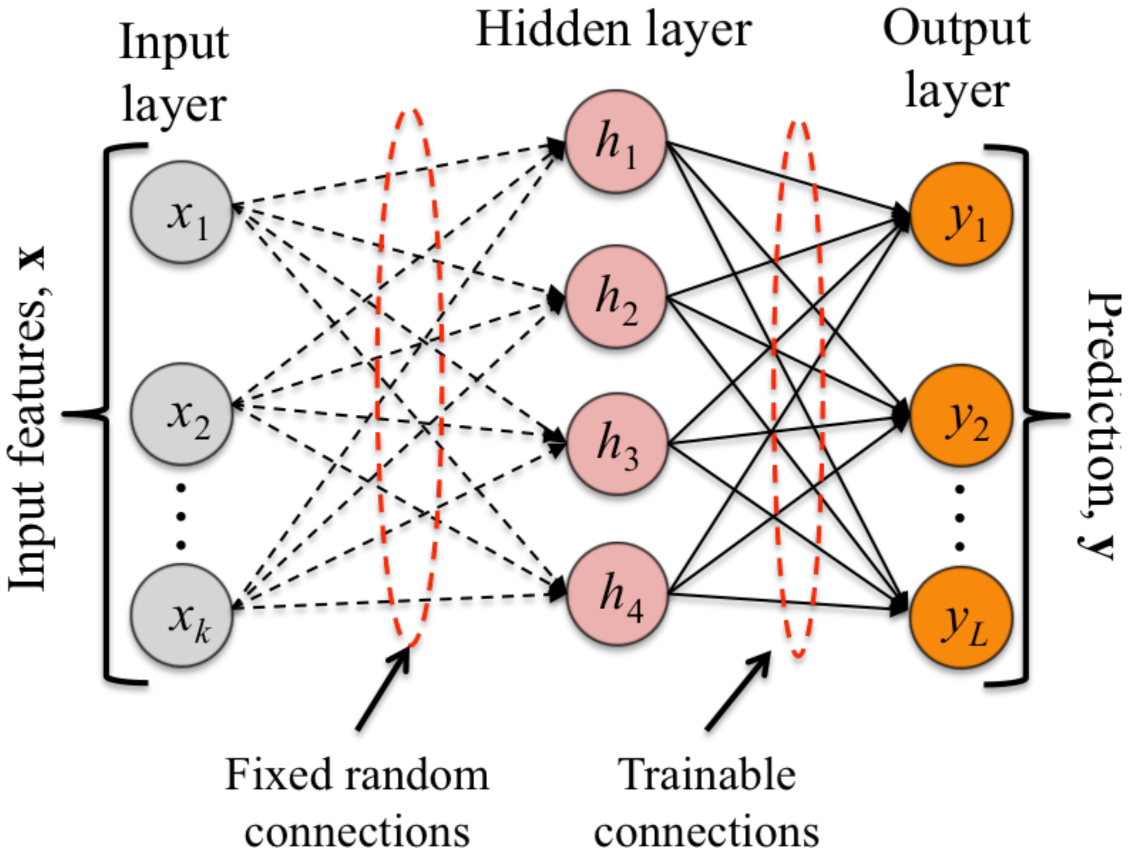
- Two fundamental ideas:
    - Randomization defines feature map lifting the input into a high-dimensional space
    - Resulting optimization problem is cast as a standard linear (regularized) least-squares

S. Scardapane and D. Wang, "Randomness in Neural Networks: an Overview," Data Mining and Knowledge Discovery, 2017.

Randomly connected neural networks

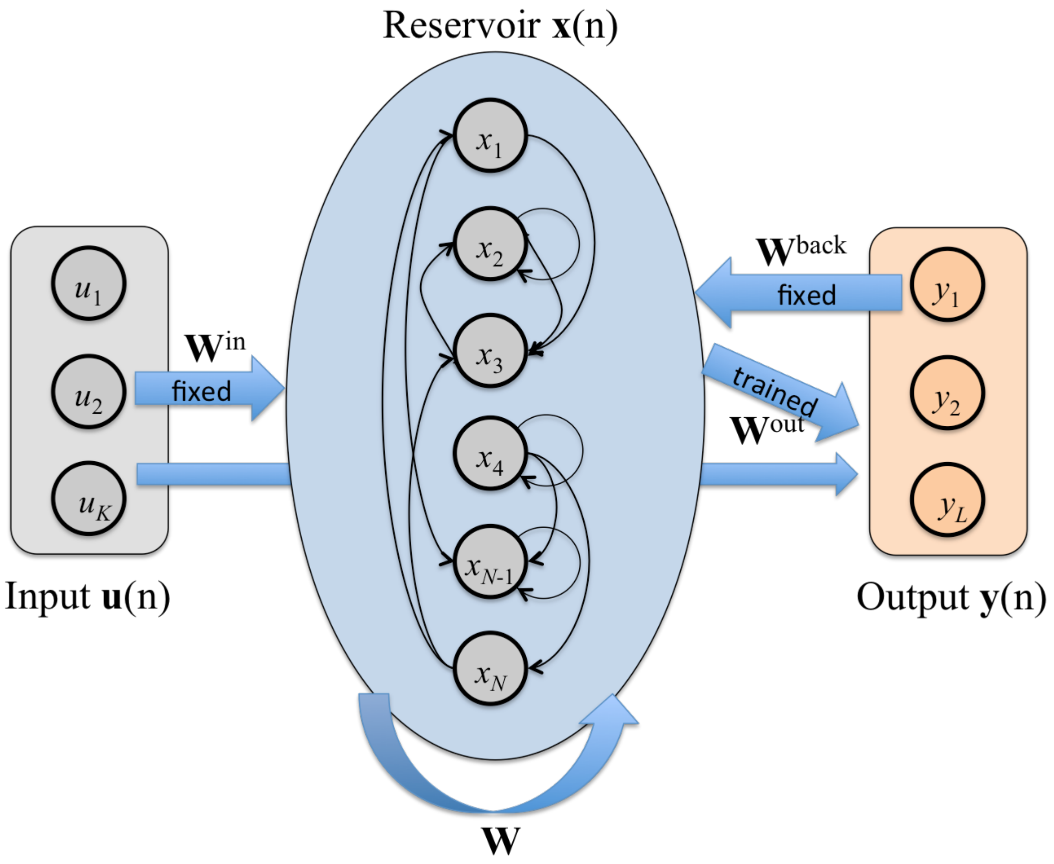
Feed-forward neural networks:

- Random Vector Functional Link Networks, RVFL
- Extreme Learning Machines, ELM



Recurrent neural networks:

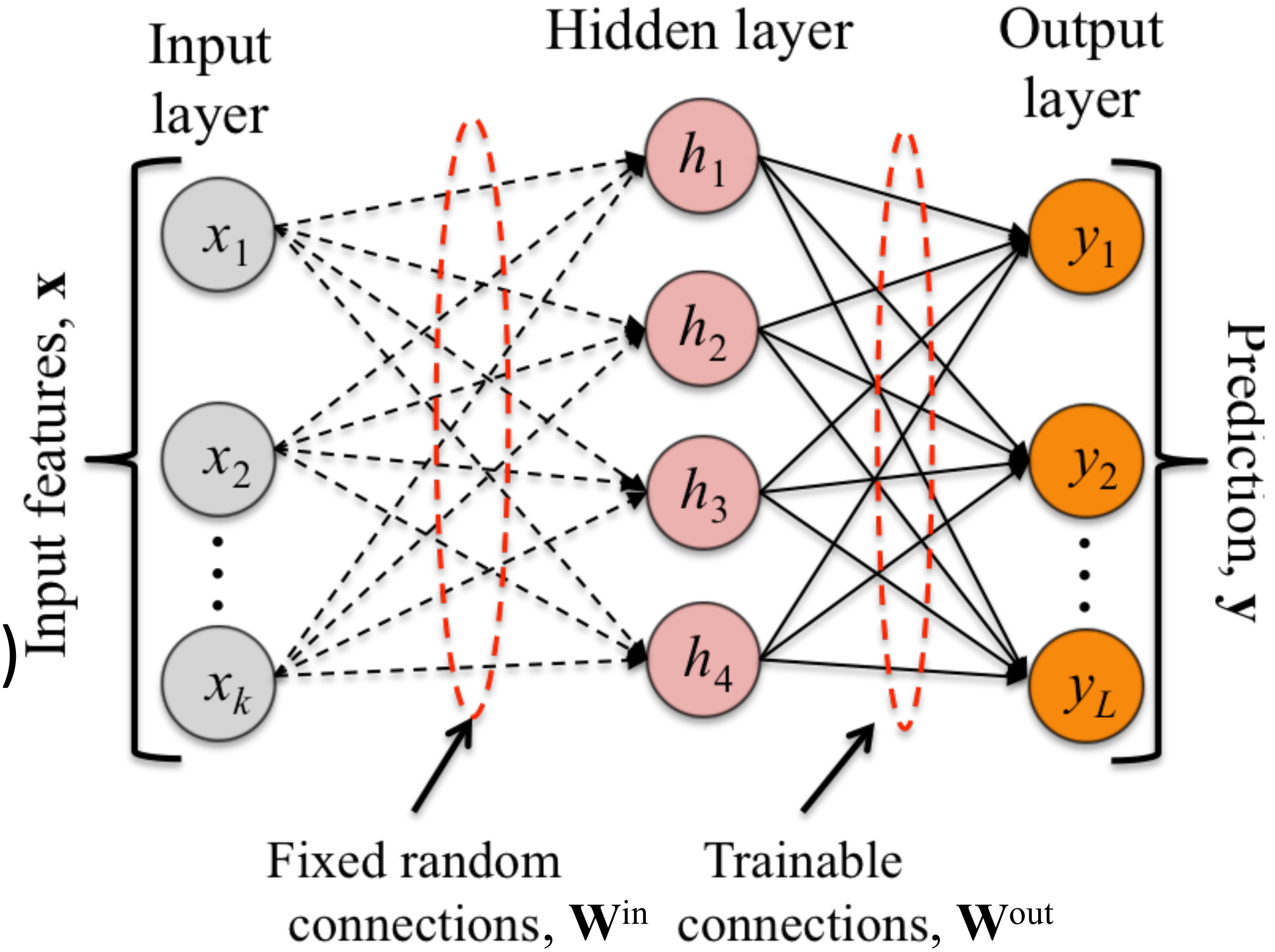
- Echo State Networks, ESN
- Liquid State Machines, LSM



Random Vector Functional Link Networks

- Three layers:
    - input
    - hidden
    - output
- Random and fixed connections
- Non-linear activation function – tanh(*x*)
- Readout connections: RLS

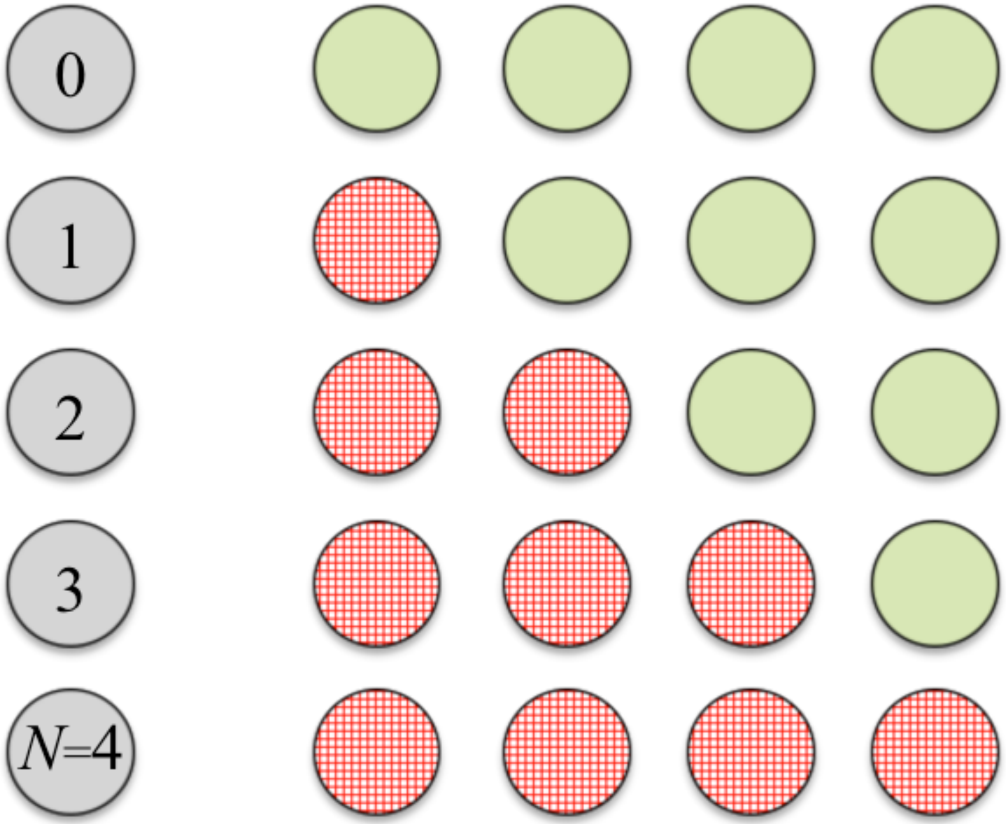
$$\mathbf{W}^{\text{out}} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^T\mathbf{y}$$



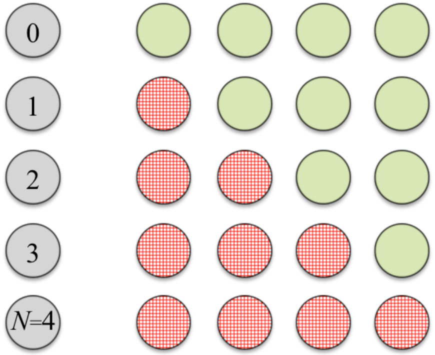
B. Igelnik, Y. Pao, “Stochastic Choice of Basis Functionsin Adaptive Function Approximation and the Functional-Link Net,” IEEE Transactions on Neural Networks, 1995.

