Structured adaptive and random spinners for fast machine learning computations

Mariusz Bojarski[1]
NVIDIA
mbojarski@nvidia.com

Anna Choromanska[1]
Courant Inst. of Math. Sciences, NYU
achoroma@cims.nyu.edu

Krzysztof Choromanski[1]
Google Brain Robotics
kchoro@google.com

Francois Fagan[1]
Columbia University
ff2316@columbia.edu

Cedric Gouy-Pailler[1]
CEA, LIST, LADIS
cedric.gouy-pailler@cea.fr

Anne Morvan[1]
CEA, LIST, LADIS
and Universite Paris-Dauphine
anne.morvan@cea.fr

Nouri Sakr[1]
Columbia University
nts2122@columbia.edu

Tamas Sarlos[1]
Google Research
stamas@google.com

Jamal Atif
Universite Paris-Dauphine
jamal.atif@dauphine.fr

Abstract

We consider an efficient computational framework for speeding up several machine learning algorithms with almost no loss of accuracy. The proposed framework relies on projections via structured matrices that we call *Structured Spinners*, which are formed as products of three structured matrix-blocks that incorporate rotations. The approach is highly generic, i.e. i) structured matrices under consideration can either be fully-randomized or learned, ii) our structured family contains as special cases all previously considered structured schemes, iii) the setting extends to the non-linear case where the projections are followed by non-linear functions, and iv) the method finds numerous applications including kernel approximations via random feature maps, dimensionality reduction algorithms, new fast cross-polytope LSH techniques, deep learning, convex optimization algorithms via Newton sketches, quantization with random projection trees, and more. The proposed framework comes with theoretical guarantees characterizing the capacity of the structured model in reference to its unstructured counterpart and is based on a general theoretical principle that we describe in the paper. As a consequence of our theoretical analysis, we provide the first theoretical guarantees for one of the most efficient existing LSH algorithms based on the $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$ structured matrix [Andoni et al., 2015]. The exhaustive experimental evaluation confirms the accuracy and efficiency of structured spinners for a variety of different applications.

1 Introduction

A striking majority of machine learning frameworks performs projections of input data via matrices of parameters, where the obtained projections are often passed to a possibly highly nonlinear function. In the case of randomized machine learning algorithms, the projection matrix is typically Gaussian with i.i.d. entries taken from $\mathcal{N}(0,1)$. Otherwise, it is learned through the optimization scheme. A plethora of machine learning algorithms admits this form. In the randomized setting, a few examples include variants of the Johnson-Lindenstrauss Transform applying random projections to reduce data dimensionality while approximately preserving Euclidean distance [Ailon and Chazelle, 2006, Liberty et al., 2008, Ailon and Liberty, 2011], kernel approximation techniques based on random feature maps produced from linear projections with Gaussian matrices followed by nonlinear mappings [Rahimi and Recht, 2007], [Le et al., 2013, Choromanski and Sindhwani, 2016, Huang et al., 2014], [Choromanska et al., 2016], LSH-based schemes [Har-Peled et al., 2012, Charikar, 2002, Terasawa and Tanaka, 2007], including the fastest known variant of the cross-polytope LSH [Andoni et al., 2015], algorithms solving convex optimization problems with random sketches of Hessian matrices [Pilanci and Wainwright, 2015, Pilanci and Wainwright, 2014], quantization techniques using random projection trees, where splitting in each node is determined by a projection of data onto Gaussian direction [Dasgupta and Freund, 2008], and many more.

[1] equal contribution

The classical example of machine learning nonlinear models where linear projections are learned is a multi-layered neural network [LeCun et al., 2015, Goodfellow et al., 2016], where the operations of linear projection via matrices with learned parameters followed by the pointwise nonlinear feature transformation are the building blocks of the network's architecture. These two operations are typically stacked multiple times to form a deep network.

The computation of projections takes $\Theta(mn|\mathcal{X}|)$ time, where $m \times n$ is the size of the projection matrix, and $|\mathcal{X}|$ denotes the number of data samples from a dataset $\mathcal{X}$. In case of high-dimensional data, this comprises a significant fraction of the overall computational time, while storing the projection matrix frequently becomes a bottleneck in terms of space complexity.

In this paper, we propose the remedy for both problems, which relies on replacing the aforementioned algorithms by their "structured variants". The projection is performed by applying a structured matrix from the family that we introduce as *Structured Spinners*. Depending on the setting, the structured matrix is either learned or its parameters are taken from a random distribution (either continuous or discrete if further compression is required). Each structured spinner is a product of three matrix-blocks that incoporate rotations. A notable member of this family is a matrix of the form $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$, where $\mathbf{D}_i$s are either random diagonal ± 1-matrices or adaptive diagonal matrices and $\mathbf{H}$ is the Hadamard matrix. This matrix is used in the fastest known cross-polytope LSH method introduced in [Andoni et al., 2015].

In the structured case, the computational speedups are significant, i.e. projections can be calculated in $o(mn)$ time, often in $O(n \log m)$ time if Fast Fourier Transform techniques are applied. At the same time, using matrices from the family of structured spinners leads to the reduction of space complexity to sub-quadratic, usually at most linear, or sometimes even constant.

The key contributions of this paper are:

- The family of structured spinners providing a highly parametrized class of structured methods and, as we show in this paper, with applications in various randomized settings such as: kernel approximations via random feature maps, dimensionality reduction algorithms, new fast cross-polytope LSH techniques, deep learning, convex optimization algorithms via Newton sketches, quantization with random projection trees, and more.

- A comprehensive theoretical explanation of the effectiveness of the structured approach based on structured spinners. Such analysis was provided in the literature before for a strict subclass of a very general family of structured matrices that we consider in this paper, i.e. the proposed family of structured spinners contains all previously considered structured matrices as special cases, including the recently introduced P-model [Choromanski and Sindhwani, 2016]. To the best of our knowledge, we are the first to theoretically explain the effectiveness of structured neural network architectures. Furthermore, we provide first theoretical guarantees for a wide range of discrete structured transforms, in particular for the fastest known cross-polytope LSH method [Andoni et al., 2015] based $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$ discrete matrices.

Our theoretical methods in the random setting apply the relatively new Berry-Esseen type Central Limit Theorem results for random vectors.

Our theoretical findings are supported by empirical evidence regarding the accuracy and efficiency of structured spinners in a wide range of different applications. Not only do structured spinners cover all already existing structured transforms as special instances, but also many other structured matrices that can be applied in all aforementioned applications.

2 Related work

This paper focuses on structured matrices, which were previously explored in the literature mostly in the context of the Johnson-Lindenstrauss Transform (JLT) [Johnson and Lindenstrauss, 1984], where the high-dimensional data is linearly transformed and embedded into a much lower dimensional space while approximately preserving the Euclidean distance between data points. Several extensions of JLT have been proposed, e.g. [Liberty et al., 2008, Ailon and Liberty, 2011, Ailon and Chazelle, 2006, Vybíral, 2011]. Most of these structured constructions involve sparse [Ailon and Chazelle, 2006, Dasgupta et al., 2010] or circulant matrices [Vybíral, 2011, Hinrichs and Vybral, 2011] providing computational speedups and space compression.

More recently, the so-called Ψ-regular structured matrices (Toeplitz and circulant matrices belong to this wider family of matrices) were used to approximate angular distances [Choromanska et al., 2016] and signed Circulant Random Matrices were used to approximate Gaussian kernels [Feng et al., 2015]. Another work [Choromanski and Sindhwani, 2016] applies structured matrices coming from the so-called *P-model*, which further generalizes the Ψ-regular family, to speed up random feature map computations of some special kernels (angular, arc-cosine and Gaussian). These techniques did not work for discrete structured constructions, such as the $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$ matrices, or their direct non-discrete modifications, since they require matrices with low (polylog) chromatic number of the corresponding coherence graphs.

Linear projections are used in the LSH setting to construct codes for given datapoints which speed up such tasks as approximate nearest neighbor search. A notable set of methods are the so-called cross-polytope techniques introduced in [Terasawa and Tanaka, 2007] and their aforementioned discrete structured variants proposed in [Andoni et al., 2015] that are based on the Walsh-Hadamard transform. Before our work, they were only experimentally verified to produce good quality codes.

Furthermore, a recently proposed technique based on the so-called *Newton Sketch* provides yet another example of application for structured matrices. The method [Pilanci and Wainwright, 2015, Pilanci and Wainwright, 2014] is used for speeding up algorithms solving convex optimization problems by approximating Hessian matrices using so-called *sketch matrices*. Initially, the sub-Gaussian sketches based on i.i.d. sub-Gaussian random variables were used. The disadvantage of the sub-Gaussian sketches lies in the fact that computing the sketch of the given matrix of size $n \times d$ requires $O(mnd)$ time, where $m \times n$ in the size of the sketch matrix. Thus the method is too slow in practice and could be accelerated with the use of structured matrices. Some structured approaches were already considered, e.g. sketches based on randomized orthonormal systems were proposed [Pilanci and Wainwright, 2015].

All previously considered methods focus on the randomized setting, whereas the structured matrix instead of being learned is fully random. In the context of adaptive setting, where the parameters are being learned instead, we focus in this paper on multi-layer neural networks. We emphasize though that our approach is much more general and extends beyond this setting. Structured neural networks were considered before, for instance in [Yang et al., 2015], where the so-called *Deep Fried Neural Convnets* were proposed. Those architectures are based on the adaptive version of the Fastfood transform used for approximating various kernels [Le et al., 2013], which is a special case of structured spinner matrices.

Deep Fried Convnets apply adaptive structured matrices for fully connected layers of the convolutional networks. The structured matrix is of the form: $\mathbf{SHG\Pi HB}$, where $\mathbf{S}$, $\mathbf{G}$, and $\mathbf{B}$ are adaptive diagonal matrices, $\mathbf{\Pi}$ is a random permutation matrix, and $\mathbf{H}$ is the Walsh-Hadamard matrix. The method reduces the storage and computational costs of matrix multiplication step from, often prohibitive, $\mathcal{O}(nd)$ down to $\mathcal{O}(n)$ storage and $\mathcal{O}(n \log d)$ computational cost, where d and n denote the size of consequitive layers of the network. At the same time, this approach does not sacrifice the network's predictive performance.

The Adaptive Fastfood approach elegantly complements previous works dedicated to address the problem of huge overparametrization of deep models with structured matrices, e.g. the method of [Denil et al., 2013] represents the parameter matrix as a product of two low rank factors and, similarly to Adaptive Fastfood, applies both at train and test time, [Sainath et al., 2013] introduces low-rank matrix factorization to reduce the size of the fully connected layers at train time, and [Li, 2013] uses low-rank factorizations with SVD after training the full model. These methods, as well as approaches that consider kernel methods in deep learning [Cho and Saul, 2009, Mairal et al., 2014, Dai et al., 2014, Huang et al., 2014], are conveniently discussed in [Yang et al., 2015].

Structured neural networks are also considered in [Sindhwani et al., 2015], where low-displacement rank matrices are applied for linear projections. The advantage of this approach over Deep Fried Convnets is due to the high parametrization of the family of low-displacement rank matrices allowing the adjustment of the number of parameters learned based on accuracy and speedup requirements.

The class of structured spinners proposed in this work is more general than Deep Fried Convnets or low displacement rank matrices, but it also provides much easier structured constructions, such as $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$ matrices, where $\mathbf{D}_i$s are adaptive diagonal matrices. Furthermore, to the best of our knowledge we are the first to prove theoretically that structured neural networks learn good quality models, by analyzing the capacity of the family of structured spinners.

3 The family of *Structured Spinners*

Before introducing the family of structured spinners, we explain notation. If not specified otherwise, matrix $\mathbf{D}$ is a random diagonal matrix with diagonal entries taken independently at random from $\{-1, +1\}$. By $\mathbf{D}_{t_1,...,t_n}$ we denote the diagonal matrix with diagonal equal to $(t_1, ..., t_n)$. For a matrix $\mathbf{A} = \{a_{i,j}\}_{i,j=1,...,n} \in \mathbb{R}^{n \times n}$, we denote by $\|\mathbf{A}\|_F$ its Frobenius norm, i.e. $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j \in \{1,...,n\}} a_{i,j}^2}$, and by $\|\mathbf{A}\|_2$ its spectral norm, i.e. $\|\mathbf{A}\|_2 = \sup_{\mathbf{x} \neq 0} \frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_2}$. We denote by $\mathbf{H}$ the L_2-normalized Hadamard matrix. We say that $\mathbf{r}$ is a random Rademacher vector if every element of $\mathbf{r}$ is chosen independently at random from $\{-1, +1\}$.

For a vector $\mathbf{r} \in \mathbb{R}^k$ and $n > 0$ let $\mathbf{C}(\mathbf{r}, n) \in \mathbb{R}^{n \times nk}$ be a matrix, where the first row is of the form $(\mathbf{r}^T, 0, ..., 0)$ and each subsequent row is obtained from the previous one by right-shifting in a circulant manner the previous one by k. For a sequence of matrices $\mathbf{W}^1, ..., \mathbf{W}^n \in \mathbb{R}^{k \times n}$ we denote by $\mathbf{V}(\mathbf{W}^1, ..., \mathbf{W}^n) \in \mathbb{R}^{nk \times n}$ a matrix obtained by vertically stacking matrices: $\mathbf{W}^1, ..., \mathbf{W}^n$.

