Summary of Paper: Binary embeddings with structured hashed projections

Presenter: Joseph Tobin

Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

Binary embeddings with structured hashed projections (2016)

- Authors: Anna Choromanska, Krzysztof Choromanski, Mariusz Bojarski, Tony Jebara, Sanjiv Kumar, Yann LeCun
- We consider the hashing mechanism for constructing binary embeddings, that involves pseudo-random projections followed by nonlinear (sign function) mappings. The pseudo-random projection is described by a matrix, where not all entries are independent random variables but instead a fixed "budget of randomness" is distributed across the matrix. Such matrices can be efficiently stored in sub-quadratic or even linear space, provide reduction in randomness usage (i.e. number of required random values), and very often lead to computational speed ups. We prove several theoretical results showing that projections via various structured matrices followed by nonlinear mappings accurately preserve the angular distance between input high-dimensional vectors. To the best of our knowledge, these results are the first that give theoretical ground for the use of general structured matrices in the nonlinear setting. In particular, they generalize previous extensions of the Johnson-Lindenstrauss lemma and prove the plausibility of the approach that was so far only heuristically confirmed for some special structured matrices. Consequently, we show that many structured matrices can be used as an efficient information compression mechanism. Our findings build a better understanding of certain deep architectures, which contain randomly weighted and untrained layers, and yet achieve high performance on different learning tasks. We empirically verify our theoretical findings and show the dependence of learning via structured hashed projections on the performance of neural network as well as nearest neighbor classifier.

Motivation: Dimensionality Reduction

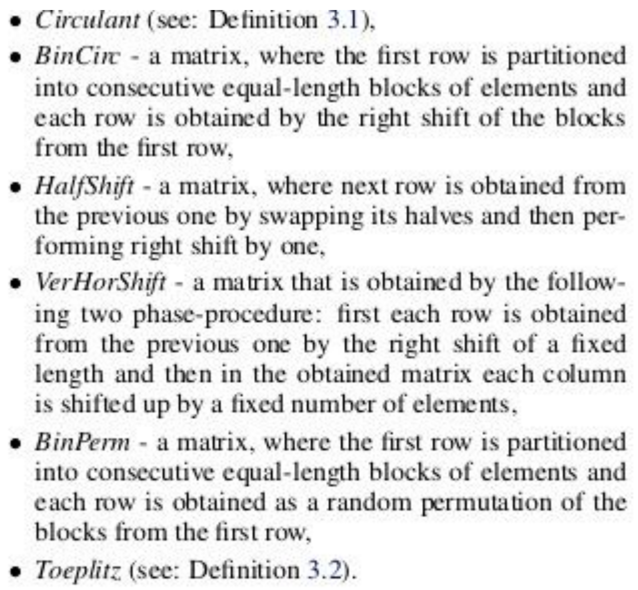
- Hughes phenomenon
  - As dimensions increase, prediction accuracy decreases
- Feature extraction and dimension reduction
  - Reduce dimensionality of data while keeping most important features
  - Linear solutions: PCA
  - Non linear solutions: low-dimensional embeddings (binary embeddings)
- Especially important for high-dimensional data and deep learning

Introductory Information

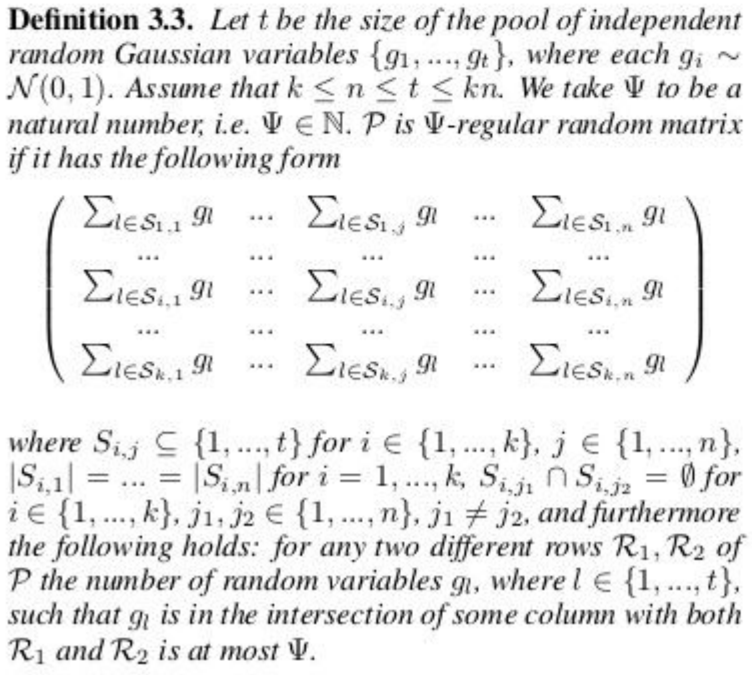
- Using linear projections with completely random Gaussian weights, instead of learned ones, studied in Giryes et al., 2015
- This paper uniquely considers structured matrices whih can be stored much more efficiently
  - Sub-quadratic or even linear space
- Terms
  - Hamming distance
  - Angular distnace