Density-based encoding

- Connections input to hidden layers
  - Projection (lifting) to HD space
  - Random projection
- Transform scalars to HD vectors via density-based encoding
  - Thermometer codes
    - Chris's Lecture for Module 8
- Binding operation with HD for weight and mapped scalar

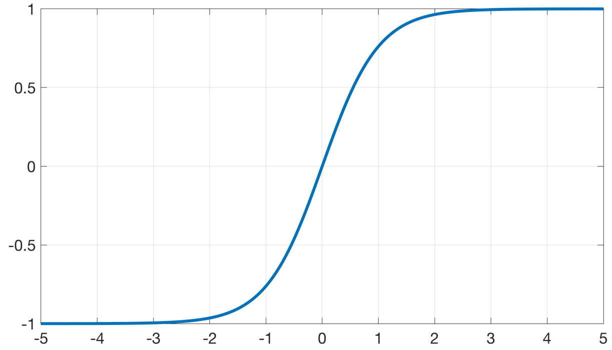


Randomized neural networks via HD computing/VSA

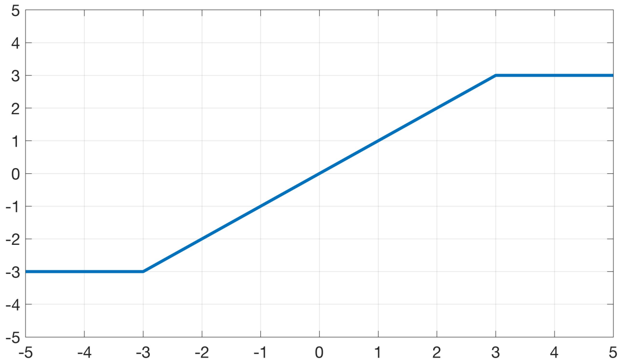


- Weight matrix is random and bipolar
  - Interpreted as a set of HD vectors
- Binding operation with HD vectors for weight and mapped scalar
  - Associating each feature with its HD vector
- Bundling all associations -> linear activation of hidden layer
- Non-linear activation function -> bundling operation in a limited range
  - Clipping as a nonlinearity function

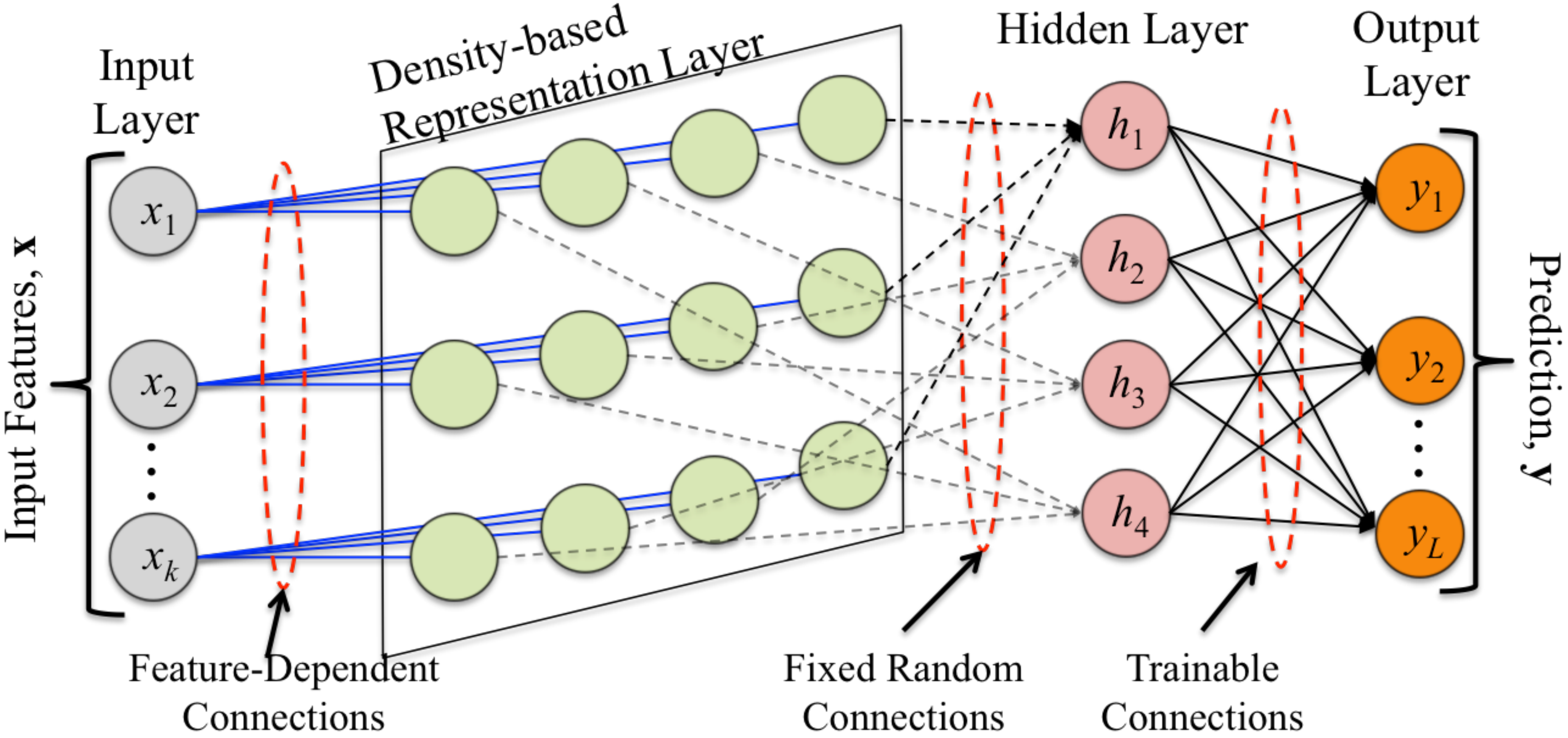
tanh(x)



clipping(x), k=3



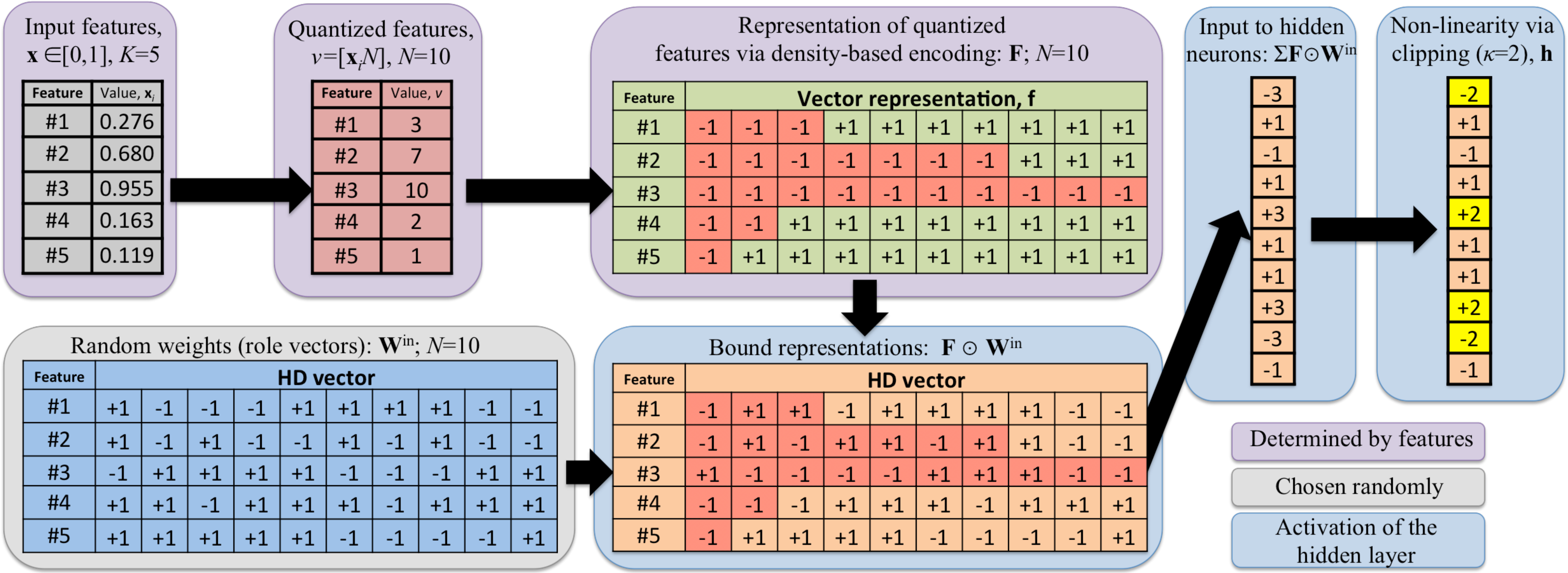
intRVFL architecture



D. Kleyko, M. Kheffache, E. P. Frady, U. Wiklund, and E. Osipov, “Density Encoding Enables Resource-Efficient Randomly Connected Neural Networks,” IEEE Transactions on Neural Networks and Learning Systems, 2021.

intRVFL example

- Hidden layer (Reservoir) contains only integers in the limited range [-*k*, *k*]
  - One neuron requires $\log_2(2k+1)$ bits -> $k=3$ – 3 bits



Input features, $\mathbf{x} \in [0,1]$, $K=5$

Feature	Value, $\mathbf{x}_i$
#1	0.276
#2	0.680
#3	0.955
#4	0.163
#5	0.119

Quantized features, $v=[\mathbf{x}_i N]$, $N=10$

Feature	Value, *v*
#1	3
#2	7
#3	10
#4	2
#5	1

Representation of quantized features via density-based encoding: **F**; $N=10$

Feature	Vector representation, f									
#1	-1	-1	-1	+1	+1	+1	+1	+1	+1	+1
#2	-1	-1	-1	-1	-1	-1	-1	+1	+1	+1
#3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
#4	-1	-1	+1	+1	+1	+1	+1	+1	+1	+1
#5	-1	+1	+1	+1	+1	+1	+1	+1	+1	+1

Random weights (role vectors): $\mathbf{W}^{\text{in}}$; $N=10$

Feature	HD vector									
#1	+1	-1	-1	-1	+1	+1	+1	+1	-1	-1
#2	+1	-1	+1	-1	-1	+1	-1	+1	-1	-1
#3	-1	+1	+1	+1	+1	-1	-1	-1	+1	+1
#4	+1	+1	-1	+1	+1	+1	-1	+1	+1	+1
#5	+1	+1	+1	+1	+1	-1	-1	-1	-1	+1

Bound representations: $\mathbf{F} \odot \mathbf{W}^{\text{in}}$

Feature	HD vector									
#1	-1	+1	+1	-1	+1	+1	+1	+1	-1	-1
#2	-1	+1	-1	+1	+1	-1	+1	+1	-1	-1
#3	+1	-1	-1	-1	-1	+1	+1	+1	-1	-1
#4	-1	-1	-1	+1	+1	+1	-1	+1	+1	+1
#5	-1	+1	+1	+1	+1	-1	-1	-1	-1	+1

Input to hidden neurons: $\Sigma \mathbf{F} \odot \mathbf{W}^{\text{in}}$

-3
+1
-1
+1
+3
+1
+1
+3
-3
-1

Non-linearity via clipping ($\kappa=2$), **h**

-2
+1
-1
+1
+2
+1
+1
+2
-2
-1

Determined by features

Chosen randomly

Activation of the hidden layer

intRVFL evaluation: classification on 121 datasets

- 121 datasets for classification from UCI Machine Learning Repository
- Number of examples: min – 10; max – 130064; median – 683;
- Number of features: min – 3; max – 262; median – 16;
- Number of classes: min – 2; max – 100; median – 3;

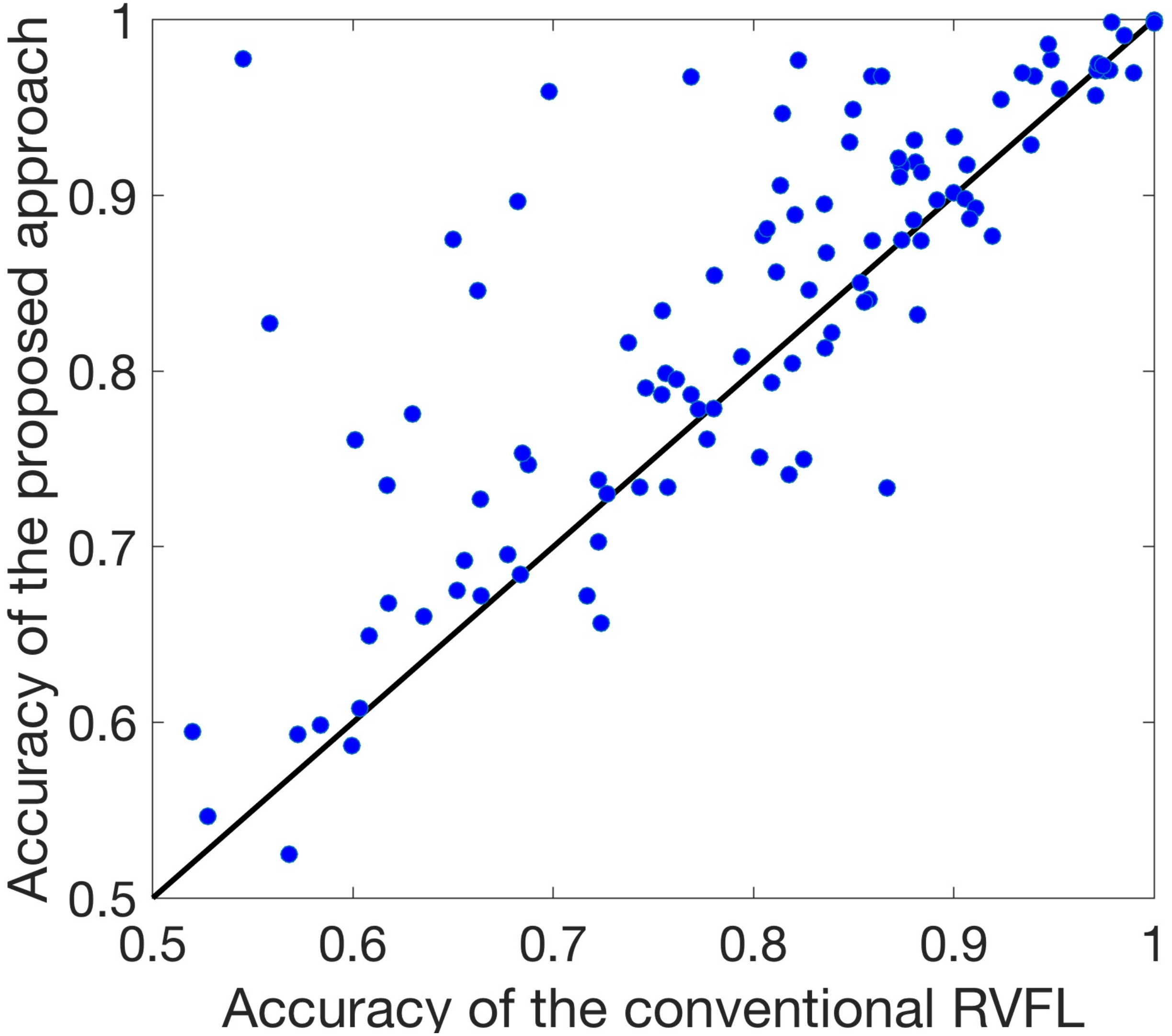
- Features were normalized to be in [0, 1] range

- Average accuracy of the best classifier (Random Forest): 0.82
- Average accuracy of the linear classifier: 0.73

M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?” Journal of Machine Learning Research, 2014.

