The implicit bias phenomenon in deep learning

Holger Rauhut
Department of Mathematics
Ludwig-Maximilians-Universität München

FAU MoD Seminar
December 3, 2024

Collaborators:
B. Bah, H. Chou, J. Maly, U. Terstiege, R. Ward, M. Westdickenberg

Mathematics of Deep Learning

Why does deep learning work?

Can we understand the inner workings of deep learning?

What can we prove about deep learning?

Mathematics of Deep Learning

Why does deep learning work?

Can we understand the inner workings of deep learning?

What can we prove about deep learning?

Mathematical aspects:

- Optimization: understanding algorithms ((stochastic) gradient descent) for learning neural networks
  Design of fast and energy efficient algorithms
- Generalization properties of deep neural networks (performance on unseen data)
- Approximation theory of deep neural networks
- Stability properties ("adversarial noise", stability under perturbations, ...)
- Network architectures for specific tasks (inverse problems in imaging, graph convolutional networks,...)

Mathematics of Deep Learning

Why does deep learning work?

Can we understand the inner workings of deep learning?

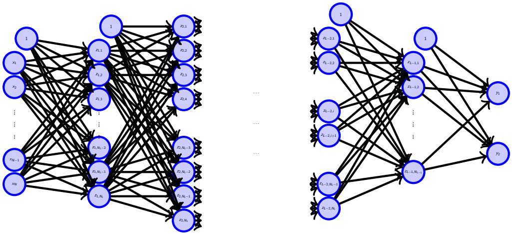
What can we prove about deep learning?

Mathematical aspects:

- Optimization: understanding algorithms ((stochastic) gradient descent) for learning neural networks
  Design of fast and energy efficient algorithms
- Generalization properties of deep neural networks (performance on unseen data)
- Approximation theory of deep neural networks
- Stability properties ("adversarial noise", stability under perturbations, ...)
- Network architectures for specific tasks (inverse problems in imaging, graph convolutional networks,...)

This talk: Convergence and Implicit bias of optimization algorithms and role of sparsity / networks of low complexity

Learning deep neural networks



Deep neural network $f : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$

$$f(x) = g_N \circ g_{N-1} \circ \cdots \circ g_1(x) = g_N(g_{N-1}(\cdots g_1(x) \cdots)),$$

with layers $g_j : \mathbb{R}^{d_{j-1}} \to \mathbb{R}^{d_j}$ with $d_0 = d_x$, $d_N = d_y$:

$$g_j(x) = \sigma(W_j x + b_j) \quad \text{with } W_j \in \mathbb{R}^{d_j \times d_{j-1}}, b_j \in \mathbb{R}^{d_j},$$

$\sigma : \mathbb{R} \to \mathbb{R}$: activation function acting componentwise

Supervised learning

Given input/output pairs $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ find parameters $W_1, \ldots, W_N$ of neural network $f = f_{W_1,\ldots,W_N}$ such that

$$f(x_\ell) \approx y_\ell, \quad \ell = 1, \ldots, m.$$

Supervised learning

Given input/output pairs $(x_1, y_1), \dots, (x_m, y_m) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ find parameters $W_1, \dots, W_N$ of neural network $f = f_{W_1,\dots,W_N}$ such that

$$f(x_\ell) \approx y_\ell, \quad \ell = 1, \dots, m.$$

Empirical risk minimization

Given a loss function $\ell : \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \to \mathbb{R}$ find the parameters of the neural network as the minimizer of the empirical loss functional

$$L(W_1, \dots, W_N) = \frac{1}{m} \sum_{\ell=1}^{m} \ell(f_{W_1,\dots,W_N}(x_\ell), y_\ell)$$

Gradient Descent and Stochastic Gradient Descent

Task: Minimization of $L(W_1, \ldots, W_N) = \frac{1}{m}\sum_{\ell=1}^{m} \ell(f_{W_1,\ldots,W_N}(x_\ell), y_\ell)$

Gradient Descent (GD):
Initialization: $W_1^0, \ldots, W_N^0$

$$W_j^{k+1} = W_j^k - \eta_k \nabla_{W_j} L(W_1^k, \ldots, W_N^k), \quad j = 1, \ldots, N$$

with appropriate step sizes $\eta_0, \eta_1, \ldots$

Gradient Descent and Stochastic Gradient Descent

Task: Minimization of $L(W_1, \ldots, W_N) = \frac{1}{m} \sum_{\ell=1}^{m} \ell(f_{W_1,\ldots,W_N}(x_\ell), y_\ell)$

Gradient Descent (GD):
Initialization: $W_1^0, \ldots, W_N^0$

$$W_j^{k+1} = W_j^k - \eta_k \nabla_{W_j} L(W_1^k, \ldots, W_N^k), \quad j = 1, \ldots, N$$

with appropriate step sizes $\eta_0, \eta_1, \ldots$

Stochastic Gradient Descent (SGD):
Initialization: $\vec{W}^0 = (W_1^0, \ldots, W_N^0)$
Iterate for $k = 0, 1, 2, \ldots$:
Stochastic approximation V_j^k: $\mathbb{E}[V_j^k | \vec{W}^k] = \nabla_{W_j} L(W_1^k, \ldots, W_N^k)$

$$W_j^{k+1} = W_j^k - \eta_k V_j^k, \quad j = 1, \ldots, N$$

Common example for stochastic gradient: Mini-batch gradient
Pick random subset $J \subset \{1, \ldots, m\}$ of size q and set

$$V_j^k = \frac{1}{q} \sum_{\ell \in J} \nabla_{W_j} \ell(f_{W_1,\ldots,W_N}(x_\ell), y_\ell)$$

Convergence of (S)GD to minimizers?

Convergence of (S)GD to global minimizer can be shown under suitable conditions on stepsize for convex loss functions.

