Gradient flow for linear neural networks

Ulrich Terstiege,

September 23rd, 2026

LMU Munich

Setup

Given data

$$X = [x_1, \cdots, x_m] \in \mathbb{R}^{d_x \times m}, \qquad Y = [y_1, \cdots, y_m] \in \mathbb{R}^{d_y \times m},$$

and neural network architecture f depending on weights w and input data, want to find specific w such that (empirical) loss ℓ of $f(w, X) := [f(w, x_1), \cdots, f(w, x_m)]$ and Y is small.

Setup

Given data

$$X = [x_1, \cdots, x_m] \in \mathbb{R}^{d_x \times m}, \qquad Y = [y_1, \cdots, y_m] \in \mathbb{R}^{d_y \times m},$$

and neural network architecture f depending on weights w and input data, want to find specific w such that (empirical) loss ℓ of $f(w, X) := [f(w, x_1), \cdots, f(w, x_m)]$ and Y is small.

Typical approach: minimize $\ell(f(w, X), Y) := \sum_i \ell(f(w, x_i), y_i)$ via (stochastic) gradient descent.

Setup

Given data

$$X = [x_1, \ \cdots, \ x_m] \in \mathbb{R}^{d_x \times m}, \qquad Y = [y_1, \ \cdots, \ y_m] \in \mathbb{R}^{d_y \times m},$$

and neural network architecture f depending on weights w and input data, want to find specific w such that (empirical) loss ℓ of $f(w, X) := [f(w, x_1), \ \cdots, \ f(w, x_m)]$ and Y is small.

Typical approach: minimize $\ell(f(w, X), Y) := \sum_i \ell(f(w, x_i), y_i)$ via (stochastic) gradient descent.

In the limit stepsize $\to 0$, this leads to the gradient flow

$$\dot{w} = -\nabla_w \ell(f(w, X), Y), \ w(0) = w_0.$$

Setup

Given data

$$X = [x_1, \ \cdots, \ x_m] \in \mathbb{R}^{d_x \times m}, \qquad Y = [y_1, \ \cdots, \ y_m] \in \mathbb{R}^{d_y \times m},$$

and neural network architecture f depending on weights w and input data, want to find specific w such that (empirical) loss ℓ of $f(w, X) := [f(w, x_1), \ \cdots, \ f(w, x_m)]$ and Y is small.

Typical approach: minimize $\ell(f(w, X), Y) := \sum_i \ell(f(w, x_i), y_i)$ via (stochastic) gradient descent.

In the limit stepsize $\to 0$, this leads to the gradient flow

$$\dot{w} = -\nabla_w \ell(f(w, X), Y), \ w(0) = w_0.$$

- Is this defined for all $t \geq 0$?
- Does it converge? To a critical point or even to a minimizer of $\ell(f(w, X), Y)$?

Fully connected linear networks

A deep linear network represents one matrix through a product of layer matrices:

$$x \longmapsto W_N W_{N-1} \cdots W_1 x,$$

where for layer widths $d_0 = d_x, d_1, \ldots, d_{N-1}, d_N = d_y$

$$W_j \in \mathbb{R}^{d_j \times d_{j-1}},$$

Fully connected linear networks

A deep linear network represents one matrix through a product of layer matrices:

$$x \longmapsto W_N W_{N-1} \cdots W_1 x,$$

where for layer widths $d_0 = d_x, d_1, \ldots, d_{N-1}, d_N = d_y$

$$W_j \in \mathbb{R}^{d_j \times d_{j-1}},$$

Recall $X = [x_1 \ \cdots \ x_m] \in \mathbb{R}^{d_x \times m}, \qquad Y = [y_1 \ \cdots \ y_m] \in \mathbb{R}^{d_y \times m}.$

Fully connected linear networks

A deep linear network represents one matrix through a product of layer matrices:

$$x \longmapsto W_N W_{N-1} \cdots W_1 x,$$

where for layer widths $d_0 = d_x, d_1, \ldots, d_{N-1}, d_N = d_y$

$$W_j \in \mathbb{R}^{d_j \times d_{j-1}},$$

Recall $X = [x_1 \ \cdots \ x_m] \in \mathbb{R}^{d_x \times m}, \qquad Y = [y_1 \ \cdots \ y_m] \in \mathbb{R}^{d_y \times m}.$

Definition (Loss function)

$$\mathcal{L}_N(W_1, \ldots, W_N) := \frac{1}{2} \| Y - W_N \cdots W_1 X \|_F^2 ,$$