Evaluation: results unlimited resources

- Grid search:
  - Number of hidden neurons (*N*) was varied in the range [50, 1500] with step 50
  - Regularization parameter was in the range $2^{[-10,5]}$
  - κ varied between {1, 3, 5, 7}
- Floating point read-out matrix via RLS
- Average accuracy conventional RVFL: **0.76**
- Average accuracy intRVFL: **0.80**
- Extra experiments:
  - RVFL with direct weights to input features: 0.76
  - RVFL with quantized features: 0.76
  - RVFL with optimized input projection: 0.71
  - RVFL with *N* for proposed approach: 0.75
  - intRVFL with RVFL's *N*: 0.78



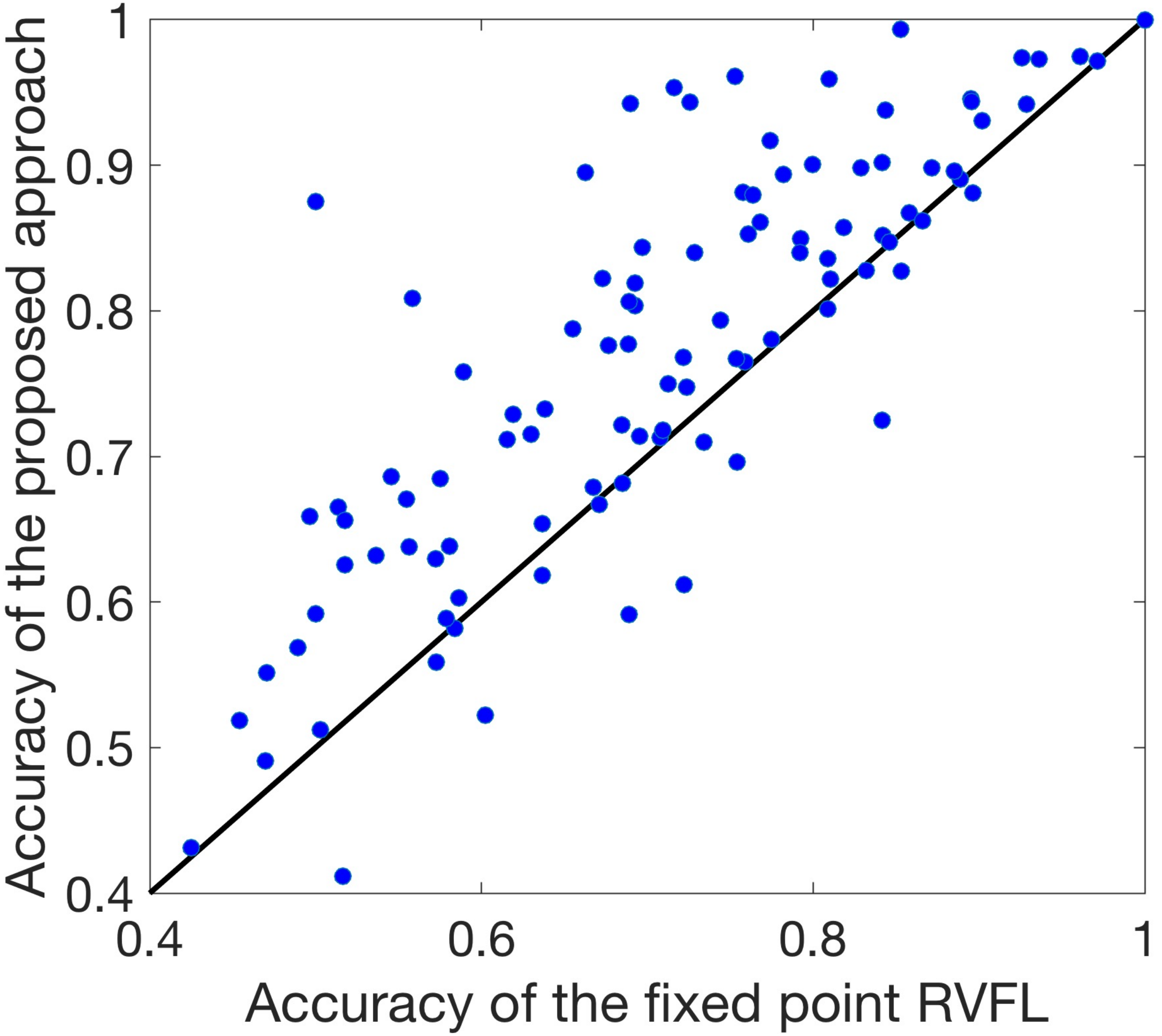
Evaluation: results limited resources

- Fixed energy budget on FPGA
  - “Poorman’s” bounded optimality
  - Effectively it limits *n*

- Finite precision RVFL (8-bits)

- Average accuracy fixed point RVFL: **0.65**

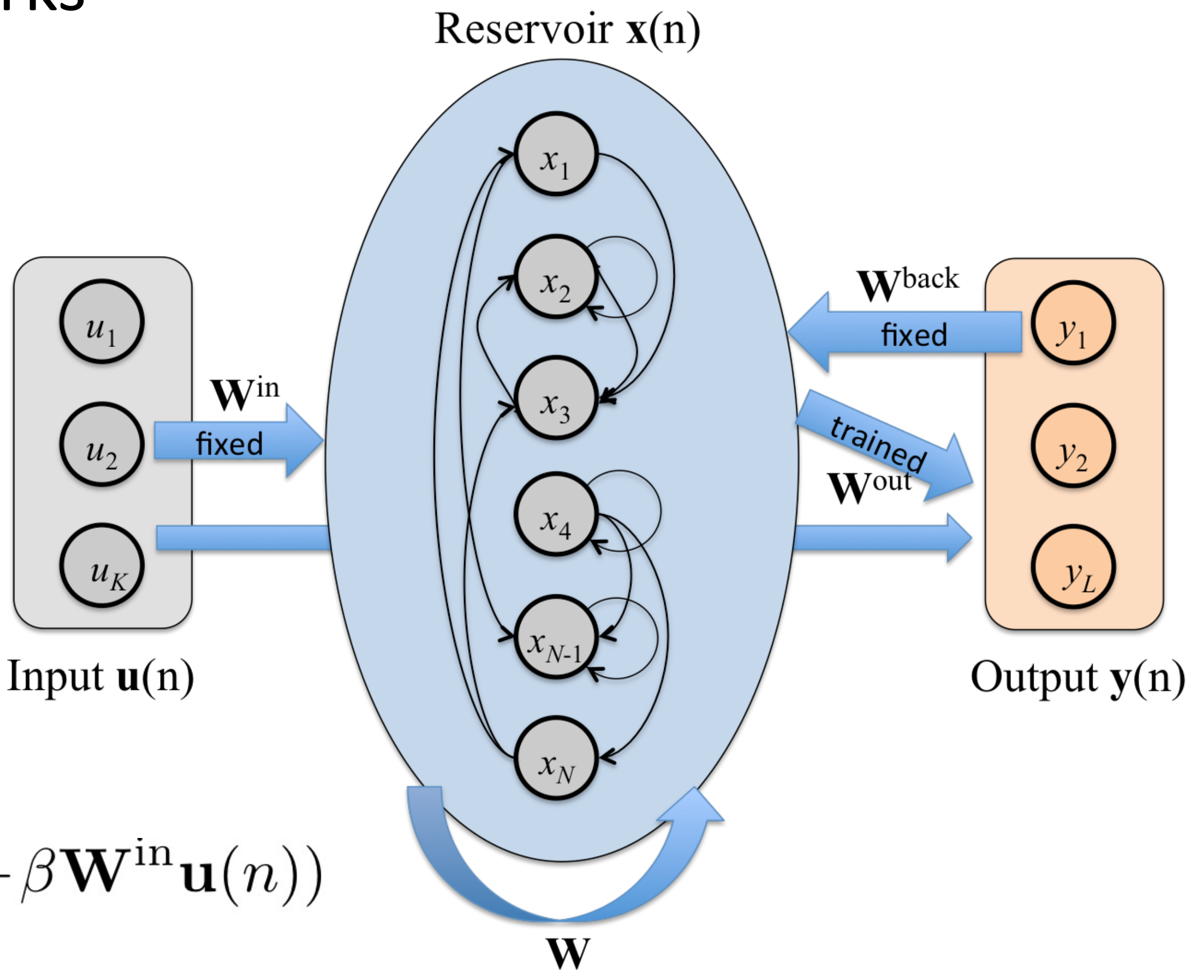
- Average accuracy intRVFL: **0.73**



A. Rosato, R. Altilio, and M. Panella, “Finite Precision Implementation of Random Vector Functional-Link Networks,” in International Conference on Digital Signal Processing, 2017.

Echo State Networks

- An approach to Recurrent Neural Networks
- Three layers
    - input
    - hidden
    - output
- Non-linear activation function – tanh(*x*)
- Random and fixed connections
- Recurrent connections between hidden neurons, **W**



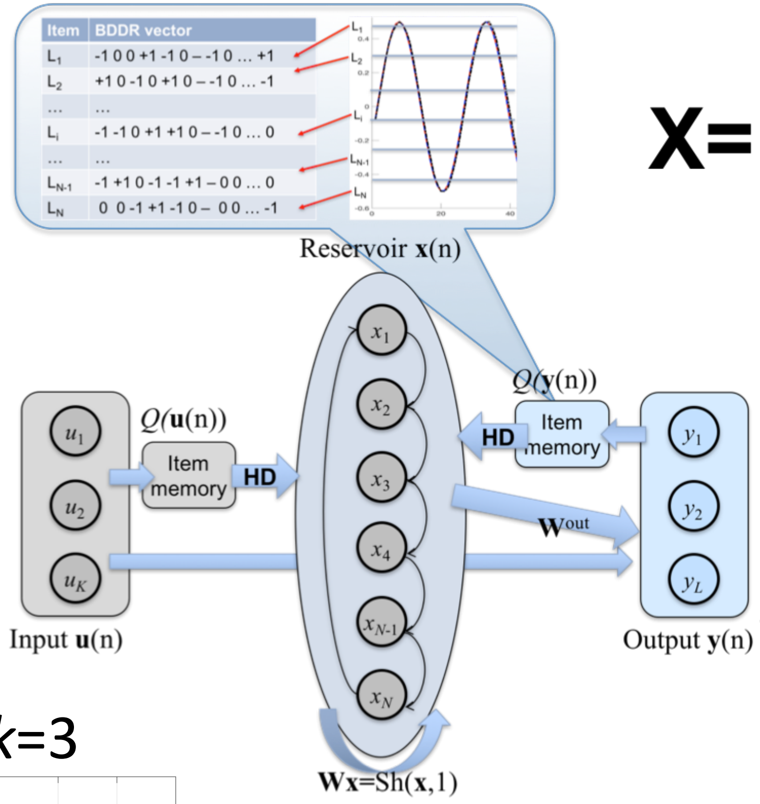
$$\mathbf{x}(n) = (1-\alpha)\mathbf{x}(n-1) + \alpha \tanh(\gamma \mathbf{W}\mathbf{x}(n-1) + \beta \mathbf{W}^{\text{in}}\mathbf{u}(n))$$

H. Jaeger, "Tutorial on Training Recurrent Neural Networks, Covering BPTT, RTRL, EKF and the Echo State Network Approach," Technical Report GMD Report 159, German National Research Center for Information Technology, 2002.