Convergence of (S)GD to minimizers?

Convergence of (S)GD to global minimizer can be shown under suitable conditions on stepsize for convex loss functions.

For neural networks, the corresponding loss functions are non-convex.

Convergence of (S)GD to minimizers?

Convergence of (S)GD to global minimizer can be shown under suitable conditions on stepsize for convex loss functions.

For neural networks, the corresponding loss functions are non-convex.

Nevertheless, (S)GD usually converges – at least to local minimizers (with good generalization properties)

Convergence of (S)GD to minimizers?

Convergence of (S)GD to global minimizer can be shown under suitable conditions on stepsize for convex loss functions.

For neural networks, the corresponding loss functions are non-convex.

Nevertheless, (S)GD usually converges – at least to local minimizers (with good generalization properties)

Can we understand convergence behavior of (S)GD in the context of deep learning?

Implicit Bias – Some Puzzling Experiments

Tests with various convolutional networks on CIFAR-10 dataset with
$m = 50\,000$ training samples (Zhang 2017); training via SGD

Architecture	**#params (p)**	$\frac{\mathbf{p}}{\mathbf{m}}$	**Training loss**	**Test accuracy**
multi-layer perceptron	1 209 866	24.2	0.00	51.51%
Alexnet	1 387 786	27.8	0.00	76.97%
Inception	1 649 402	33	0.00	85.75%
Wide Resnet	8 950 000	179	0.00	88.21%

More network parameters than training data!

- Training error always zero on various network architectures (network fits training data exactly)
- Generalization error decreases with increasing number of parameters
  $\rightarrow$ Counterintuitive to traditional statistics (overfitting)

see also: Zhang, Bengio, Hardt, Recht, Vinyals (2016; 2021). Understanding deep learning (still) requires rethinking generalization. Communications of the ACM 64:3. pp. 107–115.

Overparameterization and Implicit Bias

- Overparameterized scenario: many networks exist that interpolate the data exactly
- Empirical loss has many global minimizers (with zero loss)
- Employed optimization algorithm (including initialization and hyperparameters such as learning rate) influences the computed minimizers, i.e., leads to an implicit bias!

Overparameterization and Implicit Bias

- Overparameterized scenario: many networks exist that interpolate the data exactly
- Empirical loss has many global minimizers (with zero loss)
- Employed optimization algorithm (including initialization and hyperparameters such as learning rate) influences the computed minimizers, i.e., leads to an implicit bias!

Understanding generalization error in deep learning requires understanding of optimization algorithms for learning:

In general, this phenomenon is far from being understood.

Working hypothesis and Simplification

Working hypothesis:
Implicit bias of (stochastic) gradient descent towards solutions of low complexity (for small initialization)

Working hypothesis and Simplification

Working hypothesis:
Implicit bias of (stochastic) gradient descent towards solutions of low complexity (for small initialization)

For a first understanding reduce to simple optimization problems that have similar characteristics as deep learning models:

- Many global minimizers
- Factorization / Compositional structure

$\rightarrow$ implicit bias towards low rank / sparsity

General idea of implicit bias

Hope/expect that limit $W_\infty = \lim_{t\to\infty} W(t)$ of gradient flow / (stochastic) gradient descent satisfies

$$\min_W R(W) \quad \text{subject to} \quad f_W(x_j) = y_j \quad \text{for all } j = 1, \ldots, m$$

Regularizer R depends on algorithm, network architecture, initialization and possibly step sizes

General idea of implicit bias

Hope/expect that limit $W_\infty = \lim_{t\to\infty} W(t)$ of gradient flow / (stochastic) gradient descent satisfies

$$\min_W R(W) \quad \text{subject to} \quad f_W(x_j) = y_j \quad \text{for all } j = 1, \ldots, m$$

Regularizer R depends on algorithm, network architecture, initialization and possibly step sizes

Hypotheses

- For suitable initialization and step sizes R promotes solutions of low complexity
- Real-world data distributions can be modeled well with neural networks with such low complexity structures, leading to good generalization

Model problem: Sparse recovery

For $A \in \mathbb{R}^{m \times n}$ with $m < n$ and $y \in \mathbb{R}^m$ consider

$$\mathcal{L}(x) = \frac{1}{2}\|Ax - y\|_2^2$$

L^1 has many global minimizers: all solutions x of $Ax = y$

Model problem: Sparse recovery

For $A \in \mathbb{R}^{m \times n}$ with $m < n$ and $y \in \mathbb{R}^m$ consider

$$\mathcal{L}(x) = \frac{1}{2}\|Ax - y\|_2^2$$

L^1 has many global minimizers: all solutions x of $Ax = y$

Factorization: $x = w^{(N)} \odot \cdots \odot w^{(2)} \odot w^{(1)}$ with vectors $w^{(j)} \in \mathbb{R}^n$ and Hadamard product $(v \odot w)_i = v_i w_i$.

$$\begin{aligned} L^N(w^{(1)}, \ldots, w^{(N)}) &= \mathcal{L}(w^{(N)} \odot \cdots \odot w^{(1)}) \\ &= \frac{1}{2}\|A(w^{(N)} \odot \cdots \odot w^{(2)} \odot w^{(1)}) - y\|_2^2 \end{aligned}$$

Minimize L^N via gradient descent / gradient flow!
Properties of limit?