Fully connected linear networks

A deep linear network represents one matrix through a product of layer matrices:

$$x \longmapsto W_N W_{N-1} \cdots W_1 x,$$

where for layer widths $d_0 = d_x, d_1, \ldots, d_{N-1}, d_N = d_y$

$$W_j \in \mathbb{R}^{d_j \times d_{j-1}},$$

Recall $X = [x_1 \ \cdots \ x_m] \in \mathbb{R}^{d_x \times m}, \qquad Y = [y_1 \ \cdots \ y_m] \in \mathbb{R}^{d_y \times m}.$

Definition (Loss function)

$$\mathcal{L}_N(W_1, \ldots, W_N) := \frac{1}{2} \| Y - W_N \cdots W_1 X \|_F^2 ,$$

Setting

$$W := W_N \cdots W_1,$$

obtain

$$\mathcal{L}_N(W_1, \ldots, W_N) = \mathcal{L}_1(W).$$

Gradient flow

Training is modeled by the gradient flow

$$\dot{W}_j = -\nabla_{W_j} \mathcal{L}_N(W_1, \ldots, W_N), \qquad j = 1, \ldots, N.$$

Calculating derivaties gives

$$\nabla_{W_j} \mathcal{L}_N = W_{j+1}^\top \cdots W_N^\top \, \nabla \mathcal{L}_1(W) \, W_1^\top \cdots W_{j-1}^\top,$$

and $\nabla \mathcal{L}_1(W) = WXX^\top - YX^\top$.

Gradient flow

Training is modeled by the gradient flow

$$\dot{W}_j = -\nabla_{W_j} \mathcal{L}_N(W_1, \ldots, W_N), \qquad j = 1, \ldots, N.$$

Calculating derivaties gives

$$\nabla_{W_j} \mathcal{L}_N = W_{j+1}^\top \cdots W_N^\top \, \nabla \mathcal{L}_1(W) \, W_1^\top \cdots W_{j-1}^\top,$$

and $\nabla \mathcal{L}_1(W) = WXX^\top - YX^\top$.

Decreasing loss

$$\frac{d}{dt} \mathcal{L}_N(W_1(t), \ldots, W_N(t)) = -\sum_{j=1}^{N} \left\| \nabla_{W_j} \mathcal{L}_N \right\|_F^2 \leq 0.$$

The loss decreases, but the factorized objective is nonconvex and has many equivalent parameterizations.

Conserved Gram differences and Balancedness

Fact: Along every factor gradient flow,

$$W_{j+1}^\top(t)W_{j+1}(t) - W_j(t)W_j^\top(t)$$

remains constant.

Conserved Gram differences and Balancedness

Fact: Along every factor gradient flow,

$$W_{j+1}^\top(t)W_{j+1}(t) - W_j(t)W_j^\top(t)$$

remains constant. Hence also the quantities

$$\|W_j(t)\|_F^2 - \|W_i(t)\|_F^2$$

are constant.

Conserved Gram differences and Balancedness

Fact: Along every factor gradient flow,

$$W_{j+1}^\top(t)W_{j+1}(t) - W_j(t)W_j^\top(t)$$

remains constant. Hence also the quantities

$$\|W_j(t)\|_F^2 - \|W_i(t)\|_F^2$$

are constant.

Definition (Balanced factors)

A tuple $(W_1, \dots, W_N)$ is *balanced* if

$$W_{j+1}^\top W_{j+1} = W_j W_j^\top, \qquad j = 1, \dots, N-1.$$

Note: Balanced initial data stay balanced.

Every trajectory converges to a critical point

Theorem (Global existence and convergence)

Assume $XX^\top$ has full rank. For every initialization, the factor flow is defined and bounded for all $t \geq 0$, and

$$(W_1(t), \ldots, W_N(t)) \longrightarrow (W_1^\star, \ldots, W_N^\star),$$

where $(W_1^\star, \ldots, W_N^\star)$ is a critical point of $\mathcal{L}_N$. Consequently, $W(t) = W_N(t) \cdots W_1(t)$ also converges.

Note: Balancedness is not assumed. The theorem guarantees convergence to a critical point, not yet to a global minimizer.

Theorem 5 in B. Bah, H. Rauhut, U. Terstiege, M. Westdickenberg: Learning deep linear neural networks: riemannian gradient flows and convergence to global minimizers. Information and Inference: A Journal of the IMA 11 (1), pp. 307–353.

