On the Expressive Power of Deep Fully Circulant Neural Networks

Alexandre Araujo [1 2] **Benjamin Negrevergne** [2] **Yann Chevaleyre** [2] **Jamal Atif** [2]

Abstract

In this paper, we study deep fully circulant neural networks, that is deep neural networks in which all weight matrices are circulant ones. We show that these networks outperform the recently introduced deep networks with other types of structured layers. Besides introducing principled techniques for training these models, we provide theoretical guarantees regarding their expressivity. Indeed, we prove that the function space spanned by circulant networks of bounded depth includes the one spanned by dense networks with specific properties on their rank. We conduct a thorough experimental study to compare the performance of deep fully circulant networks with state of the art models based on structured matrices and with dense models. We show that our models achieve better accuracy than their structured alternatives while required 2x fewer weights as the next best approach. Finally we train deep fully circulant networks to build a compact and accurate models on a real world video classification dataset with over 3.8 million training examples.

1. Introduction

Building compact matrix representations consists in replacing dense matrices in neural networks by more structured ones with fewer weights (e.g. low rank matrices, Toeplitz matrices). In the context of deep learning, compact representations have gained much attention over the past years as a way to compress models or to reduce memory requirements. Despite a number of successful results, it remains unclear whether deep, fully structured networks are expressive enough to replace deep feedforward networks. In particular several questions remain open: "what is the expressive power of compact representations compared to the original ones?", "what do we lose by replacing all dense matrices with compact structured matrices?" and "how to effectively train deep fully structured networks?". In this paper we provide a principled answer to these questions for the particular case of deep neural networks with circulant matrices (a.k.a. circulant networks).

Recent circulant networks leverage results from linear algebra stating that any matrix A in $\mathbb{C}^{n \times n}$ can be decomposed as the product of $2n-1$ diagonal and circulant matrices (Müller-Quade et al., 1998). This has been in particular exploited in (Moczulski et al., 2015).

More recently, Thomas et al. (2018) have generalized these works by proposing neural networks with low-displacement rank matrices (LDR), that are structured matrices that encompasses several other types of structured matrices such as Toeplitz-like, Vandermonde-like, Cauchy-like and more notably circulant networks. To obtain this result, LDR represents a structured matrix using two displacement operators and a low-rank residual. However the LDR framework suffers from several shortcomings which are mainly due to its generality. First, the training procedure for learning LDR matrices is highly involved and implies many complex mathematical objects such as Krylov matrices. Then, as acknowledged by the authors, the number of parameters required to represent a given structured matrix (e.g. a Toeplitz matrix) in practice is unnecessarily high (higher than required in theory). Finally and more importantly, we experimentally observed that stacking LDR layers into a deep LDR networks yields poor performances.

In this paper, we focus on circulant networks which fit in the LDR framework, but demonstrate much better performances in practice. Indeed, thanks to a solid theoretical analysis and thorough experiments, we were able to train deep (up to 40 layers) circulant neural networks, and apply this framework in the context of a large scale real-world application (the *YouTube-8M* video classification dataset). Note that we are able to train *fully structured networks* (i.e. networks with structured layers only) hence demonstrating that circulant layers are able to model complex relations between inputs and outputs. This contrasts with previous experiments in which only one or few dense layers were replaced inside a large redundant network such as VGG. To the best of our knowledge, this is the first time such networks can be trained.

[1]Wavestone, Paris, France [2]Université Paris-Dauphine, PSL Research University, LAMSADE, CNRS, UMR 7243, Paris, France. Correspondence to: Alexandre Araujo <alexandre.araujo@wavestone.com>.

In addition, we endow circulant networks with strong theoretical guarantees by showing that circulant networks with ReLU activations and with bounded width and small depth can approximate any dense neural network with arbitrary precision. More precisely, we prove that for a deep ReLU network, $\mathcal{N}$ of width n and depth l built from weight matrices A_i of rank k_i, there exists a Deep Fully Circulant ReLU network $\mathcal{N}'$ of width n and of depth $9(\sum_i^l k_i)$ approximating $\mathcal{N}$ with arbitrary precision, under some technical conditions. To demonstrate this, we first need to prove that any matrix of rank k can be approximated by an alternating product of $O(k)$ circulant and diagonal matrices. This result improves the one of (Huhtanen & Perämäki, 2015), and constitutes the cornerstone of our theoretical analysis. Indeed, it helps us demonstrate that the class of functions represented by bounded depth circulant networks is a strict superset of the class of functions represented by dense networks whose sum of ranks ($\sum k_i$) is bounded. This means that any dense network whose sum of ranks is bounded can be approximated by a bounded-depth circulant network, but the converse is not true.

To sum up, our main contributions are the following.

1. We conduct a theoretical analysis to provide deep — fully structured — circulant networks with strong theoretical guarantees regarding their expressivity.
2. We describe a theoretically sound initialization procedure for these networks. It drastically simplifies their training and removes a large number of complex hyperparameters which made competing approaches difficult to tune (e.g. ACDC in (Moczulski et al., 2015)).
3. We provide a number of empirical insights to explain the behaviour of deep fully circulant networks. More specifically, we show the impact of the number of the non-linearities in the network on the convergence rate and the final accuracy of the network. We also show that it is easy to control the accuracy for the number of factors in the matrix decomposition, and hence the model size.
4. We combine all these insights to train deep fully structured circulant networks for a large scale video classification problem. (The *YouTube-8M* video classification problem). This constitutes a significant improvement over previous approaches which typically replaced a single fully connected layer in a large network by a structured one.

2. Related Work

Structured matrices exhibit a number of good properties which have been exploited by deep learning practitioners, mainly to compress large neural networks architectures into smaller ones. For example (Hinrichs & Vybíral, 2011) have demonstrated that a single circulant matrix can be used to approximate the Johson-Lindenstrauss transform, often used in machine learning to perform dimensionality reduction. Building on this result Cheng et al. proposed to replace the weight matrix of a fully connected layer by a circulant matrix effectively replacing the complex transform modeled by the fully connected layer by a simple dimensionality reduction. Despite the reduction of expressivity, the resulting network demonstrated good accuracy using only a fraction of its original size (90% reduction).

Later, several researchers have investigated the use of more expressive structures. For example (Moczulski et al., 2015) have introduced the AFDF framework which is based on sets of diagonal and circulant matrices (instead of just one) and is similar to the theoretical framework we discuss in this paper. However (Moczulski et al., 2015) chose to run their empirical analysis on an alternative framework ACDC and did not train any network based on AFDF for practical reasons. ACDC is based on a cosine transform which yields real matrices instead of complex ones, easier to process on GPUs. However, the cosine transform does not yield any circulant matrix and is difficult to link with the work by (Huhtanen & Perämäki, 2015), which provides the theoretical ground of their approach.

In addition to circulant matrices, a wealth of structured matrices have been investigated in the literature, e.g. Toeplitz (Sindhwani et al., 2015), Vandermonde (Sindhwani et al., 2015) or Fastfood transforms (Yang et al., 2015) . More recently (Thomas et al., 2018) have proposed a very general structured matrix framework using low-displacement rank (LDR) matrices. The LDR framework represents a structured matrix through two displacement operators and a low-rank residual. Learning the weight parameters of a neural network amounts in this framework to learn the displacement operators and the residual. An interesting property of LDR matrices is the link between the rank of the residual, called the displacement rank, and the structure of the learned matrix. Hence controlling the displacement rank amounts to constrain the structuredness of the linear layers. The authors further provide theoretical guarantees regarding the VC dimension of multilayer networks with LDR matrices. The LDR framework is more general than the AFDF framework and our framework and provides a number of theoretical results that we refine for the particular case of circulant networks. More importantly, they reveal themselves to be hard to train, in particular for networks with several LDR stacked layers. We hypothesize that this is due to the involved complex mathematical objects (Krylov matrices).