Model problem: Sparse recovery

For $A \in \mathbb{R}^{m \times n}$ with $m < n$ and $y \in \mathbb{R}^m$ consider

$$\mathcal{L}(x) = \frac{1}{2}\|Ax - y\|_2^2$$

L^1 has many global minimizers: all solutions x of $Ax = y$

Factorization: $x = w^{(N)} \odot \cdots \odot w^{(2)} \odot w^{(1)}$ with vectors $w^{(j)} \in \mathbb{R}^n$ and Hadamard product $(v \odot w)_i = v_i w_i$.

$$\begin{aligned} L^N(w^{(1)}, \ldots, w^{(N)}) &= \mathcal{L}(w^{(N)} \odot \cdots \odot w^{(1)}) \\ &= \frac{1}{2}\|A(w^{(N)} \odot \cdots \odot w^{(2)} \odot w^{(1)}) - y\|_2^2 \end{aligned}$$

Minimize L^N via gradient descent / gradient flow!
Properties of limit?

Compressed sensing task: Compute sparse solution of $Ax = y$!
Standard approach: ℓ_1-minimization

$$\min \|x\|_1 \quad \text{subject to } Ax = y$$

Loss functions on factorizations

Gradient descent/flow for loss functions:

$$\mathcal{L}(x) := \frac{1}{2}\|Ax - y\|_2^2,$$
$$L^N(w^{(1)}, \dots, w^{(N)}) := \mathcal{L}(w^{(N)} \odot \cdots \odot w^{(1)}),$$
$$L_{\pm}^N(u^{(1)}, \dots, u^{(N)}, v^{(1)}, \cdots, v^{(N)}) := \mathcal{L}\left(\bigodot_{k=1}^{N} u^{(k)} - \bigodot_{k=1}^{N} v^{(k)}\right)$$

Hadamard product $(w^{(1)} \odot w^{(2)})_j = w_j^{(1)} w_j^{(2)}$

Gradient flow

"Non-factorized" gradient flow $x(t) = -\nabla\mathcal{L}(x(t))$ with $x(0) = 0$ converges to least squares solution

$$x_\infty = \lim_{t\to\infty} x(t) = \arg\min_{z:Az=y} \|z\|_2.$$

Gradient flow

"Non-factorized" gradient flow $x(t) = -\nabla\mathcal{L}(x(t))$ with $x(0) = 0$ converges to least squares solution

$$x_\infty = \lim_{t\to\infty} x(t) = \arg\min_{z:Az=y} \|z\|_2.$$

Gradient flow for overparameterized loss functionals, with initialization scale $\alpha > 0$,

$$\frac{d}{dt}w^{(k)}(t) = -\nabla_{w^{(k)}}\mathcal{L}^N(w^{(1)}(t), \ldots, w^{(N)}(t)), \quad w^{(k)}(0) = w_0 > 0,$$

$$\frac{d}{dt}u^{(k)}(t) = -\nabla_{u^{(k)}}\mathcal{L}^N_\pm(u^{(1)}(t), \ldots, u^{(N)}(t), v^{(1)}(t), \ldots, v^{(N)}(t)),$$

$$\frac{d}{dt}v^{(k)}(t) = -\nabla_{v^{(k)}}\mathcal{L}^N_\pm(u^{(1)}(t), \ldots, u^{(N)}(t), v^{(1)}(t), \ldots, v^{(N)}(t)),$$

$$u^{(k)}(0) = u_0 > 0, v^{(k)}(0) = v_0 > 0, k = 1, \ldots, N$$

Convergence of $\widetilde{x}(t) := w^{(N)}(t) \odot \cdots \odot w^{(1)}(t)$ and $\widehat{x}(t) := \bigodot_{k=1}^N u^{(k)}(t) - \bigodot_{k=1}^N v^{(k)}(t)$?
Properties of limit?

Simplification for identical initialization

For identical initialization $w^{(k)}(0) = w_0 > 0$ and $u^{(k)}(0) = u_0 > 0, v^{(k)}(0) = v_0 > 0$ for all $k = 1, \ldots, N$, it holds

$$w^{(1)}(t) = \cdots = w^{(N)}(t)$$
$$u^{(1)}(t) = \cdots = u^{(N)}(t), \quad v^{(1)}(t) = \cdots = v^{(N)}(t).$$

Therefore,

$$\widetilde{x}(t) = w^{(1)}(t)^{\odot N} = w(t)^{\odot N}$$
$$\widehat{x}(t) = u^{(1)}(t)^{\odot N} - v^{(1)}(t)^{\odot N} = u(t)^{\odot N} - v(t)^{\odot N}$$

where $w(t)$ and $u(t), v(t)$ are the gradient flows for

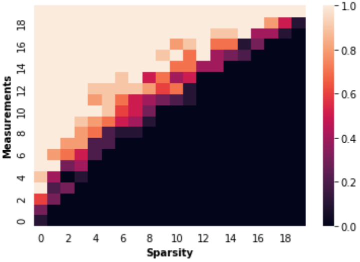
$$L(w) = \mathcal{L}(w^{\odot N}) = \frac{1}{2}\|Aw^{\odot N} - y\|_2^2,$$
$$L_{\pm}(u, v) = \mathcal{L}(u^{\odot N} - v^{\odot N})$$

Again, we set

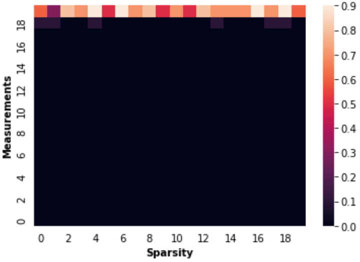
$$\widetilde{x}(t) = w^{\odot N}(t), \quad \widehat{x}(t) = u^{\odot N}(t) - v^{\odot N}(t).$$

In the following we will use $w_0 = u_0 = v_0 = \alpha(1, \ldots, 1)^T$.