Why the theorem holds

1. Loss dissipation bounds $\|W(t)X\|_F$. Since $XX^\top$ is invertible,

$$\|W(t)\|_F \leq \left(\|W(t)X - Y\|_F + \|Y\|_F\right) \left\|X^\top (XX^\top)^{-1}\right\|_F .$$

Why the theorem holds

1. Loss dissipation bounds $\|W(t)X\|_F$. Since $XX^\top$ is invertible,

$$\|W(t)\|_F \leq \left(\|W(t)X - Y\|_F + \|Y\|_F\right) \left\|X^\top (XX^\top)^{-1}\right\|_F .$$

2. Conserved Gram differences yield constants $C_i, \widetilde{C}_i > 0$ such that

$$\|W_i(t)\|_F \leq C_i \|W(t)\|_F^{1/N} + \widetilde{C}_i.$$

Thus every factor remains bounded and no finite-time blow-up occurs.

Why the theorem holds

1. Loss dissipation bounds $\|W(t)X\|_F$. Since $XX^\top$ is invertible,

$$\|W(t)\|_F \leq \left(\|W(t)X - Y\|_F + \|Y\|_F\right) \left\|X^\top (XX^\top)^{-1}\right\|_F .$$

2. Conserved Gram differences yield constants $C_i, \widetilde{C}_i > 0$ such that

$$\|W_i(t)\|_F \leq C_i \|W(t)\|_F^{1/N} + \widetilde{C}_i.$$

   Thus every factor remains bounded and no finite-time blow-up occurs.

3. $\mathcal{L}_N$ is a real polynomial. By the Łojasiewicz convergence theorem, every bounded gradient-flow trajectory converges to a single critical point.

Łojasiewicz convergence theorem

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R}$ be analytic, and let $x : [0, \infty) \to \mathbb{R}^n$ be a bounded solution of the gradient-flow equation

$$\dot{x}(t) = -\nabla f(x(t)).$$

Then there exists a critical point $x_\infty \in \mathbb{R}^n$ of f such that

$$x(t) \longrightarrow x_\infty \qquad \text{as } t \to \infty.$$

Product dynamics under balancedness

Define, for $Z \in \mathbb{R}^{d_y \times d_x}$,

Definition (Depth-dependent operator)

$$\mathcal{A}_W(Z) := \sum_{j=1}^{N} (WW^\top)^{\frac{N-j}{N}} Z (W^\top W)^{\frac{j-1}{N}},$$

Closed equation for the product

For balanced initial factors,

$$\dot{W}(t) = -\mathcal{A}_{W(t)}\big(\nabla \mathcal{L}_1(W(t))\big).$$

The fixed-rank manifold

Definition

For $0 \leq k \leq \min(d_x, d_y)$ let

$$\mathcal{M}_k := \{W \in \mathbb{R}^{d_y \times d_x} : \mathrm{rank}(W) = k\}.$$

The fixed-rank manifold

Definition

For $0 \leq k \leq \min(d_x, d_y)$ let

$$\mathcal{M}_k := \{W \in \mathbb{R}^{d_y \times d_x} : \operatorname{rank}(W) = k\}.$$

Fact: $\mathcal{M}_k$ is a smooth manifold of dimension $k(d_x + d_y - k)$.

The fixed-rank manifold

Definition

For $0 \leq k \leq \min(d_x, d_y)$ let

$$\mathcal{M}_k := \{W \in \mathbb{R}^{d_y \times d_x} : \operatorname{rank}(W) = k\}.$$

Fact: $\mathcal{M}_k$ is a smooth manifold of dimension $k(d_x + d_y - k)$.

Theorem

Assume $XX^\top$ has full rank and the initial factors are balanced. If $W(0) \in \mathcal{M}_k$, then $W(t) \in \mathcal{M}_k$ for every finite $t \geq 0$.

The fixed-rank manifold

Definition

For $0 \leq k \leq \min(d_x, d_y)$ let

$$\mathcal{M}_k := \{W \in \mathbb{R}^{d_y \times d_x} : \operatorname{rank}(W) = k\}.$$

Fact: $\mathcal{M}_k$ is a smooth manifold of dimension $k(d_x + d_y - k)$.

Theorem

Assume $XX^\top$ has full rank and the initial factors are balanced. If $W(0) \in \mathcal{M}_k$, then $W(t) \in \mathcal{M}_k$ for every finite $t \geq 0$.

Note: The limiting matrix may have rank below k because $\mathcal{M}_k$ is not closed in $\mathbb{R}^{d_y \times d_x}$.