Besides structured matrices, a variety of techniques have been proposed to build more compact deep learning models. These include *model distillation* (Hinton et al., 2015), Tensor Train (Novikov et al., 2015), Low-rank decomposition (Denil et al., 2013), to mention a few. However, Circulant networks show good performances in several contexts (the interested reader can refer to the results reported in (Moczulski et al., 2015; Thomas et al., 2018)), hence the interest of this work.

3. A primer on circulant matrices and a new result

An n-by-n circulant matrix C is a matrix where each row is a cyclic right shift of the previous one as illustrated below.

$$C = circ(c) = \begin{bmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ c_2 & c_1 & c_0 & & c_3 \\ \vdots & & & \ddots & \vdots \\ c_{n-1} & c_{n-2} & c_{n-3} & & c_0 \end{bmatrix}$$

Circulant matrices enjoy several nice properties, both in terms of expressivity and computational efficiency. From the computational perspective, a circulant n-by-n matrix C can be represented using only n coefficients. Thus, it is far more compact than a full matrix that requires n^2 coefficients. Besides, any circulant matrix C can be written as the product FDF^* where is F is the discrete fourier matrix (DFT), F^* is its conjugate, and D is a diagonal matrix. A consequence of this property is that the product between a circulant matrix C and a vector x reduces to a simple element-wise product between the vector c and x in the Fourier domain (which is generally performed efficiently on GPU devices). This results in a reduced complexity from $O(n^2)$ to $O(n\,log(n))$.

From the expressivity perspective, circulant matrices can model a variety of linear transforms (e.g. random projections (Hinrichs & Vybíral, 2011)). And when they are combined with diagonal matrices, they can be used to represent any arbitrary transform (Schmid et al., 2000; Huhtanen & Perämäki, 2015). In the sequel, we build upon the tight-bound result by (Huhtanen & Perämäki, 2015), that we recall hereafter.

Theorem 1. *(Reformulation (Huhtanen & Perämäki, 2015)) For any given matrix $A \in \mathbb{C}^{n\times n}$, for any $\epsilon > 0$ and for any matrix norm $\|\cdot\|$, there exists a sequence of matrices $B_1 \dots B_{2n-1}$ where B_i is a circulant matrix if i is odd, and a diagonal matrix otherwise, such that $\|B_1 B_2 \dots B_{2n-1} - A\| < \epsilon$.*

This result provides a bound on the number of factors required to approximate a matrix A that does not depend on its rank. In our following theorem, we express the number of factors required to approximate A as a function of the rank of A. This is useful when one deals with low-rank matrices, which is common in machine learning problems.

Theorem 2. *(Rank-based circulant decomposition) Let $A \in \mathbb{C}^{n\times n}$ be a matrix of rank at most k. Assume that n can be divided by k. For any $\epsilon > 0$, there exists a sequence of $4k+1$ matrices $B_1, \dots, B_{4k+1}$, where B_i is a circulant matrix if i is odd, and a diagonal matrix otherwise, such that*

$$\left\| A - \prod_{i=1}^{4k+1} B_i \right\|_F < \epsilon$$

The proof is in supplementary material. Such a result will prove particularly useful for our design of circulant networks.

4. Building deep fully circulant neural networks

4.1. Theoretical Properties

There has already been some recent theoretical works on circulant networks, in which 2-layer networks of unbounded width were shown to be universal approximators(Zhao et al., 2017). These results are of limited interest, because the networks used in practice are of bounded width. Unfortunately, nothing is known about the theoretical properties of circulant networks in this case. In particular, the following questions remained unanswered up to now: Are circulant networks with bounded width universal approximators? What kind of functions can Deep Fully Circulant ReLU networks with bounded-width and small depth approximate?

In this section, we first introduce some necessary definitions and then provide a theoretical analysis of their approximation capabilities.

Definition 1 (Deep ReLU network)**.** *Let $f_{A,b}(x) = \Phi(Ax+b)$ for any matrices $A \in \mathbb{C}^{n\times n}$, any $b \in \mathbb{R}^n$ and $\Phi(x) = ReLU(x)$. A* Deep ReLU network *is a function $f_{A_l,b_l} \circ \dots \circ f_{A_1,b_1}$, where $A_1 \dots A_l$ are arbitrary $n \times n$ matrices and where l and n are the depth and the width of the network respectively.*

In order to characterize the expressivity of ReLU networks, we introduce the definition concept of *total rank* as follows:

Definition 2. *The* total rank *of a deep ReLU network $f_{A_l,b_l} \circ \dots \circ f_{A_1,b_1}$ is the sum of the ranks of the matrices $A_1 \dots A_l$.*

We now introduce Deep Fully Circulant ReLU networks, similarly to (Moczulski et al., 2015).

Definition 3 (Deep Fully Circulant ReLU network)**.** *Let $f_{DC,b}(x) = \Phi(DCx+b)$ with $D \in \mathbb{C}^{n\times n}$, $C \in \mathbb{C}^{n\times n}$, $b \in \mathbb{R}^n$ and $\Phi \in \{ReLU, leaky\text{-}ReLU, Identity\}$. A* Deep

Fully Circulant ReLU network *is a function* $f_{A_l,b_l} \circ \ldots \circ f_{A_1,b_1}$*, where* $D_1 \ldots D_l$ *are diagonal matrices,* $C_1 \ldots C_l$ *are circulant matrices and where* l *and* n *are the depth and the width of the network respectively.*

To show that bounded-width Deep Fully Circulant ReLU networks are universal approximators, we first need a proposition relating standard deep neural networks to Deep Fully Circulant ReLU networks.

Lemma 1. *Let* $\mathcal{N} : \mathbb{R}^n \to \mathbb{C}^n$ *be a deep ReLU network of width* n *and depth* l*, and let* $\mathcal{X} \subset \mathbb{R}^n$ *be a compact set. For any* $\epsilon > 0$*, there exists a Deep Fully Circulant ReLU network* $\mathcal{N}'$ *of width* n *and of depth* $(2n-1)l$ *such that* $\|\mathcal{N}(x) - \mathcal{N}'(x)\| < \epsilon$ *for all* $x \in \mathcal{X}$*. Moreover, this result still holds if we restrict ourselves to Deep Fully Circulant ReLU networks without any Identity activation function.*

The proof is in the supplementary material. We can now state the universal approximation corollary:

Corollary 1. *Bounded width Deep Fully Circulant ReLU networks are universal approximators in the following sense: for any continuous function* $f : [0,1]^n \to \mathbb{R}_+$ *of bounded supremum norm, for any* $\epsilon > 0$*, there exists a Deep Fully Circulant ReLU network* $\mathcal{N}_\epsilon$ *of width* $n+3$ *such that* $\forall x \in [0,1]^{n+3}$, $|f(x_1 \ldots x_n) - (\mathcal{N}_\epsilon(x))_1| < \epsilon$*, where* $(\cdot)_i$ *represents the* i^{th} *component of a vector.*

A number of work provided empirical evidences that small depth Deep Fully Circulant ReLU networks result in good performance (e.g. (Moczulski et al., 2015; Araujo et al., 2018; Cheng et al., 2015)). The following proposition, whose proof is also in supplemental material, studies the approximation properties of these small depth networks.

Theorem 3. ***(Main result**: Rank-based expressive power of circulant networks)*
Let $\mathcal{N} : f_{A_l,b_l} \circ \ldots \circ f_{A_1,b_1}$ *be a deep ReLU network of width* n*, depth* l *and a total rank* k*. Assume* n *is a power of* 2*. Let* $\mathcal{X} \subset \mathbb{R}^n$ *be a compact set. For any* $\epsilon > 0$*, there exists a deep diagonal-circulant ReLU network* $\mathcal{N}'$ *of width* n *such that* $\|\mathcal{N}(x) - \mathcal{N}'(x)\| < \epsilon$ *for all* $x \in \mathcal{X}$*. In addition, the depth of* $\mathcal{N}'$ *is bounded by* $9k$*. Moreover, if the rank of each matrix* A_i *divides* n*, then the depth of* $\mathcal{N}'$ *is bounded by* $l + \sum_{i=1}^{l} 4.rank(A_i)$*.*

Note that in the theorem, we require that n is a power of 2. We conjecture that the result still holds even without this condition, but we were not able to prove it.

