Cheap Orthogonal Constraints in Neural Networks: A Simple Parametrization of the Orthogonal and Unitary Group

Mario Lezcano-Casado
Mathematical Institute

David Martínez-Rubio
Department of Computer Science

UNIVERSITY OF OXFORD

June 12, 2019

Optimization with orthogonal constraints

We study the optimization of neural networks with orthogonal constraints

$$B \in \mathbb{R}^{n\times n}, \quad B^\intercal B = \mathrm{I}$$

Optimization with orthogonal constraints

We study the optimization of neural networks with orthogonal constraints

$$B \in \mathbb{R}^{n \times n}, \quad B^\intercal B = \mathrm{I}$$

Motivation:

Optimization with orthogonal constraints

We study the optimization of neural networks with orthogonal constraints

$$B \in \mathbb{R}^{n\times n}, \quad B^\intercal B = \mathrm{I}$$

Motivation:

- Orthogonal matrices have eigenvalues with norm 1.

Optimization with orthogonal constraints

We study the optimization of neural networks with orthogonal constraints

$$B \in \mathbb{R}^{n\times n}, \quad B^\intercal B = \mathrm{I}$$

Motivation:

- Orthogonal matrices have eigenvalues with norm 1.
  - Convenient for **exploding and vanishing gradient problems** within RNNs.
  - They constitute a implicit regularization method.

Optimization with orthogonal constraints

We study the optimization of neural networks with orthogonal constraints

$$B \in \mathbb{R}^{n \times n}, \quad B^\intercal B = \mathrm{I}$$

Motivation:

- Orthogonal matrices have eigenvalues with norm 1.
  - Convenient for **exploding and vanishing gradient problems** within RNNs.
  - They constitute a implicit regularization method.
- They are the basic building block for matrix factorizations like SVD or QR.

Optimization with orthogonal constraints

We study the optimization of neural networks with orthogonal constraints

$$B \in \mathbb{R}^{n \times n}, \quad B^\intercal B = \mathrm{I}$$

Motivation:

- Orthogonal matrices have eigenvalues with norm 1.
  - Convenient for **exploding and vanishing gradient problems** within RNNs.
  - They constitute a implicit regularization method.
- They are the basic building block for matrix factorizations like SVD or QR.
  - They allow for the implementation of factorized linear layers.

Optimization with orthogonal constraints

$\min_{B \in \mathbf{SO}(n)} f(B)$ is equivalent to solving $\min_{A \in \mathbf{Skew}(n)} f(\exp(A))$

Optimization with orthogonal constraints

$\underbrace{\min_{B \in \mathrm{SO}(n)} f(B)}_{\text{constrained problem.}}$ is equivalent to solving $\underbrace{\min_{A \in \mathrm{Skew}(n)} f(\exp(A))}_{\text{unconstrained problem.}}$

Optimization with orthogonal constraints

$\underbrace{\min_{B \in \mathrm{SO}(n)} f(B)}_{\text{constrained problem.}}$ is equivalent to solving $\underbrace{\min_{A \in \mathrm{Skew}(n)} f(\exp(A))}_{\text{unconstrained problem.}}$

- The matrix exponential **maps skew-symmetric matrices to orthogonal matrices**.

Optimization with orthogonal constraints

$$\underbrace{\min_{B \in \mathrm{SO}(n)} f(B)}_{\text{constrained problem.}} \quad \text{is equivalent to solving} \quad \underbrace{\min_{A \in \mathrm{Skew}(n)} f(\exp(A))}_{\text{unconstrained problem.}}$$

- The matrix exponential **maps skew-symmetric matrices to orthogonal matrices**.
- Compute the exponential to optimize over the **unconstrained** space of skew symmetric matrices.