Structured Matrices

- Matrix-vector products fast



- *Circulant* (see: Definition 3.1),
- *BinCirc* - a matrix, where the first row is partitioned into consecutive equal-length blocks of elements and each row is obtained by the right shift of the blocks from the first row,
- *HalfShift* - a matrix, where next row is obtained from the previous one by swapping its halves and then performing right shift by one,
- *VerHorShift* - a matrix that is obtained by the following two phase-procedure: first each row is obtained from the previous one by the right shift of a fixed length and then in the obtained matrix each column is shifted up by a fixed number of elements,
- *BinPerm* - a matrix, where the first row is partitioned into consecutive equal-length blocks of elements and each row is obtained as a random permutation of the blocks from the first row,
- *Toeplitz* (see: Definition 3.2).



Definition 3.3. *Let t be the size of the pool of independent random Gaussian variables $\{g_1, ..., g_t\}$, where each $g_i \sim \mathcal{N}(0,1)$. Assume that $k \leq n \leq t \leq kn$. We take Ψ to be a natural number, i.e. $\Psi \in \mathbb{N}$. $\mathcal{P}$ is Ψ-regular random matrix if it has the following form*

$$\begin{pmatrix} \sum_{l \in \mathcal{S}_{1,1}} g_l & \dots & \sum_{l \in \mathcal{S}_{1,j}} g_l & \dots & \sum_{l \in \mathcal{S}_{1,n}} g_l \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{l \in \mathcal{S}_{i,1}} g_l & \dots & \sum_{l \in \mathcal{S}_{i,j}} g_l & \dots & \sum_{l \in \mathcal{S}_{i,n}} g_l \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{l \in \mathcal{S}_{k,1}} g_l & \dots & \sum_{l \in \mathcal{S}_{k,j}} g_l & \dots & \sum_{l \in \mathcal{S}_{k,n}} g_l \end{pmatrix}$$

where $S_{i,j} \subseteq \{1, ..., t\}$ for $i \in \{1, ..., k\}$, $j \in \{1, ..., n\}$, $|S_{i,1}| = ... = |S_{i,n}|$ for $i = 1, ..., k$, $S_{i,j_1} \cap S_{i,j_2} = \emptyset$ for $i \in \{1, ..., k\}$, $j_1, j_2 \in \{1, ..., n\}$, $j_1 \neq j_2$, and furthermore the following holds: for any two different rows $\mathcal{R}_1, \mathcal{R}_2$ of $\mathcal{P}$ the number of random variables g_l, where $l \in \{1, ..., t\}$, such that g_l is in the intersection of some column with both $\mathcal{R}_1$ and $\mathcal{R}_2$ is at most Ψ.

Hashing Methdods

- Preprocessing step:balance vectors by distriubting their mass uniformly over all the coordinates without changing L2 distances
- Hashing step: pseudo-random project followed by nonlinear mapping
- Short and full hash

$$\underbrace{x \xrightarrow{\mathcal{R}} x_{\mathcal{R}} \xrightarrow{\mathcal{H}} x_{\mathcal{H}} \xrightarrow{\mathcal{D}} x_{\mathcal{D}}}_{\text{preprocessing}} \underbrace{\xrightarrow{\mathcal{P}_{\Psi}} x_{\mathcal{P}_{\Psi}} \xrightarrow{\phi} h(x)}_{\text{hashing}} \in \mathbb{R}^k. \quad (1)$$

$$\underbrace{x \xrightarrow{\mathcal{D}} x_{\mathcal{D}}}_{\text{preprocessing}} \underbrace{\xrightarrow{\mathcal{P}_{\Psi}} x_{\mathcal{P}_{\Psi}} \xrightarrow{\phi} h(x)}_{\text{hashing}} \in \mathbb{R}^k.$$

Unbiasedness of the estimator

- Normalized approximate angle between points p and r

$$\tilde{\theta}^n_{p,r} = \frac{1}{2k}\|h(p) - h(r)\|_1$$

Lemma 4.1. *Let $\mathcal{M}$ be a Ψ-regular hashing model (either extended or a short one) and $\|p\|_2 = \|r\|_2 = 1$. Then $\tilde{\theta}^n_{p,r}$ is an unbiased estimator of $\frac{\theta_{p,r}}{\pi}$, i.e. $E(\tilde{\theta}^n_{p,r}) = \frac{\theta_{p,r}}{\pi}$.*