Each structured matrix $\mathbf{G}_{struct} \in \mathbb{R}^{n \times n}$ from the family of structured spinners is a product of three main structured components/blocks, i.e.:

$$\mathbf{G}_{struct} = \mathbf{M}_3 \mathbf{M}_2 \mathbf{M}_1, \quad (1)$$

where matrices $\mathbf{M}_1, \mathbf{M}_2$ and $\mathbf{M}_3$ satisfy conditions:

Condition 1: Matrices: $\mathbf{M}_1$ and $\mathbf{M}_2\mathbf{M}_1$ are $(\delta(n), p(n))$-balanced isometries.
Condition 2: $\mathbf{M}_2 = \mathbf{V}(\mathbf{W}^1, ..., \mathbf{W}^n)\mathbf{D}_{\rho_1,...,\rho_n}$ for some (Δ_F, Δ_2)-smooth set: $\mathbf{W}^1, ..., \mathbf{W}^n \in \mathbb{R}^{k \times n}$ and some i.i.d sub-Gaussian random variables $\rho_1, ..., \rho_n$ with sub-Gaussian norm K.
Condition 3: $\mathbf{M}_3 = \mathbf{C}(\mathbf{r}, n)$ for $\mathbf{r} \in \mathbb{R}^k$, where $\mathbf{r}$ is random Rademacher/Gaussian in the random setting and is learned in the adaptive setting.

Matrix $\mathbf{G}_{struct}$ is a structured spinner with parameters: $\delta(n), p(n), K, \Lambda_F, \Lambda_2$. We explain the introduced conditions below.

Definition 1 ($(\delta(n), p(n))$-balanced matrices) *A randomized matrix $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ is $(\delta(n), p(n))$-balanced if for every $\boldsymbol{x} \in \mathbb{R}^m$ with $\|\boldsymbol{x}\|_2 = 1$ we have: $\mathbb{P}[\|\boldsymbol{M}\boldsymbol{x}\|_\infty > \frac{\delta(n)}{\sqrt{n}}] \leq p(n)$.*

Remark 1 *One can take as $\boldsymbol{M}_1$ a matrix $\boldsymbol{H}\boldsymbol{D}_1$ since, as we will show in the Supplement, matrix $\boldsymbol{H}\boldsymbol{D}_1$ is $(\log(n), 2ne^{-\frac{\log^2(n)}{8}})$-balanced.*

Definition 2 ((Δ_F, Δ_2)-smooth sets) *A deterministic set of matrices $\boldsymbol{W}^1, ..., \boldsymbol{W}^n \in \mathbb{R}^{k \times n}$ is (Λ_F, Λ_2)-smooth if:*

- *$\|\boldsymbol{W}_1^i\|_2 = .. = \|\boldsymbol{W}_n^i\|_2$ for $i = 1, ..., n$, where $\boldsymbol{W}_j^i$ stands for the j^{th} column of $\boldsymbol{W}^i$,*
- *for $i \neq j$ and $l = 1, ..., n$ we have: $(\boldsymbol{W}_l^i)^T \cdot \boldsymbol{W}_l^j = 0$,*
- *$\max_{i,j} \|(\boldsymbol{W}^j)^T \boldsymbol{W}^i\|_F \leq \Lambda_F$ and $\max_{i,j} \|(\boldsymbol{W}^j)^T \boldsymbol{W}^i\|_2 \leq \Lambda_2$.*

Remark 2 *If the unstructured matrix $\boldsymbol{G}$ has rows taken from the general multivariate Gaussian distribution with diagonal covariance matrix $\Sigma \neq \boldsymbol{I}$ then one needs to rescale vectors $\mathbf{r}$ accordingly. For clarity, we assume here that $\Sigma = \boldsymbol{I}$ and we present our theoretical results for that setting.*

All structured matrices previously considered are special cases of a wider family of structured spinners (for clarity, we will explicitly show it for some important special cases). We have:

Lemma 1 *The following matrices: $\boldsymbol{G}_{circ}\boldsymbol{D}_2\boldsymbol{H}\boldsymbol{D}_1$, $\sqrt{n}\boldsymbol{H}\boldsymbol{D}_3\boldsymbol{H}\boldsymbol{D}_2\boldsymbol{H}\boldsymbol{D}_1$ and $\sqrt{n}\boldsymbol{H}\boldsymbol{D}_{g_1,...,g_n}\boldsymbol{H}\boldsymbol{D}_2\boldsymbol{H}\boldsymbol{D}_1$, where $\boldsymbol{G}_{circ}$ is Gaussian circulant, are valid structured spinners for $\delta(n) = \log(n)$, $p(n) = 2ne^{-\frac{\log^2(n)}{8}}$, $K = 1$, $\Lambda_F = O(\sqrt{n})$ and $\Lambda_2 = O(1)$. The same is true if one replaces $\boldsymbol{G}_{circ}$ by a Gaussian Hankel or Toeplitz matrix.*

3.1 The role of three blocks $\mathbf{M_1}$, $\mathbf{M_2}$, and $\mathbf{M_3}$

The role of blocks $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$ can be intuitively explained. Matrix $\mathbf{M}_1$ makes vectors "balanced", so that there is no dimension that carries too much of the L_2-norm of the vector. The balanceness property was already applied in the structured setting [Ailon and Chazelle, 2006].

The role of $\mathbf{M}_2$ is more subtle and differs between adaptive and random settings. In the random setting, the cost of applying the structured mechanism is the loss of independence. For instance, the dot products of the rows of a circulant Gaussian matrix with a given vector $\mathbf{x}$ are no longer independent, as it is the case in the fully random setup. Those dot products can be expressed as a dot product of a fixed Gaussian row with different vectors $\mathbf{v}$. Matrix $\mathbf{M}_2$ makes these vectors close to orthogonal. In the adaptive setup, the "close to orthogonality" property is replaced by the independence property.

Finally, matrix $\mathbf{M}_3$ defines the capacity of the entire structured transform by providing a vector of parameters (either random or to be learned). The near-independence of the aforementioned dot products in the random setting is now implied by the near-orthogonality property achieved by $\mathbf{M}_2$ and the fact that the projections of the Gaussian vector or the random Rademacher vector onto "almost orthogonal directions" are "close to independent". The role of the three matrices is described pictorially in Figure 1.

3.2 Stacking together *Structured Spinners*

We described structured spinners as square matrices, but in practice we are not restricted to those, i.e. one can construct an $m \times n$ structured spinner for $m \leq n$ from the square $n \times n$ structured spinner by taking its first m rows. We can then stack vertically these independently constructed $m \times n$ matrices to obtain an $k \times n$ matrix for both: $k \leq n$ and $k > n$. We think about m as another parameter of the model that tunes the "structuredness" level, i.e. larger values of m indicate more structured approach while smaller values lead to more random matrices ($m = 1$ case is the fully unstructured one).

4 Theoretical results

We now show that structured spinners can replace their unstructured counterparts in many machine learning

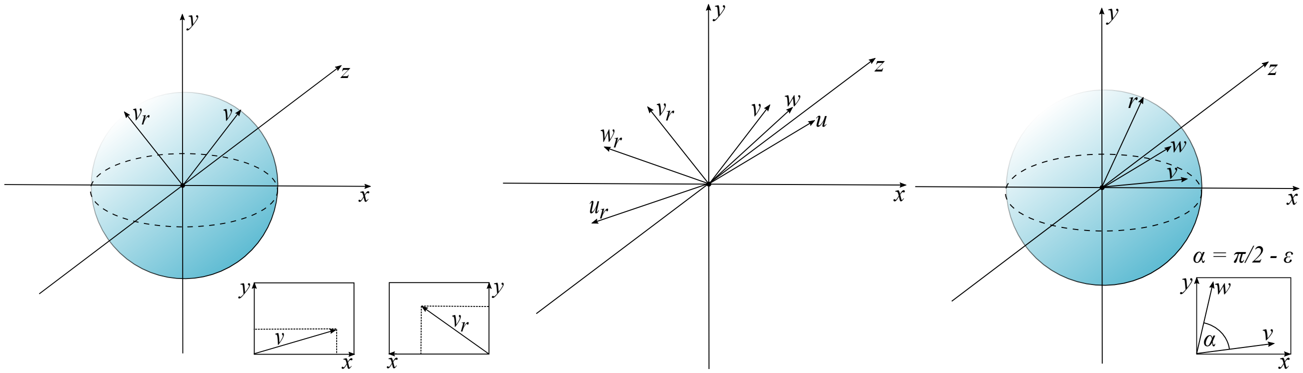

Figure 1: Pictorial explanation of the role of three matrix-blocks in the construction of the structured spinner. Left picture: $\mathbf{M}_1$ rotates $\mathbf{v}$ such that the rotated version $\mathbf{v}_r$ is balanced. Middle picture: $\mathbf{M}_2$ transforms vectors $\mathbf{v}, \mathbf{w}, \mathbf{u}$ such that their images $\mathbf{v}_r, \mathbf{w}_r, \mathbf{u}_r$ are near-orthogonal. Right picture: The projections of the random vector $\mathbf{r}$ onto such two near-orthogonal vectors $\mathbf{v}$, $\mathbf{w}$ are near-independent.

algorithms with minimal loss of accuracy.

Let $\mathcal{A}_{\mathcal{G}}$ be a machine learning algorithm applied to a fixed dataset $\mathcal{X} \subseteq \mathbb{R}^n$ and parametrized by a set $\mathcal{G}$ of matrices $\mathbf{G} \in \mathcal{R}^{m \times n}$, where each $\mathbf{G}$ is either learned or Gaussian with independent entries taken from $\mathcal{N}(0,1)$. Assume furthermore, that $\mathcal{A}_{\mathcal{G}}$ consists of functions $f_1, ..., f_s$, where each f_i applies a certain matrix $\mathbf{G}_i$ from $\mathcal{G}$ to vectors from some linear space $\mathcal{L}_i$ of dimensionality at most d. Note that for a fixed dataset $\mathcal{X}$ function f_i is a function of a random vector

$$\mathbf{q}_{f_i} = ((\mathbf{G}_i\mathbf{x}^1)^T, ..., (\mathbf{G}_i\mathbf{x}^{d_i})^T)^T \in \mathbb{R}^{d_i \cdot m},$$

where $dim(\mathcal{L}_i) = d_i \leq d$ and $\mathbf{x}^1, ..., \mathbf{x}^{d_i}$ stands for some fixed basis of $\mathcal{L}_i$.

Denote by f'_i the structured counterpart of f_i, where $\mathbf{G}_i$ is replaced by the structured spinner (for which vector $\mathbf{r}$ is either learned or random). We will show that $f'_i s$ "resemble" $f_i s$ distribution-wise. Surprisingly, we will show it under very weak conditions regarding $f_i s$, In particular, they can be nondifferentiable, even non-continuous.

Note that the above setting covers a wide range of machine learning algorithms. In particular:

Remark 3 *In the kernel approximation setting with random feature maps one can match each pair of vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ to a different $f = f_{\boldsymbol{x},\boldsymbol{y}}$. Each f computes the approximate value of the kernel for vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Thus in that scenario $s = \binom{|\mathcal{X}|}{2}$ and $d = 2$ (since one can take: $\mathcal{L}_{f(\boldsymbol{x},\boldsymbol{y})} = span(\boldsymbol{x}, \boldsymbol{y})$).*

Remark 4 *In the vector quantization algorithms using random projection trees one can take $s = 1$ (the algorithm $\mathcal{A}$ itself is a function f outputting the partitioning of space into cells) and $d = d_{intrinsic}$, where $d_{intrinsic}$ is an intrinsic dimensionality of a given dataset $\mathcal{X}$ (random projection trees are often used if $d_{intrinsic} \ll n$).*

4.1 Random setting

We need the following definition.

Definition 3 *A set $\mathcal{S}$ is b-convex if it is a union of at most b pairwise disjoint convex sets.*

Fix a funcion $f_i : \mathbb{R}^{d_i \cdot m} \rightarrow \mathcal{V}$, for some domain $\mathcal{V}$. Our main result states that for any $\mathcal{S} \subseteq \mathcal{V}$ such that $f_i^{-1}(\mathcal{S})$ is measurable and b-convex for b not too large, the probability that $f_i(\mathbf{q}_{f_i})$ belongs to $\mathcal{S}$ is close to the probability that $f'_i(\mathbf{q}_{f'_i})$ belongs to $\mathcal{S}$.