The Riemannian metric induced by depth

Let T_W be the tangent space of $\mathcal{M}_k$ (as a submanifold of $\mathbb{R}^{d_y \times d_x}$) at W.

For $W \in \mathcal{M}_k$, the operator $\mathcal{A}_W$ has

$$\mathrm{im}(\mathcal{A}_W) = T_W\mathcal{M}_k, \qquad \ker(\mathcal{A}_W) = (T_W\mathcal{M}_k)^{\perp}.$$

Its restriction

$$\overline{\mathcal{A}}_W : T_W\mathcal{M}_k \to T_W\mathcal{M}_k$$

is self-adjoint and positive definite.

Definition (Metric on $\mathcal{M}_k$)

For $Z_1, Z_2 \in T_W\mathcal{M}_k$, set

$$g_W(Z_1, Z_2) := \left\langle \overline{\mathcal{A}}_W^{-1} Z_1, Z_2 \right\rangle_F .$$

The product flow is a Riemannian gradient flow

For differentiable $f : \mathbb{R}^{d_y \times d_x} \to \mathbb{R}$, we set

$$\nabla^g f(W) = \mathcal{A}_W(\nabla f(W)) \in T_W \mathcal{M}_k,$$

since, for every $Z \in T_W \mathcal{M}_k$,

$$g_W(\nabla^g f(W), Z) = \langle \nabla f(W), Z \rangle_F = Df(W)[Z].$$

The product flow is a Riemannian gradient flow

For differentiable $f : \mathbb{R}^{d_y \times d_x} \to \mathbb{R}$, we set

$$\nabla^g f(W) = \mathcal{A}_W(\nabla f(W)) \in T_W \mathcal{M}_k,$$

since, for every $Z \in T_W \mathcal{M}_k$,

$$g_W(\nabla^g f(W), Z) = \langle \nabla f(W), Z \rangle_F = Df(W)[Z].$$

Theorem

Assume $XX^\top$ has full rank and the factor flow starts balanced. If $W(0) \in \mathcal{M}_k$, then the product is the unique global solution on $\mathcal{M}_k$ of

$$\dot{W} = -\nabla^g \mathcal{L}_1(W), \qquad W(0) \in \mathcal{M}_k.$$

Strict saddles and almost-sure avoidance

Definition (Strict saddle)

A critical point x of a C^2-function $F : \mathbb{R}^n \to \mathbb{R}$ is a strict saddle point if $\operatorname{Hess} F(x)$ has a negative eigenvalue.

Strict saddles and almost-sure avoidance

Definition (Strict saddle)

A critical point x of a C^2-function $F : \mathbb{R}^n \to \mathbb{R}$ is a strict saddle point if $\operatorname{Hess} F(x)$ has a negative eigenvalue.

Theorem (Gradient flows avoid strict saddles)

let F be C^2. Assume its gradient flow $\psi_t(a)$ (i.e. $\dot{\psi}_t(a) = -\nabla F(\psi_t(a))$ and $\psi_0(a) = a$) exists for every $a \in \mathbb{R}^n$ and $t \geq 0$. Then

$$\{a \in \mathbb{R}^n : \lim_{t \to \infty} \psi_t(a) \text{ is a strict saddle of } F\}$$

has Lebesgue measure zero.

Critical points on each rank stratum

Let

$$Q := YX^\top (XX^\top)^{-1/2} = \sum_{i=1}^{q} \sigma_i u_i v_i^\top, \qquad \sigma_1 \geq \cdots \geq \sigma_q > 0.$$

Theorem (Critical-point classification)

The critical points of $\mathcal{L}_1|_{\mathcal{M}_k}$ are exactly

$$W_J = \left(\sum_{j \in J} \sigma_j u_j v_j^\top \right) (XX^\top)^{-1/2}, \qquad |J| = k.$$

Their loss values are

$$\mathcal{L}_1(W_J) = \frac{1}{2} \left(\operatorname{tr}(YY^\top) - \sum_{j \in J} \sigma_j^2 \right).$$

A critical point is globally minimizing on $\mathcal{M}_k$ exactly when J selects the k largest singular values. If $k > q$, $\mathcal{L}_1|_{\mathcal{M}_k}$ has no critical point.

Transfer from products to factors

Theorem (Strict-saddle structure)

Assume $XX^\top$ has full rank.