Optimization with orthogonal constraints

$$\underbrace{\min_{B \in \mathrm{SO}(n)} f(B)}_{\text{constrained problem.}} \quad \text{is equivalent to solving} \quad \underbrace{\min_{A \in \mathrm{Skew}(n)} f(\exp(A))}_{\text{unconstrained problem.}}$$

- The matrix exponential **maps skew-symmetric matrices to orthogonal matrices**.
- Compute the exponential to optimize over the **unconstrained** space of skew symmetric matrices.
  - **No orthogonality needs to be enforced.**

Optimization with orthogonal constraints

$$\underbrace{\min_{B \in \mathrm{SO}(n)} f(B)}_{\text{constrained problem.}} \quad \text{is equivalent to solving} \quad \underbrace{\min_{A \in \mathrm{Skew}(n)} f(\exp(A))}_{\text{unconstrained problem.}}$$

- The matrix exponential **maps skew-symmetric matrices to orthogonal matrices**.
- Compute the exponential to optimize over the **unconstrained** space of skew symmetric matrices.
  - **No orthogonality needs to be enforced.**
  - It has **negligible overhead** in your neural network.

Optimization with orthogonal constraints

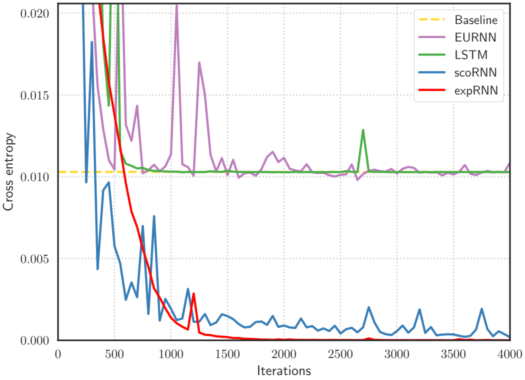
$$\underbrace{\min_{B\in \mathrm{SO}(n)} f(B)}_{\text{constrained problem.}} \quad \text{is equivalent to solving} \quad \underbrace{\min_{A\in \mathrm{Skew}(n)} f(\exp(A))}_{\text{unconstrained problem.}}$$

- The matrix exponential **maps skew-symmetric matrices to orthogonal matrices**.
- Compute the exponential to optimize over the **unconstrained** space of skew symmetric matrices.
  - **No orthogonality needs to be enforced.**
  - It has **negligible overhead** in your neural network.
  - General purpose optimizers can be used (SGD, ADAM, ADAGRAD, ...).

Optimization with orthogonal constraints

$$\underbrace{\min_{B \in \mathrm{SO}(n)} f(B)}_{\text{constrained problem.}} \quad \text{is equivalent to solving} \quad \underbrace{\min_{A \in \mathrm{Skew}(n)} f(\exp(A))}_{\text{unconstrained problem.}}$$

- The matrix exponential **maps skew-symmetric matrices to orthogonal matrices**.
- Compute the exponential to optimize over the **unconstrained** space of skew symmetric matrices.
  - **No orthogonality needs to be enforced.**
  - It has **negligible overhead** in your neural network.
  - General purpose optimizers can be used (SGD, ADAM, ADAGRAD, ...).
  - **No new extremal points** are created in the main parametrization region.



Cross entropy in the copying problem for $L = 2000$.

The copying problem uses synthetic data of the form:

	Random numbers	Wait for L steps	Recall
Input:	`14221`	`-------`	`:-----`
Output:	`-----`	`-------`	`14221`

MODEL	N	# PARAM	VALID.	TEST
EXPRNN	224	$\approx 83K$	5.34	5.30
EXPRNN	322	$\approx 135K$	**4.42**	**4.38**
EXPRNN	425	$\approx 200K$	5.52	5.48
SCORNN	224	$\approx 83K$	9.26	8.50
SCORNN	322	$\approx 135K$	8.48	7.82
SCORNN	425	$\approx 200K$	7.97	7.36
LSTM	84	$\approx 83K$	15.42	14.30
LSTM	120	$\approx 135K$	13.93	12.95
LSTM	158	$\approx 200K$	13.66	12.62
EURNN	158	$\approx 83K$	15.57	18.51
EURNN	256	$\approx 135K$	15.90	15.31
EURNN	378	$\approx 200K$	16.00	15.15
RGD	128	$\approx 83K$	15.07	14.58
RGD	192	$\approx 135K$	15.10	14.50
RGD	256	$\approx 200K$	14.96	14.69

RNNs trained on a speech prediction task on the TIMIT dataset.
It shows the best validation MSE accuracy.