Theorem 1 (structured random setting) *Let $\mathcal{A}$ be a randomized algorithm using unstructured Gaussian matrices $\boldsymbol{G}$ and let s, d and $f_i s$ be as at the beginning of the section. Replace the unstructured matrix $\boldsymbol{G}$ by one of structured spinners defined in Section 3 with blocks of m rows each. Then for n large enough, $\epsilon = o_{md}(1)$ and fixed f_i with probability p_{succ} at least:*

$$1-2p(n)d-2\binom{md}{2}e^{-\Omega(\min(\frac{\epsilon^2 n^2}{K^4\Lambda_F^2\delta^4(n)},\frac{\epsilon n}{K^2\Lambda_2\delta^2(n)}))} \tag{2}$$

with respect to the random choices of $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ the following holds for any $\mathcal{S}$ such that $f_i^{-1}(\mathcal{S})$ is measurable and b-convex:

$$|\mathbb{P}[f_i(\boldsymbol{q}_{f_i}) \in \mathcal{S}] - \mathbb{P}[f'_i(\boldsymbol{q}_{f'_i}) \in \mathcal{S}]| \leq b\eta,$$

where the the probabilities in the last formula are with respect to the random choice of $\boldsymbol{M}_3$, $\eta = \frac{\delta^3(n)}{n^{\frac{2}{5}}}$, and $\delta(n), p(n), K, \Lambda_F, \Lambda_2$ are as in the definition of structured spinners from Section 3.

Remark 5 *The theorem does not require any strong regularity conditions regarding $f_i s$ (such as differentiability or even continuity). In practice, b is often a small constant. For instance, for the angular kernel approximation where $f_i s$ are non-continuous and for $\mathcal{S}$-singletons, we can take $b = 1$ (see Supplement).*

Now let us think of f_i and f_i' as random variables, where randomness is generated by vectors $\mathbf{q}_{f_i}$ and $\mathbf{q}_{f_i'}$ respectively. Then, from Theorem 1, we get:

Theorem 2 *Denote by F_X the cdf of the random variable X and by ϕ_X its characteristic function. If f_i is convex or concave in respect to $\boldsymbol{q}_{f_i}$, then for every t the following holds: $|F_{f_i}(t) - F_{f_i'}(t)| = O(\frac{\delta^3(n)}{n^{\frac{2}{5}}})$. Furthermore, if f_i is bounded then: $|\phi_{f_i}(t) - \phi_{f_i'}(t)| = O(\frac{\delta^3(n)}{n^{\frac{2}{5}}})$.*

Theorem 1 implies strong accuracy guarantees for the specific structured spinners. As a corollary we get:

Theorem 3 *Under assumptions from Theorem 1 the probability p_{succ} from Theorem 1 reduces to: $1 - 4ne^{-\frac{\log^2(n)}{8}}d - 2\binom{md}{2}e^{-\Omega(\frac{\epsilon^2 n}{\log^4(n)})}$ for the structured matrices $\sqrt{n}\boldsymbol{HD}_3\boldsymbol{HD}_2\boldsymbol{HD}_1$, $\sqrt{n}\boldsymbol{HD}_{g_1,\ldots,g_n}\boldsymbol{HD}_2\boldsymbol{HD}_1$ as well as for the structured matrices of the form $\boldsymbol{G}_{struct}\boldsymbol{D}_2\boldsymbol{HD}_1$, where $\boldsymbol{G}_{struct}$ is Gaussian circulant, Gaussian Toeplitz or Gaussian Hankel matrix.*

As a corollary of Theorem 3, we obtain the following result showing the effectiveness of the cross-polytope LSH with structured matrices $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$ that was only heuristically confirmed before [Andoni et al., 2015].

Theorem 4 *Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ be two unit L_2-norm vectors. Let $\boldsymbol{v}_{\boldsymbol{x},\boldsymbol{y}}$ be the vector indexed by all $(2m)^2$ ordered pairs of canonical directions $(\pm\boldsymbol{e}_i, \pm\boldsymbol{e}_j)$, where the value of the entry indexed by $(\boldsymbol{u}, \boldsymbol{w})$ is the probability that: $h(\boldsymbol{x}) = \boldsymbol{u}$ and $h(\boldsymbol{y}) = \boldsymbol{w}$, and $h(\boldsymbol{v})$ stands for the hash of $\boldsymbol{v}$. Then with probability at least: $p_{success} = 1 - 8ne^{-\frac{\log^2(n)}{8}} - 2\binom{2m}{2}e^{-\Omega(\frac{\epsilon^2 n}{\log^4(n)})}$ the version of the stochastic vector $\boldsymbol{v}^1_{\boldsymbol{x},\boldsymbol{y}}$ for the unstructured Gaussian matrix $\boldsymbol{G}$ and its structured counterpart $\boldsymbol{v}^2_{\boldsymbol{x},\boldsymbol{y}}$ for the matrix $\boldsymbol{HD}_3\boldsymbol{HD}_2\boldsymbol{HD}_1$ satisfy: $\|\boldsymbol{v}^1_{\boldsymbol{x},\boldsymbol{y}} - \boldsymbol{v}^2_{\boldsymbol{x},\boldsymbol{y}}\|_\infty \leq \log^3(n)n^{-\frac{2}{5}} + c\epsilon$, for n large enough, where $c > 0$ is a universal constant. The probability above is taken with respect to random choices of $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$.*

For angles in the range $[0, \frac{\pi}{3}]$ the result above leads to the same asymptotics of the probabilities of collisions as these in Theorem 1 of [Andoni et al., 2015] given for the unstructured cross-polytope LSH.

The proof for the discrete structured setting applies Berry-Esseen-type results for random vectors (details are in the Supplement) showing that for n large enough ± 1 random vectors $\mathbf{r}$ act similarly to Gaussian vectors.

4.2 Adaptive setting

The following theorem explains that structured spinners can be used to replace unstructured fully connected neural network layers performing dimensionality reduction (such as hidden layers in certain autoencoders) provided that input data has low intrinsic dimensionality. These theoretical findings were confirmed in experiments that will be presented in the next section. We will use notation from Theorem 1.

Theorem 5 *Consider a matrix $\boldsymbol{M} \in \mathbb{R}^{m \times n}$ encoding the weights of connections between a layer l_0 of size n and a layer l_1 of size m in some learned unstructured neural network model. Assume that the input to layer l_0 is taken from the d-dimensional space $\mathcal{L}$ (although potentially embedded in a much higher dimensional space). Then with probability at least*

$$1 - 2p(n)d - 2\binom{md}{2}e^{-\Omega(\min(\frac{t^2n^2}{K^4\Lambda_F^2\delta^4(n)}, \frac{tn}{K^2\Lambda_2\delta^2(n)}))} \qquad (3)$$

for $t = \frac{1}{md}$ and with respect to random choices of $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, there exists a vector $\boldsymbol{r}$ defining $\boldsymbol{M}_3$ (see: definition of the structured spinner) such that the structured spinner $\boldsymbol{M}^{struct} = \boldsymbol{M}_3\boldsymbol{M}_2\boldsymbol{M}_1$ equals to $\boldsymbol{M}$ on $\mathcal{L}$.

5 Experiments

In this section we consider a wide range of different applications of structured spinners: locality-sensitive hashing, kernel approximations, and finally neural networks. Experiments with Newton sketches are deferred to the Supplement. Experiments were conducted using Python. In particular, NumPy is linked against a highly optimized BLAS library (Intel MKL). Fast Fourier Transform is performed using numpy.fft and Fast Hadamard Transform is using ffht from [Andoni et al., 2015]. To have a fair comparison, we have set up: OMP_NUM_THREADS = 1 so that every experiment is done on a single thread. Every parameter of the structured spinner matrix is computed in advance, such that obtained speedups take only matrix-vector products into account. All figures should be read in color.

5.1 Locality-Sensitive Hashing (LSH)

In the first experiment, we consider cross-polytope LSH. In Figure 2, we compare collision probabilities for the low dimensional case ($n = 256$), where for each interval, collision probability has been computed for 20000 points. Results are shown for one hash function (averaged over 100 runs). We report results for a random 256×64 Gaussian matrix $\mathbf{G}$ and five other types of matrices from a family of structured spinners (descending order of number of parameters): $\mathbf{G}_{circ}\mathbf{K}_2\mathbf{K}_1$, $\mathbf{G}_{Toeplitz}\mathbf{D}_2\mathbf{HD}_1$, $\mathbf{G}_{skew-circ}\mathbf{D}_2\mathbf{HD}_1$, $\mathbf{HD}_{g_1,\ldots,g_n}\mathbf{HD}_2\mathbf{HD}_1$, and $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$, where $\mathbf{K}_i$, $\mathbf{G}_{Toeplitz}$, and $\mathbf{G}_{skew-circ}$ are respectively a Kronecker matrix with discrete entries, Gaussian Toeplitz and Gaussian skew-circulant matrices.

All matrices from the family of structured spinners show high collision probabilities for small distances and

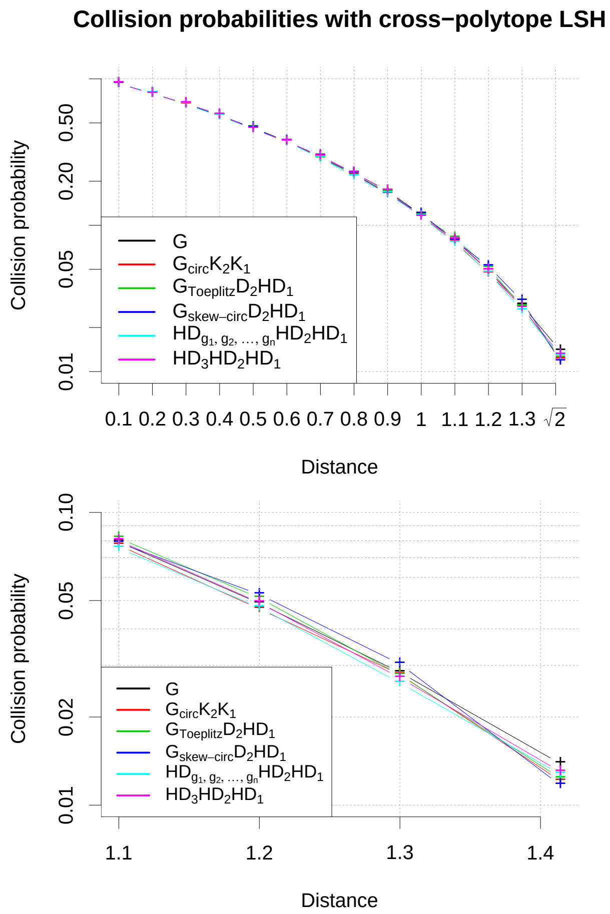

Figure 2: Cross-polytope LSH - collision probabilities. (bottom) A zoom on higher distances enables to distinguish the curves which are almost superposed.

low ones for large distances. As theoretically predicted, structured spinners do not lead to accuracy losses. All considered matrices give almost identical results.

5.2 Kernel approximation

In the second experiment, we approximate the Gaussian and angular kernels using Random Fourier features. The Gaussian random matrix (with i.i.d. Gaussian entries) can be used to sample random Fourier features with a specified σ. This Gaussian random matrix is replaced with specific matrices from a family of structured spinners for Gaussian and angular kernels. The obtained feature maps are compared. To test the quality of the structured kernels' approximations, we compute Gram-matrix reconstruction error as in [Choromanski and Sindhwani, 2016] : $\frac{||\mathbf{K}-\tilde{\mathbf{K}}||_F}{||\mathbf{K}||_F}$, where $\mathbf{K}, \tilde{\mathbf{K}}$ are respectively the exact and approximate Gram-matrices, as a function of the number of random features. When number of random features k is greater than data dimensionality n, we apply block-mechanism described in 3.2.

For the Gaussian kernel, $\mathbf{K}_{ij} = e^{\frac{-||\mathbf{x}_i-\mathbf{x}_j||_2^2}{2\sigma^2}}$ and for the angular kernel, $\mathbf{K}_{ij} = 1-\frac{\theta}{\pi}$ with $\theta = cos^{-1}(\frac{\mathbf{x}_i^T\mathbf{x}_j}{||\mathbf{x}_i||_2||\mathbf{x}_j||_2})$. For the approximation, $\tilde{\mathbf{K}}_{i,j} = \frac{1}{\sqrt{d'}}s(\mathbf{A}\mathbf{x}_i)^T\frac{1}{\sqrt{d'}}s(\mathbf{A}\mathbf{x}_j)$ where $s(x) = e^{\frac{-ix}{\sigma}}$ and $\tilde{\mathbf{K}}_{i,j} = 1 - \frac{\mathrm{d_H}(s(\mathbf{A}\mathbf{x}_i),s(\mathbf{A}\mathbf{x}_j))}{d'}$ where $s(x) = \mathrm{sign}(x)$ respectively. In both cases, function s is applied pointwise. d_H stands for the Hamming distance and x_i, x_j are points from the dataset.