Integer Echo State Networks

- Reservoir contains only integers in the limited range [-*k*, *k*]

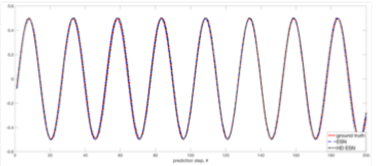
- One neuron requires $\log_2(2k+1)$ bits
    - *k*=3 – 3 bits



Item	BDDR vector
L_1	-1 0 0 +1 -1 0 – -1 0 … +1
L_2	+1 0 -1 0 +1 0 – -1 0 … -1
…	…
L_i	-1 -1 0 +1 +1 0 – -1 0 … 0
…	…
L_{N-1}	-1 +1 0 -1 -1 +1 – 0 0 … 0
L_N	0 0 -1 +1 -1 0 – 0 0 … -1

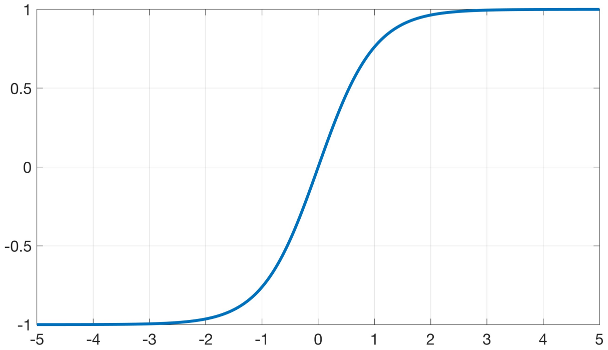
$$\mathbf{X}=\begin{pmatrix} 3 & 3 & 3 & -2 & -1 & -2 & -1 \\ 3 & 3 & 2 & 3 & -3 & -2 & -1 \\ 0 & 2 & 3 & 3 & 3 & -2 & -3 \\ -3 & -1 & 1 & 2 & 3 & 2 & -1 \\ 3 & -2 & -2 & 0 & 1 & 2 & 1 \\ 3 & 3 & -1 & -1 & 1 & 2 & 3 \end{pmatrix}$$

Y=

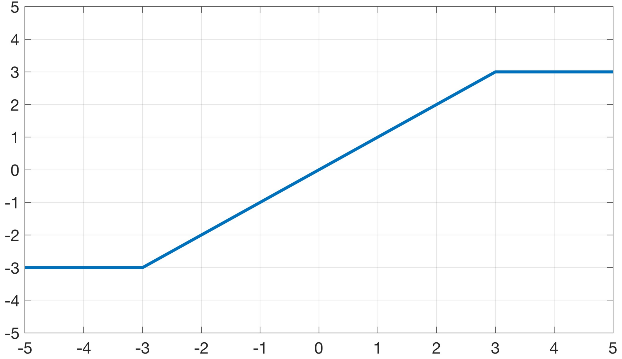


$$\mathbf{W}=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

tanh(*x*)



clipping(*x*), *k*=3

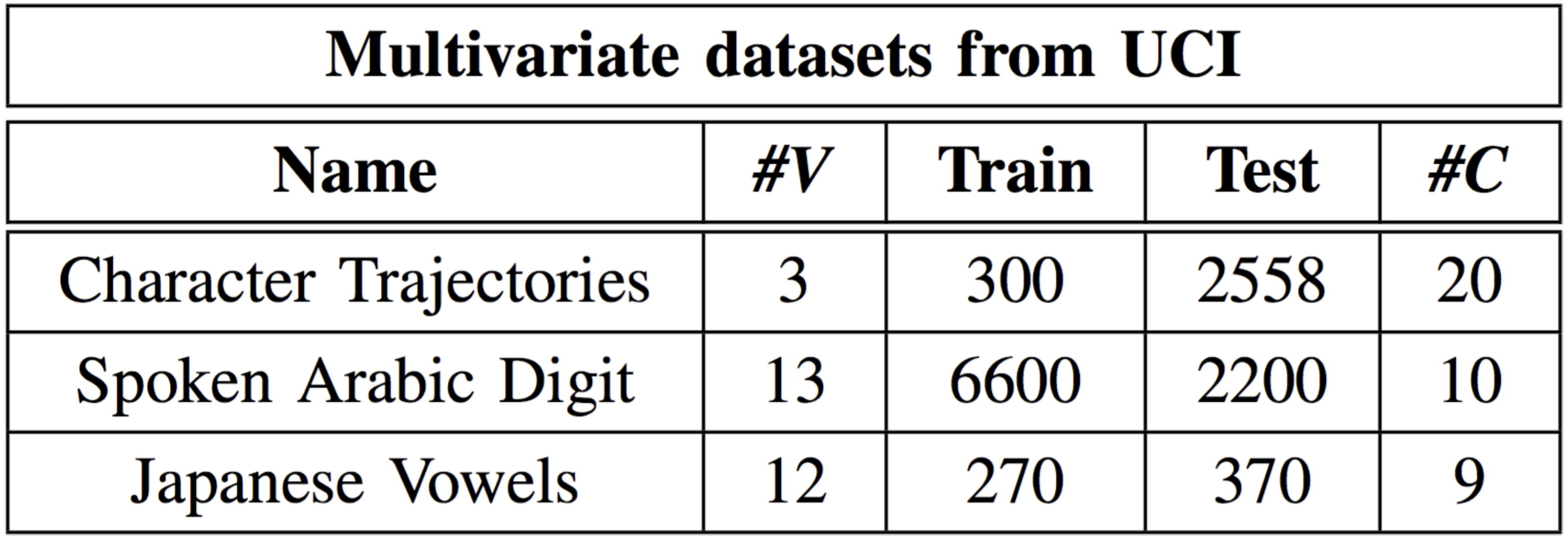


A. Rodan, P. Tino, "Minimum Complexity Echo State Network," IEEE Transactions on Neural Networks, 2011.

D. Kleyko, E. P. Frady, M. Kheffache, E. Osipov, "Integer Echo State Networks: Efficient Reservoir Computing for Digital Hardware," IEEE Transactions on Neural Networks and Learning Systems, 2020.

intESN evaluation: Time-series classification

- 800 neurons
- 3 datasets
- 3.9 times faster than ESN



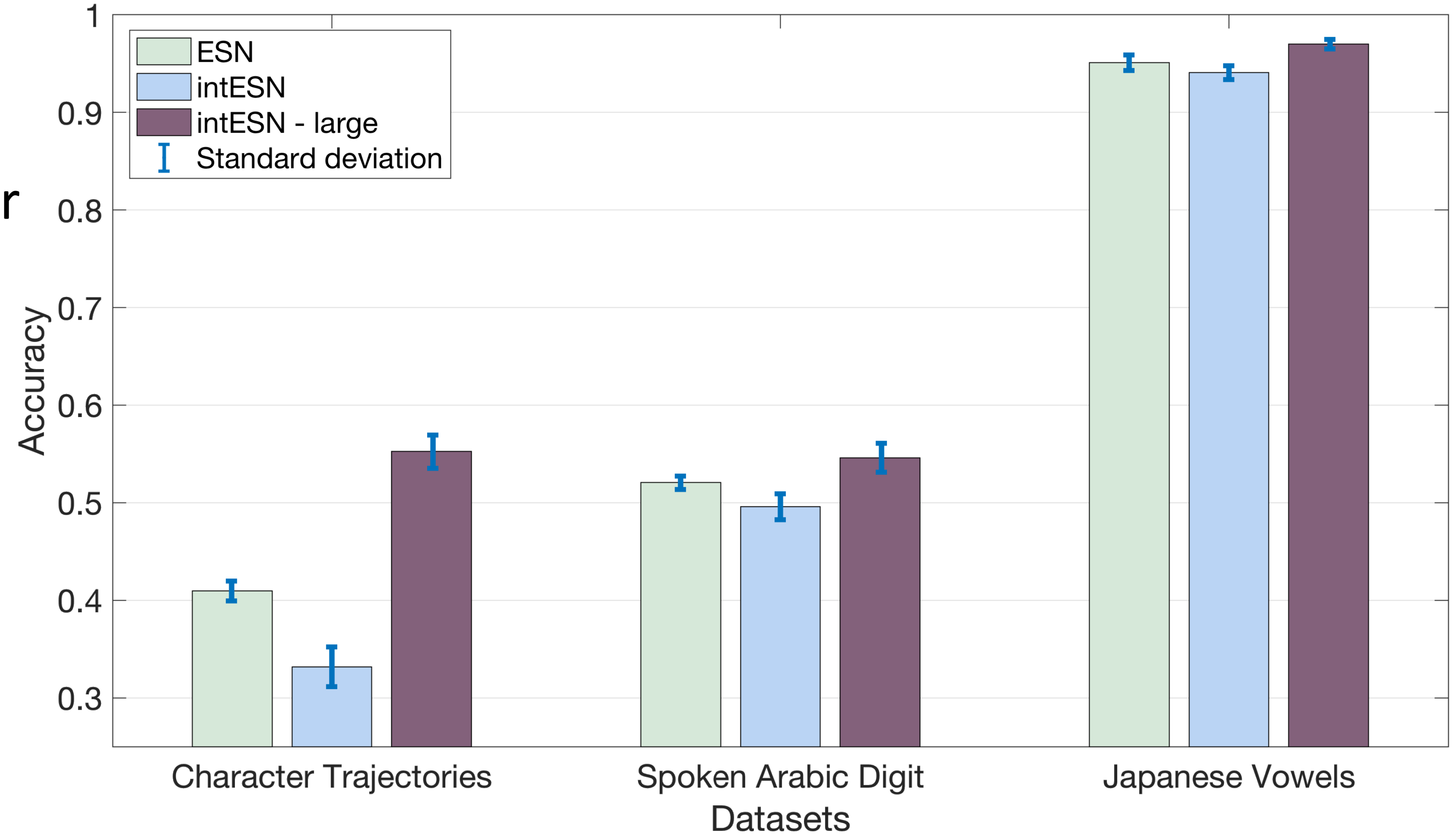
Multivariate datasets from UCI

Name	***#V***	**Train**	**Test**	***#C***
Character Trajectories	3	300	2558	20
Spoken Arabic Digit	13	6600	2200	10
Japanese Vowels	12	270	370	9

D. Dua, C. Graff, “UCI Machine Learning Repository,” University of California, Irvine, School of Information and Computer Sciences, 2019.

intESN evaluation: Time-series classification

- 800 neurons
- 3 datasets
- 3.9 times faster than ESN



D. Dua, C. Graff, “UCI Machine Learning Repository,” University of California, Irvine, School of Information and Computer Sciences, 2019.

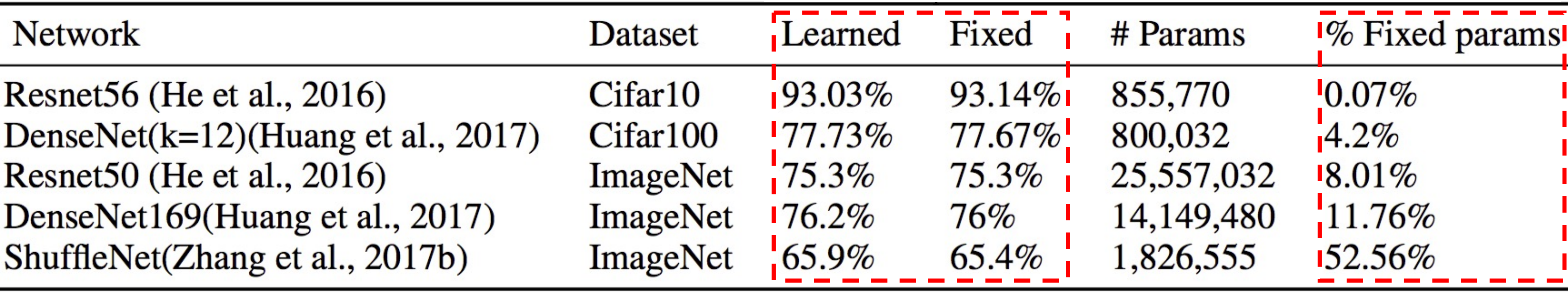
Random vector for each class

- Neural netowrks can learn useful representation without modifying the weights of the output layer

- Hadamard matrix as a weight matrix

$$H_1 = [1],$$
$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$
$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

$$\begin{bmatrix} H & H \\ H & -H \end{bmatrix}$$



Network	Dataset	Learned	Fixed	# Params	% Fixed params
Resnet56 (He et al., 2016)	Cifar10	93.03%	93.14%	855,770	0.07%
DenseNet(k=12)(Huang et al., 2017)	Cifar100	77.73%	77.67%	800,032	4.2%
Resnet50 (He et al., 2016)	ImageNet	75.3%	75.3%	25,557,032	8.01%
DenseNet169(Huang et al., 2017)	ImageNet	76.2%	76%	14,149,480	11.76%
ShuffleNet(Zhang et al., 2017b)	ImageNet	65.9%	65.4%	1,826,555	52.56%

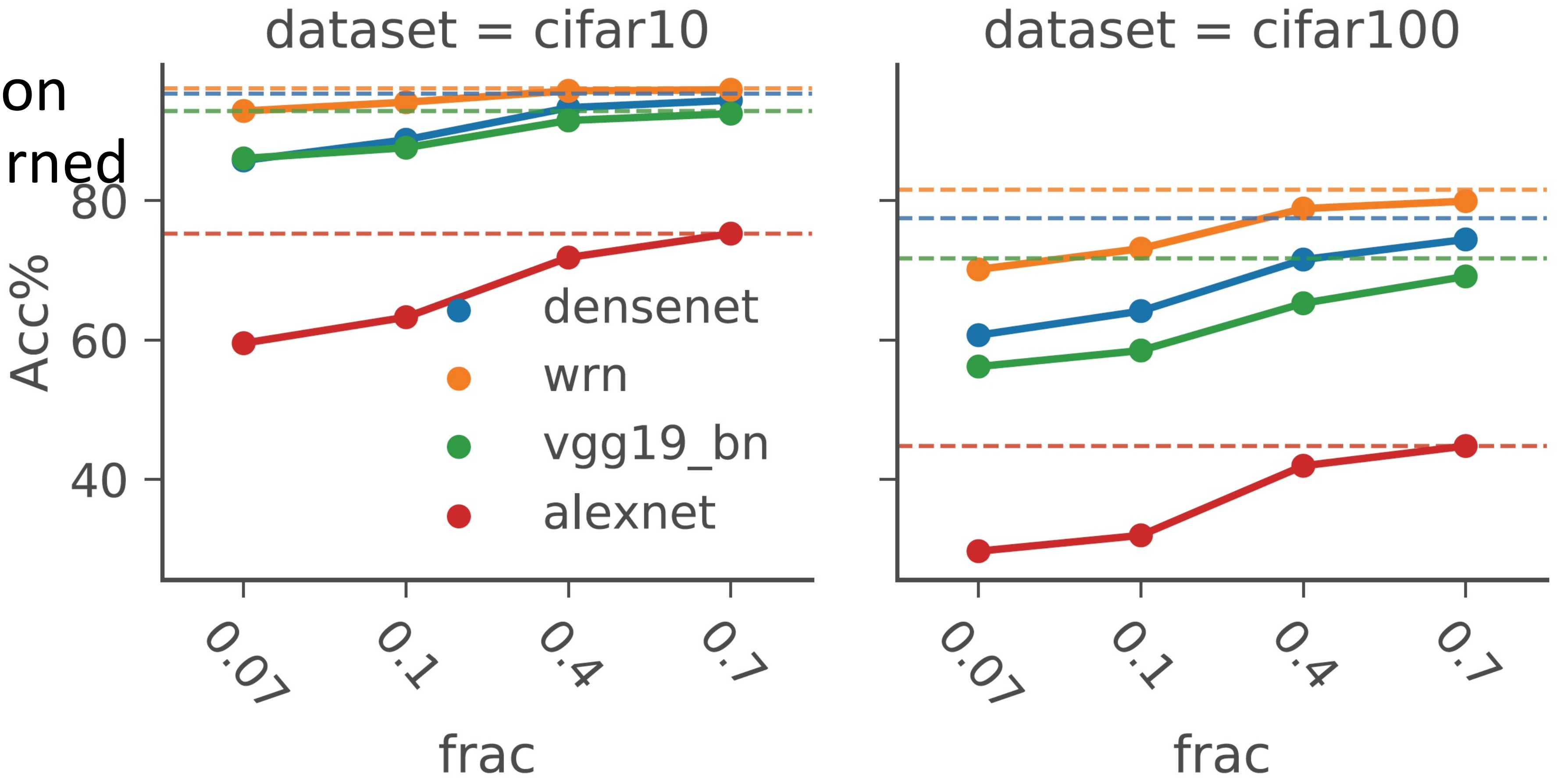
E. Hoffer, I. Hubara, and D. Soudry, “Fix Your Classifier: the Marginal Value of Training the Last Weight Layer,” in International Conference on Learning Representations (ICLR), 2018.