Unbiasedness of Estimator

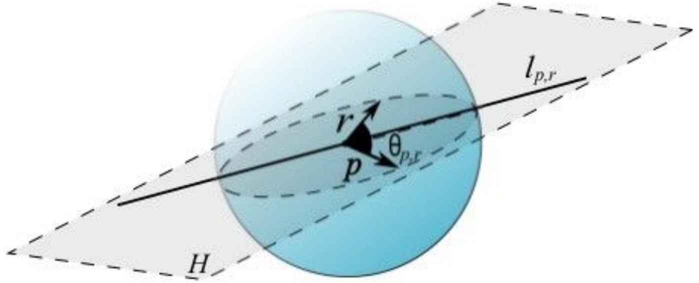


Figure 1: Two vectors: p, r spanning two-dimensional hyperplane H and with the angular distance $\theta_{p,r}$ between them. We have: $l_{p,r} = g^i_{\mathcal{D},H,\perp}$. Line $l_{p,r}$ is dividing $\theta_{p,r}$ and thus g^i contributes to $\|h(p) - h(r)\|_1$.

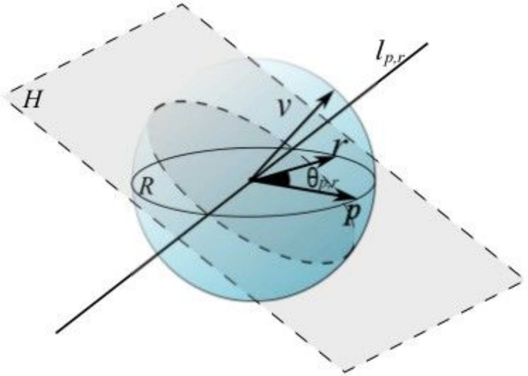
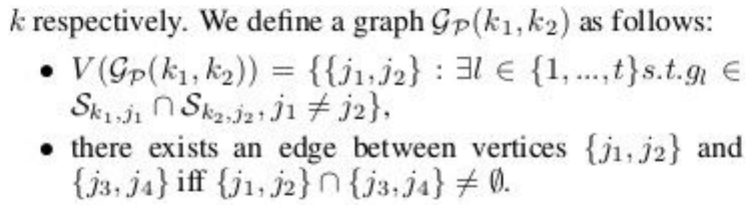


Figure 2: Similar setting to the one presented on Figure 1. Vector v represents L_2-normalized version of g^i and is perpendicular to the two-dimensional plane R. The intersection $R \cap H$ of that plane with the 2-dimensional plane H spanned by p, r is a line $l_{p,r}$ that this time is outside $\mathcal{U}_{p,r}$. Thus g^i does not contribute to $\|h(p) - h(r)\|_1$.

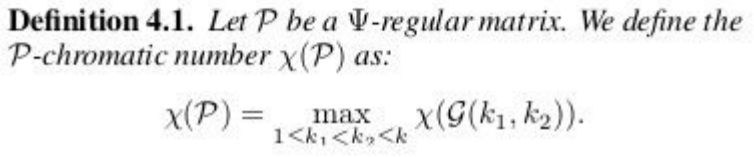
P-chromatic number

- Encode dependencies between entries of structured matrix in compact form with quantitative ways to measure deps.



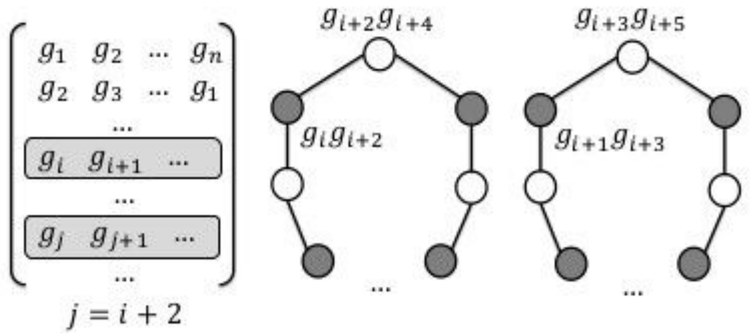
k respectively. We define a graph $\mathcal{G}_{\mathcal{P}}(k_1, k_2)$ as follows:

- $V(\mathcal{G}_{\mathcal{P}}(k_1, k_2)) = \{\{j_1, j_2\} : \exists l \in \{1, ..., t\} s.t. g_l \in \mathcal{S}_{k_1, j_1} \cap \mathcal{S}_{k_2, j_2}, j_1 \neq j_2\}$,
- there exists an edge between vertices $\{j_1, j_2\}$ and $\{j_3, j_4\}$ iff $\{j_1, j_2\} \cap \{j_3, j_4\} \neq \emptyset$.