We used two datasets: G50C (550 points, $n = 50$) and USPST (test set, 2007 points, $n = 256$). The results for the USPST dataset are given in the Supplement. For Gaussian kernel, bandwidth σ is set to 17.4734 for G50C and to 9.4338 for USPST. The choice of σ comes from [Choromanski and Sindhwani, 2016] in order to have comparable results. The results are averaged over 10 runs and the following matrices have been tested: Gaussian random matrix $\mathbf{G}$, $\mathbf{G}_{circ}\mathbf{K}_2\mathbf{K}_1$, $\mathbf{G}_{Toeplitz}\mathbf{D}_2\mathbf{H}\mathbf{D}_1$, $\mathbf{G}_{skew-circ}\mathbf{D}_2\mathbf{H}\mathbf{D}_1$, $\mathbf{H}\mathbf{D}_{g_1,...,g_n}\mathbf{H}\mathbf{D}_2\mathbf{H}\mathbf{D}_1$ and $\mathbf{H}\mathbf{D}_3\mathbf{H}\mathbf{D}_2\mathbf{H}\mathbf{D}_1$.

Figure 5 shows results for the G50C dataset. In case of G50C dataset, for both kernels, all matrices from the family of structured spinners perform similarly to a random Gaussian matrix. $\mathbf{H}\mathbf{D}_3\mathbf{H}\mathbf{D}_2\mathbf{H}\mathbf{D}_1$ performs better than all other matrices for a wide range of sizes of random feature maps. In case of USPST dataset (see: Supplement), for both kernels, all matrices from the family of structured spinners again perform similarly to a random Gaussian matrix (except $\mathbf{G}_{circ}\mathbf{K}_2\mathbf{K}_1$ which gives relatively poor results) and $\mathbf{H}\mathbf{D}_3\mathbf{H}\mathbf{D}_2\mathbf{H}\mathbf{D}_1$ is giving the best results. Finally, the efficiency of structured spinners does not depend on the dataset.

Table 1 shows substantial speedups obtained by the structured spinner matrices. The speedups are computed as time($\mathbf{G}$)/time($\mathbf{T}$), where time($\mathbf{G}$) and time($\mathbf{T}$) are the runtimes for respectively a random Gaussian matrix and a structured spinner matrix.

5.3 Neural networks

Finally, we performed experiments with neural networks using two different network architectures. The first one is a fully-connected network with two fully connected layers (we call it MLP), where we refer to the size of the hidden layer as h, and the second one is a convolutional network with following architecture:

- Convolution layer with filter size 5×5, 4 feature maps + ReLU + Max Pooling (region 2×2 and step 2×2)
- Convolution layer with filter size 5×5, 6 feature maps + ReLU + Max Pooling (region 2×2 and step 2×2)
- Fully-connected layer (h outputs) + ReLU
- Fully-connected layer (10 outputs)
- LogSoftMax.

Experiments were performed on the MNIST data set. In both experiments, we re-parametrized each matrix of weights of fully connected layers with a structured

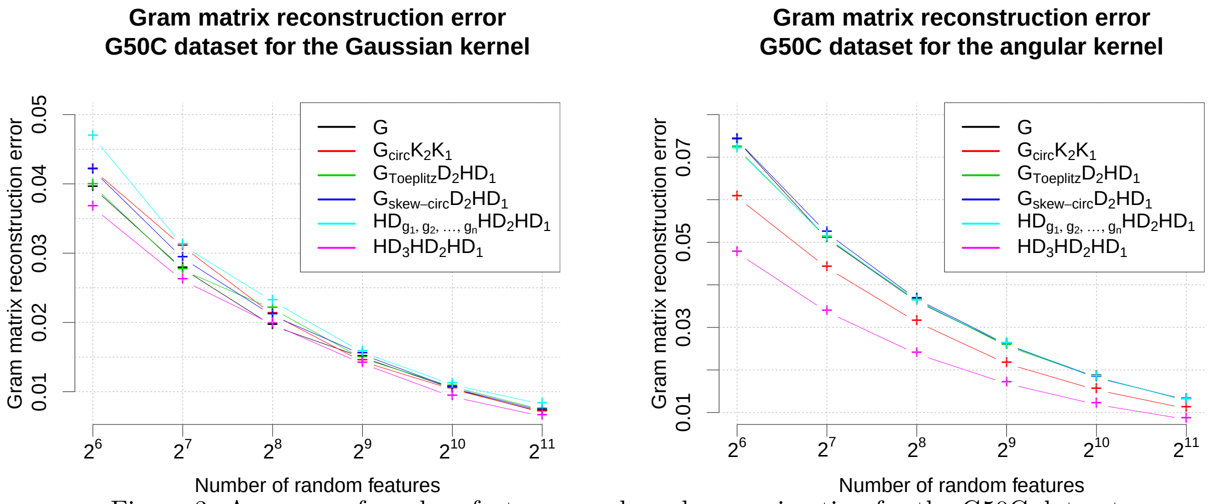

Figure 3: Accuracy of random feature map kernel approximation for the G50C dataset.

MATRIX DIM.	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
$\mathbf{G}_{Toeplitz}\mathbf{D}_2\mathbf{HD}_1$	x1.4	x3.4	x6.4	x12.9	x28.0	x42.3	x89.6
$\mathbf{G}_{skew-circ}\mathbf{D}_2\mathbf{HD}_1$	x1.5	x3.6	x6.8	x14.9	x31.2	x49.7	x96.5
$\mathbf{HD}_{g_1,\ldots,g_n}\mathbf{HD}_2\mathbf{HD}_1$	x2.3	x6.0	x13.8	x31.5	x75.7	x137.0	x308.8
$\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$	x2.2	x6.0	x14.1	x33.3	x74.3	x140.4	x316.8

Table 1: Speedups for Gaussian kernel approximation via structured spinners.

h	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}
unstructured	42.9	51.9	72.7	99.9	163.9	350.5	716.7	1271.5	2317.4
$\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$	109.2	121.3	109.7	114.2	117.4	123.9	130.6	214.3	389.8

Table 2: Running time (in $[\mu s]$) for the MLP - unstructured matrices vs structured spinners.

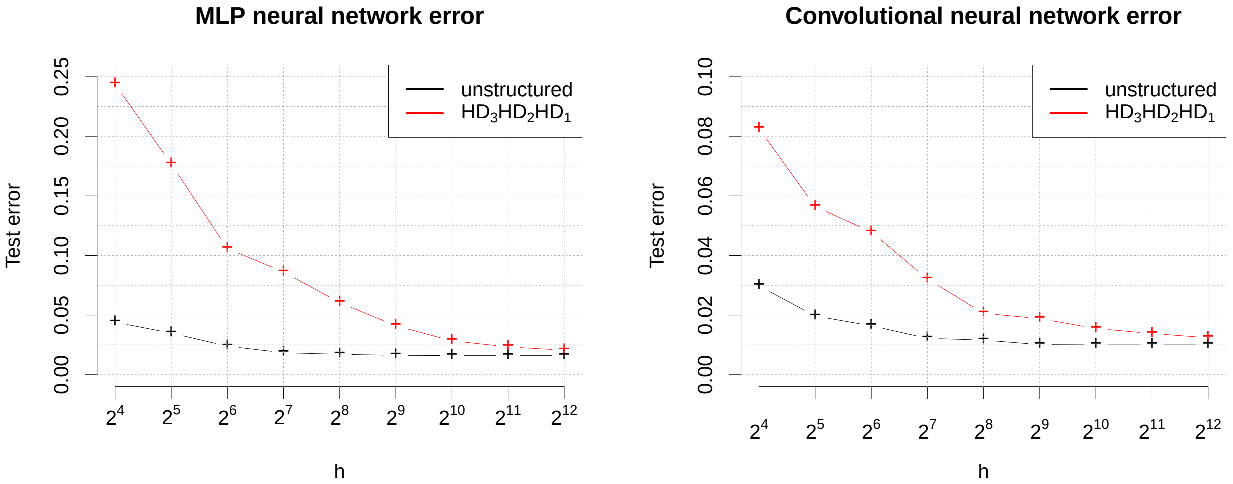

Figure 4: Test error for MLP (top) and convolutional network (bottom).

$\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$ matrix from a family of structured spinners. We compare this setting with the case where the unstructured parameter matrix is used. Note that in case when we use $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$ only linear number of parameters is learned (the Hadamard matrix is deterministic and even does not need to be explicitly stored, instead Walsh-Hadamard transform is used). Thus the network has significantly less parameters than in the unstructured case, e.g. for the MLP network we have $\mathcal{O}(h)$ instead of $\mathcal{O}(\text{input size} \times h)$ parameters.

In Figure 4 and Table 2 we compare respectively the test error and running time of the unstructured and structured approaches. Figure 4 shows that for large enough h, neural networks with structured spinners achieve similar performance to those with unstructured projections, while at the same time using structured spinners lead to significant computational savings as shown in Table 2. As mentioned before, the $\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$-neural network is a simpler construction than the Deep Friend Convnet, however one can replace it with any structured spinner to obtain compressed neural network architecture of a good capacity.

References

[Ailon and Chazelle, 2006] Ailon, N. and Chazelle, B. (2006). Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *STOC*.

[Ailon and Liberty, 2011] Ailon, N. and Liberty, E. (2011). An almost optimal unrestricted fast Johnson-Lindenstrauss transform. In *SODA*.

[Andoni et al., 2015] Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I. P., and Schmidt, L. (2015). Practical and optimal LSH for angular distance. In *NIPS*.

[Bentkus, 2003] Bentkus, V. (2003). On the dependence of the Berry–Esseen bound on dimension. *Journal of Statistical Planning and Inference*, 113(2):385–402.

[Brent et al., 2014] Brent, R. P., Osborn, J. H., and Smith, W. D. (2014). Bounds on determinants of perturbed diagonal matrices. *arXiv:1401.7084*.

[Charikar, 2002] Charikar, M. (2002). Similarity estimation techniques from rounding algorithms. In *STOC*.

[Cho and Saul, 2009] Cho, Y. and Saul, L. K. (2009). Kernel methods for deep learning. In *NIPS*.

[Choromanska et al., 2016] Choromanska, A., Choromanski, K., Bojarski, M., Jebara, T., Kumar, S., and LeCun, Y. (2016). Binary embeddings with structured hashed projections. In *ICML*.

[Choromanski and Sindhwani, 2016] Choromanski, K. and Sindhwani, V. (2016). Recycling randomness with structure for sublinear time kernel expansions. In *ICML*.

[Dai et al., 2014] Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M.-F., and Song, L. (2014). Scalable kernel methods via doubly stochastic gradients. In *NIPS*.

[Dasgupta et al., 2010] Dasgupta, A., Kumar, R., and Sarlos, T. (2010). A sparse johnson: Lindenstrauss transform.

[Dasgupta and Freund, 2008] Dasgupta, S. and Freund, Y. (2008). Random projection trees and low dimensional manifolds. In *STOC*.

[Denil et al., 2013] Denil, M., Shakibi, B., Dinh, L., Ranzato, M., and Freitas, N. D. (2013). Predicting parameters in deep learning. In *NIPS*.

[Feng et al., 2015] Feng, C., Hu, Q., and Liao, S. (2015). Random feature mapping with signed circulant matrix projection. In *IJCAI*.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning. Book in preparation for MIT Press.

[Har-Peled et al., 2012] Har-Peled, S., Indyk, P., and Motwani, R. (2012). Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of Computing*, 8(14):321–350.

[Hinrichs and Vybral, 2011] Hinrichs, A. and Vybral, J. (2011). Johnson-lindenstrauss lemma for circulant matrices. *Random Struct. Algorithms*, 39(3):391–398.

[Huang et al., 2014] Huang, P.-S., Avron, H., Sainath, T., Sindhwani, V., and Ramabhadran, B. (2014). Kernel methods match deep neural networks on timit. In *ICASSP*.

[Johnson and Lindenstrauss, 1984] Johnson, W. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability*, volume 26 of *Contemporary Mathematics*, pages 189–206.

[Le et al., 2013] Le, Q., Sarlós, T., and Smola, A. (2013). Fastfood-computing hilbert space expansions in loglinear time. In *ICML*.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[Li, 2013] Li, J. (2013). Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*.

[Liberty et al., 2008] Liberty, E., Ailon, N., and Singer, A. (2008). Dense fast random projections and lean Walsh transforms. In *RANDOM*.

[Mairal et al., 2014] Mairal, J., Koniusz, P., Harchaoui, Z., and Schmid, C. (2014). Convolutional kernel networks. In *NIPS*.

[Pilanci and Wainwright, 2014] Pilanci, M. and Wainwright, M. J. (2014). Randomized sketches of convex programs with sharp guarantees. In *ISIT*.

[Pilanci and Wainwright, 2015] Pilanci, M. and Wainwright, M. J. (2015). Newton sketch: A linear-time optimization algorithm with linear-quadratic convergence. *CoRR*, abs/1505.02250.

[Rahimi and Recht, 2007] Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In *NIPS*.

[Sainath et al., 2013] Sainath, T. N., Kingsbury, B., Sindhwani, V., Arisoy, E., and Ramabhadran, B. (2013). Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *ICASSP*.

[Sindhwani et al., 2015] Sindhwani, V., Sainath, T. N., and Kumar, S. (2015). Structured transforms for small-footprint deep learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3088–3096.

[Terasawa and Tanaka, 2007] Terasawa, K. and Tanaka, Y. (2007). Spherical LSH for approximate nearest neighbor search on unit hypersphere. In *WADS*.