Learning next to nothing

- Fix fractions of convolutional layers of deep CNNs

- Allow only a small portion of the weights to be learned

- Performance can be on a par with learning all of them



A. Rosenfeld and J. K. Tsotsos, “Intriguing Properties of Randomly Weighted Networks: Generalizing while Learning Next to Nothing,” in IEEE Conference on Computer and Robot Vision (CRV), 2019, pp. 9–16.

Use of HD Computing/VSA primitives in neural networks design

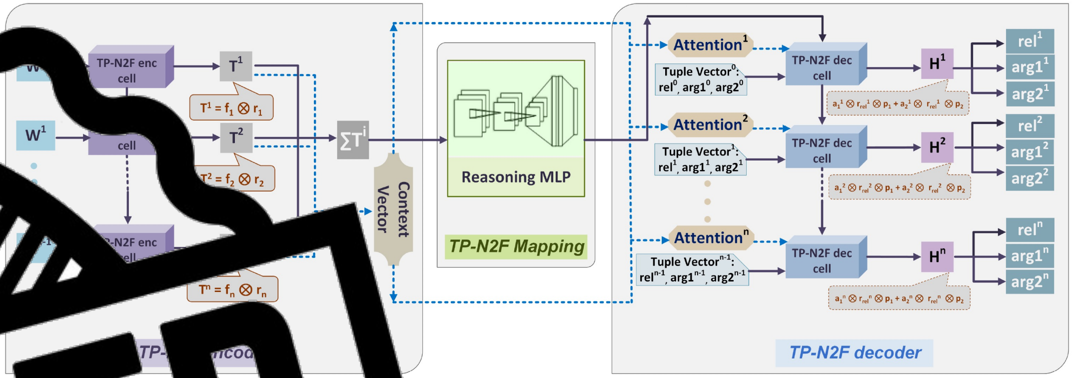
Composite data structures: Natural-to Formal-Language Generation

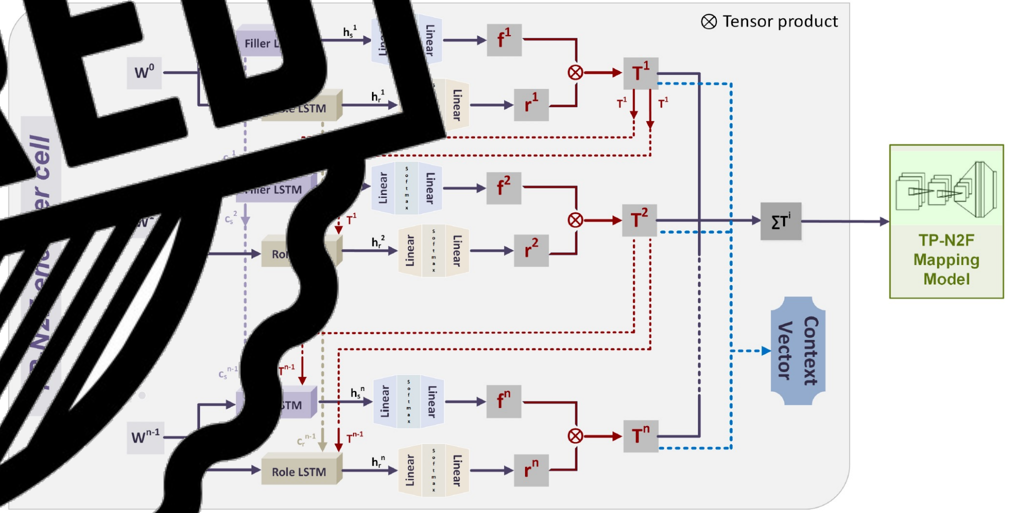
- Tensor Product Representations-based binding
  - Claim: the use of TPRs allows explicit [illegible] of relational structure to support reasoning

- Represent input data as superposition of tensors representing role-filler pairs
  - Both codebooks are learned

- Represent output data as [illegible]

- The model is pretty complicated
  - But claimed to obtain good results on 2 datasets



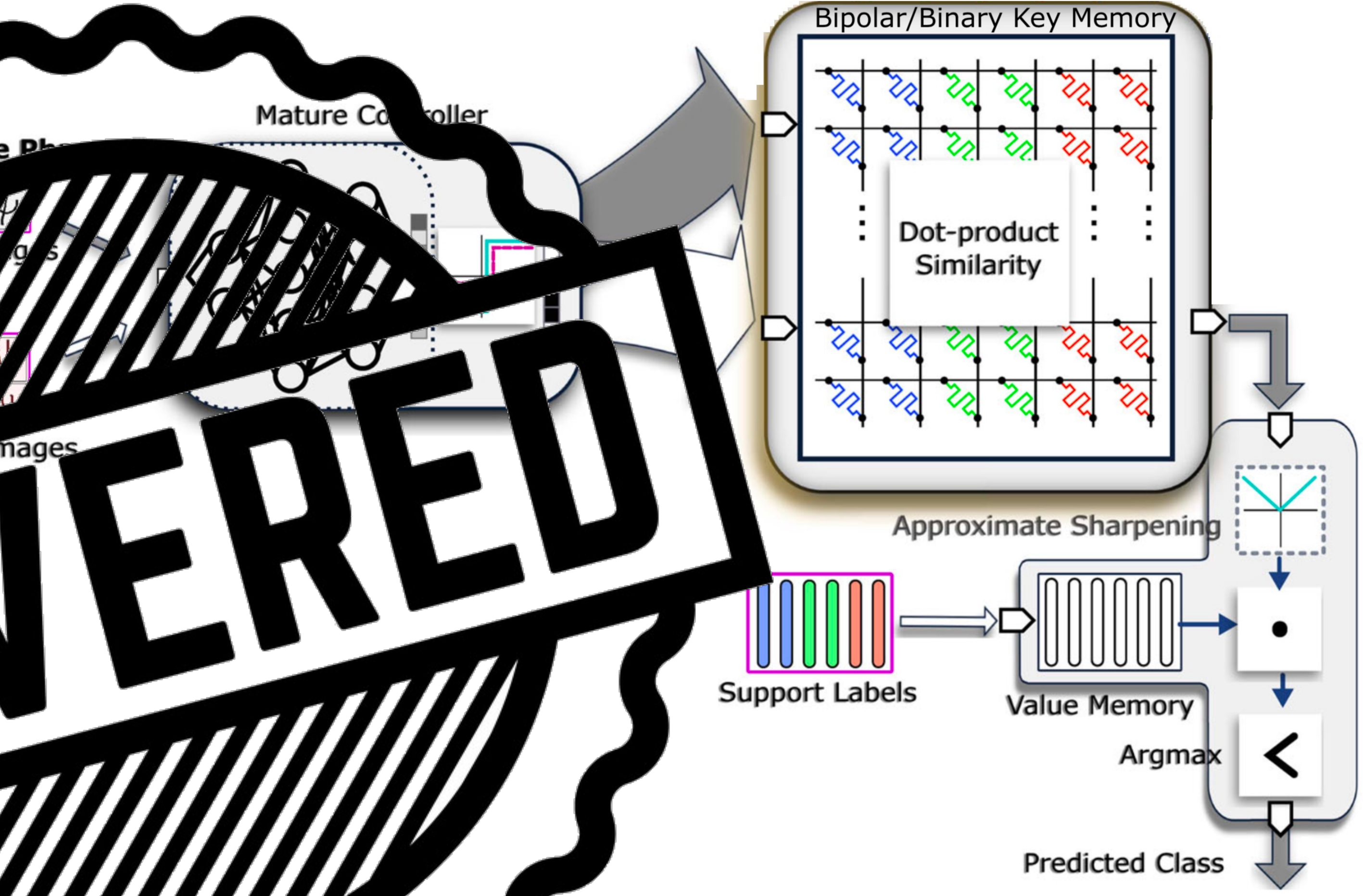


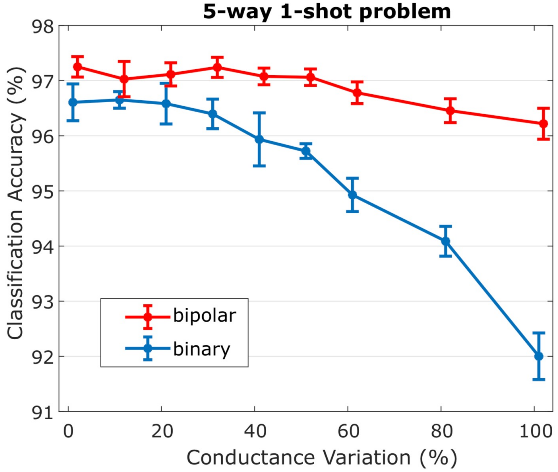
K. Chen, Q. Huang, H. Palangi, P. Smolensky, K. D. Forbus, J. Gao, “Mapping Natural-language Problems to Formal-language Solutions using Structured Neural Representations”, International Conference on Machine Learning (ICML), 2020.

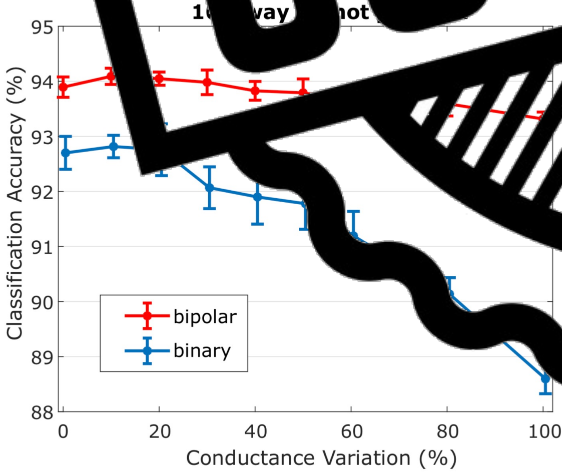
Binary HD vectors from images

- Memory-augmented networks
    - Controller – Convolutional network
    - Key-value memory
        - Content addressable memory
        - Explicit memory
    - Few-shot learning

- Evaluation
    - Omniglot dataset
    - Phase-change memory devices





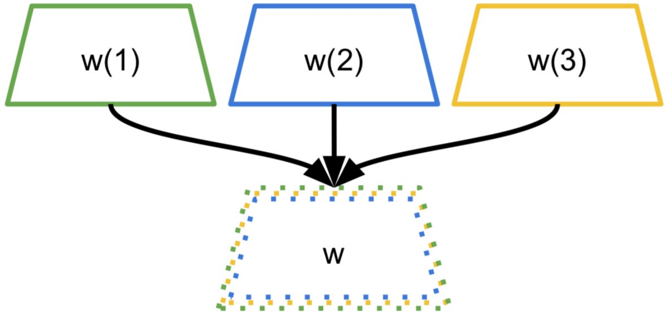


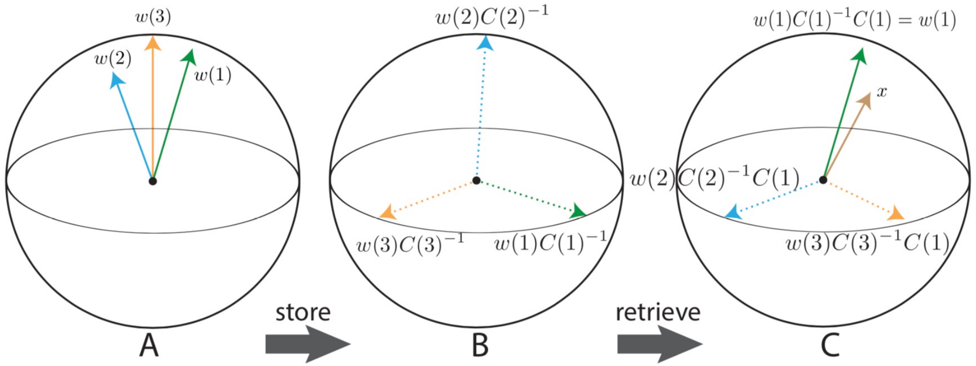
G. Karuna[illegible]tne, M. [illegible]chmuck, M. Le Gallo, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi, "R[illegible] high-[illegible]ensional Memory-augmented Neural Networks", Nature Communications, 2021.