1. *Every critical point of $\mathcal{L}_1|_{\mathcal{M}_k}$ that is not globally minimizing on $\mathcal{M}_k$ is a strict saddle.*
2. *If $(W_1, \ldots, W_N)$ is critical for $\mathcal{L}_N$ and $W = W_N \cdots W_1$ has rank k, then W is critical for $\mathcal{L}_1|_{\mathcal{M}_k}$.*
3. *If this product W is not globally minimizing on $\mathcal{M}_k$, then the factor tuple is a strict saddle of $\mathcal{L}_N$.*

Main global-convergence result

Let $r = \min_{0 \le j \le N} d_j$, $q = \operatorname{rank}(Q)$, and $\bar{r} = \min\{r, q\}$.

Theorem (Almost-everywhere convergence)

Assume $XX^\top$ has full rank.

1. *For almost every initialization, the factor flow converges to a critical tuple whose product $W^\star$ globally minimizes $\mathcal{L}_1$ on $\mathcal{M}_k$, where*

$$k = \operatorname{rank}(W^\star) \in \{0, \dots, \bar{r}\}$$

   may depend on the initialization.

2. *If $N = 2$, then for almost every initialization the factor flow converges to a global minimizer of $\mathcal{L}_2$.*

"Almost every" refers to Lebesgue measure on the product of all layer-matrix spaces.

Balanced product flow

Theorem (Generic behavior on $\mathcal{M}_k$)

Assume $XX^\top$ has full rank. For almost every $W(0) \in \mathcal{M}_k$, the Riemannian flow

$$\dot{W} = -\nabla^g \mathcal{L}_1(W)$$

converges either

- *to a global minimum of $\mathcal{L}_1|_{\mathcal{M}_k}$, or*
- *to a critical point on a lower-rank stratum $\mathcal{M}_\ell$ with $\ell < k$.*

Linear convolutional networks

A linear convolutional network (LCN) is a linear neural network in which every layer is a convolutional map.

A convolutional layer

$$\alpha_i : \mathbb{R}^{d_{i-1}} \longrightarrow \mathbb{R}^{d_i}$$

with filter

$$w^{(i)} = \left(w_1^{(i)}, \ldots, w_{k_i}^{(i)}\right) \in \mathbb{R}^{k_i},$$

filter width k_i and stride s_i is given by

$$\big(\alpha_i(x)\big)_j = \sum_{n=1}^{k_i} w_n^{(i)}\, x_{(j-1)s_i+n}, \qquad j = 1, \ldots, d_i.$$

Example

Recall $\big(\alpha_i(x)\big)_j = \sum_{n=1}^{k_i} w_n^{(i)}\, x_{(j-1)s_i+n}, \qquad j = 1, \ldots, d_i.$

Consider

$$\alpha : \mathbb{R}^8 \longrightarrow \mathbb{R}^3, \qquad w = (w_1, w_2, w_3, w_4),$$

with filter width $k = 4$ and stride $s = 2$.

Then for

$$W = \begin{pmatrix} w_1 & w_2 & w_3 & w_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & w_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & w_3 & w_4 \end{pmatrix} \in \mathbb{R}^{3\times 8}.$$

we have

$$\alpha(x) = Wx.$$

Architecture of a linear convolutional network

Let

$$\Pi_{(d_{i-1},d_i),k_i,s_i} : \mathbb{R}^{k_i} \longrightarrow \mathbb{R}^{d_i \times d_{i-1}}$$

denote the map sending a filter $w^{(i)}$ to its corresponding convolution matrix.

For a network of depth N, write

$$\mathbf{d} = (d_0, \ldots, d_N), \qquad \mathbf{k} = (k_1, \ldots, k_N), \qquad \mathbf{s} = (s_1, \ldots, s_N),$$

with

$$d_i = \frac{d_{i-1} - k_i}{s_i} + 1, \qquad i = 1, \ldots, N.$$

The end-to-end linear map is

$$W = W^{(N)} \cdots W^{(1)}, \qquad W^{(i)} = \Pi_{(d_{i-1},d_i),k_i,s_i}\left(w^{(i)}\right).$$

Set of realizable end-to-end matrices is

$$\mathcal{M}_{(\mathbf{d},\mathbf{k},\mathbf{s})} = \left\{ W \in \mathbb{R}^{d_N \times d_0} \;\middle|\; W = \prod_{i=1}^{N} \Pi_{(d_{i-1},d_i),k_i,s_i}\left(w^{(i)}\right), \quad w^{(i)} \in \mathbb{R}^{k_i} \right\}.$$

Compositions of convolutions are convolions again

A depth-N linear convolutional network is a composition of convolutional layers.