[Vybíral, 2011] Vybíral, J. (2011). A variant of the Johnson-Lindenstrauss lemma for circulant matrices. *Journal of Functional Analysis*, 260(4):1096–1105.

[Yang et al., 2015] Yang, Z., Moczulski, M., Denil, M., de Freitas, N., Smola, A., Song, L., and Wang, Z. (2015). Deep fried convnets. In *ICCV*.

Structured adaptive and random spinners for fast machine learning computations (Supplementary Material)

In the Supplementary material we prove all theorems presented in the main body of the paper.

5.4 Structured machine learning algorithms with *Structured Spinners*

We prove now Lemma 1, Remark 1, as well as Theorem 1 and Theorem 3.

5.4.1 Proof of Remark 1

This result first appeared in [Ailon and Chazelle, 2006]. The following proof was given in [Choromanski and Sindhwani, 2016], we repeat it here for completeness. We will use the following standard concentration result.

Lemma 2 *(Azuma's Inequality) Let $X_1, ..., X_n$ be a martingale and assume that $-\alpha_i \leq X_i \leq \beta_i$ for some positive constants $\alpha_1, ..., \alpha_n, \beta_1, ..., \beta_n$. Denote $X = \sum_{i=1}^{n} X_i$. Then the following is true:*

$$\mathbb{P}[|X - \mathbb{E}[X]| > a] \leq 2e^{-\frac{a^2}{2\sum_{i=1}^{n}(\alpha_i+\beta_i)^2}} \tag{4}$$

Proof: Denote by $\tilde{\mathbf{x}}^j$ an image of $\mathbf{x}^j$ under transformation $\mathbf{HD}$. Note that the i^{th} dimension of $\tilde{\mathbf{x}}^j$ is given by the formula: $\tilde{x}_i^j = h_{i,1}x_1^j + ... + h_{i,n}x^{j,n}$, where $h_{l,u}$ stands for the l^{th} element of the u^{th} column of the randomized Hadamard matrix $\mathbf{HD}$. First, we use Azuma's Inequality to find an upper bound on the probability that $|\tilde{x}_i^j| > a$, where $a = \frac{\log(n)}{\sqrt{n}}$. By Azuma's Inequality, we have:

$$\mathbb{P}[|h_{i,1}x_1^j + ... + h_{i,n}x^{j,n}| \geq a] \leq 2e^{-\frac{\log^2(n)}{8}}. \tag{5}$$

We use: $\alpha_i = \beta_i = \frac{1}{\sqrt{n}}$. Now we take the union bound over all n dimensions and the proof is completed. □

5.4.2 *Structured Spinners*-equivalent definition

We will introduce here an equivalent definition of the model of structured spinners that is more technical (thus we did not give it in the main body of the paper), yet more convenient to work with in the proofs.

Note that from the definition of structured spinners we can conclude that each structured matrix $\mathbf{G}_{struct} \in \mathbb{R}^{n \times n}$ from the family of structured spinners is a product of three main structured blocks, i.e.:

$$\mathbf{G}_{struct} = \mathbf{B}_3\mathbf{B}_2\mathbf{B}_1, \tag{6}$$

where matrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ satisfy two conditions that we give below.

Condition 1: Matrices: $\mathbf{B}_1$ and $\mathbf{B}_2\mathbf{B}_1$ are $(\delta(n), p(n))$-balanced isometries.
Condition 2: Pair of matrices $(\mathbf{B}_2, \mathbf{B}_3)$ is $(K, \Lambda_F, \Lambda_2)$-random.

Below we give the definition of $(K, \Lambda_F, \Lambda_2)$-randomness.

Definition 4 (**$(K, \Lambda_F, \Lambda_2)$-randomness**) *A pair of matrices $(\boldsymbol{Y}, \boldsymbol{Z}) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$ is $(K, \Lambda_F, \Lambda_2)$-random if there exists $\boldsymbol{r} \in \mathbb{R}^k$, and a set of linear isometries $\phi = \{\phi_1, ..., \phi_n\}$, where $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}^k$, such that:*

- *$\boldsymbol{r}$ is either a ± 1-vector with i.i.d. entries or Gaussian with identity covariance matrix,*
- *for every $\boldsymbol{x} \in \mathbb{R}^n$ the j^{th} element $(\boldsymbol{Zx})_j$ of $\boldsymbol{Zx}$ is of the form: $\boldsymbol{r}^T \cdot \phi_j(\boldsymbol{x})$,*
- *there exists a set of i.i.d. sub-Gaussian random variables $\{\rho_1, ..., \rho_n\}$ with sub-Gaussian norm at most K, mean 0, the same second moments and a (Λ_F, Λ_2)-smooth set of matrices $\{\boldsymbol{W}^i\}_{i=1,...,n}$ such that for every $\boldsymbol{x} = (x_1, ..., x_n)^T$, we have: $\phi_i(\boldsymbol{Yx}) = \boldsymbol{W}^i(\rho_1 x_1, ..., \rho_n x_n)^T$.*

5.4.3 Proof of Lemma 1

Proof: Let us first assume the $\mathbf{G}_{circ}\mathbf{D}_2\mathbf{HD}_1$-setting (analysis for Toeplitz Gaussian or Hankel Gaussian is completely analogous). In that setting, it is easy to see that one can take $\mathbf{r}$ to be a Gaussian vector (this vector corresponds to the first row of $\mathbf{G}_{circ}$). Furthermore linear mappings ϕ_i are defined as: $\phi_i((x_0, x_1, ..., x_{n-1})^T) = (x_{n-i}, x_{n-i+1}, ..., x_{i-1})^T$, where operations on indices are modulo n. The value of $\delta(n)$ and $p(n)$ come from the fact that matrix $\mathbf{HD}_1$ is used as a $(\delta(n), p(n))$-balanced matrix and from Remark 1. In that setting, sequence $(\rho_1, ..., \rho_n)$ is discrete and corresponds to the diagonal of $\mathbf{D}_2$. Thus we have: $K = 1$. To calculate Λ_F and Λ_2, note first that matrix $\mathbf{W}^1$ is defined as $\mathbf{I}$ and subsequent $\mathbf{W}^i$s are given as circulant shifts of the previous ones (i.e. each row is a circulant shift of the previous row). That observation comes directly from the circulant structure of $\mathbf{G}_{circ}$. Thus we have: $\Lambda_F = O(\sqrt{n})$ and $\Lambda_2 = O(1)$. The former is true since each $\mathbf{A}^{i,j}$ has $O(n)$ nonzero entries and these are all 1s. The latter is true since each nontrivial $\mathbf{A}^{i,j}$ in that setting is an isometry (this is straightforward from the definition of $\{\mathbf{W}^i\}_{i=1,...,n}$). Finally, all other conditions regarding $\mathbf{W}^i$-matrices are clearly satisfied (each column of each $\mathbf{W}^i$ has unit L_2 norm and corresponding columns from different $\mathbf{W}^i$ and $\mathbf{W}^j$ are clearly orthogonal).

Now let us consider the setting, where the structured matrix is of the form: $\sqrt{n}\mathbf{HD}_3\mathbf{HD}_2\mathbf{HD}_1$. In that case, $\mathbf{r}$ corresponds to a discrete vector (namely, the diagonal of $\mathbf{D}_3$). Linear mappings ϕ_i are defined as: $\phi_i((x_1,...,x_n)^T) = (\sqrt{n}h_{i,1}x_1,...,\sqrt{n}h_{i,n}x_n)^T$, where $(h_{i,1},...,h_{i,n})^T$ is the i^{th} row of $\mathbf{H}$. One can also notice that the set $\{\mathbf{W}^i\}_{i=1,...,n}$ is defined as: $w^i_{a,b} = \sqrt{n}h_{i,a}h_{a,b}$. Let us first compute the Frobenius norm of the matrix $\mathbf{A}^{i,j}$, defined based on the aforementioned sequence $\{\mathbf{W}^i\}_{i=1,...,n}$. We have:

$$\begin{aligned}\|\mathbf{A}^{i,j}\|_F^2 &= \sum_{l,t\in\{1,...,n\}}(\sum_{k=1}^{n} w^j_{k,l}w^i_{k,t})^2 \\ &= n^2\sum_{l,t\in\{1,...,n\}}(\sum_{k=1}^{n} h_{j,k}h_{k,l}h_{i,k}h_{k,t})^2 \qquad (7)\end{aligned}$$

To compute the expression above, note first that for $r_1 \neq r_2$ we have:

$$\begin{aligned}\theta &= \sum_{k,l} h_{r_1,k}h_{r_1,l}h_{r_2,k}h_{r_2,l} \\ &= \sum_k h_{r_1,k}h_{r_2,k}\sum_l h_{r_1,l}h_{r_2,l} = 0, \qquad (8)\end{aligned}$$

where the last equality comes from fact that different rows of H are orthogonal. From the fact that $\theta = 0$ we get:

$$\begin{aligned}\|\mathbf{A}^{i,j}\|_F^2 &= n^2\sum_{r=1,...,n}\sum_{k,l} h^2_{i,r}h^2_{j,r}h^2_{r,k}h^2_{r,l} \\ &= n\cdot n^2(\frac{1}{\sqrt{n}})^8\cdot n^2 = n. \qquad (9)\end{aligned}$$

Thus we have: $\Lambda_F \leq \sqrt{n}$.

Now we compute $\|\mathbf{A}^{i,j}\|_2$. Notice that from the definition of $\mathbf{A}^{i,j}$ we get that

$$\mathbf{A}^{i,j} = \mathbf{E}^{i,j}\mathbf{F}^{i,j}, \qquad (10)$$

where the l^{th} row of $\mathbf{E}^{i,j}$ is of the form $(h_{j,1}h_{1,l},...,h_{j,n}h_{n,l})$ and the t^{th} column of $\mathbf{F}^{i,j}$ is of the form $(h_{i,1}h_{1,t},...,h_{i,n}h_{n,t})^T$. Thus one can easily verify that $\mathbf{E}^{i,j}$ and $\mathbf{H}^{i,j}$ are isometries (since $\mathbf{H}$ is) thus $\mathbf{A}^{i,j}$ is also an isometry and therefore $\Lambda_2 = 1$. As in the previous setting, remaining conditions regarding matrices $\mathbf{W}^i$ are trivially satisfied (from the basic properties of Hadamard matrices). That completes the proof. □

5.4.4 Proof of Theorem 1

Let us briefly give an overview of the proof before presenting it in detail. Challenges regarding proving accuracy results for structured matrices come from the fact that, for any given $\mathbf{x}\in\mathbb{R}^n$, different dimensions of $\mathbf{y} = \mathbf{G}_{struct}\mathbf{x}$ are no longer independent (as it is the case for the unstructured setting). For matrices from the family of structured spinners we can, however, show that with high probability different elements of $\mathbf{y}$ correspond to projections of a given vector $\mathbf{r}$ (see Section 3) into directions that are close to orthogonal. The "close-to-orthogonality" characteristic is obtained with the use of the Hanson-Wright inequality that focuses on concentration results regarding quadratic forms involving vectors of sub-Gaussian random variables. If $\mathbf{r}$ is Gaussian, then from the well-known fact that projections of the Gaussian vector into orthogonal directions are independent, we can conclude that dimensions of $\mathbf{y}$ are "close to independent". If $\mathbf{r}$ is a discrete vector then we need to show that for n large enough, it "resembles" the Gaussian vector. This is where we need to apply the aforementioned techniques regarding multivariate Berry-Esseen-type central limit theorem results.

Proof: We will use notation from Section 3 and previous sections of the Supplement. We assume that the model with structured matrices stacked vertically, each of m rows, is applied. Without loss of generality, we can assume that we have just one block since different blocks are chosen independently. Let $\mathbf{G}_{struct}$ be a matrix from the family of structured spinners. Let us assume that $\mathbf{G}_{struct}$ is used by a function f operating in the d-dimensional space and let us denote by $\mathbf{x}^1,\ldots,\mathbf{x}^d$ some fixed orthonormal basis of that space. Our first goal is to compute: $\mathbf{y}^1 = \mathbf{G}_{struct}\mathbf{x}^1,...,\mathbf{y}^d = \mathbf{G}_{struct}\mathbf{x}^d$. Denote by $\tilde{\mathbf{x}}^i$ the linearly transformed version of $\mathbf{x}$ after applying block $\mathbf{B}_1$, i.e. $\tilde{\mathbf{x}}^i = \mathbf{B}_1\mathbf{x}^i$. Since $\mathbf{B}_1$ is $(\delta(n),p(n))$-balanced), we conclude that with probability at least: $p_{balanced} \geq 1 - dp(n)$ each element of each $\tilde{\mathbf{x}}^i$ has absolute value at most $\frac{\delta(n)}{\sqrt{n}}$. We shortly say that each $\tilde{\mathbf{x}}^i$ is $\delta(n)$-balanced. We call this event $\mathcal{E}_{balanced}$.