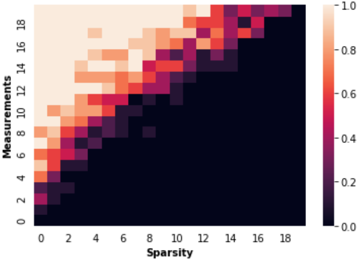
Numerics for positive case (Gaussian measurements)



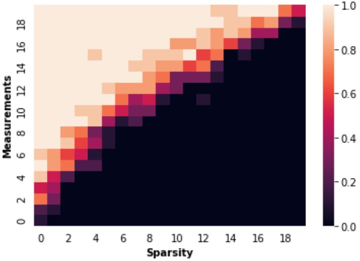
ℓ_1 minimization on $\mathbb{R}^n_+$



GD on $\mathcal{L}^N$ with $N = 1$

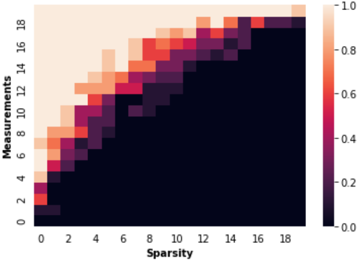


GD on $\mathcal{L}^N$ with $N = 2$

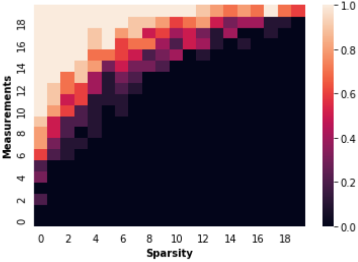


GD on $\mathcal{L}^N$ with $N = 3$

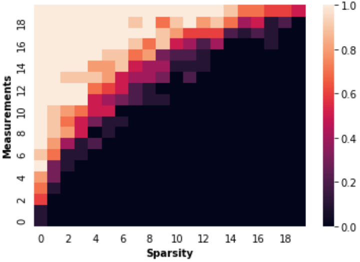
Numerical experiments for general case



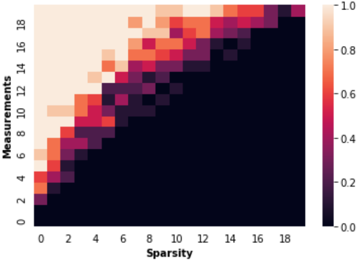
ℓ_1 minimization



GD on $\mathcal{L}_{\pm}^{N}$ with $N = 2$



GD on $\mathcal{L}_{\pm}^{N}$ with $N = 3$



GD on $\mathcal{L}_{\pm}^{N}$ with $N = 4$

Convergence to approximate ℓ_1-minimizer: positive case

Theorem (Chou, Maly, R 2022)

Let $N \geq 2$ and assume $S_+ = \{z \geq 0 : Az = y\}$ is not empty. Then the limit $\widetilde{x}_\infty = \lim_{t\to\infty} \widetilde{x}(t) = \lim_{t\to\infty} w^{\odot N}(t)$ exists and $\widetilde{x}_\infty \in S_+$. Moreover, let

$$Q = \min_{z \in S_+} \|z\|_1, \quad \beta_1 = \|\widetilde{x}(0)\|_1 = \alpha\sqrt{N}, \quad \beta_{\min} = \min_{n\in[N]} \widetilde{x}_n(0) = \alpha.$$

If $\beta_1 < Q$, then

$$\|\widetilde{x}_\infty\|_1 - Q \leq \epsilon Q,$$

where ϵ is given as

$$\epsilon = \begin{cases} \dfrac{\log(\beta_1/\beta_{\min})}{\log(Q/\beta_1)} & \text{if } N = 2, \\ \dfrac{N}{2} \cdot \dfrac{\beta_1^{1-\frac{2}{N}} - \beta_{\min}^{1-\frac{2}{N}}}{Q^{1-\frac{2}{N}} - \beta_1^{1-\frac{2}{N}}} & \text{if } N > 2. \end{cases}$$

Note: If $N > 2$ and $\beta_1^{1-2/N} \leq Q^{1-2/N}/2$ then $\epsilon \leq N(\beta_1/Q)^{1-2/N}$

A general framework for characterizing the implicit bias

Approach by Gunasekar, Lee, Soudry, Srebro (2018):
Suppose that a flow $x : [0, \infty) \to \mathbb{R}^n$ satisfies

$$\frac{d}{dt}x(t) = -H(x(t))^{-1}\nabla\mathcal{L}(x(t))$$

for some matrix valid function $H = \nabla^2 F : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ for some $F : \mathbb{R}^n \to \mathbb{R}$. Loss of the form $\mathcal{L}(x) = \frac{1}{m}\sum_{\ell=1}^{m} \ell((Ax)_j, y_j)$

Bregman divergence

$$D_F(x, z) = F(x) - F(z) - \langle\nabla F(z), x - z\rangle$$

Theorem (Gunasekar, Lee, Soudry, Srebro, 2018)

If $x_\infty = \lim_{t\to\infty} x(t)$ exists and $\mathcal{L}(x_\infty) = 0$ then x_∞ is minimizer of

$$\min_x D_F(x, x(0)) \quad \textit{subject to } Ax = y.$$

Bregman divergence

For

$$F(x) = \begin{cases} \frac{1}{2}\sum_{k=1}^n x_k \log(x_k) - x_k & \text{if } N = 2, \\ -\frac{N}{2(N-2)}\sum_{k=1}^n x_k^{2/N} & \text{if } N > 2 \end{cases}$$

the Bregman divergence is

$$D_F(z,x) = \begin{cases} \frac{1}{2}\sum\limits_{k=1}^n z_k \log(z_k/x_k) + \frac{1}{2}\sum_{k=1}^n (x_k - z_k) & \text{if } N = 2, \\ \frac{1}{2(N-2)}\sum\limits_{k=1}^n \left((N-2)x_k^{\frac{2}{N}} + 2z_k x_k^{\frac{2}{N}-1} - Lz_k^{\frac{2}{N}} \right) & \text{if } N > 2 \end{cases}$$