Fact: Such a composition is a again a convolution.

Compositions of convolutions are convolions again

A depth-N linear convolutional network is a composition of convolutional layers.

Fact: Such a composition is a again a convolution.

More precisely, the composition of N convolutions $\alpha_1, \ldots, \alpha_N$ with filter sizes $\mathbf{k} = (k_1, \ldots, k_N)$ and strides $\mathbf{s} = (s_1, \ldots, s_N)$ is a convolution α_v with stride s_v and filter width k_s given by

$$s_v = \prod_{i=1}^{N} s_i \quad \text{and} \quad k_v = k_1 + \sum_{i=2}^{N} (k_i - 1) \prod_{m=1}^{i-1} s_m.$$

We denote by $v = \pi_{(\mathbf{d},\mathbf{k},\mathbf{s})}(w^{(1)}, ..., w^{(N)})$ the end-to-end filter.

K. Kohn, T. Merkh, G. Montufar, and M. Trager. "Geometry of linear convolutional networks". In: SIAM Journal on Applied Algebra and Geometry 6.3 (2022)

Main convergence theorem

For $\boldsymbol{w} = (w^1, ..., w^N)$, let $\Pi(\boldsymbol{w}) = \prod_{i=1}^{N} \Pi_{(d_{i-1},d_i),k_i,s_i}\left(w^{(i)}\right)$.

Given data $X = [x_1, \ldots, x_m]$ and $Y = [y_1, \ldots, y_m]$, consider empirical loss

$$\mathcal{L}(\boldsymbol{w}) = \sum_{r=1}^{m} \ell\big(\Pi(\boldsymbol{w})x_r, y_r\big), \qquad \dot{\boldsymbol{w}}(t) = -\nabla\mathcal{L}(\boldsymbol{w}(t)).$$

Main convergence theorem

For $\boldsymbol{w} = (w^1, ..., w^N)$, let $\Pi(\boldsymbol{w}) = \prod_{i=1}^{N} \Pi_{(d_{i-1},d_i),k_i,s_i}\left(w^{(i)}\right)$.

Given data $X = [x_1, \ldots, x_m]$ and $Y = [y_1, \ldots, y_m]$, consider empirical loss

$$\mathcal{L}(\boldsymbol{w}) = \sum_{r=1}^{m} \ell\big(\Pi(\boldsymbol{w})x_r, y_r\big), \qquad \dot{\boldsymbol{w}}(t) = -\nabla\mathcal{L}(\boldsymbol{w}(t)).$$

Theorem

Assume $XX^\top$ is invertible and let ℓ be the square loss. For every initialization $\boldsymbol{w}(0)$, the gradient-flow trajectory

$$\dot{\boldsymbol{w}}(t) = -\nabla\mathcal{L}(\boldsymbol{w}(t))$$

is defined for all $t \geq 0$, remains bounded, and converges to a critical point:

$$\boldsymbol{w}(t) \longrightarrow \boldsymbol{w}_\infty, \qquad \nabla\mathcal{L}(\boldsymbol{w}_\infty) = 0.$$

J. Diederen, H. Rauhut, U. Terstiege: Convergence of gradient flow for learning convolutional neural networks, https://arxiv.org/pdf/2601.08547

The theorem also holds for other losses, e.g.:

- $\ell_p(z, y) = \frac{1}{p}\|z - y\|_p^p$ for $p \geq 2$
- pseudo-Huber loss;
- generalized Huber loss with parameters $\alpha > 0$, $\beta > -2$;
- log cosh loss.

Proof strategy: boundedness is the hard part

The proof follows the same high-level template as for fully connected deep linear networks:

$$\boxed{\text{loss decreases}} \Longrightarrow \boxed{\text{final filter bounded}} \Longrightarrow \boxed{\text{all layer filters bounded}} \Longrightarrow \boxed{\text{convergence}}.$$

The last implication is supplied by Łojasiewicz' theorem .

Thus the architecture-specific problem is:

How does a bound on the end-to-end convolution control all individual filters?

Proof strategy: boundedness is the hard part

The proof follows the same high-level template as for fully connected deep linear networks:

$$\boxed{\text{loss decreases}} \Longrightarrow \boxed{\text{final filter bounded}} \Longrightarrow \boxed{\text{all layer filters bounded}} \Longrightarrow \boxed{\text{convergence}}.$$

The last implication is supplied by Łojasiewicz' theorem .

Thus the architecture-specific problem is:

How does a bound on the end-to-end convolution control all individual filters?