This result refines Proposition 1, showing that a Deep Fully Circulant ReLU network of bounded width and small depth can approximate a Deep ReLU network of low total rank. Note that the converse is not true: because n-by-n circulant matrices can be of rank n, approximating a Deep Fully Circulant ReLU network of depth 1 can require a deep ReLU network of total rank equals to n.

For the sake of clarity, we highlight the significance of these results in the following property:

Property 1. *Let* $\mathcal{R}_{k,n}$ *be the set of all functions* $f : \mathbb{R}^n \to \mathbb{R}^n$ *representable by deep ReLU networks of total rank at most* k *and let* $\mathcal{C}_{l,n}$ *the set of all functions* $f : \mathbb{R}^n \to \mathbb{R}^n$ *representable by deep diagonal-circulant networks of depth at most* l*, then:*

$$\forall k, \exists l, \forall n\, \mathcal{R}_{k,n} \subsetneq \mathcal{C}_{l,n}$$
$$\forall l, \nexists k, \forall n\, \mathcal{C}_{l,n} \subseteq \mathcal{R}_{k,n}$$

We illustrate the meaning of this Property 1 using Figure 4.1. As we can see, the set $\mathcal{R}_{k,n}$ of all the functions representable by a deep ReLU network of total rank k is strictly included in the set $\mathcal{C}_{9k}$ of all circulant networks of depth $9k$ (as by Theorem 3).

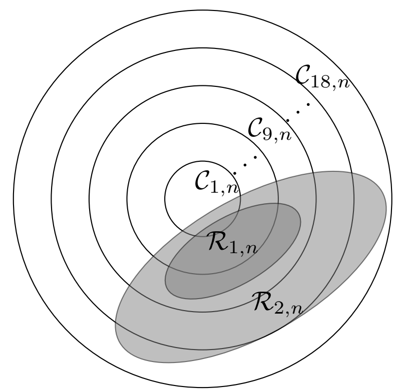


Figure 4.1. Illustration of Property 1 for a fixed width n.

Finally, what if we choose to use small depth networks to approximate deep ReLU networks where matrices are not of low rank? To answer this question, we first need to show the negative impact of replacing matrices by their low rank approximators in neural networks:

Proposition 4. *Let* $\mathcal{N} = f_{A_l,b_l} \circ \ldots \circ f_{A_1,b_1}$ *be a Deep ReLU network, where* $A_i \in \mathbb{C}^{n\times n}, b_i \in \mathbb{C}^n$ *for all* $i \in [l]$*. Let* $\bar{A}_i$ *be the matrix obtained by an SVD approximation of rank* k *of matrix* A_i*. Let* $\sigma_{i,j}$ *be the* j^{th} *singular value of* A_i*. Define* $\bar{\mathcal{N}} = f_{\bar{A}_l,b_l} \circ \ldots \circ f_{\bar{A}_1,b_1}$*. Then, for any* $x \in \mathbb{C}^n$*, we have:*

$$\left\|\mathcal{N}(x) - \bar{\mathcal{N}}(x)\right\| \leq \frac{\left(\sigma_{max,1}^{l} - 1\right) R\sigma_{max,k}}{\sigma_{max,1} - 1}$$

where R *is an upper bound on norm of the output of any layer in* $\mathcal{N}$*, and* $\sigma_{max,j} = \max_i \sigma_{i,j}$*.*

Proposition 4 shows that we can approximate matrices in a neural network by low rank matrices, and control the approximation error. In general, the term $\sigma_{max,1}^{l}$ could seem large, but in practice, it is likely that most singular values in deep neural network are small in order to avoid divergent behaviors. We can now prove the result on Deep Fully Circulant ReLU networks:

Corollary 2. *Consider any deep ReLU network* $\mathcal{N} = f_{A_l,b_l} \circ \ldots \circ f_{A_1,b_1}$ *of depth* l *and width* n. *Let* $\sigma_{max,j} = \max_i \sigma_{i,j}$ *where* $\sigma_{i,j}$ *is the* j^{th} *singular value of* A_i. *Let* $\mathcal{X} \subset \mathbb{R}^n$ *be a compact set. Let* k *be an integer dividing* n. *There exists a Deep Fully Circulant ReLU network* $\mathcal{N}' = f_{D_m C_m, b'_l} \circ \ldots \circ f_{D_1 C_1, b'_1}$ *of width* n *and of depth* $m = l(4k+1)$, *such that for any* $x \in \mathcal{X}$:

$$\|\mathcal{N}(x) - \mathcal{N}'(x)\| < \frac{\left(\sigma^l_{max,1} - 1\right) R\sigma_{max,k}}{\sigma_{max,1} - 1}$$

where R *is an upper bound on the norm of the outputs of each layer in* $\mathcal{N}$.

4.2. Initialization of deep circulant layers

Let us now discuss how to initialize the Deep Fully Circulant ReLU network correctly. Our initialization procedure, which is a variant of Xavier initialization, is as follows:

- For each circulant matrix $C = circ(c_1 \ldots c_n)$, each c_i is randomly drawn from $\mathcal{N}\left(0, \sigma^2\right)$, with $\sigma = \sqrt{\frac{2}{n}}$.
- For each diagonal matrix $D = diag(d_1 \ldots d_n)$, each d_i is drawn randomly and uniformly from $\{-1, 1\}$ for all i.
- All biases in the network are randomly drawn from $\mathcal{N}\left(0, \sigma'^2\right)$, for some small value of σ'.

To equip this initialization procedure with theoretical guarantees, we first state a lemma about the covariance matrix of the output of a diagonal-circulant layer.

Lemma 2. *Let* $C = circ(c_1 \ldots c_n)$ *and* $D = diag(d_1 \ldots d_n)$ *be circulant and diagonal matrices initialized with our procedure. Let* b *be the bias vector. Define* $y = DCu + b$ *and* $z = CDu + b$ *for some vector* u *in* $\mathbb{R}^n$. *Then, for all* i, *the p.d.f. of* y_i *and* z_i *are symmetric. Also:*

- *Assume* $u_1 \ldots u_n$ *is fixed. Then, we have for* $i \neq i'$:

$$cov(y_i, y_{i'}) = cov(z_i, z_{i'}) = 0$$
$$var(y_i) = var(z_i) = \sigma'^2 + \sum_j u_j^2 \sigma^2$$

- *Let* $x_1 \ldots x_n$ *be random variables in* $\mathbb{R}$ *such that the p.d.f. of* x_i *is symmetric for all* i, *and let* $u_i = ReLU(x_i)$. *We have for* $i \neq i'$:

$$cov(y_i, y_{i'}) = cov(z_i, z_{i'}) = 0$$
$$var(y_i) = var(z_i) = \sigma'^2 + \frac{1}{2} \sum_j var(x_i).\sigma^2$$

We can now write our main proposition, which states that the covariance matrix at the output of any layer in a Deep Fully Circulant ReLU network is constant.

Proposition 5. *Let* $\mathcal{N}$ *be a Deep Fully Circulant ReLU network of depth* l *initialized according to our procedure, with* $\sigma' = 0$. *Assume that all layers* 1 *to* $l-1$ *have ReLU activation functions, and that the last layer has the identity activation function. Then, for any* $x \in \mathbb{R}^n$, *the covariance matrix of* $\mathcal{N}(x)$ *is* $\frac{2.Id}{n}\|x\|_2^2$. *Moreover, note that this covariance does not depend on the depth of the network.*

With this initialization, the covariance matrix at the output of any layer does not depend on the depth of the network. This ensures that the signal is propagated accross the network without vanishing nor exploding.