Definition 4.1. *Let $\mathcal{P}$ be a Ψ-regular matrix. We define the $\mathcal{P}$-chromatic number $\chi(\mathcal{P})$ as:*

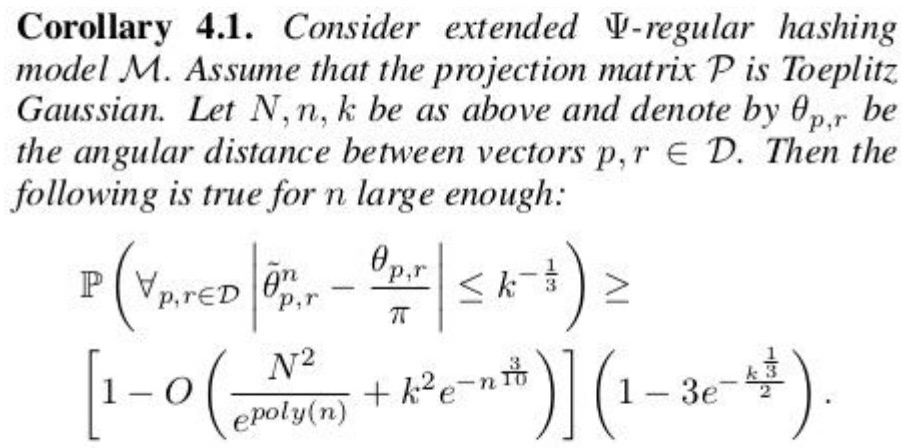
$$\chi(\mathcal{P}) = \max_{1<k_1<k_2<k} \chi(\mathcal{G}(k_1, k_2)).$$


$g_1 \; g_2 \; \cdots \; g_n$
$g_2 \; g_3 \; \cdots \; g_1$
$g_i \; g_{i+1} \; \cdots$
$g_j \; g_{j+1} \; \cdots$
$j = i + 2$
$g_{i+2}g_{i+4}$
$g_i g_{i+2}$
$g_{i+3}g_{i+5}$
$g_{i+1}g_{i+3}$

Concentration Inequalities for structured hasing with sign function

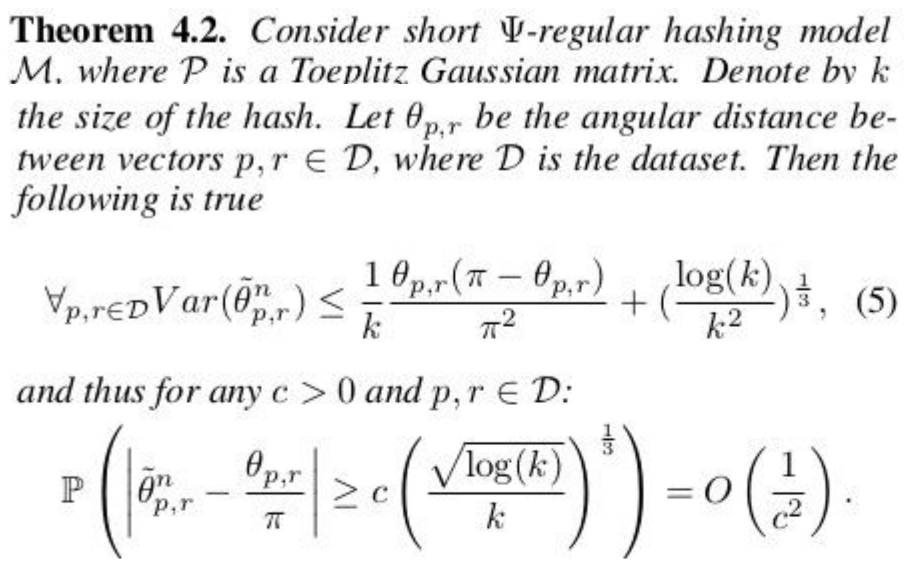
- Take two rows and project them onto linear space spanned by p and r
- Consider four coordinates obtained this way. They are Gaussian and are“almost independent” because p,r are “almost orthogonal”
- Want to prove that the directions considered are close to orthogonal with high probability
  - Use property of Gaussian vector that projects onto orthogonal directions are indepdendent
  - Use structured matrices, chromatic number, and diagonal matrices of hashin scheme