Kullback-Leibler divergence for $N = 2$

Convergence to minimizer of Bregman divergence

Theorem (Chou, Maly, R 2022)

Let $N \geq 2$ *and assume* $S_+ = \{z \geq 0 : Az = y\}$ *is not empty. Then the limit* $\widetilde{x}_\infty = \lim_{t\to\infty} \widetilde{x}(t) = \lim_{t\to\infty} w^{\odot N}(t)$ *exists and* $\widetilde{x}_\infty \in S_+$*. Moreover,*

$$\widetilde{x}_\infty \in \operatorname{argmin}_{z\in S_+} D_F(z, \widetilde{x}(0)) = \operatorname{argmin}_{z\in S_+} g_{\widetilde{x}(0)}(z)$$

where

$$g_{\widetilde{x}}(z) = \begin{cases} \sum_{k=1}^n z_k(\log(z_k) - 1 - \log(\widetilde{x}_k)) & \text{if } N = 2, \\ 2\|z\|_1 - N\sum_{k=1}^n z_k^{\frac{2}{N}} \widetilde{x}_k^{1-\frac{2}{L}} & \text{if } N > 2. \end{cases}$$

Convergence to approximate ℓ_1-minimizer: general case

Theorem (Chou, Maly, R 2022)

Let $N \geq 2$ *and assume* $S = \{z : Az = y\}$ *is not empty. Consider the flow* $(u(t), v(t))$ *and the corresponding "product flow"* $\widehat{x}(t) = u^{\odot N}(t) - v^{\odot N}(t)$. *Then the limit* $\widehat{x}_\infty = \lim_{t\to\infty} \widehat{x}(t)$ *exists and* $A\widehat{x}_\infty = y$. *Moreover, let* $Q = \min_{z\in S} \|z\|_1$ *and*

$$\beta_1 = \|u^{\odot N}(0)\|_1 + \|v^{\odot N}(0)\|_1 = 2\alpha\sqrt{N},$$
$$\beta_{\min} = \min_{k\in[N]} \min\{u_k^N(0), v_k^N(0)\} = \alpha.$$

If $\beta_1 < Q$, *then*

$$\|\widehat{x}_\infty\|_1 - Q \leq \epsilon Q,$$

where ϵ *is given as*

$$\epsilon = \begin{cases} \dfrac{\log(\beta_1/\beta_{\min})}{\log(Q/\beta_1)} & \text{if } N = 2, \\ \dfrac{N}{2} \cdot \dfrac{\beta_1^{1-\frac{2}{N}} - \beta_{\min}^{1-\frac{2}{N}}}{Q^{1-\frac{2}{N}} - \beta_1^{1-\frac{2}{N}}} & \text{if } N > 2. \end{cases}$$

General initialization

Results stated for initialization

$$w(0) = u(0) = v(0) = \alpha \mathbf{1}.$$

For general initialization $w(0), u(0), v(0) > 0$ we obtain convergence to (approximate) weighted ℓ_1-minimization with weight h depending on initialization,

$$h = w(0)^{\odot \frac{2}{L} - 1}$$

Compressive sensing from Gaussian matrices via gradient flow

Corollary (Chou, Maly, R 2022)

Choose A to be a random Gaussian matrix in $\mathbb{R}^{m \times n}$ with

$$m \geq C\rho^{-2} s \log(en/s)$$

for some constant $\rho \in (0,1)$. Then the following holds with probability at least $1 - e^{-cm}$. Let $x \in \mathbb{R}^n$ and $y = Ax$. Then the limit $\widehat{x}_\infty$ of the product flow satisfies

$$\|\widehat{x}_\infty - x\|_1 \leq \frac{1+\rho}{1-\rho} \left(2\sigma_s(x)_1 + \epsilon\right),$$

where ϵ is defined as before.

Extension to noisy measurements possible
(via so-called ℓ_1-quotient property)

Weight normalization

Previous results require small initialization scale α.
Small initialization leads to high computation time
(flow needs to escape neighborhood of saddle point zero)

Is it possible to work with larger initialization?

Weight normalization

Previous results require small initialization scale α.
Small initialization leads to high computation time
(flow needs to escape neighborhood of saddle point zero)

Is it possible to work with larger initialization?

Weight normalization
In practice, the weights are often normalized in (stochastic) gradient descent, improving stability and generalization.

Normalized gradient flow

Separate w into magnitude and direction

$$w = r\frac{v}{\|v\|} \quad \text{with } r \geq 0, v \in \mathbb{R}^n,$$

and set

$$\widetilde{\mathcal{L}}(r,v) = \mathcal{L}\left(r\frac{v}{\|v\|}\right) = \frac{1}{2}\left\|A\left(r\frac{v}{\|v\|}\right)^{\otimes N} - y\right\|_2^2$$

Normalized gradient flow

Separate w into magnitude and direction

$$w = r\frac{v}{\|v\|} \quad \text{with } r \geq 0, v \in \mathbb{R}^n,$$

and set

$$\widetilde{\mathcal{L}}(r,v) = \mathcal{L}\left(r\frac{v}{\|v\|}\right) = \frac{1}{2}\left\|A\left(r\frac{v}{\|v\|}\right)^{\otimes N} - y\right\|_2^2$$

Gradient flow with different rates for r and w:

$$\begin{aligned}\frac{d}{dt}r(t) &= -\eta_r \nabla_r \widetilde{\mathcal{L}}(r,v), \quad r(0) = r_0 > 0\\ \frac{d}{dt}v(t) &= \quad -\nabla_v \widetilde{\mathcal{L}}(r,v), \quad v(0) = \frac{1}{\sqrt{n}}\mathbf{1} > 0\end{aligned}$$

Denote $w(t) = r(t)\frac{v(t)}{\|w(t)\|_2}$ and $\widetilde{x}(t) = w(t)^{\odot N}$.