The key tools are

- conserved differences of layer norms;
- a polynomial representation of convolutions.

Conservation law from rescaling symmetry

The end-to-end filter is invariant under opposite rescaling of two layers:

$$w^{(i)} \mapsto \kappa w^{(i)}, \qquad w^{(j)} \mapsto \kappa^{-1} w^{(j)}.$$

Differentiating this symmetry along gradient flow yields

$$\boxed{\delta_{ij}(t) := \left\| w^{(i)}(t) \right\|_2^2 - \left\| w^{(j)}(t) \right\|_2^2 = \delta_{ij}(0).}$$

Consequence: It suffices to bound *one* layer norm.

But a bounded product does not by itself bound the factors:

$(\kappa w^{(i)}, \kappa^{-1} w^{(j)})$ represents the same network for every $\kappa \neq 0$.

K. Kohn, T. Merkh, G. Montúfar, and M. Trager. "Geometry of linear convolutional networks". In: SIAM Journal on Applied Algebra and Geometry 6.3 (2022)

The algebraic trick: convolution becomes polynomial multiplication

Associate to each layer filter the sparse polynomial

$$\widetilde{p}_i(z) = \sum_{j=0}^{k_i-1} w_{j+1}^{(i)} z^{j \prod_{n=1}^{i-1} s_n}.$$

If v is the final filter (of dimension k_v), then for $p_v(z) := \sum_{i=0}^{k_v-1} v_{i+1} z^i$ we have

$$\boxed{p_v(z) = \prod_{i=1}^{N} \widetilde{p}_i(z).}$$

K. Kohn, T. Merkh, G. Montúfar, and M. Trager. "Geometry of linear convolutional networks". In: SIAM Journal on Applied Algebra and Geometry 6.3 (2022)

The algebraic trick: convolution becomes polynomial multiplication

Associate to each layer filter the sparse polynomial

$$\widetilde{p}_i(z) = \sum_{j=0}^{k_i-1} w_{j+1}^{(i)} z^{j \prod_{n=1}^{i-1} s_n}.$$

If v is the final filter (of dimension k_v), then for $p_v(z) := \sum_{i=0}^{k_v-1} v_{i+1} z^i$ we have

$$\boxed{p_v(z) = \prod_{i=1}^{N} \widetilde{p}_i(z).}$$

K. Kohn, T. Merkh, G. Montúfar, and M. Trager. "Geometry of linear convolutional networks". In: SIAM Journal on Applied Algebra and Geometry 6.3 (2022)

Why useful in our situation?

A bound on the coefficients of p_v gives control of its roots. Unique factorization and conserved quantities δ_{ij} then constrain the possible layer polynomials $\widetilde{p}_i$.

Key boundedness lemma

Lemma

Fix the conserved quantities

$$\delta_i = \left\| w^{(i+1)} \right\|_2^2 - \left\| w^{(i)} \right\|_2^2.$$

If the final filter satisfies

$$\|v\|_1 \leq T,$$

then there exists

$$\tau = \tau(T, k_v, \delta_1, \ldots, \delta_{N-1})$$

such that

$$\left\| w^{(i)} \right\|_2^2 \leq \tau \qquad \forall i.$$

General structured linear networks

Let

$$N, m, d_0, \ldots, d_N \in \mathbb{N}.$$

For each $j = 1, \ldots, N$, consider fixed matrices

$$V_1^{(j)}, \ldots, V_{n_j}^{(j)} \in \mathbb{R}^{d_j \times d_{j-1}},$$

and define the linear parametrization

$$T_j : \mathbb{R}^{n_j} \longrightarrow \mathbb{R}^{d_j \times d_{j-1}}, \qquad T_j w = \sum_{k=1}^{n_j} w_k V_k^{(j)}.$$

General structured linear networks

Let

$$N, m, d_0, \dots, d_N \in \mathbb{N}.$$

For each $j = 1, \dots, N$, consider fixed matrices

$$V_1^{(j)}, \dots, V_{n_j}^{(j)} \in \mathbb{R}^{d_j \times d_{j-1}},$$

and define the linear parametrization

$$T_j : \mathbb{R}^{n_j} \longrightarrow \mathbb{R}^{d_j \times d_{j-1}}, \qquad T_j w = \sum_{k=1}^{n_j} w_k V_k^{(j)}.$$

Fix $X \in \mathbb{R}^{d_0 \times m}$, and $Y \in \mathbb{R}^{d_N \times m}$, and define for $w^{(1)} \in \mathbb{R}^{n_1}, \dots, w^{(N)} \in \mathbb{R}^{n_N}$, the empirical loss

$$\mathcal{L}(w^{(1)}, \dots, w^{(N)}) = \left\| T_N w^{(N)} \cdots T_1 w^{(1)} X - Y \right\|_F^2.$$

Consider gradient flow

$$\dot{w}^{(j)} = -\nabla_{w^{(j)}} \mathcal{L}(w^{(1)}, ..., w^{(N)}), \qquad j = 1, \dots, N.$$

Convergence?