Main Results



Corollary 4.1. *Consider extended Ψ-regular hashing model $\mathcal{M}$. Assume that the projection matrix $\mathcal{P}$ is Toeplitz Gaussian. Let N, n, k be as above and denote by $\theta_{p,r}$ be the angular distance between vectors $p, r \in \mathcal{D}$. Then the following is true for n large enough:*

$$\mathbb{P}\left(\forall_{p,r\in\mathcal{D}} \left|\tilde{\theta}^{n}_{p,r} - \frac{\theta_{p,r}}{\pi}\right| \leq k^{-\frac{1}{3}}\right) \geq \left[1 - O\left(\frac{N^2}{e^{poly(n)}} + k^2 e^{-n^{\frac{3}{10}}}\right)\right]\left(1 - 3e^{-\frac{k^{\frac{1}{3}}}{2}}\right).$$



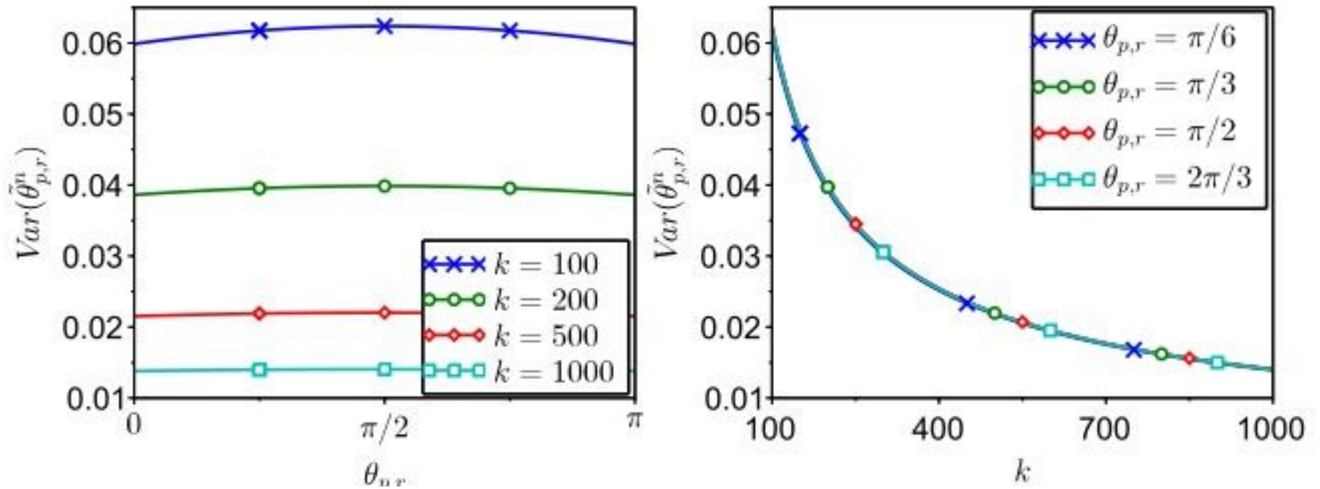
Theorem 4.2. *Consider short Ψ-regular hashing model $\mathcal{M}$, where $\mathcal{P}$ is a Toeplitz Gaussian matrix. Denote by k the size of the hash. Let $\theta_{p,r}$ be the angular distance between vectors $p, r \in \mathcal{D}$, where $\mathcal{D}$ is the dataset. Then the following is true*

$$\forall_{p,r\in\mathcal{D}} Var(\tilde{\theta}^{n}_{p,r}) \leq \frac{1}{k}\frac{\theta_{p,r}(\pi - \theta_{p,r})}{\pi^2} + \left(\frac{\log(k)}{k^2}\right)^{\frac{1}{3}}, \quad (5)$$

and thus for any $c > 0$ and $p, r \in \mathcal{D}$:

$$\mathbb{P}\left(\left|\tilde{\theta}^{n}_{p,r} - \frac{\theta_{p,r}}{\pi}\right| \geq c\left(\frac{\sqrt{\log(k)}}{k}\right)^{\frac{1}{3}}\right) = O\left(\frac{1}{c^2}\right).$$