Separating scales, i.e., $\eta_r \ll 1$, important for removing need for small initialization

Magnification of implicit regularization

Theorem (Chou, R, Ward 2023)

Let $N \geq 2$, assume that $Av = 0$ for some $v > 0$ and that $S_+ = \{z \geq 0 : Az = y\}$ is not empty. Suppose that $\widetilde{x}_\infty = \lim_{t\to\infty} \widetilde{x}(t)$ exists and denote $r_\infty = \|\widetilde{x}_\infty^{\odot 1/N}\|_2$. Define the magnification factor as

$$\rho := \frac{r_0}{r_\infty} \exp\left(\frac{r_\infty^2 - r_0^2}{\eta_r}\right).$$

Moreover, let

$$Q = \min_{z\in S_+} \|z\|_1, \quad \beta_1 = \|\widetilde{x}(0)\|_1 = r_0^N\sqrt{n}, \quad \beta_{\min} = \min_{n\in[N]} \widetilde{x}_n(0) = r_0^N.$$

If $c_N\beta_1 < Q$, with $c_2 = 1$ and $c_N = (N/2)^{N/(N-2)}$ for $N > 2$ then

$$\|\widetilde{x}_\infty\|_1 - Q \leq \epsilon(\rho^{-N}\beta_1, \rho^{-N}\beta_{\min})Q,$$

where ϵ is given as before, in particular, $\epsilon(\rho^{-N}\beta_1, \rho^{-N}\beta_{\min}) = \frac{\log(\beta_1/\beta_{\min})}{\log(\rho^N Q/\beta_1)}$ if $N = 2$ and $\epsilon(\rho^{-N}\beta_1, \rho^{-N}\beta_{\min}) = \frac{N}{2} \cdot \frac{\beta_1^{1-\frac{2}{N}} - \beta_{\min}^{1-\frac{2}{N}}}{\rho^{N-2}Q^{1-\frac{2}{N}} - \beta_1^{1-\frac{2}{N}}}$ if $N > 2$.

Model problem: Low rank matrix recovery

Task: Recover a matrix $W \in \mathbb{R}^{n_1 \times n_2}$ of rank $r \ll \min\{n_1, n_2\}$ from $m \ll n_1 n_2$ linear measurements (Candès, Recht '09; Candès, Plan '10; Gross et al '10; Kueng, Rauhut, Terstiege '17, ...)

$$y = \mathcal{A}(W) \in \mathbb{R}^m, \quad \mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m.$$

Underdetermined linear system with rank-constraint

Recovery via gradient descent on matrix factorization?

Let $W \in \mathbb{R}^{n \times n}$ of rank $r \ll n$ and

$$y = \mathcal{A}(W) \in \mathbb{R}^m, \quad \mathcal{A} : \mathbb{R}^{n \times n} \to \mathbb{R}^m, \quad m \ll n^2.$$

for a suitable linear map $\mathcal{A}$.

Deep matrix factorization (linear neural network):
Set $Z = W_N \cdots W_2 \cdot W_1$ and minimize

$$L_{\mathcal{A}}(W_1, \dots, W_N) = \|y - \mathcal{A}(W_N \cdots W_1)\|_2^2$$

via gradient descent on $(W_N, \dots, W_1)$.

Recovery via gradient descent on matrix factorization?

Let $W \in \mathbb{R}^{n\times n}$ of rank $r \ll n$ and

$$y = \mathcal{A}(W) \in \mathbb{R}^m, \quad \mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^m, \quad m \ll n^2.$$

for a suitable linear map $\mathcal{A}$.

Deep matrix factorization (linear neural network):
Set $Z = W_N \cdots W_2 \cdot W_1$ and minimize

$$L_{\mathcal{A}}(W_1, \dots, W_N) = \|y - \mathcal{A}(W_N \cdots W_1)\|_2^2$$

via gradient descent on $(W_N, \dots, W_1)$.

If $W_j \in \mathbb{R}^{n_j \times n_{j-1}}$, $r := \min_j n_j$ then

$$\operatorname{rank}(W) = \operatorname{rank}(W_N \cdots W_1) \le r.$$

Recovery via gradient descent on matrix factorization?

Let $W \in \mathbb{R}^{n \times n}$ of rank $r \ll n$ and

$$y = \mathcal{A}(W) \in \mathbb{R}^m, \quad \mathcal{A} : \mathbb{R}^{n \times n} \to \mathbb{R}^m, \quad m \ll n^2.$$

for a suitable linear map $\mathcal{A}$.

Deep matrix factorization (linear neural network):
Set $Z = W_N \cdots W_2 \cdot W_1$ and minimize

$$L_{\mathcal{A}}(W_1, \dots, W_N) = \|y - \mathcal{A}(W_N \cdots W_1)\|_2^2$$

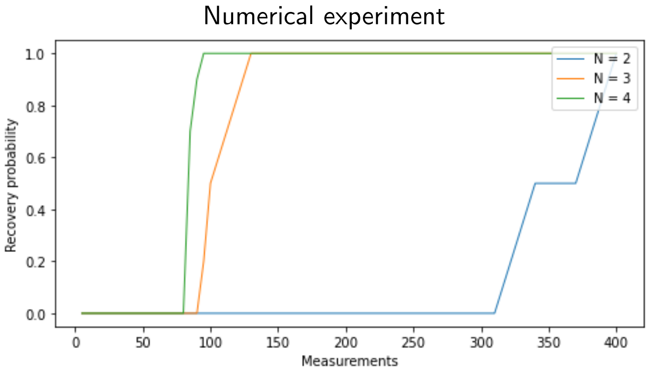
via gradient descent on $(W_N, \dots, W_1)$.