5. Empirical evaluation

This experimental section aims at answering the following questions

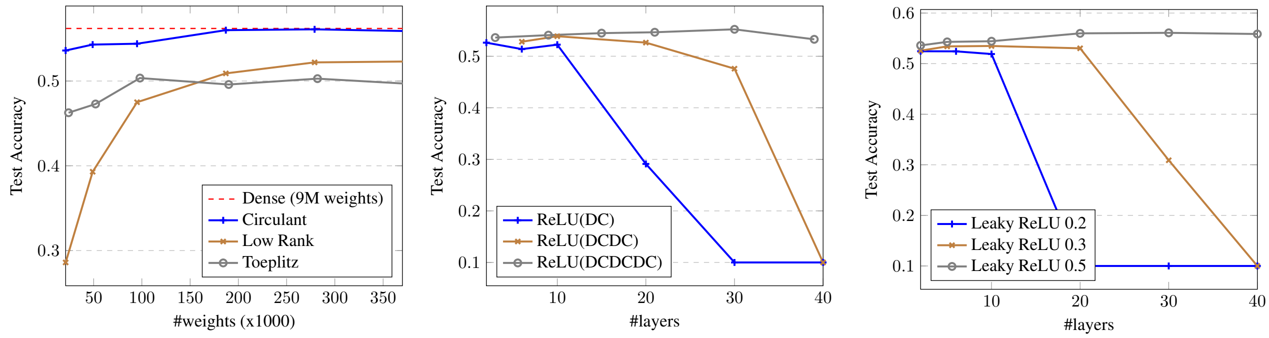
- Q1: How to train deep circulant networks and tune their hyper parameters?
- Q2: How do circulant networks compare to alternatives structured approaches?
- Q3: How do circulant networks perform in the context of large scale real-world machine learning applications?

Datasets: We conducted our experiments using two datasets. The first series of experiments (for Q1 and Q2) were conducted on the CIFAR-10 image classification dataset (Krizhevsky et al.) following the experimental protocol in (Thomas et al., 2018), and the final experiments (Q3) were conducted on the *YouTube-8M* video dataset. The *YouTube-8M* (Abu-El-Haija et al., 2016a) is a very large video classification dataset with over 3.8 million training examples and 3800 classes. For more details about the dataset, we refer to the website of the challenge[1].

Architectures: The circulant networks architectures we used for the first series of experiments are all based on simple feed forward networks in which one or more dense matrices have been replaced with one or more circulant matrices.

For the last experiment, we built a neural network based on the state-of-the-art architecture initially proposed by Abu-El-Haija et al. (2016b) and later improved by Miech et al. (2017). Remark that no convolution layer is involved in this application since the input vectors are

[1] https://research.google.com/youtube8m/workshop2018/



(a) (b) (c)

Figure 5.1. These figures demonstrates the training of Deep Fully Circulant ReLU network on CIFAR-10. Each of these figures have been build by varying the number of layers of Deep Fully Circulant ReLU network and Toeplitz networks, and the rank of low-rank networks. Figure (a) shows that circulant networks outperform concurrent structured matrices such as Toeplitz and Low Rank. The next figures demonstrate that by taking a larger depth, the network ability to converge is impacted. Figure (b) shows that by interleaving identity function instead of ReLU facilitates the convergence. The Figure (c) shows that this is also true with leaky relu with larger slope.

embeddings of video frames processed using state-of-the-art convolutional neural networks trained on ImageNet.

Unless otherwise mentioned, we consider *fully structured* networks (i.e. we replace *all* dense layers with structured layers) in order to demonstrate that our structured layers are able to model complex non-linear transforms. This is an important distinction from the previous experiments from (Moczulski et al., 2015) or (Thomas et al., 2018) that were typically ran on a large VGG network with only few layers replaced with structured ones. Since these networks are generally highly redundant, the contribution of the structured layer is difficult to quantify.

5.1. Parametrization and training of Deep Fully Circulant ReLU networks (Q1)

Finding the best trade-off between model size and model accuracy is a difficult task and training very deep networks can be challenging. In this section, we provide insights on training and tuning such networks.

Impact of the depth on accuracy In this first series of experiments, we vary the depth of the networks and observe the resulting accuracy. We then compare it with the accuracy of a dense feedforward network with 9M weights.

As we can see in Figure 5.1 (a), the accuracy of circulant networks increases with the number of layers until it reaches a maximum accuracy of 56% with 20 layers ($\sim$ 200000 weights). The best circulant network is as good as the dense network which has 9 million weights.

Training deeper circulant networks with less non-linearities. As observed in Figure 5.1 (b), training Deep Fully Circulant ReLU networks with ReLU activation functions at each layer fails for more than 15 layers. To successfully train Deep Fully Circulant ReLU networks with many layers, we discovered that reducing the non-linearity of the network helps a lot.

The first way to reduce the non-linearity of a network is to replace some of the ReLU activations by identity functions. In Figure 5.1 (b) we vary the number of ReLU activations in the network, as well as its depth. The curves entitled "ReLU(DC)" (respectively "ReLU(DCDCDC)") refer to networks in which ReLU activations appear after every layer (respectively after every group of three layers).

When the depth is small, all the networks converge and reach a similar accuracy. However, when we increase the number of layers, the networks with more ReLU activations (the less linear ones) fail to converge and exhibit poor performances. In contrast, the network with only one ReLU activation every three layers can be trained with a large number of layers and achieve the best accuracy with 30 layers.

Another way to reduce the degree of non-linearity in a network is to replace ReLU activations by leaky-ReLU with different slope parameters. This is experimented in Figure 5.1 (c).This approach achieves similar results as the previous one, since bringing the value of the slope close to one will also increase the overall linearity of the network. In this second set of experiments, we obtain a maximum accuracy of 0.56 (compared to 0.55 in the previous experiments). Overall, both techniques (one ReLU every three layers and leaky-ReLUs with a slope of 0.5) achieve nearly similar performances. We chose to use this setting in the final experiments (Q3).

Architecture	Accuracy
ACDC (Moczulski et al., 2015)	0.344
ACDC Leaky-ReLU	-
DCN ReLU	0.522
DCN Leaky-ReLU (our best approach)	**0.544**

Table 1. This table shows the accuracy of 10 layers fully structured networks with ACDC (Moczulski et al., 2015) and circulant layer. We notice that fully structured ACDC networks converge badly compared to the circulant approach.

Method	#Parameters	Accuracy
unconstrained	*9 470 986*	*0.562*
DCN $(5\ layers)$	**49 172**	**0.543**
DCN $(2\ layers)$	**21 524**	**0.536**
Circulant-like $(r = 2)$	52 234	0.526
LDR-TD $(r = 2)$	64 522	0.511
Toeplitz-like $(r = 3)$	52 234	0.496
Hankel-like $(r = 3)$	52 234	0.483
Low Rank $(r = 3)$	52 234	0.346

Table 2. This table shows the accuracy to the LDR neural network and our circulant network on the CIFAR10 dataset. LDR networks are based on one compact layer followed by a dense softmax layer. We can note that our fully structured circulant networks outperform all LDR methods with only a fraction of the weights. We only show the best score for each method. The full table is in the appendix.

5.2. Comparison with other structured approaches (Q2)

In depth comparison with ACDC. The ACDC construction was introduced by Moczulski et al. (2015), and is closely related to our approach.An important difference between the ACDC construction and our approach is that the former is implemented in practice using a Discrete Cosine Transform with ReLU activations. In our approach we rather use a Discrete Fourier Transform (DFT) [2], hence better complying with the theoretical foundation in (Huhtanen & Perämäki, 2015), and we experiment with ReLU and leaky-ReLU activations. We have argued that our approach can be used to train deep fully structured networks. We now conduct an *ablation study* in order to characterize the impact of each improvement we propose. The networks are fully circulant networks with 10 layers trained on the CIFAR-10 dataset. The results are reported in Table 1.