Note that by the definition of structured spinners, each $\mathbf{y}^i$ is of the form:

$$\mathbf{y}^i = (\mathbf{r}^T\cdot\phi_1(\mathbf{B}_2\tilde{\mathbf{x}}^i),...,\mathbf{r}^T\cdot\phi_m(\mathbf{B}_2\tilde{\mathbf{x}}^i))^T. \qquad (11)$$

For clarity and to reduce notation, we will assume that $\mathbf{r}$ is n-dimensional. To obtain results for vectors $\mathbf{r}$ of different dimensionality D, it suffices to replace in our analysis and theoretical statements n by D. Let us denote $\mathcal{A} = \{\phi_1(\mathbf{B}_2\tilde{\mathbf{x}}^1),...,\phi_m(\mathbf{B}_2\tilde{\mathbf{x}}^1),...,\phi_1(\mathbf{B}_2\tilde{\mathbf{x}}^d),...,\phi_m(\mathbf{B}_2\tilde{\mathbf{x}}^d))\}$. Our goal is to show that with high probability (in respect to random choices of $\mathbf{B}_1$ and $\mathbf{B}_2$) for all $\mathbf{v}^i,\mathbf{v}^j\in\mathcal{A}$, $i\neq j$ the following is true:

$$|(\mathbf{v}^i)^T\cdot\mathbf{v}^j| \leq t \qquad (12)$$

for some given $0 < t \ll 1$.

Fix some $t > 0$. We would like to compute the lower bound on the corresponding probability. Let us fix two vectors $\mathbf{v}^1, \mathbf{v}^2 \in \mathcal{A}$ and denote them as: $\mathbf{v}^1 = \phi_i(\mathbf{B}_2\mathbf{x})$, $\mathbf{v}^2 = \phi_j(\mathbf{B}_2\mathbf{y})$ for some $\mathbf{x} = (x_1, ..., x_n)^T$ and $\mathbf{y} = (y_1, ..., y_n)^T$. Note that we have (see denotation from Section 3):

$$\phi_i(\mathbf{B}_2\mathbf{x}) = (w^i_{11}\rho_1 x_1 + ... \\ + w^i_{1,n}\rho_n x_n, ..., w^i_{n,1}\rho_1 x_1 + ... + w^i_{n,n}\rho_n x_n)^T \quad (13)$$

and

$$\phi_j(\mathbf{B}_2\mathbf{y}) = (w^j_{11}\rho_1 y_1 + ... + w^j_{1,n}\rho_n y_n, ..., \\ w^j_{n,1}\rho_1 y_1 + ... + w^j_{n,n}\rho_n y_n)^T. \quad (14)$$

We obtain:

$$(\mathbf{v}^1)^T \cdot \mathbf{v}^2 = \sum_{l \in \{1,...,n\}, u \in \{1,...,n\}} \rho_l \rho_u (\sum_{k=1}^{n} x_l y_u w^i_{k,u} w^j_{k,l}). \quad (15)$$

We now show that, under assumptions from Theorem 1, the expected value of the expression above is 0. We have:

$$\mathbb{E}[(\mathbf{v}^1)^T \cdot \mathbf{v}^2] = \mathbb{E}[\sum_{l \in \{1,...,n\}} \rho_l^2 x_l y_l (\sum_{k=1}^{n} w^i_{k,l} w^j_{k,l})], \quad (16)$$

since $\rho_1, ..., \rho_n$ are independent and have expectations equal to 0. Now notice that if $i \neq j$ then from the assumption that corresponding columns of matrices $\mathbf{W}^i$ and $\mathbf{W}^j$ are orthogonal, we get that the above expectation is 0. Now assume that $i = j$. But then $\mathbf{x}$ and $\mathbf{y}$ have to be different and thus they are orthogonal (since they are taken from the orthonormal system transformed by an isometry). In that setting we get:

$$\mathbb{E}[(\mathbf{v}^1)^T \cdot \mathbf{v}^2] = \mathbb{E}[\sum_{l \in \{1,...,n\}} \rho_l^2 x_l y_l (\sum_{k=1}^{n} (w^i_{k,l})^2)] \\ = \tau w \sum_{l=1}^{n} x_l y_l = 0, \quad (17)$$

where τ stands for the second moment of each ρ_i, w is the squared L_2-norm of each column of $\mathbf{W}^i$ (τ and w are well defined due to the properties of structured spinners). The last inequality comes from the fact that $\mathbf{x}$ and $\mathbf{y}$ are orthogonal. Now if we define matrices $\mathbf{A}^{i,j}$ as in the definition of the model of structured spinners then we see that

$$(\mathbf{v}^1)^T \cdot \mathbf{v}^2 = \sum_{l,u \in \{1,...,n\}} \rho_l \rho_u T^{i,j}_{l,u}, \quad (18)$$

where: $T^{i,j}_{l,u} = x_l y_u A^{i,j}_{l,u}$.

Now we will use the following inequality:

Theorem 6 (Hanson-Wright Inequality) *Let $\boldsymbol{X} = (X_1, ..., X_n)^T \in \mathbb{R}^n$ be a random vector with independent components X_i which satisfy: $\mathbb{E}[X_i] = 0$ and have sub-Gaussian norm at most K for some given $K > 0$. Let $\boldsymbol{A}$ be an $n \times n$ matrix. Then for every $t \geq 0$ the following is true:*

$$\mathbb{P}[\boldsymbol{X}^T \boldsymbol{A} \boldsymbol{X} - \mathbb{E}[\boldsymbol{X}^T \boldsymbol{A} \boldsymbol{X}] > t] \\ \leq 2e^{-c\min(\frac{t^2}{K^4\|\boldsymbol{A}\|_F^2}, \frac{t}{K^2\|\boldsymbol{A}\|_2})}, \quad (19)$$

where c is some universal positive constant.

Note that, assuming $\delta(n)$-balancedness, we have: $\|\mathbf{T}^{i,j}\|_F \leq \frac{\delta^2(n)}{n}\|\mathbf{A}^{i,j}\|_F$ and $\|\mathbf{T}^{i,j}\|_2 \leq \frac{\delta^2(n)}{n}\|\mathbf{A}^{i,j}\|_2$.

Now we take $\mathbf{X} = (\rho_1, ..., \rho_n)^T$ and $\mathbf{A} = \mathbf{T}^{i,j}$ in the theorem above. Applying the Hanson-Wright inequality in that setting, taking the union bound over all pairs of different vectors $\mathbf{v}^i, \mathbf{v}^j \in \mathcal{A}$ (this number is exactly: $\binom{md}{2}$)) and the event $\mathcal{E}_{balanced}$, finally taking the union bound over all s functions f_i, we conclude that with probability at least:

$$p_{good} = 1 - p(n)ds \\ - 2\binom{md}{2} se^{-\Omega(\min(\frac{t^2n^2}{K^4\Lambda_F^2\delta^4(n)}, \frac{tn}{K^2\Lambda_2\delta^2(n)}))} \quad (20)$$

for every f any two different vectors $\mathbf{v}^i, \mathbf{v}^j \in \mathcal{A}$ satisfy: $|(\mathbf{v}^i)^T \cdot \mathbf{v}^j| \leq t$.

Note that from the fact that $\mathbf{B}_2\mathbf{B}_1$ is $(\delta(n), p(n))$-balanced and from Equation 20, we get that with probability at least:

$$p_{right} = 1 - 2p(n)ds \\ - 2\binom{md}{2} se^{-\Omega(\min(\frac{t^2n^2}{K^4\Lambda_F^2\delta^4(n)}, \frac{tn}{K^2\Lambda_2\delta^2(n)}))}. \quad (21)$$

for every f any two different vectors $\mathbf{v}^i, \mathbf{v}^j \in \mathcal{A}$ satisfy: $|(\mathbf{v}^i)^T \cdot \mathbf{v}^j| \leq t$ and furthermore each $\mathbf{v}^i$ is $\delta(n)$-balanced.

Assume now that this event happens. Consider the vector

$$\mathbf{q}' = ((\mathbf{y}^1)^T, ..., (\mathbf{y}^d)^T)^T \in \mathbb{R}^{md}. \quad (22)$$

Note that $\mathbf{q}'$ can be equivalently represented as:

$$\mathbf{q}' = (\mathbf{r}^T \cdot \mathbf{v}^1, ..., \mathbf{r}^T \cdot \mathbf{v}^{md}), \quad (23)$$

where: $\mathcal{A} = \{\mathbf{v}^1, ..., \mathbf{v}^{md}\}$. From the fact that $\phi_i\mathbf{B}_2$ and $\mathbf{B}_1$ are isometries we conclude that: $\|\mathbf{v}^i\|_2 = 1$ for $i = 1, ...$

Now we will need the following Berry-Esseen type result for random vectors:

Theorem 7 (Bentkus [Bentkus, 2003]) *Let $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ be independent vectors taken from $\mathbb{R}^k$ with common mean $\mathbb{E}[\boldsymbol{X}_i] = 0$. Let $\boldsymbol{S} = \boldsymbol{X}_1 + ... + \boldsymbol{X}_n$. Assume that the covariance operator $\boldsymbol{C}^2 = cov(\boldsymbol{S})$ is invertible. Denote $\beta_i = \mathbb{E}[\|\boldsymbol{C}^{-1}\boldsymbol{X}_i\|_2^3]$ and $\beta = \beta_1 + ... + \beta_n$. Let $\mathcal{C}$ be the set of all convex subsets of $\mathbb{R}^k$. Denote $\Delta(\mathcal{C}) = \sup_{A \in \mathcal{C}} |\mathbb{P}[S \in A] - \mathbb{P}[Z \in A]|$, where Z is the multivariate Gaussian distribution with mean 0 and covariance operator $\boldsymbol{C}^2$. Then:*

$$\Delta(\mathcal{C}) \leq ck^{\frac{1}{4}}\beta \tag{24}$$

for some universal constant c.

Denote: $\mathbf{X}_i = (r_i v_i^1, ..., r_i v_i^k)^T$ for $k = md$, $\mathbf{r} = (r_1, ..., r_n)^T$ and $\mathbf{v}^j = (v_1^j, ..., v_n^j)$. Note that $\mathbf{q}' = \mathbf{X}_1 + ... + \mathbf{X}_n$. Clearly we have: $\mathbb{E}[\mathbf{X}_i] = 0$ (the expectation is taken with respect to the random choice of $\mathbf{r}$). Furthermore, given the choices of $\mathbf{v}^1, ..., \mathbf{v}^k$, random vectors $\mathbf{X}_1, .., \mathbf{X}_n$ are independent.

Let us calculate now the covariance matrix of $\mathbf{q}'$. We have:

$$\mathbf{q}'_i = r_1 v_1^i + ... + r_n v_n^i, \tag{25}$$

where: $\mathbf{q}' = (\mathbf{q}'_1, ..., \mathbf{q}'_k)$.

Thus for i_1, i_2 we have:

$$\mathbb{E}[\mathbf{q}'_{i_1}\mathbf{q}'_{i_2}] = \sum_{j=1}^{n} v_j^{i_1} v_j^{i_2} \mathbb{E}[r_j^2] + 2 \sum_{1 \leq j_1 < j_2 \leq n} v_{j_1}^{i_1} v_{j_2}^{i_2} \mathbb{E}[r_{j_1} r_{j_2}]$$
$$= (\mathbf{v}^{i_1})^T \cdot \mathbf{v}^{i_2}, \tag{26}$$

where the last equation comes from the fact r_j are either Gaussian from $\mathcal{N}(0,1)$ or discrete with entries from $\{-1, +1\}$ and furthermore different r_js are independent.

Therefore if $i_1 = i_2 = i$, since each $\mathbf{v}^i$ has unit L_2-norm, we have that

$$\mathbb{E}[\mathbf{q}'_i \mathbf{q}'_i] = 1, \tag{27}$$

and for $i_1 \neq i_2$ we get:

$$|\mathbb{E}[\mathbf{q}'_{i_1}\mathbf{q}'_{i_2}]| \leq t. \tag{28}$$

We conclude that the covariance matrix $\Sigma_{\mathbf{q}'}$ of the distribution $\mathbf{q}'$ is a matrix with entries 1 on the diagonal and other entries of absolute value at most t.