Superposition of many neural networks into one

Problem(s):

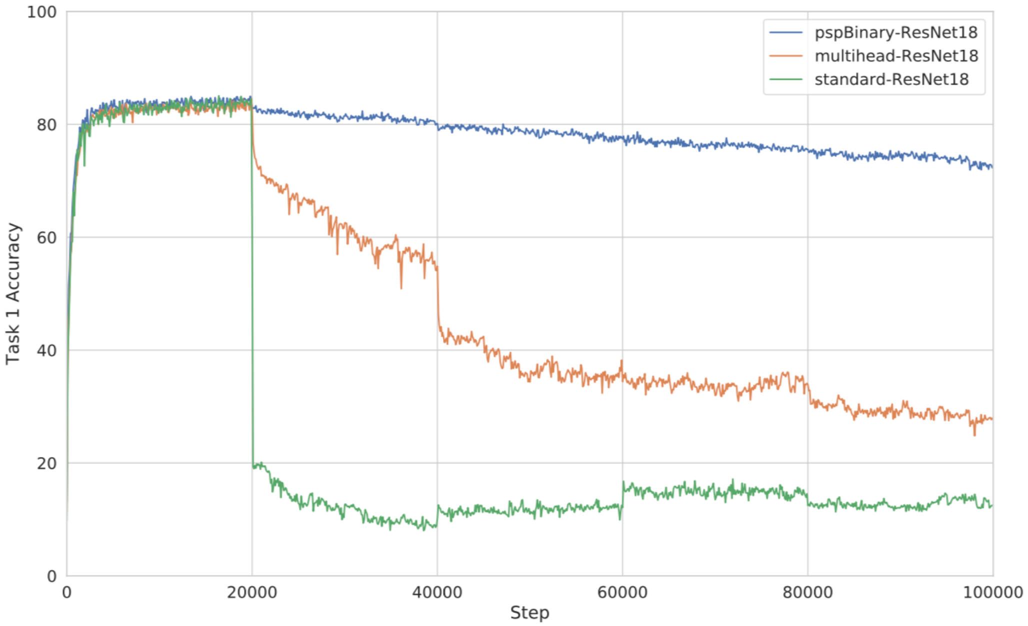
- Online learning of multiple tasks
- Catastrophic forgetting
- Memory constrained environments





Approach:

- Use random HD vectors as context for tasks
- Use binding operation to associate with task's set of parameters W_k
- Store models in superposition

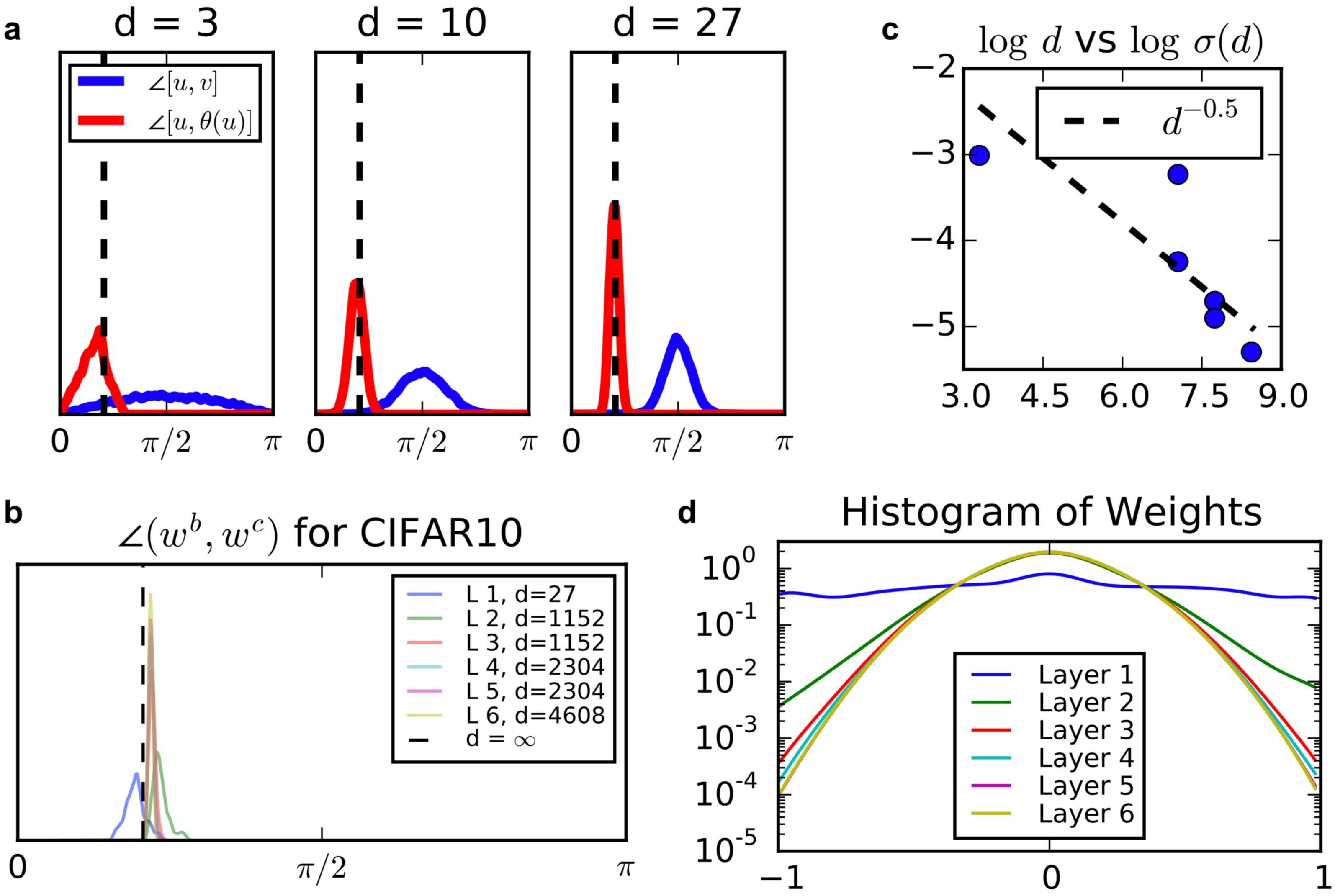


B. Cheung, A. Terekhov, Y. Chen, P. Agrawal, B. A. Olshausen, "Superposition of Many Models into One," Advances in Neural Information Processing Systems (NeurIPS), 2019.

Explaining neural networks with HD Computing/VSA

Qualitative: Binarized CNNs

- Why one can effectively capture the features in data with binary weights and activations?
  - Continuous vectors are well-approximated by binary vectors
- HD geometry
  - Binarization approximately preserves the direction of high-dimensional vectors
  - Weight-activation dot products are approximately proportional



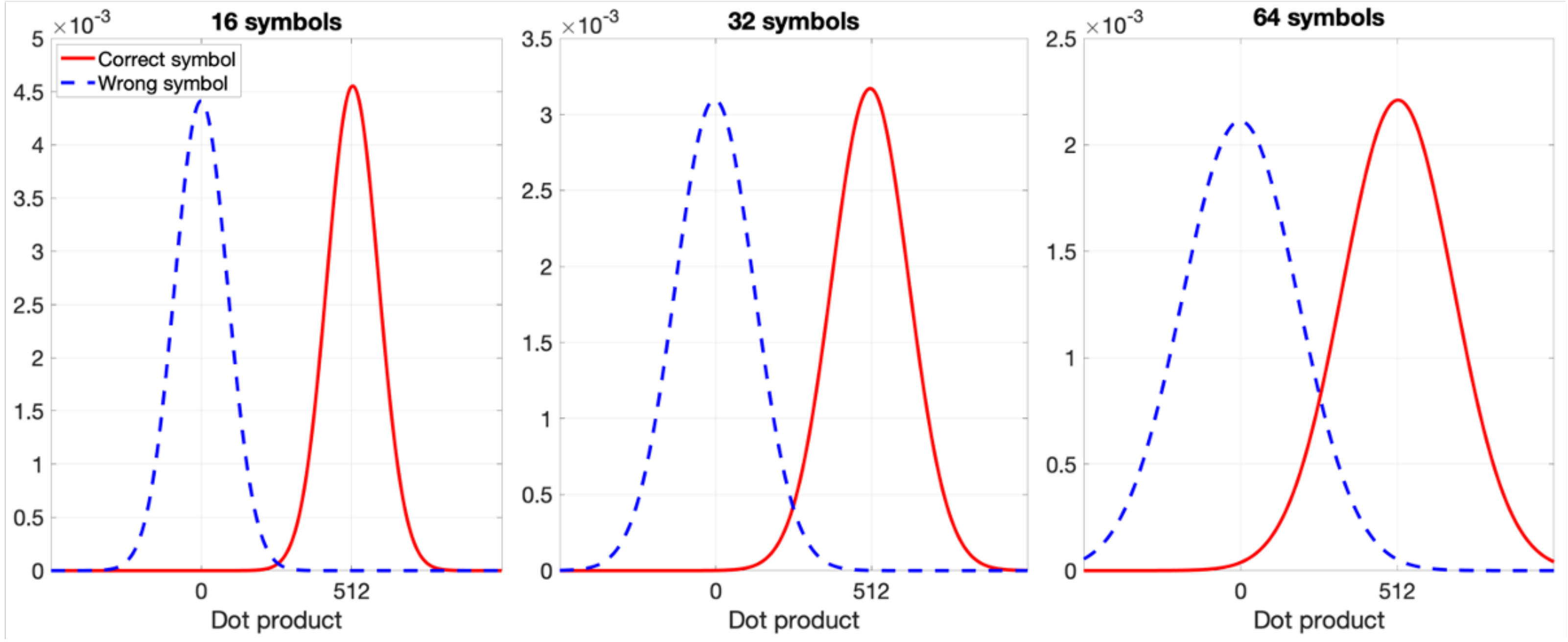
I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio, “Binarized Neural Networks,” Advances in Neural Information Processing Systems (NIPS), 2016.

A. G. Anderson, C. P. Berg, “The High-Dimensional Geometry of Binary Neural Networks,” International Conference on Learning Representations (ICLR), 2018.

Quantitative: Capacity theory

- What affects the accuracy?
  - *D*, *N*, *L*

- Can we predict retrieval accuracy?



N=512; *D*=32; *L* varies

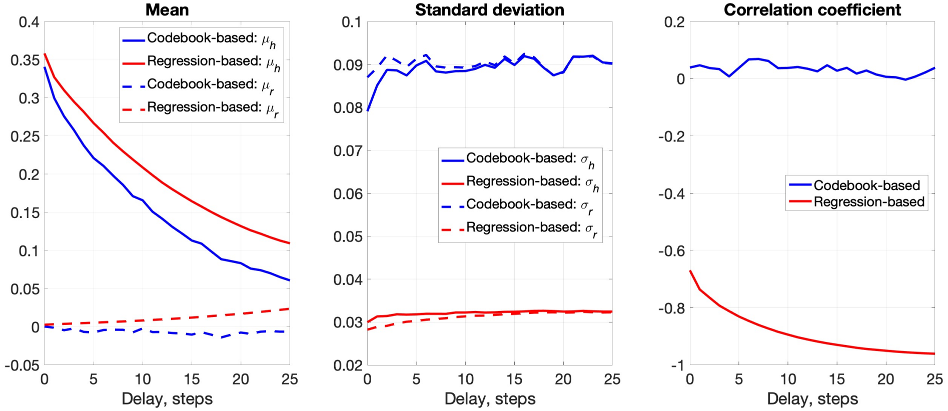
$$\int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}\sigma_h} e^{-\frac{(x-\mu_h)^2}{2\sigma_h^2}} \left(\Phi(x,\mu_r,\sigma_r)\right)^{D-1}$$ **T1**

- Assumptions:
  - The distributions are normal
  - The distributions for "distractor" symbols are all the same
  - The distributions for different symbols are independent

2018 NeCo paper presented **T1** describing the capacity of VSAs and some ESNs cases

E. P. Frady, D. Kleyko, F. T. Sommer, "A Theory of Sequence Indexing and Working Memory in Recurrent Neural Networks," Neural Computation, 2018.