Although the ACDC setting achieves good performances when combined with dense layers, it does not achieve good performances when it is used to build fully structured networks. Unlike our approach, the leaky ReLU activations do not help, and the network did not converge. Our setting (DFT) with ReLU activations achieve a good accuracy of 52% (we recall that the dense baseline achieves 56% for this problem) and the leaky ReLU activations contribute to further improve the accuracy up to 54%.

In order to demonstrate the performance of circulant networks, we devise experiments to compare them with other approaches. First, we compare circulant networks with a neural network where *all* fully connected layers have been replaced by a Toeplitz and low-rank matrices. In order to compare networks with equal number of weights, we increase the number of layers for the circulant and the Toeplitz and the rank for the low rank networks. Figure 5.1 (a) shows the performance of these three architectures on CIFAR10 dataset. The weights range from 21K to 370K weights, the circulant and Toeplitz networks range from 2 to 40 layers and the rank of the low rank networks range from 7 to 34. We notice that circulant networks outperform both Toeplitz and low rank networks and achieves similar results as the dense model.

Finally, we compare circulant networks with LDR networks (Thomas et al., 2018). We reproduce the same architectures as in Thomas et al. (2018) in our experimental setup with different ranks. The networks from Thomas et al. (2018) are composed of only one structured layer and one dense softmax layer. Table 2 shows that circulant networks outperform every type of LDR matrices with less parameters.

5.3. Circulant networks for large-scale video classification (Q3)

In this section, we demonstrate the applicability of circulant networks in the context of a large scale video classification problem based on the *YouTube-8M* dataset.

5.3.1. Experiments on the aggregated dataset with Full Circulant Networks

We run experiments with Deep Fully Circulant ReLU networks on the aggregated dataset, a more compact version of the original dataset in which video and audio input vectors have been averaged every 300 frames. We compared fully circulant networks with a dense baseline with 5.7 millions weights. The goal of this experiment is to discover a good tradeoff between depth and model accuracy. To compare the models we use the GAP metric (Global Average Precision) following the experimental protocol in Abu-El-Haija et al. (2016b), to compare our experiments.

Table 3 shows the GAP score with circulant networks with

[2] recall from section 3 that any circulant matrix C can be written as FDF^*, where F is the DFT matrix

Architecture	#Weights	Compress.	GAP@20
unconstrained	*5.7M*	-	*0.773*
4 layers	25 410	0.44%	0.599
8 layers	39 234	0.68%	0.651
32 layers	122 178	2.11%	0.685
2 layers + dense	4.45M	77.14%	0.745
4 layers + dense	4.46M	77.26%	0.747

Table 3. This table shows the GAP score for the *YouTube-8M* dataset with deep circulant networks. We can see a large increase in the score with deeper networks. The increase score can be increased by reducing the compression rate by adding a dense layer in the network.

different depths. We can see that the compression ratio offered by the circulant architectures is high. This comes at the cost of a little decrease of GAP measure.

5.3.2. Preliminary Results with Circulant deep-bag-of-frames

For the sake of completeness, we also report preliminary experiments on the state-of-the-art deep-bag-of-frame architectures trained on the complex un-aggregated dataset. These architectures are hybrid architectures, instead of fully circulant networks. Thus, they are composed of a much bigger number of weights, which explains the high accuracy we get, compared to fully circulant architectures.

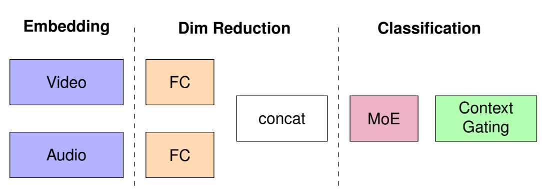


Figure 5.2. This figure shows the state-of-the-art neural network architecture, initially proposed by Abu-El-Haija et al. (2016b) and later improved by Miech et al. (2017), used in our experiment.

The Deep-Bag-of-Frame architecture can be decomposed into three blocks of layers, as illustrated in Figure 5.2. The first block of layers, composed of the Deep Bag-of-Frames embedding, is meant to model an embedding of these frames in order to make a simple representation of each video. A second block of layers reduces the dimensionality of the output of the embedding and merges the resulting output with a concatenation operation. Finally, the classification block uses a combination of Mixtures-of-Experts (MoE) (Jordan & Jacobs, 1993; Abu-El-Haija et al., 2016a) and Context Gating (Miech et al., 2017) to calculate the final class probabilities.

Architecture	#Weights	Compress.	GAP@20
unconstrained	*45M*	-	*0.846*
Circulant DBoF	36M	18.4	0.838
Circulant FC	41M	9.2	0.845
Circulant MoE	12M	72.0	0.805

Table 4. This table shows the performance of the state-of-the-art architecture on the *YouTube-8M* dataset with different layer represented with our diagonal-circulant decomposition.

Hyper-parameters All our experiments are developed with TensorFlow Framework (Abadi et al., 2015). We trained our models with the CrossEntropy loss and used Adam optimizer with a 0.0002 learning rate and a 0.8 exponential decay every 4 million examples. We used a fully connected layer of size 8192 for the video DBoF and 4096 for the audio. The fully connected layers used for dimensionality reduction have a size of 512 neurons. We used 4 mixtures for the MoE Layer.

This series of experiments aims at understanding the effect of circulant-diagonal ReLU decomposition over different layers with 1 factor. Table 5 shows the result in terms of number of weights, size of the model (MB) and GAP. We also compute the compression ratio with respect to the dense model. The compact fully connected layer achieves a compression rate of 9.5 while having a very similar performance, whereas the compact DBoF and MoE achieve a higher compression rate at the expense of accuracy.

6. Conclusion

This paper deals with a class of compact neural networks: deep networks in which all weight matrices are either diagonal or circulant matrices. Up to our knowledge, training such networks with a large number of layers had not been done before.We also endowed this kind of models with theoretical guarantees, hence enriching and refining previous theoretical work from the literature. More importantly, we showed that deep circulant networks outperform their competing structured alternatives, including the very recently general approach based on low-displacement rank matrices. Our results suggest that stacking circulant layers with non linearities improves the convergence rate and the final accuracy of the network. Formally proving these statements constitutes the future directions of this work.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, A. P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. Youtube-8m: A large-scale video classification benchmark. In *arXiv:1609.08675*, 2016a.

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016b.

Araujo, A., Negrevergne, B., Chevaleyre, Y., and Atif, J. Training compact deep learning models for video classification using circulant matrices. In *The 2nd Workshop on YouTube-8M Large-Scale Video Understanding at ECCV 2018*, 2018.

Cheng, Y., Yu, F. X., Feris, R. S., Kumar, S., Choudhary, A., and Chang, S. F. An exploration of parameter redundancy in deep networks with circulant projections. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2857–2865, Dec 2015.

Denil, M., Shakibi, B., Dinh, L., Ranzato, M. A., and de Freitas, N. Predicting parameters in deep learning. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2148–2156. Curran Associates, Inc., 2013.

Hanin, B. Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations. *ArXiv e-prints*, August 2017.

Hinrichs, A. and Vybíral, J. Johnson-lindenstrauss lemma for circulant matrices. *Random Structures & Algorithms*, 39(3):391–398, 2011.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Huhtanen, M. and Perämäki, A. Factoring matrices into the product of circulant and diagonal matrices. *Journal of Fourier Analysis and Applications*, 21(5):1018–1033, Oct 2015. ISSN 1531-5851. doi: 10.1007/s00041-015-9395-0.

Jordan, M. I. and Jacobs, R. A. Hierarchical mixtures of experts and the em algorithm. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 2, pp. 1339–1344 vol.2, Oct 1993. doi: 10.1109/IJCNN.1993.716791.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research).

Miech, A., Laptev, I., and Sivic, J. Learnable pooling with context gating for video classification. *CoRR*, abs/1706.06905, 2017.