Main Results



- Dependence of upper bound on the variance of the normalized approximate angle

Numerical Experiments

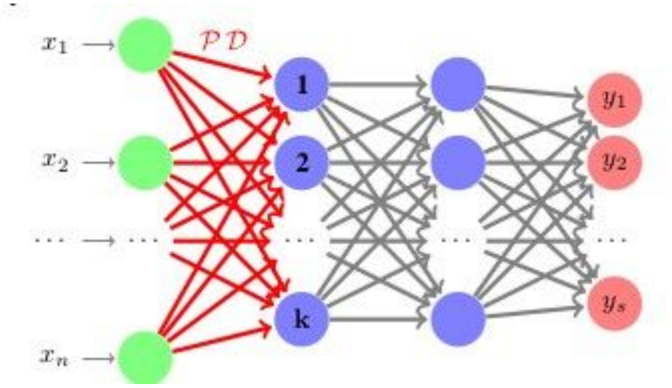
- Show the dependence of performance on the size of the hash and the reduction factor n/k for different structured matrices
- Show performance of different structured matrices when used with neural nets and 1-NN classifier
- Performed experiments on MNIST dataset
- Data was preprocessed with short hasing scheme before being given to input of network

Numerical Experiments

- First hidden layer has random untrained weights
  - Only second layer and output layer are trained using SGD
- Weights in first hidden layer correspond to entries in “preprocessed structured matrix”
- Baseline refers to network with one hidden layer with 100 hidden units and all parameters are trained
- Top performers: Random, BinPerm, HalfShift
- All matrices except BinPerm lead to biggest memory savings and require smallest “budget of randomness”

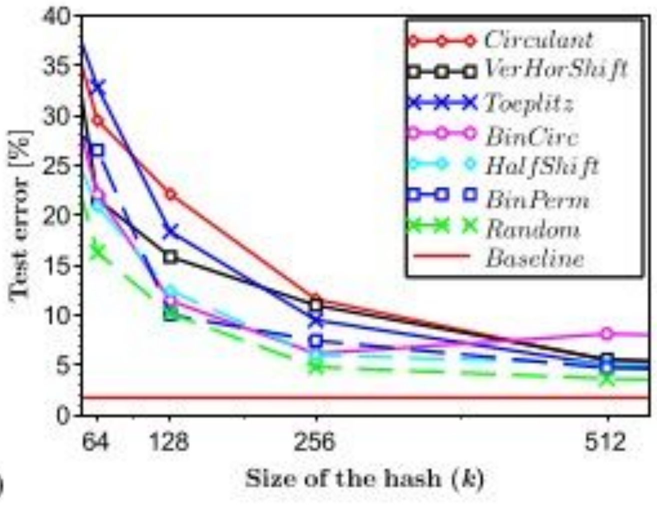
Numerical Experiments

- Fully connected network with randomized input
- D is random diagonal matrix with values from {-1, 1}
- P is a structured matrix (corresponds to red edges)



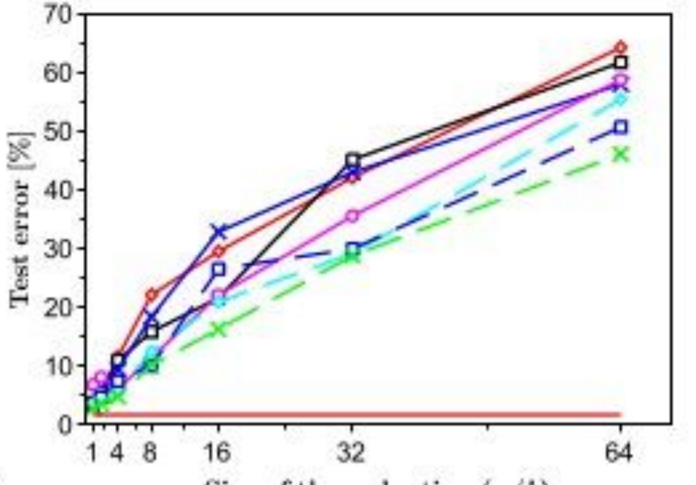
Numerical Experiments (Neural Net)

- Mean test error versus size of hash (k)



Numerical Experiments (Neural Net)

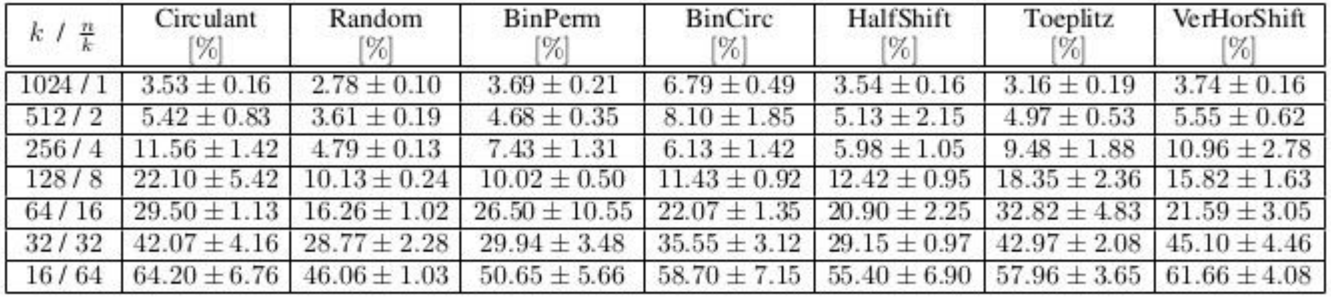
- Mean test error versus size of reduction (n/k) for the network



b)