Extensions of **T1**



- Need to extend the theory:

- **T1**: $\int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}\sigma_h} e^{-\frac{(x-\mu_h)^2}{2\sigma_h^2}} (\Phi(x, \mu_r, \sigma_r))^{D-1}$

- **T2**:
  - The distributions are normal
  - ~~The distributions for “distractor” symbols are all the same~~
  - The distributions for different symbols are independent

$$\int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}\boldsymbol{\sigma}_i} e^{-\frac{(x-\boldsymbol{\mu}_i)^2}{2\boldsymbol{\sigma}_i^2}} \prod_{j=1, j\neq i}^{D-1} \Phi(x, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)$$

- **T3**:
  - The distributions are normal
  - ~~The distributions for “distractor” symbols are all the same~~
  - ~~The distributions for different symbols are independent~~

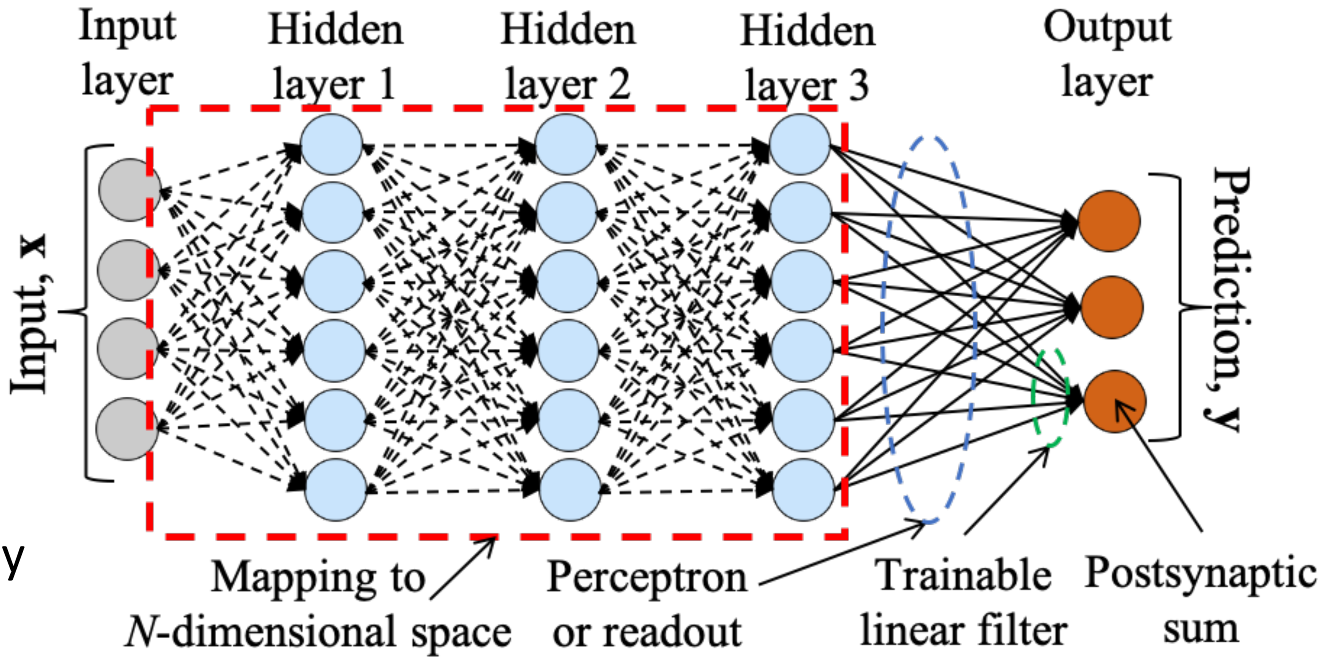
$$\int_{-\infty}^{\infty} d\mathbf{x}_1 \int_{-\infty}^{\mathbf{x}_1} d\mathbf{x}_2 \cdots \int_{-\infty}^{\mathbf{x}_1} d\mathbf{x}_D \, \mathcal{N}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

D. Kleyko, A. Rosato, E. P. Frady, M. Panella, F. T. Sommer, “Perceptron Theory for Predicting the Accuracy of Neural Networks,” arXiv, 2020.

From memory buffer to classification with neural networks

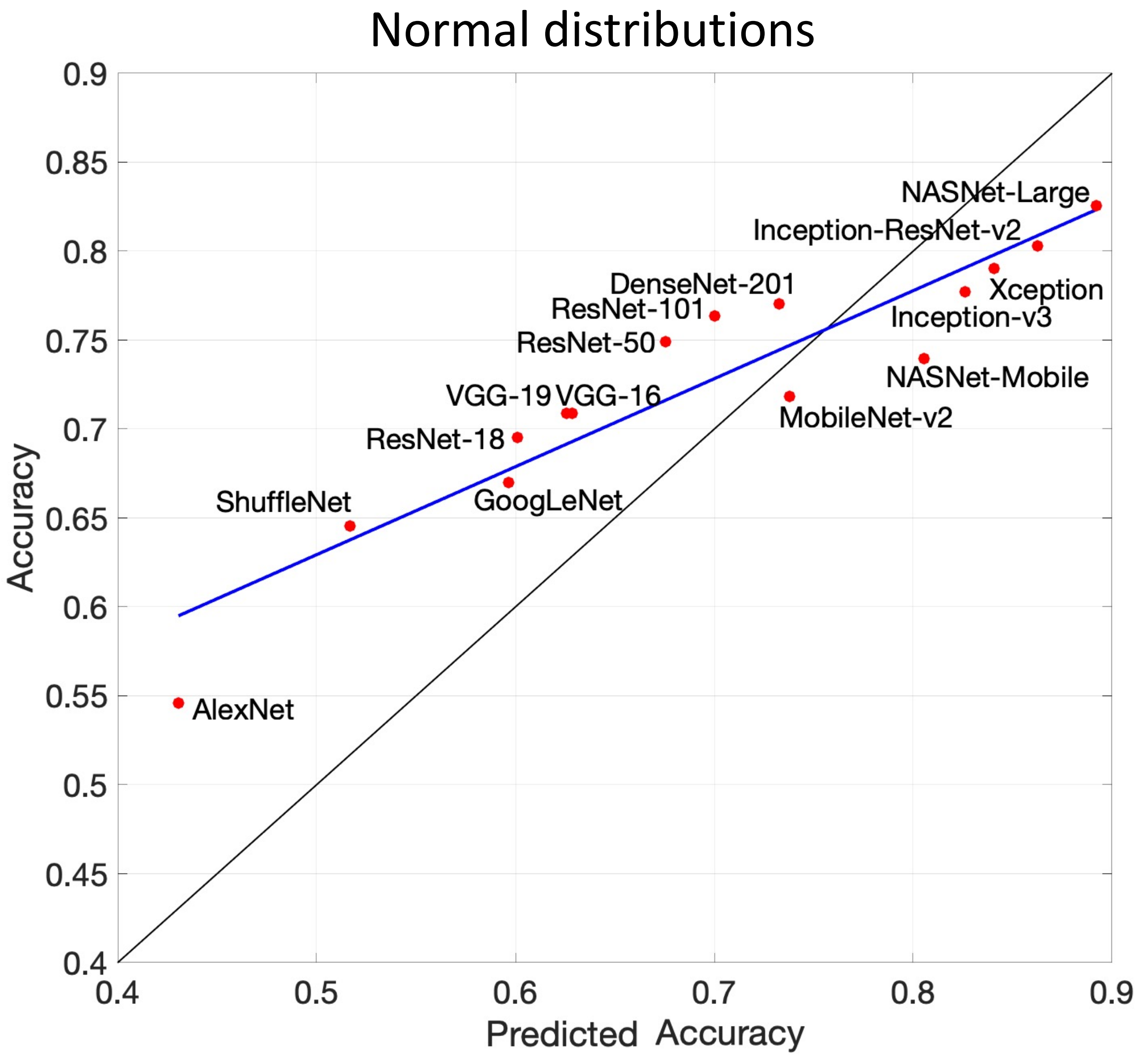
- Desire to get under the hood of neural networks

- Dissect the holistic functionality of neural network into two parts:
  - Multi-layer encoding stage –> corresponds to **x** in the memory buffer task
  - Single-layer classification by perceptron -> similar to the regression-based perceptron in the memory buffer task

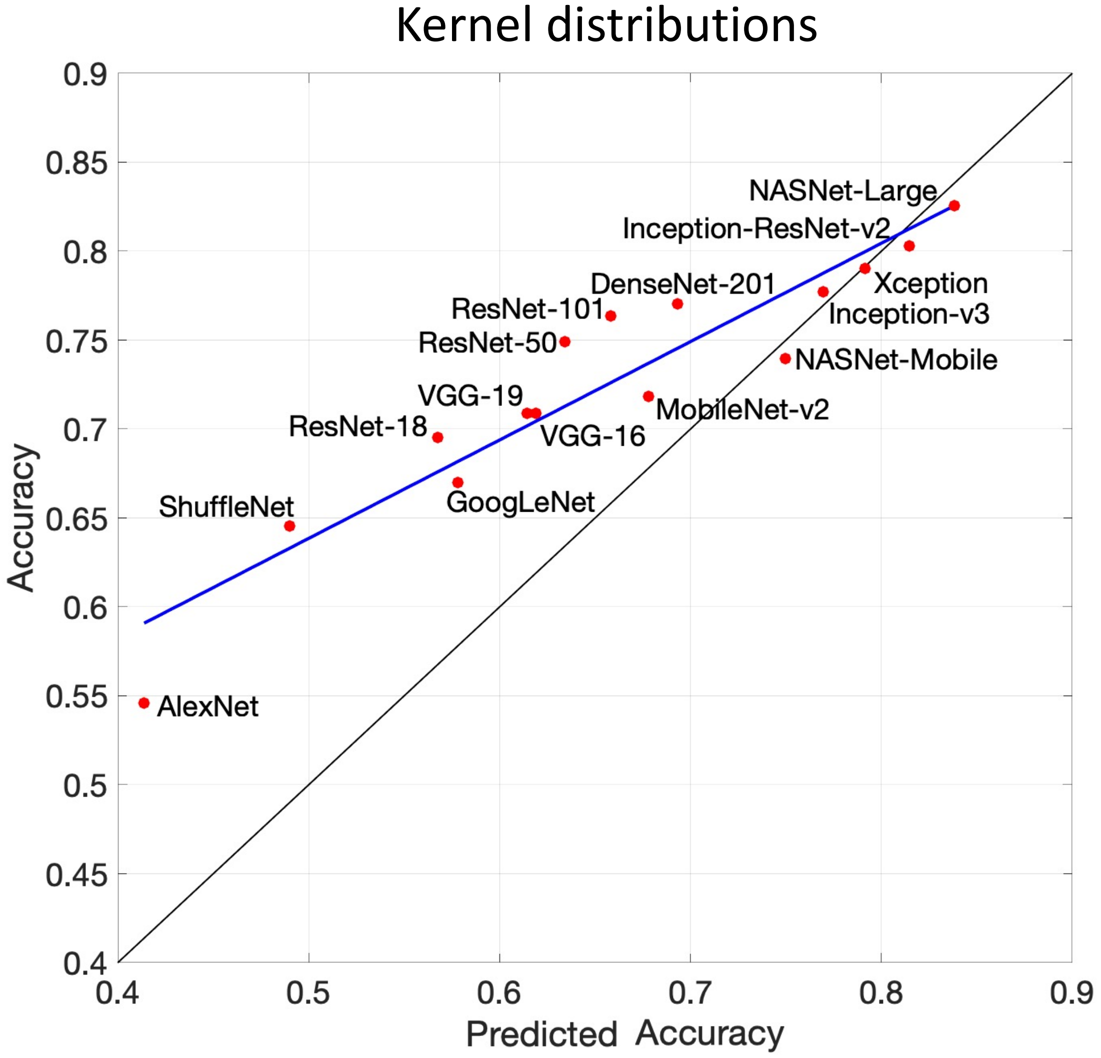


V. Papyan, X. Y. Han, D. L. Donoho, “Prevalence of Neural Collapse During the Terminal Phase of Deep Learning Training,” Proceedings of the National Academy of Sciences, 2020.