Moczulski, M., Denil, M., Appleyard, J., and de Freitas, N. Acdc: A structured efficient linear layer. *arXiv preprint arXiv:1511.05946*, 2015.

Müller-Quade, J., Aagedal, H., Beth, T., and Schmid, M. Algorithmic design of diffractive optical systems for information processing. *Physica D: Nonlinear Phenomena*, 120(1-2):196–205, 1998.

Novikov, A., Podoprikhin, D., Osokin, A., and Vetrov, D. P. Tensorizing neural networks. In *Advances in Neural Information Processing Systems*, pp. 442–450, 2015.

Schmid, M., Steinwandt, R., Müller-Quade, J., Rötteler, M., and Beth, T. Decomposing a matrix into circulant and diagonal factors. *Linear Algebra and its Applications*, 306(1-3):131–143, 2000.

Sindhwani, V., Sainath, T., and Kumar, S. Structured transforms for small-footprint deep learning. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 3088–3096. Curran Associates, Inc., 2015.

Thomas, A., Gu, A., Dao, T., Rudra, A., and Ré, C. Learning compressed transforms with low displacement rank. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 9066–9078. Curran Associates, Inc., 2018.

Yang, Z., Moczulski, M., Denil, M., d. Freitas, N., Smola, A., Song, L., and Wang, Z. Deep fried convnets. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1476–1483, Dec 2015. doi: 10.1109/ICCV.2015.173.

Zhao, L., Liao, S., Wang, Y., Li, Z., Tang, J., and Yuan, B. Theoretical properties for neural networks with weight matrices of low displacement rank. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 4082–4090, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Supplemental Material: The Expressive Power of Deep Neural Networks with Circulant Matrices

7. Technical Lemmas and Proofs

Lemma 3. *Let $A_l, \ldots A_1 \in \mathbb{C}^{n\times n}$, $b \in \mathbb{C}^n$ and let $\mathcal{X} \subset \mathbb{R}^n$ be a compact set. There exists $\beta_l \ldots \beta_1 \in \mathbb{C}^n$ such that for all $x \in \mathcal{X}$ we have $f_{A_l,\beta_l} \circ \ldots \circ f_{A_1,\beta_1}(x) = ReLU\left(A_l A_{l-1} \ldots A_1 x + b\right)$.*

Proof. of lemma 3. Define $\Omega = \max_{x\in\mathcal{X}, j\in[l]} \left\|\prod_{k=1}^{j} A_k x\right\|_\infty$. Define $h_j(x) = A_j x + \beta_j$. Let $\beta_1 = \Omega \mathbf{1}_n$ where $\mathbf{1}_n$ is the n-vector of ones. Clearly, for all $x \in \mathcal{X}$ we have $h_1(x) \geq 0$, so $ReLU \circ h_1(x) = h_1(x)$. More generally, for all $j < n-1$ define $\beta_{j+1} = \mathbf{1}_n\Omega - A_{j+1}\beta_j$. It is easy to see that for all $j < n$ we have $h_j \circ \ldots \circ h_1(x) = A_j A_{j-1} \ldots A_1 x + \mathbf{1}_n\Omega$. This guarantees that for all $j < n$, $h_j \circ \ldots \circ h_1(x) = ReLU \circ h_j \circ \ldots \circ ReLU \circ h_1(x)$. Finally, define $\beta_l = b - A_l\beta_{l-1}$. We have, $ReLU \circ h_l \circ \ldots \circ ReLU \circ h_1(x) = ReLU\left(A_j \ldots A_1 x + b\right)$. □

Proof. of proposition 1. Assume $\mathcal{N} = f_{A_l,b_l} \circ \ldots \circ f_{A_1,b_1}$. By theorem 1, for any $\epsilon' > 0$, any matrix A_i, there exists a sequence of $2n-1$ matrices $C_{i,n}D_{i,n-1}C_{i,n-1}\ldots D_{i,1}C_{i,1}$ such that $\left|\prod_{j=0}^{n-1} D_{i,n-j}C_{i,n-j} - A_i\right| < \epsilon'$, where $D_{i,1}$ is the identity matrix. By lemma 3, we know that there exists $\{\beta_{ij}\}_{i\in[l],j\in[n]}$ such that for all $i \in [l]$, $f_{D_{in}C_{in},\beta_{in}} \circ \ldots \circ f_{D_{i1}C_{i1},\beta_{i1}}(x) = ReLU\left(D_{in}C_{in}\ldots C_{i1}x + b_i\right)$.

Now if ϵ' tends to zero, $\|f_{D_{in}C_{in},\beta_{in}} \circ \ldots \circ f_{D_{i1}C_{i1},\beta_{i1}} - ReLU\left(A_i x + b_i\right)\|$ will also tend to zero for any $x \in \mathcal{X}$, because the ReLU function is continuous and $\mathcal{X}$ is compact. Let $\mathcal{N}' = f_{D_{1n}C_{1n},\beta_{1n}} \circ \ldots \circ f_{D_{i1}C_{i1},\beta_{i1}}$. Again, because all functions are continuous, for all $x \in \mathcal{X}$, $\|\mathcal{N}(x) - \mathcal{N}'(x)\|$ tends to zero as ϵ' tends to zero. □

Proof. of corollary 1. It has been shown recently in (Hanin, 2017) that for any continuous function $f : [0,1]^n \to \mathbb{R}_+$ of bounded supremum norm, for any $\epsilon > 0$, there exists a dense neural network $\mathcal{N}$ with an input layer of width n, an output layer of width 1, hidden layers of width $n+3$ and ReLU activations such that $\forall x \in [0,1]^n$, $|f(x) - \mathcal{N}(x)| < \epsilon$. From $\mathcal{N}$, we can easily build a deep ReLU network $\mathcal{N}'$ of width exactly $n+3$, such that $\forall x \in [0,1]^{n+3}$, $|f(x_1 \ldots x_n) - (\mathcal{N}'(x))_1| < \epsilon$. Thanks to lemma 1, this last network can be approximated arbitrarily well by a Deep Fully Circulant ReLU network of width $n+3$. □

Proof. of proposition 3 . Let $k_1 \ldots k_l$ be the ranks of matrices $A_1 \ldots A_l$, which are n-by-n matrices. For all i, there exists $k'_i \in \{k_i \ldots 2k_i\}$ such that k'_i is a power of 2. Due to the fact that n is also a power of 2, k'_i divides n. By theorem 6, for all i each matrix A_i can be decomposed as an alternating product of diagonal-circulant matrices $B_{i,1}\ldots B_{i,4k'_i+1}$ such that $\left\|A_i - B_{i,1} \times \ldots \times B_{i,4k'_i+1}\right\| < \epsilon$. Using the exact same technique as in lemma 1, we can build a $DeepFullyCirculantReLUnetwork$ $\mathcal{N}'$ using matrices $B_{1,1}\ldots B_{l,4k'_l+1}$, such that $\|\mathcal{N}(x) - \mathcal{N}'(x)\| < \epsilon$ for all $x \in \mathcal{X}$. The total number of layers is $\sum_i (4k'_i + 1) \leq l + 8\sum_i k_i \leq l + 8$.total rank ≤ 9.total rank.

□

Proposition 6. *Let $A \in \mathbb{C}^{n\times n}$ a matrix of rank k. Assume that n can be divided by k. For any $\epsilon > 0$, there exists a sequence of $4k+1$ matrices $B_1, \ldots, B_{4k+1}$, where B_i is a circulant matrix if i is odd, and a diagonal matrix otherwise, such that*

$$\left\|A - \prod_{i=1}^{4k+1} B_i\right\|_F < \epsilon$$

Proof. of proposition 6. Let $U\Sigma V^T$ be the SVD decomposition of M where U, V and Σ are $n \times n$ matrices. Because M is of rank k, the last $n-k$ columns of U and V are null. In the following, we will first decompose U into a product of matrices WRO, where R and O are respectively circulant and diagonal matrices, and W is a matrix which will be further decomposed into a product of diagonal and circulant matrices. Then, we will apply the same decomposition technique to V. Ultimately, we will get a product of $4k+2$ matrices alternatively diagonal and circulant.