For $t = o_k(1)$ small enough and from the $\delta(n)$-balancedness of vectors $\mathbf{v}^1, ..., \mathbf{v}^k$ we can conclude that:

$$\mathbb{E}[\|\mathbf{C}^{-1}\mathbf{X}_i\|_2^3] = O(\mathbb{E}[\|\mathbf{X}_i\|_2^3]) = O(\sqrt{(\frac{k}{n})^3}\delta^3(n)), \tag{29}$$

Now, using Theorem 7, we conclude that

$$\sup_{A \in \mathcal{C}} |\mathbb{P}[\mathbf{q}' \in A] - \mathbb{P}[Z \in A]| = O(k^{\frac{1}{4}} n \cdot \frac{k^{\frac{3}{2}}}{n^{\frac{3}{2}}}\delta^3(n))$$
$$= O(\frac{\delta^3(n)}{\sqrt{n}}k^{\frac{7}{4}}), \tag{30}$$

where Z is taken from the multivariate Gaussian distribution with covariance matrix $\mathbf{I} + \mathbf{E}$ and $\mathcal{C}$ is the set of all convex sets. Now if we apply the above inequality to the pairwise disjoint convex sets $A_1, ..., A_j$, where $A_1 \cup ... \cup A_j = f_i^{-1}(\mathcal{S})$ and $l \leq b$ (such sets exist form the b-convexity of $f_i^{-1}(\mathcal{S})$), take $\eta = \frac{\delta^3(n)}{\sqrt{n}}k^{\frac{7}{4}}$, $\epsilon = t = o_{md}(1)$ and take n large enough, the statement of the theorem follows. □

5.4.5 Proof of Theorem 2

Proof: Let us assume that f_i is a convex function of $\mathbf{q}_{f_i}$ (if f_i is concave then the proof completely analogous). For any $t \in \mathbb{R}$ let $\mathcal{S}_t = \{\mathbf{q}_{f_i} : f_i(\mathbf{q}_{f_i}) \leq t\}$ for f_i and $\mathcal{S}_t = \{\mathbf{q}_{f'_i} : f'_i(\mathbf{q}_{f'_i}) \leq t\}$ for f'_i. From the convexity assumption we get that $\mathcal{S}_t$ is a convex set. Thus we can directly apply Theorem 1 and the result regarding cdf functions follows. To obtain the result regarding the characteristic functions, notice first that we have:

$$\phi_X(t) = \int_{-1}^{1} \mathbb{P}[cos(tX) > s]ds + i \int_{-1}^{1} \mathbb{P}[sin(tX) > s]ds \tag{31}$$

The event $\{cos(tX) > s\}$ for $t \neq 0$ is equivalent to: $X \in \cup_{I \in \mathcal{I}} I$ for some family of intervals $\mathcal{I}$. Similar observation is true for the event $\{sin(tX) > s\}$.

In our scenario, from the fact that f_i is bounded, we conclude that the corresponding families $\mathcal{I}$ are finite. Furthermore, the probability of belonging to a particular interval can be expressed by the values of the cdf function in the endpoints of that interval. From this observation and the result on cdfs that we have just obtained, the result for the characteristic functions follows immediately. □

5.4.6 Proof of Theorem 3

Proof: This comes directly from Theorem 1 and Lemma 1. □

5.4.7 Proof of Theorem 4

Proof: For clarity we will assume that the structured matrix consists of just one block of m rows and will compare its performance with the unstructured variant of m rows (the more general case when the structured matrix is obtained by stacking vertically many blocks is analogous since the blocks are chosen independently).

Consider the two-dimensional linear space $\mathcal{H}$ spanned by $\mathbf{x}$ and $\mathbf{y}$. Fix some orthonormal basis $\mathcal{B} = \{\mathbf{u}^1, \mathbf{u}^2\}$ of $\mathcal{H}$. Take vectors $\mathbf{q}$ and $\mathbf{q}'$. Note that they are $2m$-dimensional, where m is the number of rows of the block used in the structured setting. From Theorem 3 we conclude that will probability at least $p_{success}$, where $p_{success}$ is as in the statement of the theorem the following holds for any convex $2m$-dimensional set

A:

$$|\mathbb{P}[\mathbf{q}(\epsilon) \in A] - \mathbb{P}[\mathbf{q}' \in A]| \leq \eta, \tag{32}$$

where $\eta = \frac{\log^3(n)}{n^{\frac{2}{5}}}$. Take two corresponding entries of vectors $\mathbf{v}^1_{\mathbf{x},\mathbf{y}}$ and $\mathbf{v}^2_{\mathbf{x},\mathbf{y}}$ indexed by a pair $(\mathbf{e}_i, \mathbf{e}_j)$ for some fixed $i, j \in \{1, ..., m\}$ (for the case when the pair is not of the form $(\mathbf{e}, \mathbf{e}_j)$, but of a general form: $(\pm\mathbf{e}_i, \pm\mathbf{e}_j)$ the analysis is exactly the same). Call them p^1 and p^2 respectively. Our goal is to compute $|p^1 - p^2|$. Notice that p^1 is the probability that $h(\mathbf{x}) = \mathbf{e}_i$ and $h(\mathbf{y}) = \mathbf{e}_j$ for the unstructured setting and p^2 is that probability for the structured variant.

Let us consider now the event $E^1 = \{h(\mathbf{x}) = \mathbf{e}_i \wedge h(\mathbf{y}) = \mathbf{e}_j\}$, where the setting is unstructured. Denote the corresponding event for the structured setting as E^2. Denote $\mathbf{q} = (q_1, ..., q_{2m})$. Assume that $\mathbf{x} = \alpha_1\mathbf{u}^1 + \alpha_2\mathbf{u}^2$ for some scalars $\alpha_1, \alpha_2 > 0$. Denote the unstructured Gaussian matrix by $\mathbf{G}$. We have:

$$\mathbf{G}\mathbf{x} = \alpha_1\mathbf{G}\mathbf{u}^1 + \alpha_2\mathbf{G}\mathbf{u}^2 \tag{33}$$

Note that we have: $\mathbf{G}\mathbf{u}^1 = (q_1, ..., q_m)^T$ and $\mathbf{G}\mathbf{u}^2 = (q_{m+1}, ..., q_{2m})^T$. Denote by $A(\mathbf{e}_i)$ the set of all the points in $\mathbb{R}^m$ such that their angular distance to $\mathbf{e}_i$ is at most the angular distance to all other $m-1$ canonical vectors. Note that this is definitely the convex set. Now denote:

$$\begin{aligned} Q(\mathbf{e}_i) = \{&(q_1, ..., q_{2m})^T \in \mathbb{R}^{2m} : \\ &\alpha_1(q_1, ..., q_m)^T + \alpha_2(q_{m+1}, ..., q_{2m})^T \in A(\mathbf{e}_i)\}. \end{aligned} \tag{34}$$

Note that since $A(\mathbf{e}_i)$ is convex, we can conclude that $Q(\mathbf{e}_i)$ is also convex. Note that

$$\{h(\mathbf{x}) = \mathbf{e}_i\} = \{\mathbf{q} \in Q(\mathbf{e}_i)\}. \tag{35}$$

By repeating the analysis for the event $\{h(\mathbf{y}) = \mathbf{e}_j\}$, we conclude that:

$$\{h(\mathbf{x}) = \mathbf{e}_i \wedge h(\mathbf{y}) = \mathbf{e}_j\} = \{\mathbf{q} \in Y(\mathbf{e}_i, \mathbf{e}_j)\} \tag{36}$$

for convex set $Y(\mathbf{e}_i, \mathbf{e}_j) = Q(\mathbf{e}_i) \cap Q(\mathbf{e}_j)$. Now observe that

$$|p^1 - p^2| = |\mathbb{P}[\mathbf{q} \in Y(\mathbf{e}_i, \mathbf{e}_j)] - \mathbb{P}[\mathbf{q}' \in Y(\mathbf{e}_i, \mathbf{e}_j)]| \tag{37}$$

Thus we have:

$$\begin{aligned} |p^1 - p^2| \leq\ &|\mathbb{P}[\mathbf{q} \in Y(\mathbf{e}_i, \mathbf{e}_j)] - \mathbb{P}[\mathbf{q}(\epsilon) \in Y(\mathbf{e}_i, \mathbf{e}_j)]| \\ &+ |\mathbb{P}[\mathbf{q}(\epsilon) \in Y(\mathbf{e}_i, \mathbf{e}_j)] - \mathbb{P}[\mathbf{q}' \in Y(\mathbf{e}_i, \mathbf{e}_j)]| \end{aligned} \tag{38}$$

Therefore we have:

$$|p^1 - p^2| \leq |\mathbb{P}[\mathbf{q} \in Y(\mathbf{e}_i, \mathbf{e}_j)] - \mathbb{P}[\mathbf{q}(\epsilon) \in Y(\mathbf{e}_i, \mathbf{e}_j)]| + \eta. \tag{39}$$

Thus we just need to upper-bound:

$$\xi = |\mathbb{P}[\mathbf{q} \in Y(\mathbf{e}_i, \mathbf{e}_j)] - \mathbb{P}[\mathbf{q}(\epsilon) \in Y(\mathbf{e}_i, \mathbf{e}_j)]|. \tag{40}$$

Denote the covariance matrix of the distribution $\mathbf{q}(\epsilon)$ as $\mathbf{I} + \mathbf{E}$. Note that $\mathbf{E}$ is equal to 0 on the diagonal and the absolute value of all other off-diagonal entries is at most ϵ.

Denote $k = 2m$. We have

$\xi = |A - B|$,

where $A = \frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{\det(I+E)}} \int_{Y(\mathbf{e}_i, \mathbf{e}_j)} e^{-\frac{\mathbf{x}^T(\mathbf{I}+\mathbf{E})^{-1}\mathbf{x}}{2}} d\mathbf{x}$

and $B = \frac{1}{(2\pi)^{\frac{k}{2}}} \int_{Y(\mathbf{e}_i, \mathbf{e}_j)} e^{-\frac{\mathbf{x}^T\mathbf{x}}{2}} d\mathbf{x}$.

Expanding: $(\mathbf{I} + \mathbf{E})^{-1} = \mathbf{I} - \mathbf{E} + \mathbf{E}^2 - ...$, noticing that $|\det(I + E) - 1| = O(\epsilon^{2m})$, and using the above formula, we easily get:

$$\xi = O(\epsilon). \tag{41}$$

That completes the proof. □

5.4.8 b-convexity for angular kernel approximation

Let us now consider the setting, where linear projections are used to approximate angular kernels between paris of vectors via random feature maps. In this case, the linear projection is followed by the pointwise nonlinear mapping, where the applied nonlinear mapping is a sign function. The angular kernel is retrieved from the Hamming distance between $\{-1, +1\}$-hashes obtained in such a way. Note that in this case we can assign to each pair $\mathbf{x}, \mathbf{y}$ of vectors from a database a function $f_{\mathbf{x},\mathbf{y}}$ that outputs the binary vector which length is the size of the hash and with these indices turned on for which the hashes of $\mathbf{x}$ and $\mathbf{y}$ disagree. Such a binary vector uniquely determines the Hadamard distance between the hashes. Notice that for a fixed-length hash $f_{\mathbf{x},\mathbf{y}}$ produces only finitely many outputs. If $\mathcal{S}$ is a set-singleton consisting of one of the possible outputs, then one can notice (straightforwardly from the way the hash is created) that $f^{-1}_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ is an intersection of the convex sets (as a function of $\mathbf{q}_{f_{\mathbf{x},\mathbf{y}}}$). Thus it is convex and thus for sets $\mathcal{S}$ which are singletons we can take $b = 1$.

5.4.9 Proof of Theorem 5

In this section, we show that by learning vector $\mathbf{r} \in \mathbb{R}^k$ from the definition above, one can approximate well any matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ learned by the neural network, providing that the size k or $\mathbf{r}$ is large enough in comparison with the number of projections and the intrinsic dimensionality d of the data $\mathcal{X}$.

Take the parametrized structured spinner matrix $\mathbf{M}_{struct} \in \mathbb{R}^{m \times n}$ with a learnable vector $\mathbf{r}$. Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ be a matrix learned in the unstructured setting.

Let $\mathcal{B} = \{\mathbf{x}^1, ..., \mathbf{x}^d\}$ be some orthonormal basis of the linear space, where data $\mathcal{X}$ is taken from.

Proof: Note that from the definition of the parametrized structured spinner model we can conclude that with probability at least $p_1 = 1 - p(n)$ with respect to the choices of $\mathbf{M}_1$ and $\mathbf{M}_2$ each $\mathbf{M}_{struct}\mathbf{x}^i$ is of the form:

$$\mathbf{M}_{struct}\mathbf{x}^i = (\mathbf{r}^T \cdot \mathbf{z}_1(\mathbf{q}^i), ..., \mathbf{r}^T \cdot \mathbf{z}_m(\mathbf{q}^i))^T, \qquad (42)$$

where each $\mathbf{z}_j(\mathbf{q}^i)$ is of the form:

$$\mathbf{z}_j(\mathbf{q}^i) = (w^j_{1,1}\rho_1 q^i_1 + w^j_{1,n}\rho_n q^i_n, ..., w^j_{k,1}\rho_1 q^i_1 + w^j_{k,n}\rho_n q^i_n)^T \qquad (43)$$

and $\mathcal{B}' = \{\mathbf{q}^1, ..., \mathbf{q}^d\}$ is an orthonormal basis such that: $\|\mathbf{q}^i\|_\infty \leq \frac{\delta(n)}{\sqrt{n}}$ for $i = 1, ..., n$.

Note that the system of equations:

$$\mathbf{M}^{struct}\mathbf{x}^i = \mathbf{M}\mathbf{x}^i \qquad (44)$$

for $i = 1, ..., d$ has the solution in $\mathbf{r}$ if the vectors from the set $\mathcal{A} = \{\mathbf{z}_j(\mathbf{q}^i) : j = 1, ..., m, i = 1, ...d\}$ are independent.