Pretrained deep networks on ImageNet with **T2**



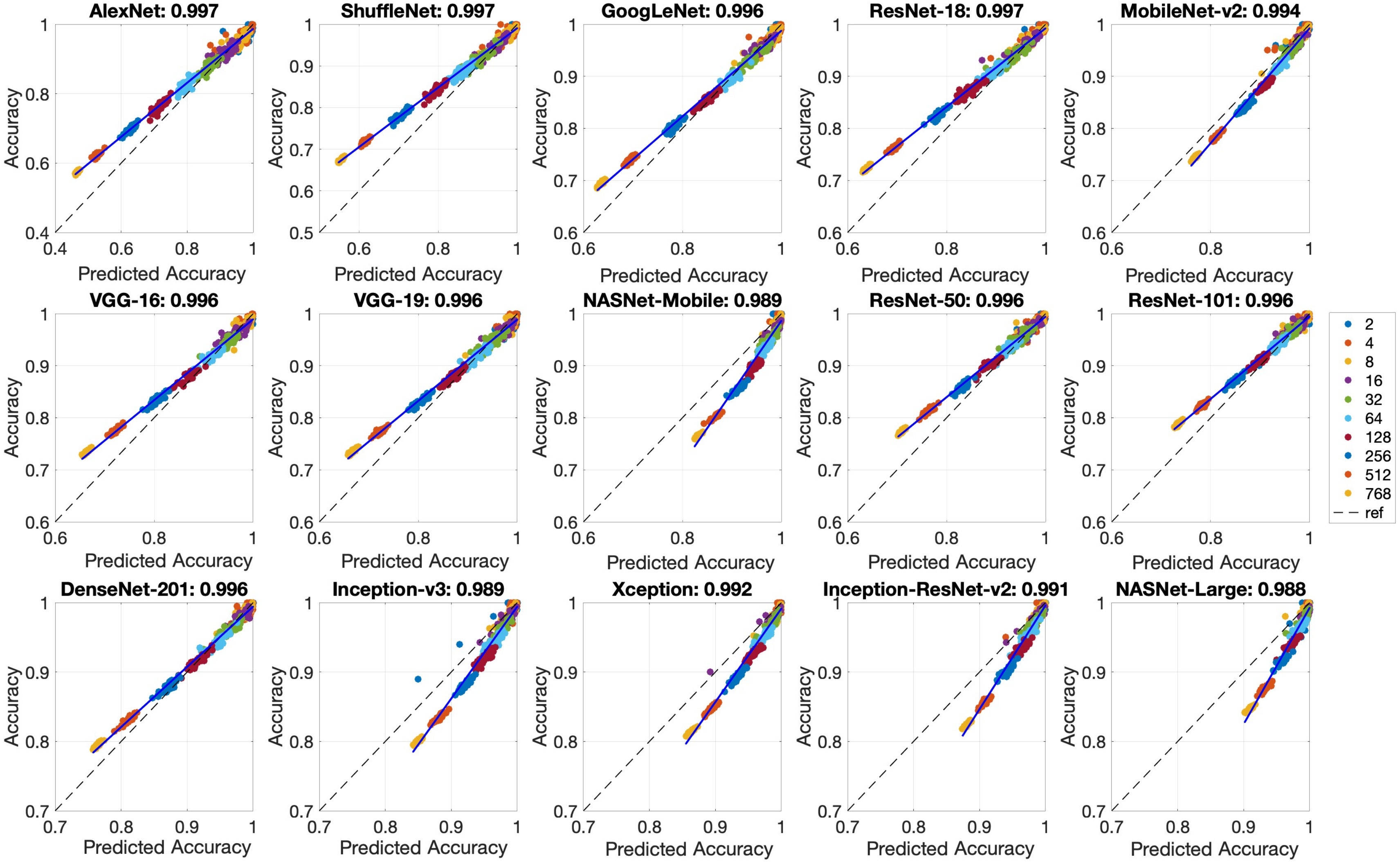
Correlation coefficient: 0.933



Correlation coefficient: 0.940

ImageNet subproblems on individual models

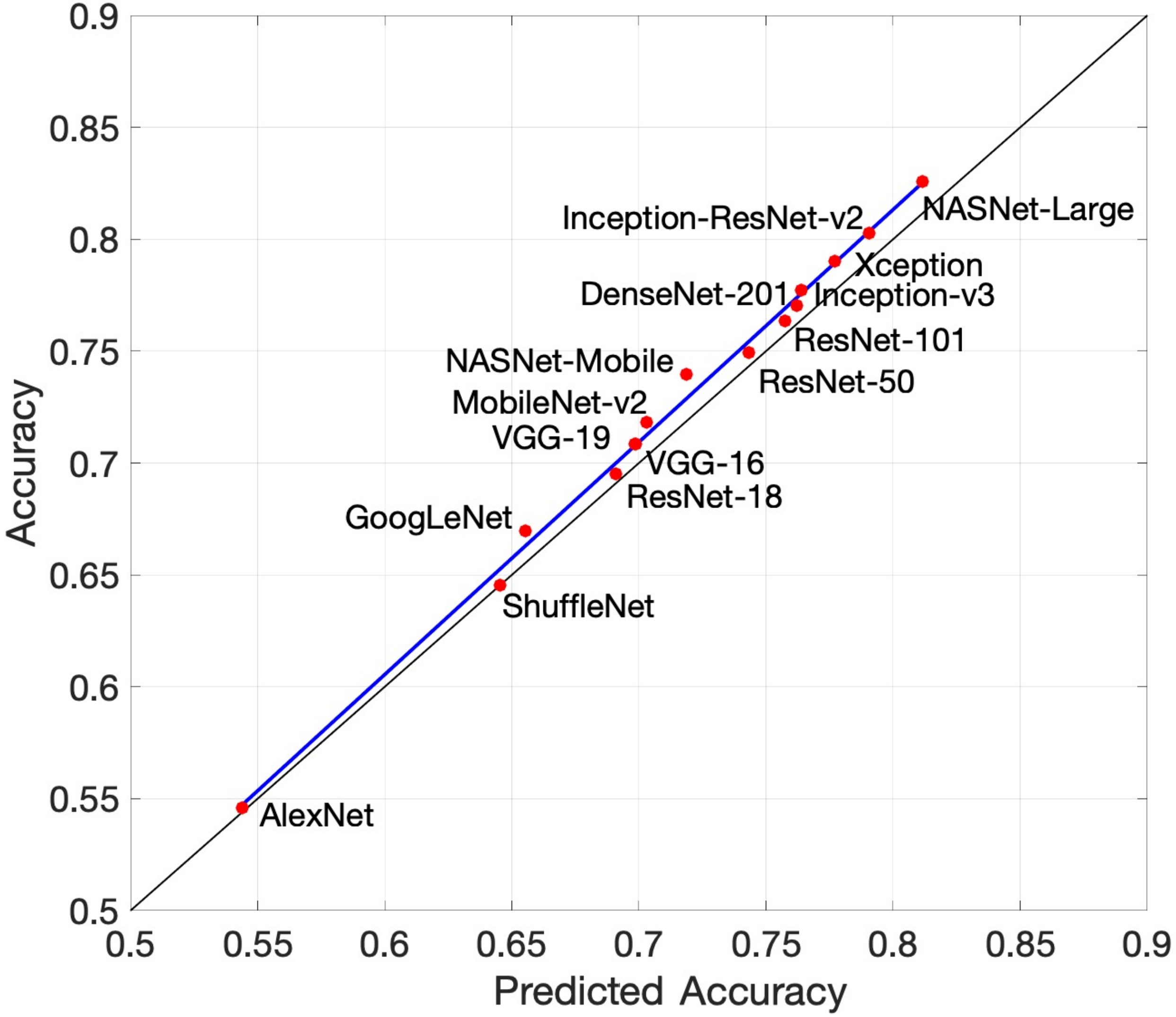
- We want to remove the bias
- Sub-problems of different size by randomly sampling from the ImageNet
- Prediction for sub-problems
- Bias for each model



ImageNet adjusted predicted accuracy

- Compensation for each network is based on the sub-problems on individual networks

- The compensations have almost removed bias and unsystematic deviations between the accuracies

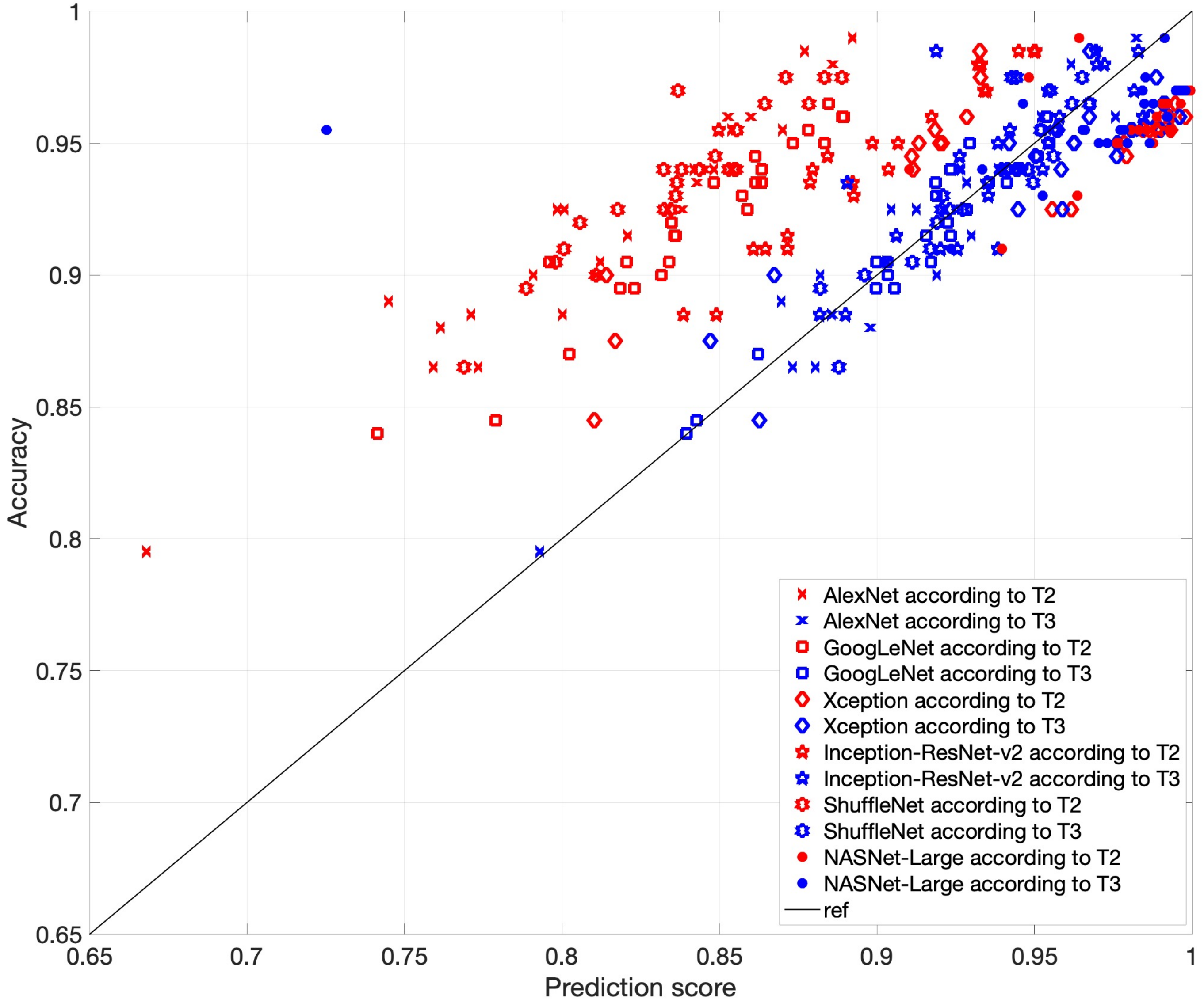


Correlation coefficient: 0.998

ImageNet subproblems with **T3**

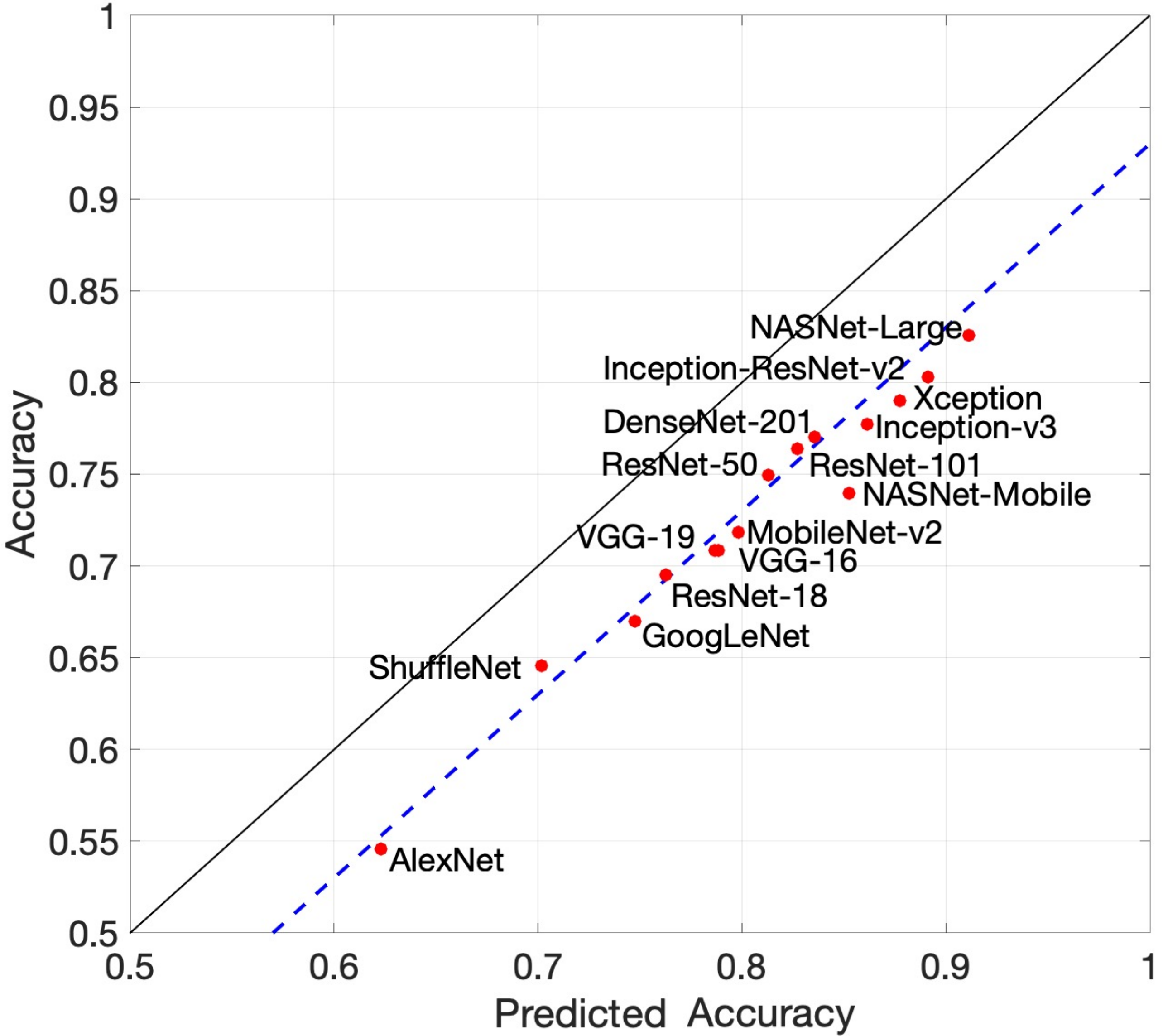
- Compensation needs to observe accuracies of smaller sub-problems

- Solution: independence assumption
  - Numerical integration is challenging
  - Sub-problems of size 4



Predictions via Monte Carlo sampling

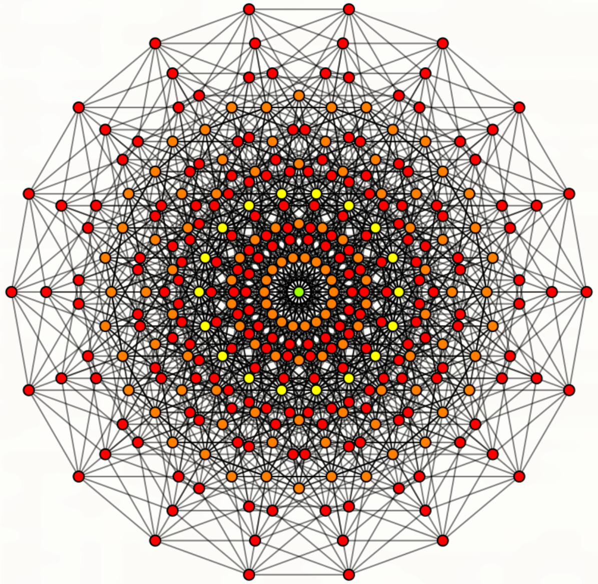
- It is hard to calculate integral in **T3**
- We try to estimate it by MC sampling
  - Correlation coefficient: 0.98
- Constant offset
  - No normalization constant



Takeaways

The only rule is, there are no rules:

- HD vectors as input to neural networks
- Neural networks for producing HD vectors
- HD Computing/VSA connections to randomized neural networks
- Use of HD Computing/VSA primitives in neural networks design
- Explaining neural networks with HD Computing/VSA



Relations to neural networks

Denis Kleyko