Let $R = circ(r_1 \dots r_n)$. Let O be a $n \times n$ diagonal matrix where $O_{i,i} = 1$ if $i \leq k$ and 0 otherwise. The k first columns of the product RO will be equal to that of R, and the $n-k$ last colomns of RO will be zeros. For example, if $k=2$, we have:

$$RO = \begin{pmatrix} r_1 & r_n & 0 & \cdots & 0 \\ r_2 & r_1 & & & \\ r_3 & r_2 & \vdots & & \vdots \\ \vdots & \vdots & & & \\ r_n & r_{n-1} & 0 & \cdots & 0 \end{pmatrix}$$

Let us define k diagonal matrices $D_i = diag(d_{i1} \dots d_{in})$ for $i \in [k]$. For now, the values of d_{ij} are unknown, but we will show how to compute them. Let $W = \sum_{i=1}^{k} D_i S^{i-1}$. Note that the $n-k$ last columns of the product WRO will be zeros. For example, with $k=2$, we have:

$$W = \begin{bmatrix} d_{1,1} & & & & d_{2,1} \\ d_{2,2} & d_{1,2} & & & \\ & d_{2,3} & \ddots & & \\ & & \ddots & & \\ & & & d_{2,n} & d_{1,n} \end{bmatrix}$$

$$WRO = \begin{pmatrix} r_1 d_{11} + r_n d_{21} & r_n d_{11} + r_{n-1} d_{21} & 0 & \cdots & 0 \\ r_2 d_{12} + r_1 d_{22} & r_1 d_{12} + r_n d_{22} & & & \\ & & \vdots & & \vdots \\ \vdots & \vdots & & & \\ r_n d_{1n} + r_{n-1} d_{2n} & r_{n-1} d_{1n} + r_{n-2} d_{2n} & 0 & \cdots & 0 \end{pmatrix}$$

We want to find the values of d_{ij} such that $WRO = U$. We can formulate this as linear equation system. In case $k=2$, we get:

$$\begin{pmatrix} r_n & r_1 & & & & & & \\ r_{n-1} & r_n & & & & & & \\ & & r_1 & r_2 & & & & \\ & & r_n & r_1 & & & & \\ & & & & r_2 & r_3 & & \\ & & & & r_1 & r_2 & & \\ & & & & & & \ddots & \\ & & & & & & & \ddots \end{pmatrix} \times \begin{pmatrix} d_{2,1} \\ d_{1,1} \\ d_{2,2} \\ d_{1,2} \\ d_{2,3} \\ d_{1,3} \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} U_{1,1} \\ U_{1,2} \\ U_{2,1} \\ U_{2,2} \\ \\ \\ \vdots \\ \\ \end{pmatrix}$$

The i^{th} bloc of the bloc-diagonal matrix is a Toeplitz matrix induced by a subsequence of length k of $(r_1, \dots r_n, r_1 \dots r_n)$. Set $r_j = 1$ for all $j \in \{k, 2k, 3k, \dots n\}$ and set $r_j = 0$ for all other values of j. Then it is easy to see that each bloc is a permutation of the identity matrix. Thus, all blocs are invertible. This entails that the block diagonal matrix above is also invertible. So by solving this set of linear equations, we find $d_{1,1} \dots d_{k,n}$ such that $WRO = U$. We can apply the same idea to factorize $V = W'.R.O$ for some matrix W'. Finally, we get

$$A = U\Sigma V^T = WRO\Sigma O^T R^T W'^T$$

Thanks to Theorem 1, W and W' can both be factorized in a product of $2k-1$ circulant and diagonal matrices. Note that $O\Sigma O^T$ is diagonal, because all three are diagonal. Overall, A can be represented with a product of $4k+2$ matrices, alternatively diagonal and circulant. □

Proof. of proposition 4 Let $x_0 \in \mathbb{C}^n$ and $\bar{x}_0 = x_0$. For all $i \in [l]$, define $x_i = ReLU\left(A_i x_{i-1} + b\right)$ and $\bar{x}_i = ReLU\left(\bar{A}_i \bar{x}_{i-1} + b\right)$. By lemma 4, we have

$$\|x_i - \bar{x}_i\| \leq \sigma_{i,k+1} \|x_{i-1}\| + \sigma_{i,1} \|x_{i-1} - \bar{x}_{i-1}\|$$

Observe that for any sequence $a_0, a_1 \ldots$ defined recurrently by $a_0 = 0$ and $a_i = r a_{i-1} + s$, the recurrence relation can be unfold as follows: $a_i = \frac{s(r^i - 1)}{r-1}$. We can apply this formula to bound our error as follows:

$$\|x_l - \bar{x}_l\| \leq \frac{\left(\sigma_{max,1}^l - 1\right)\sigma_{max,k}\max_i \|x_i\|}{\sigma_{max,1} - 1}$$

□

Lemma 4. *Let $A \in \mathbb{C}^{n\times n}$ with singular values $\sigma_1 \ldots \sigma_n$, and let $x, \bar{x} \in \mathbb{C}^n$. Let $\bar{A}$ be the matrix obtained by a SVD approximation of rank k of matrix A. Then we have:*

$$\left\|ReLU\left(Ax+b\right) - ReLU\left(\bar{A}\bar{x}+b\right)\right\| \leq \sigma_{k+1}\|x\| + \sigma_1 \|\bar{x} - x\|$$

Proof. Recall that $\|A\|_2 = \sup_z \frac{\|Az\|_2}{\|z\|_2} = \sigma_1 = \left\|\bar{A}\right\|_2$, because σ_1 is the greatest singular value of both A and $\bar{A}$. Also, note that $\left\|A - \bar{A}\right\|_2 = \sigma_{k+1}$. Let us bound the formula without ReLUs:

$$\begin{aligned}
\left\|(Ax+b) - \left(\bar{A}\bar{x}+b\right)\right\| &= \left\|(Ax+b) - \left(\bar{A}\bar{x}+b\right)\right\| \\
&= \left\|Ax - \bar{A}x - \bar{A}\left(\bar{x} - x\right)\right\| \\
&\leq \left\|\left(A - \bar{A}\right)x\right\| + \left\|\bar{A}\right\|_2 \|\bar{x} - x\| \\
&\leq \|x\|\,\sigma_{k+1} + \sigma_1 \|\bar{x} - x\|
\end{aligned}$$

Finally, it is easy to see that for any pair of vectors $a, b \in \mathbb{C}^n$, we have $\|ReLU(a) - ReLU(b)\| \leq \|a - b\|$. This concludes the proof. □

Proof. of corollary 2. Let $\bar{\mathcal{N}} = f_{\bar{A}_l,b_l} \circ \ldots \circ f_{\bar{A}_1,b_1}$, where each $\bar{A}_i$ is the matrix obtained by an SVD approximation of rank k of matrix A_i. With Proposition 4, we have an error bound on $\|\mathcal{N}(x) - \bar{\mathcal{N}}(x)\|$. Now each matrix $\bar{A}_i$ can be replaced by a product of k diagonal-circulant matrices. By theorem 3, this product yields a Deep Fully Circulant ReLU network of depth $m = l(4k+1)$, strictly equivalent to $\bar{\mathcal{N}}$ on $\mathcal{X}$. The result follows. □

Proof. of proposition 5 Let $\mathcal{N} = f_{D_l,C_l} \circ \ldots \circ f_{D_1,C_1}$ be a l layer Deep Fully Circulant ReLU network. All matrices are initialized as described in the statement of the proposition. Let $y = D_1 C_1 x$. Lemma 5 shows that $cov(y_i, y_{i'}) = 0$ for $i \neq i'$ and $var(y_i) = \frac{2}{n}\|x\|_2^2$. For any $j \leq l$, define $z^j = f_{D_j,C_j} \circ \ldots \circ f_{D_1,C_1}(x)$. By a recursive application of lemma 5, we get that then $cov(z_i^j, z_{i'}^j) = 0$ and $var(z_i^j) = \frac{2}{n}\|x\|_2^2$. □