If $W_j \in \mathbb{R}^{n_j \times n_{j-1}}$, $r := \min_j n_j$ then

$$\operatorname{rank}(W) = \operatorname{rank}(W_N \cdots W_1) \leq r.$$

Implicit bias (recovery) in the setting $W_j \in \mathbb{R}^{n \times n}$ for all $j = 1, \dots, N$?

Low rank matrix recovery via deep matrix factorization



Recovery of $X \in \mathbb{R}^{20\times 20}$ of rank 2 from Gaussian random measurements

Satisfying theory not yet available

More work on implicit bias of gradient descent/flow

- Analysis of (S)GD for two-layer diagonal networks (sparse recovery)
  Evan, Pesme, Gunasekar, Flammarion (2023)
- Recovery of positive semidefinite matrices from commuting set of measurements A_j, $y_j = \mathrm{tr}(A_j^T X)$, for gradient flow on factorization $W = UU^T$; convergence to nuclear norm minimizer (Problem: Commuting measurements A_j very restrictive!)
  Gunasekar, Woodworth, Bhojanapalli, Neyshabur, Srebro 2017
  Arora, Cohen, Hu, Luo 2019
- Recovery of positive semidefinite matrices from Gaussian measurements for gradient flow on factorization $W = UU^T$
  Stöger, Soltanolkotabi 2021
- Implicit bias of GD for classification with fully connected and convolutional neuronal networks
  Soudry, Hoffer, Nacson, Gunasekar, N. Srebro 2018 Gunasekar, Lee, Soudry, Srebro 2018
- Dynamics and implicit bias for GD on matrix estimation problems
  Chou, Maly, Rauhut 2020
- Early alignment for gradient flow on two-layer ReLU-networks
  Flammarion, Boursier 2024

Product flow for matrix factorization

For a general loss $\mathcal{L} : \mathbb{R}^{d_0 \times d_N} \to \mathbb{R}$ consider

$$L^N(W_1, \ldots, W_N) = \mathcal{L}(W_N \cdots W_1), \quad W_j \in \mathbb{R}^{d_{j-1} \times d_j}$$

and associated gradient flow

$$\frac{d}{dt} W_j(t) = -\nabla_{W_j} L^N(W_1(t), \ldots, W_N(t)).$$

Product flow

$$W(t) = W_N(t) \cdots W_1(t)$$

Under balancedness: $W_{j+1}(0)^T W_{j+1}(0) = W_j(0) W_j(0)^T$ it holds

$$\frac{d}{dt} W = -\sum_{j=1}^{N} (WW^T)^{\frac{N-j}{N}} \cdot \nabla \mathcal{L}^1(W) \cdot (W^T W)^{\frac{j-1}{N}}.$$

For $W, Z \in \mathbb{R}^{d_0 \times d_N}$ introduce the map

$$\mathcal{A}_W(Z) = \mathcal{A}_W^N(Z) = \sum_{j=1}^{N} (WW^T)^{\frac{N-j}{N}} \cdot Z \cdot (W^T W)^{\frac{j-1}{N}}.$$

Riemannian manifold of rank r matrices

Rank of $W = W_N \cdots W_1$, $W_j \in \mathbb{R}^{d_j \times d_{j-1}}$ at most $r = \min_{j=0,\dots,N} d_j$

$\mathcal{M}_k$: manifold or matrices $W \in \mathbb{R}^{d_y \times d_x}$ of rank k

Tangent space of $\mathcal{M}_k$ at $W \in \mathcal{M}_k$:

$$T_W(\mathcal{M}_k) = \left\{ WA + BW : A \in \mathbb{R}^{d_x \times d_x}, B \in \mathbb{R}^{d_y \times d_y} \right\}.$$

Riemannian manifold of rank r matrices

Rank of $W = W_N \cdots W_1$, $W_j \in \mathbb{R}^{d_j \times d_{j-1}}$ at most $r = \min_{j=0,\dots,N} d_j$
$\mathcal{M}_k$: manifold or matrices $W \in \mathbb{R}^{d_y \times d_x}$ of rank k
Tangent space of $\mathcal{M}_k$ at $W \in \mathcal{M}_k$:

$$T_W(\mathcal{M}_k) = \left\{ WA + BW : A \in \mathbb{R}^{d_x \times d_x}, B \in \mathbb{R}^{d_y \times d_y} \right\}.$$

Theorem (Bah, Rauhut, Terstiege, Westdickenberg 2020)

Let $N \geq 2$. For $W \in \mathcal{M}_k$, the restriction $\bar{\mathcal{A}}_W : T_W(\mathcal{M}_r) \to T_W(\mathcal{M}_k)$ of $\mathcal{A}_W$ to $T_W(\mathcal{M}_r)$ is self-adjoint and positive definite, hence invertible.
For $W \in \mathbb{R}^{d_y \times d_x}$, the bilinear map

$$g_W(Z_1, Z_2) := \langle \bar{\mathcal{A}}_W^{-1}(Z_1), Z_2 \rangle_F, \quad Z_1, Z_2 \in T_W(\mathcal{M}_k),$$

defines a Riemannian metric on $\mathcal{M}_k$ of class C^1.