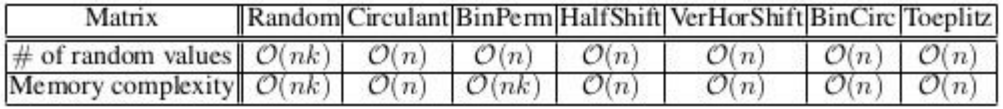
Numerical Experiments

Table 1: Mean and std of the test error versus the size of the hash (k) / size of the reduction (n/k) for the network.



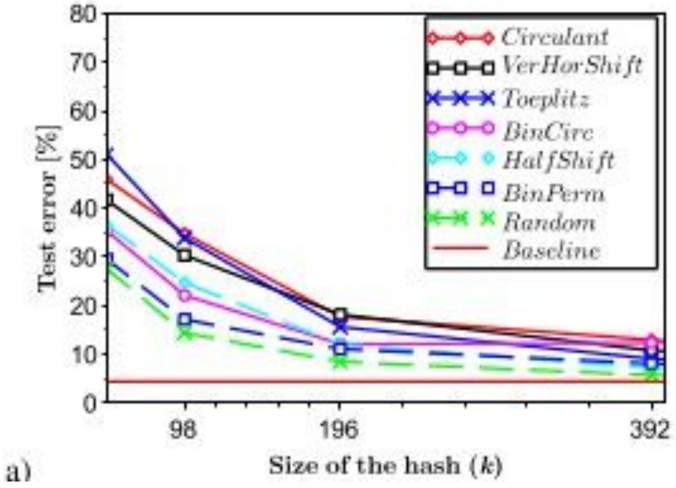
k / $\frac{n}{k}$	Circulant [%]	Random [%]	BinPerm [%]	BinCirc [%]	HalfShift [%]	Toeplitz [%]	VerHorShift [%]
1024 / 1	3.53 ± 0.16	2.78 ± 0.10	3.69 ± 0.21	6.79 ± 0.49	3.54 ± 0.16	3.16 ± 0.19	3.74 ± 0.16
512 / 2	5.42 ± 0.83	3.61 ± 0.19	4.68 ± 0.35	8.10 ± 1.85	5.13 ± 2.15	4.97 ± 0.53	5.55 ± 0.62
256 / 4	11.56 ± 1.42	4.79 ± 0.13	7.43 ± 1.31	6.13 ± 1.42	5.98 ± 1.05	9.48 ± 1.88	10.96 ± 2.78
128 / 8	22.10 ± 5.42	10.13 ± 0.24	10.02 ± 0.50	11.43 ± 0.92	12.42 ± 0.95	18.35 ± 2.36	15.82 ± 1.63
64 / 16	29.50 ± 1.13	16.26 ± 1.02	26.50 ± 10.55	22.07 ± 1.35	20.90 ± 2.25	32.82 ± 4.83	21.59 ± 3.05
32 / 32	42.07 ± 4.16	28.77 ± 2.28	29.94 ± 3.48	35.55 ± 3.12	29.15 ± 0.97	42.97 ± 2.08	45.10 ± 4.46
16 / 64	64.20 ± 6.76	46.06 ± 1.03	50.65 ± 5.66	58.70 ± 7.15	55.40 ± 6.90	57.96 ± 3.65	61.66 ± 4.08

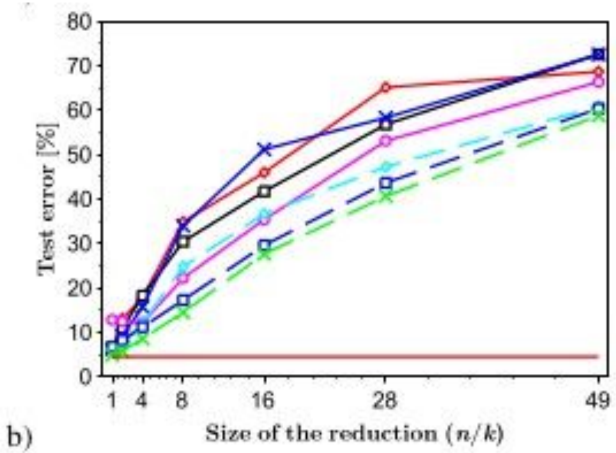
Table 2: Memory complexity and number of required random values for structured matrices and *Random* matrix.