Lemma 5. *Let $c_1 \ldots c_n, d_1 \ldots d_n, b_1 \ldots b_n$ be random variables in $\mathbb{R}$ such that $c_i \sim \mathcal{N}(0, \sigma^2)$, $b_i \sim \mathcal{N}(0, \sigma'^2)$ and $d_i \sim \{-1, 1\}$ uniformly. Define $C = circ(c_1 \ldots c_n)$ and $D = diag(d_1 \ldots d_n)$. Define $y = DCu$ and $z = CDu$ for some vector u in $\mathbb{R}^n$. Also define $\bar{y} = y + b$ and $\bar{z} = z + b$. Then, for all i, the p.d.f. of y_i, $\bar{y}_i$, z_i and $\bar{z}_i$ are symmetric. Also:*

- *Assume $u_1 \dots u_n$ is fixed. Then, we have for $i \neq i'$:*

$$
\begin{aligned}
cov(y_i, y_{i'}) &= cov(z_i, z_{i'}) = cov(\bar{y}_i, \bar{y}_{i'}) = cov(\bar{z}_i, \bar{z}_{i'}) = 0 \\
var(y_i) &= var(z_i) = \sum_j u_j^2 \sigma^2 \\
var(\bar{y}_i) &= var(\bar{z}_i) = \sigma'^2 + \sum_j u_j^2 \sigma^2
\end{aligned}
$$

- *Let $x_1 \dots x_n$ be random variables in $\mathbb{R}$ such that the p.d.f. of x_i is symmetric for all i, and let $u_i = ReLU(x_i)$. We have for $i \neq i'$:*

$$
\begin{aligned}
cov(y_i, y_{i'}) &= cov(z_i, z_{i'}) = cov(\bar{y}_i, \bar{y}_{i'}) = cov(\bar{z}_i, \bar{z}_{i'}) = 0 \\
var(y_i) &= var(z_i) = \frac{1}{2} \sum_j var(x_i).\sigma^2 \\
var(\bar{y}_i) &= var(\bar{z}_i) = \sigma'^2 + \frac{1}{2} \sum_j var(x_i).\sigma^2
\end{aligned}
$$

Proof. of lemma 5 By an abuse of notation, we write $c_0 = c_n, c_{-1} = c_{n-1}$ and so on. First, note that: $y_i = \sum_{j=1}^n c_{j-i} u_j d_j$ and $z_i = \sum_{j=1}^n c_{j-i} u_j d_i$. Observe that each term $c_{j-i} u_j d_j$ and $c_{j-i} u_j d_i$ have symmetric p.d.f. because of d_i and d_j. Thus, y_i and z_i have symmetric p.d.f. Now let us compute the covariance. □

$$
\begin{aligned}
cov(y_i, y_{i'}) &= \sum_{j,j'=1}^n cov\left(c_{j-i} u_j d_j, c_{j'-i'} u_{j'} d_{j'}\right) \\
&= \sum_{j,j'=1}^n \mathbb{E}\left[c_{j-i} u_j d_j c_{j'-i'} u_{j'} d_{j'}\right] - \mathbb{E}\left[c_{j-i} u_j d_j\right] \mathbb{E}\left[c_{j'-i'} u_{j'} d_{j'}\right]
\end{aligned}
$$

Observe that $\mathbb{E}\left[c_{j-i} u_j d_j\right] = \mathbb{E}\left[c_{j-i} u_j\right] \mathbb{E}\left[d_j\right] = 0$ because d_j is independent from $c_{j-i} u_j$. Also, observe that if $j \neq j'$ then $\mathbb{E}\left[d_j d_{j'}\right] = 0$ and thus $\mathbb{E}\left[c_{j-i} u_j d_j c_{j'-i'} u_{j'} d_{j'}\right] = \mathbb{E}\left[d_j d_{j'}\right] \mathbb{E}\left[c_{j-i} u_j c_{j'-i'} u_{j'}\right] = 0$. Thus, the only non null terms are those for which $j = j'$. We get:

$$
\begin{aligned}
cov(y_i, y_{i'}) &= \sum_{j=1}^n \mathbb{E}\left[c_{j-i} u_j d_j c_{j-i'} u_j d_j\right] \\
&= \sum_{j=1}^n \mathbb{E}\left[c_{j-i} c_{j-i'} u_j^2\right]
\end{aligned}
$$

Assume u is a fixed vector. Then, $var(y_i) = \sum_{j=1}^n u_j^2 \sigma^2$ and $cov(y_i, y_{i'}) = 0$ for $i \neq i'$ because c_{j-i} is independent from $c_{j-i'}$.

Now assume that $u_j = ReLU(x_j)$ where x_j is a r.v. Clearly, u_j^2 is independent from c_{j-i} and $c_{j-i'}$. Thus:

$$
cov(y_i, y_{i'}) = \sum_{j=1}^n \mathbb{E}\left[c_{j-i} c_{j-i'}\right] \mathbb{E}\left[u_j^2\right]
$$

For $i \neq i'$, then c_{j-i} and $c_{j-i'}$ are independent, and thus $\mathbb{E}\left[c_{j-i} c_{j-i'}\right] = \mathbb{E}\left[c_{j-i}\right] \mathbb{E}\left[c_{j-i'}\right] = 0$. Therefor, $cov(y_i, y_{i'}) = 0$ if $i \neq i'$. Let us compute the variance. We get $var(y_i) = \sum_{j=1}^n var(c_{j-i}).\mathbb{E}\left[u_j^2\right]$. Because the p.d.f. of x_j is symmetric, $\mathbb{E}\left[x_j^2\right] = 2\mathbb{E}\left[u_j^2\right]$ and $\mathbb{E}\left[x_j\right] = 0$. Thus, $var(y_i) = \frac{1}{2} \sum_{j=1}^n var(c_{j-i}).\mathbb{E}\left[x_j^2\right] = \frac{1}{2} \sum_{j=1}^n var(c_{j-i}).var(x_j)$.

Finally, note that $cov(\bar{y}_i, \bar{y}_{i'}) = cov(y_i, y_{i'}) + cov(b_i, b_{i'})$. This yields the covariances of $\bar{y}$.

To derive $cov(z_i, z_{i'})$ and $cov(\bar{z}_i, \bar{z}_{i'})$, the required calculus is nearly identical. We let the reader check by himself.

8. Additional experimental results on CIFAR-10

Method	**#Parameters**	**Accuracy**
unconstrained	*9 470 986*	*0.5620*
DC networks $(layer = 2)$	**21 524**	**0.5360**
DC networks $(layer = 5)$	**49 172**	**0.5430**
Hankel like $(r = 1)$	39 946	0.4628
Hankel like $(r = 2)$	46 090	0.4805
Hankel like $(r = 3)$	52 234	0.4831
LDR-TD $(r = 2)$	64 522	0.5115
LDR-TD $(r = 3)$	70 666	0.4739
Toeplitz like $(r = 1)$	39 946	0.4796
Toeplitz like $(r = 2)$	46 090	0.4832
Toeplitz like $(r = 3)$	52 234	0.4960
Low rank $(r = 1)$	39 946	0.2135
Low rank $(r = 2)$	46 090	0.2955
Low rank $(r = 3)$	52 234	0.3468
Circulant sparsity $(r = 1)$	46 090	0.5111
Circulant sparsity $(r = 2)$	52 234	0.5260
Circulant sparsity $(r = 3)$	58 378	0.5127

Table 5. This table shows the accuracy for the LDR neural network and our circulant network on the CIFAR10 dataset.