Riemannian manifold of rank r matrices

Rank of $W = W_N \cdots W_1$, $W_j \in \mathbb{R}^{d_j \times d_{j-1}}$ at most $r = \min_{j=0,\dots,N} d_j$

$\mathcal{M}_k$: manifold or matrices $W \in \mathbb{R}^{d_y \times d_x}$ of rank k

Tangent space of $\mathcal{M}_k$ at $W \in \mathcal{M}_k$:

$$T_W(\mathcal{M}_k) = \left\{ WA + BW : A \in \mathbb{R}^{d_x \times d_x}, B \in \mathbb{R}^{d_y \times d_y} \right\}.$$

Theorem (Bah, Rauhut, Terstiege, Westdickenberg 2020)

Let $N \geq 2$. For $W \in \mathcal{M}_k$, the restriction $\bar{\mathcal{A}}_W : T_W(\mathcal{M}_r) \to T_W(\mathcal{M}_k)$ of $\mathcal{A}_W$ to $T_W(\mathcal{M}_r)$ is self-adjoint and positive definite, hence invertible.

For $W \in \mathbb{R}^{d_y \times d_x}$, the bilinear map

$$g_W(Z_1, Z_2) := \langle \bar{\mathcal{A}}_W^{-1}(Z_1), Z_2 \rangle_F, \quad Z_1, Z_2 \in T_W(\mathcal{M}_k),$$

defines a Riemannian metric on $\mathcal{M}_k$ of class C^1.

Explicit formula for Riemannian metric

$$g_W(Z_1, Z_2) = \frac{\sin(\pi/N)}{\pi} \int_0^\infty \operatorname{tr}\left((tI + WW^T)^{-1} Z_1 (tI + W^T W)^{-1} Z_2^T \right) t^{1/N} dt$$

Riemannian gradient flow

Riemannian gradient associated to metric g

$$\nabla^g \mathcal{L}(W) = \mathcal{A}_W (\nabla \mathcal{L}(W)),$$

where $\nabla \mathcal{L}$ is standard gradient of $\mathcal{L}$, i.e.,

$$g_W(\nabla^g \mathcal{L}(W), Z) = \langle \nabla \mathcal{L}(W), Z \rangle_F \quad \text{for all } Z \in T_W(\mathcal{M}_r),$$

Assuming balancedness and $W(0) \in \mathcal{M}_k$ we recover the flow for $W(t)$ as Riemannian gradient flow on $\mathcal{M}_k$

$$\frac{d}{dt} W(t) = -\nabla^g \mathcal{L}(W(t)) = -\mathcal{A}_{W(t)} (\nabla \mathcal{L}(W(t))).$$

Note: If $W(0) \in \mathcal{M}_k$ then $W(t) \in \mathcal{M}_k$ for all $t \geq 0$.

Implicit bias towards solutions of large intrinsic volume

Riemannian volume form for g: For $W \in \mathbb{R}^{n\times n}$ of full rank n with singular value decomposition $W = U\Sigma V^T$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$,

$$\sqrt{\det g}\, dW = \underbrace{N^{\frac{n(n-1)}{2}} \det(\Sigma^2)^{\frac{1-N}{2N}} \mathrm{van}(\Sigma^{2/N})}_{=: v(W)} d\Sigma dU dV$$

where dU, dV denote Haar measure on $O(n)$ and $\mathrm{van}(\Sigma^{2/N})$ is Vandermonde determinant of the diagonal of $\Sigma^{2/N}$:

$$\mathrm{van}(\Sigma^{2/N}) = \prod_{1 \le i < j \le n} (\sigma_i^{2/N} - \sigma_j^{2/N}).$$

Implicit bias towards solutions of large intrinsic volume

Riemannian volume form for g: For $W \in \mathbb{R}^{n\times n}$ of full rank n with singular value decomposition $W = U\Sigma V^T$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$,

$$\sqrt{\det g}\, dW = \underbrace{N^{\frac{n(n-1)}{2}} \det(\Sigma^2)^{\frac{1-N}{2N}} \operatorname{van}(\Sigma^{2/N})}_{=: v(W)} d\Sigma dU dV$$

where dU, dV denote Haar measure on $O(n)$ and $\operatorname{van}(\Sigma^{2/N})$ is Vandermonde determinant of the diagonal of $\Sigma^{2/N}$:

$$\operatorname{van}(\Sigma^{2/N}) = \prod_{1\leq i<j\leq n} (\sigma_i^{2/N} - \sigma_j^{2/N}).$$

Numerical experiments on small matrix completion problems by Cohen et al. (2022) indicate implicit bias of gradient flow towards solutions with large intrinsic Riemannian volume $v(W)$.

Note: $v(W) = \infty$ for W of rank $r < n$.

N. Cohen, G. Menon, Z. Veraszto (2022). Deep Linear Networks for Matrix Completion – An Infinite Depth Limit. arXiv:2210.12497

Open Questions

- Extensions from gradient flow to (stochastic) gradient descent (work in progress)
- Matrix case
- Nonlinear networks (work on ReLU-networks in progress)
- ...

General question

- Do we really need to start with network structures having millions or billions of learnable weights?
- Can we exploit insights on bias to low complexity network structures when designing algorithms / networks?
- High-dimensionality required because of intrinsic hardness of learning?

Thanks very much for your attention!

References:

B. Bah, H. Rauhut, U. Terstiege, M. Westdickenberg (2022). Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers. Information and Inference 11(1):307–353.

M. Nguegnang, H. Rauhut, U. Terstiege (2024). Convergence of gradient descent for learning linear neural network. Advances in Continuous and Discrete Models, article number 23.

H.-H. Chou, C. Gieshoff, J. Maly, H. Rauhut (2024). Gradient Descent for Deep Matrix Factorization: Dynamics and Implicit Bias towards Low Rank. Applied and Computational Harmonic Analysis (68), 101595.

H.-H. Chou, J. Maly, H. Rauhut (2023). More is Less: Inducing Sparsity via Overparameterization. Information and Inference 12(3), 1437-1460.

H.-H. Chou, H. Rauhut, R. Ward (2024). Robust Implicit Regularization via Weight Normalization, Information and Inference 13(3), iaae022.