Matrix	Random	Circulant	BinPerm	HalfShift	VerHorShift	BinCirc	Toeplitz
# of random values	$\mathcal{O}(nk)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Memory complexity	$\mathcal{O}(nk)$	$\mathcal{O}(n)$	$\mathcal{O}(nk)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

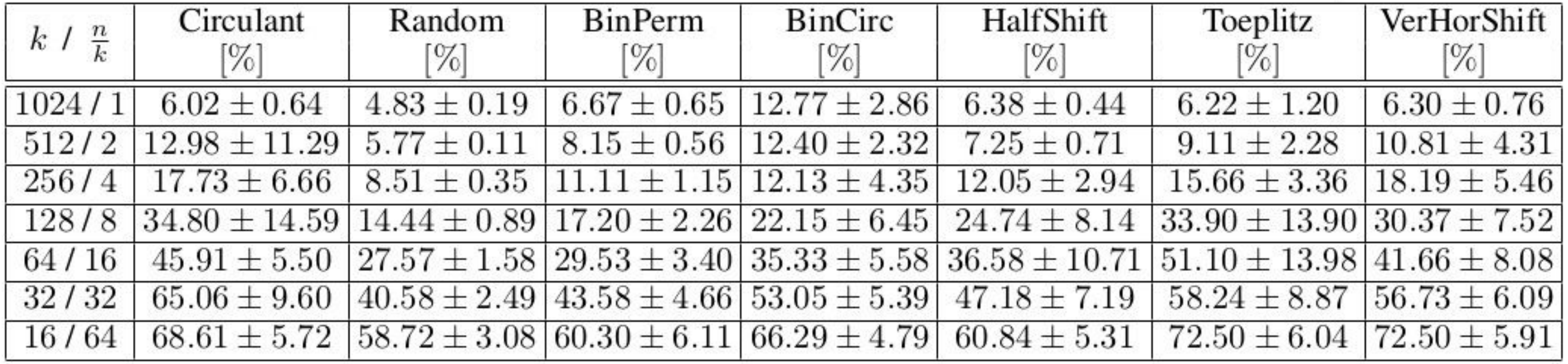
Numerical Results (1-NN)





Numerical Results (1-NN)

Table 4: Mean and std of the test error versus the size of the hash (k) / size of the reduction (n/k) for 1-NN.



k / $\frac{n}{k}$	Circulant [%]	Random [%]	BinPerm [%]	BinCirc [%]	HalfShift [%]	Toeplitz [%]	VerHorShift [%]
1024 / 1	6.02 ± 0.64	4.83 ± 0.19	6.67 ± 0.65	12.77 ± 2.86	6.38 ± 0.44	6.22 ± 1.20	6.30 ± 0.76
512 / 2	12.98 ± 11.29	5.77 ± 0.11	8.15 ± 0.56	12.40 ± 2.32	7.25 ± 0.71	9.11 ± 2.28	10.81 ± 4.31
256 / 4	17.73 ± 6.66	8.51 ± 0.35	11.11 ± 1.15	12.13 ± 4.35	12.05 ± 2.94	15.66 ± 3.36	18.19 ± 5.46
128 / 8	34.80 ± 14.59	14.44 ± 0.89	17.20 ± 2.26	22.15 ± 6.45	24.74 ± 8.14	33.90 ± 13.90	30.37 ± 7.52
64 / 16	45.91 ± 5.50	27.57 ± 1.58	29.53 ± 3.40	35.33 ± 5.58	36.58 ± 10.71	51.10 ± 13.98	41.66 ± 8.08
32 / 32	65.06 ± 9.60	40.58 ± 2.49	43.58 ± 4.66	53.05 ± 5.39	47.18 ± 7.19	58.24 ± 8.87	56.73 ± 6.09
16 / 64	68.61 ± 5.72	58.72 ± 3.08	60.30 ± 6.11	66.29 ± 4.79	60.84 ± 5.31	72.50 ± 6.04	72.50 ± 5.91

Conclusion

- Shows construction of hash projections that will preserve angular distance between inputs
- Theoretical results revolve around mapping data to lower-dimensional space with structured linear projections followed by previously-unconsidered sign nonlinearity
- Empirically verify theoretical results for nearest neighbor classifier

Additional References

- http://yann.lecun.com/exdb/publis/pdf/lecun-06.pdf
- https://arxiv.org/abs/1406.2661
- http://www.deeplearningbook.org/
- https://en.wikipedia.org/wiki/Stability_(learning_theory)