Construct a matrix $\mathbf{G} \in \mathbb{R}^{md \times k}$, where rows are vectors from $\mathcal{A}$. We want to show that $rank(\mathbf{G}) = md$. It suffices to show that $det(\mathbf{G}\mathbf{G}^T) \neq 0$. Denote $\mathbf{B} = \mathbf{G}\mathbf{G}^T$. Note that $B_{i,j} = (\mathbf{v}^i)^T\mathbf{v}^j$, where $\mathcal{A} = \{\mathbf{v}^1, ..., \mathbf{v}^{md}\}$. Take two vectors $\mathbf{v}^a, \mathbf{v}^b \in \mathcal{A}$. Note that from the definition of $\mathcal{A}$ we get:

$$(\mathbf{v}^a)^T\mathbf{v}^b = \sum_{l \in \{1,...,n\}, u \in \{1,...,n\}} \rho_l \rho_u x_l y_u (\sum_{s=1}^{k} w^i_{s,l} w^j_{s,u}) \qquad (45)$$

for some i, j and some vectors $\mathbf{x} = (x_1, ..., x_n)^T$, $\mathbf{y} = (y_1, ..., y_n)^T$. Furthermore,

- $i = j$ and $\mathbf{x} = \mathbf{y}$ if $a = b$,
- $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = 1$,
- $\mathbf{x}^T\mathbf{y} = 0$ or $\mathbf{x} = \mathbf{y}$ and $i \neq j$ for $a \neq b$.

We also have:

$$\mathbb{E}[(\mathbf{v}^a)^T\mathbf{v}^b] = \mathbb{E}[\sum_{l \in \{1,...,n\}} \rho_l^2 x_l y_l (\sum_{s=1}^{k} w^i_{s,l} w^j_{s,u})]. \qquad (46)$$

From the previous observations and the properties of matrices $\mathbf{W}^1, ..., \mathbf{W}^n$ we conclude that the entries of the diagonal of $\mathbf{B}$ are equal to 1. Furthermore, all other entries are 0 on expectation. Using Hanson-Wright inequality, we conclude that for any $t > 0$ we have: $|B_{i,j}| \leq t$ for all $i \neq j$ with probability at least:

$$p_{succ} = 1 - 2p(n)d - 2\binom{md}{2} e^{-c\min(\frac{t^2n^2}{K^4\Lambda_F^2\delta^4(n)}, \frac{tn}{K^2\Lambda_2\delta^2(n)})}.$$

If this is the case, we let $\tilde{\mathbf{B}} \in \mathbb{R}^{(md)\times(md)}$ be a matrix with diagonal entries $\tilde{\mathbf{B}}_{i,i} = 0$ and off-diagonal entries $\tilde{\mathbf{B}}_{\mathbf{i},\mathbf{j}} = -\mathbf{B}_{i,j}$. Furthermore, let $\mathbf{B}^* \in \mathbb{R}^{(md)\times(md)}$ be a matrix with diagonal entries $\mathbf{B}^*_{i,i} = 0$ and off-diagonal entries $\mathbf{B}^*_{i,j} = t$.

Following a similar argument as in [Brent et al., 2014], note that $\mathbf{B}^* = t(\mathbf{J} - \mathbf{I})$ where $\mathbf{J}$ is the matrix of all ones (thus of rank 1) and $\mathbf{I}$ is the identity matrix. Then the eigenvalues of $\mathbf{B}^*$ are $t(md - 1)$ with multiplicity 1 and $t(0 - 1)$ with multiplicity $(md - 1)$. We, thereby, are able to explicitly compute $\det(\mathbf{I} - \mathbf{B}^*) = (1 - t(md - 1))(1 + t)^{md-1}$.

If $\rho(\mathbf{B}^*) \leq 1$, we can apply Theorem 1 of [Brent et al., 2014] by replacing $\mathbf{F}$ with $\mathbf{B}^*$ and $\mathbf{E}$ with $\tilde{\mathbf{B}}$. For the convenience of the reader, we state their theorem here: Let $\mathbf{F} \in \mathbb{R}^{n\times n}$ with non-negative entries and $\rho(F) \leq 1$. Let $\mathbf{E} \in \mathbb{R}^{n\times n}$ with entries $| e_{i,j} | \leq f_{i,j}$, then $\det(\mathbf{I} - \mathbf{E}) \geq \det(\mathbf{I} - \mathbf{F})$.

That is: if $\rho(\mathbf{B}^*) \leq 1$, then

$$det(\mathbf{I} - \mathbf{B}^*) = (1 - t(md - 1))(1 + t)^{md-1} \leq det(\mathbf{I} - \tilde{\mathbf{B}}) = det(\mathbf{B}). \qquad (47)$$

The final step is to observe that: $\rho(\mathbf{B}^*) \leq 1 \iff \max\{| t(md - 1) |, | -t |\} = t(md - 1) \leq 1 \iff t \leq \frac{1}{md-1}$. Using this result, we, hence, see that $det(\mathbf{B}) \geq (1 - t(md - 1))(1 + t)^{md-1} \geq 0$, in particular $det(\mathbf{B}) > 0$ for $t = \frac{1}{md}$. That completes the proof. □

5.4.10 Additional experiments

This experiment focuses on the Newton sketch approach [Pilanci and Wainwright, 2015], a generic optimization framework. It guarantees super-linear convergence with exponentially high probability for self-concordant functions, and a reduced computational complexity compared to the original second-order Newton method. The method relies on using a sketched version of the Hessian matrix, in place of the original one. In the subsequent experiment we show that matrices from the family of strucured spinners can be used for this purpose, thus can speed up several convex optimization problems solvers.

We consider the unconstrained large scale logistic regression problem, i.e. given a set of n observations $\{(a_i, y_i)\}_{i=1..n}$, with $a_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, find

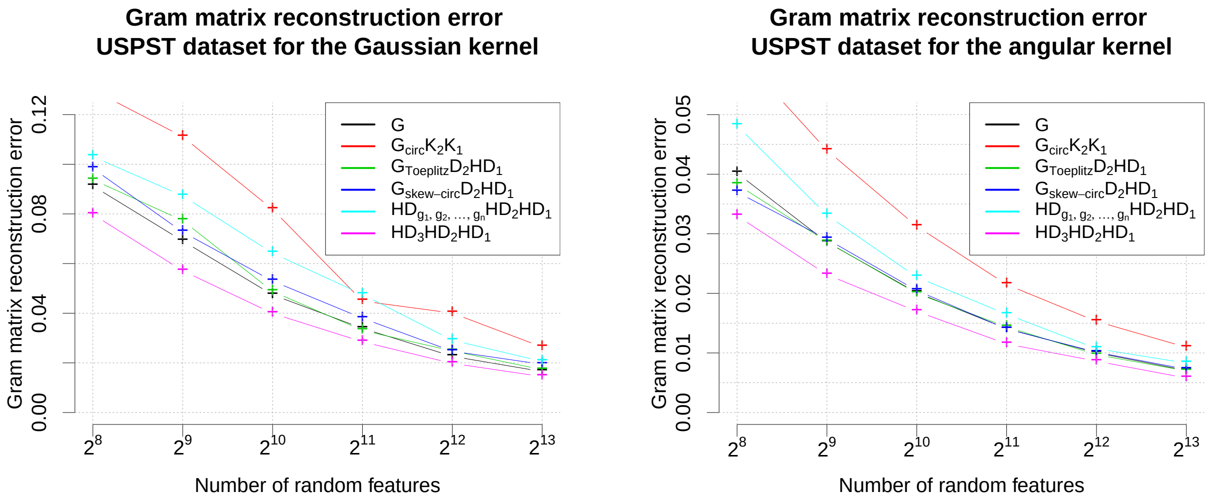

Figure 5: Accuracy of random feature map kernel approximation for the USPST dataset.

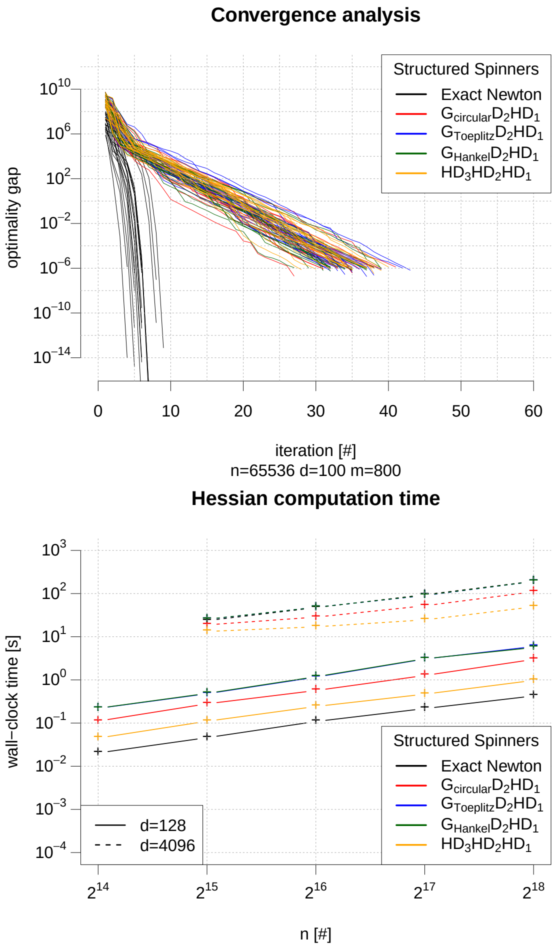

Figure 6: Numerical illustration of the convergence (top) and computational complexity (bottom) of the Newton sketch algorithm with various structured spinners. (left) Various sketching structures are compared in terms of the convergence against iteration number. (bottom) Wall-clock times of structured spinners are compared in various dimensionality settings.

$x \in \mathbb{R}^d$ minimizing the cost function

$$f(x) = \sum_{i=1}^{n} \log(1 + \exp(-y_i a_i^T x)) \ .$$

The Newton approach to solving this optimization problem entails solving at each iteration the least squares equation $\nabla^2 f(x^t)\Delta^t = -\nabla f(x^t)$, where

$$\nabla^2 f(x^t) = \\ A^T \mathrm{diag}\left(\frac{1}{1+\exp(-a_i^T x)}(1 - \frac{1}{1+\exp(-a_i^T x)})\right) A \\ \in \mathbb{R}^{d\times d}$$

is the Hessian matrix of $f(x^t)$, $A = [a_1^T a_2^T \cdots a_n^T] \in \mathbb{R}^{n\times d}$, $\Delta^t = x^{t+1} - x^t$ is the increment at iteration t and $\nabla f(x^t) \in \mathbb{R}^d$ is the gradient of the cost function. In [Pilanci and Wainwright, 2015] it is proposed to consider the sketched version of the least square equation, based on a Hessian square root of $\nabla^2 f(x^t)$, denoted $\nabla^2 f(x^t)^{1/2} = \mathrm{diag}\left(\frac{1}{1+\exp(-a_i^T x)}(1 - \frac{1}{1+\exp(-a_i^T x)})\right)^{1/2} A \in \mathbb{R}^{n\times d}$. The least squares problem at each iteration t is of the form:

$$\left((S^t \nabla^2 f(x^t)^{1/2})^T S^t \nabla^2 f(x^t)^{1/2}\right) \Delta^t = -\nabla f(x^t) \ ,$$

where $S^t \in \mathbb{R}^{m\times n}$ is a sequence of isotropic sketch matrices. Let's finally recall that the gradient of the cost function is

$$\nabla f(x^t) = \sum_{i=1}^{n} \left(\frac{1}{1+\exp(-y_i a_i^T x)} - 1\right) y_i a_i \ .$$

In our experiment, the goal is to find $x \in \mathbb{R}^d$, which minimizes the logistic regression cost, given a dataset

$\{(a_i, y_i)\}_{i=1..n}$, with $a_i \in \mathbb{R}^d$ sampled according to a Gaussian centered multivariate distribution with covariance $\Sigma_{i,j} = 0.99^{|i-j|}$ and $y_i \in \{-1, 1\}$, generated at random. Various sketching matrices $S^t \in \mathbb{R}^{m \times n}$ are considered.

In Figure 6 we report the convergence of the Newton sketch algorithm, as measured by the optimality gap defined in [Pilanci and Wainwright, 2015], versus the iteration number. As expected, the structured sketched versions of the algorithm do not converge as quickly as the exact Newton-sketch approach, however various matrices from the family of structured spinners exhibit equivalent convergence properties as shown in the figure.

When the dimensionality of the problem increases, the cost of computing the Hessian in the exact Newton-sketch approach becomes very large [Pilanci and Wainwright, 2015], scaling as $\mathcal{O}(nd^2)$. The complexity of the structured Newton-sketch approach with the matrices from the family of structured spinners is instead only $\mathcal{O}(dn\log(n) + md^2)$. Figure 6 also illustrates the wall-clock times of computing single Hessian matrices and confirms that the increase in number of iterations of the Newton sketch compared to the exact sketch is compensated by the efficiency of sketched computations, in particular Hadamard-based sketches yield improvements at the lowest dimensions.