Does this flow always converge to a critical point?

Convergence?

Does this flow always converge to a critical point?

No!

Counterexample: For

$$L(w^{(1)}, w^{(2)}) = \frac{1}{2} \left\| \begin{pmatrix} w_1^{(2)} & 0 & 0 \\ w_2^{(2)} & w_1^{(2)} & 0 \\ 0 & 0 & w_1^{(2)} \end{pmatrix} \cdot \begin{pmatrix} w_1^{(1)} & w_2^{(1)} & 0 \\ 0 & w_1^{(1)} & 0 \\ 0 & 0 & w_1^{(1)} \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \right\|_F^2$$

and suitable initialization (e.g. $1/2$ for the diagonal entries $w_1^{(1)}$, $w_1^{(2)}$ and 2 for $w_2^{(1)}$ and $w_2^{(2)}$) trajectories are unbounded.

Convergence?

Does this flow always converge to a critical point?

No!

Counterexample: For

$$L(w^{(1)}, w^{(2)}) = \frac{1}{2} \left\| \begin{pmatrix} w_1^{(2)} & 0 & 0 \\ w_2^{(2)} & w_1^{(2)} & 0 \\ 0 & 0 & w_1^{(2)} \end{pmatrix} \cdot \begin{pmatrix} w_1^{(1)} & w_2^{(1)} & 0 \\ 0 & w_1^{(1)} & 0 \\ 0 & 0 & w_1^{(1)} \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \right\|_F^2$$

and suitable initialization (e.g. $1/2$ for the diagonal entries $w_1^{(1)}$, $w_1^{(2)}$ and 2 for $w_2^{(1)}$ and $w_2^{(2)}$) trajectories are unbounded.

But the loss still converges to 0!

Generalized critical values

Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable.

Let

$$\mathrm{Crit}(f) := \{\theta \in \mathbb{R}^d : \nabla f(\theta) = 0\},$$

be the set of critical points of f and

$$K_0(f) := \{f(\theta) : \theta \in \mathrm{Crit}(f)\}.$$

be the set of its critical values.

Generalized critical values

Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable.

Let

$$\mathrm{Crit}(f) := \{\theta \in \mathbb{R}^d : \nabla f(\theta) = 0\},$$

be the set of critical points of f and

$$K_0(f) := \{f(\theta) : \theta \in \mathrm{Crit}(f)\}.$$

be the set of its critical values.

Asymptotic critical values

A value $c \in \mathbb{R}$ belongs to $K_\infty(f)$ if there exists a sequence $(\theta_n)_{n\in\mathbb{N}} \subset \mathbb{R}^d$ such that

$$\|\theta_n\| \to \infty, \qquad f(\theta_n) \to c, \qquad \|\theta_n\| \, \|\nabla f(\theta_n)\| \to 0.$$

The generalized critical values are

$$K(f) := K_0(f) \cup K_\infty(f).$$

Dichotomy for gradient flows

Theorem (Kranz, Gallon, Dereich, Jentzen)

Let ψ be a C^1 definable activation and let $x \mapsto \mathcal{N}_w(x)$ be a neural network map with activation function ψ (and identity in the last layer).

Let

$$f : \mathbb{R}^{d_0} \to \mathbb{R}^{d_N}, \qquad \ell : \mathbb{R}^{d_N} \times \mathbb{R}^{d_N} \to [0, \infty)$$

be C^1 definable functions, and let μ be a probability measure on $\mathbb{R}^{d_x}$ with expected loss

$$\mathcal{L}(w) = \mathbb{E}_{x \sim \mu}\Big[\ell\big(\mathcal{N}_w(x), f(x)\big)\Big].$$

Assume that either $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, or $d\mu(x) = p(x)\,dx$ for a C^1 density p such that $\mathcal{L}$ is definable, or that μ is a convex combination of these two cases.

Assume moreover that ℓ, f, and ψ have locally Lipschitz derivatives. Let $d(\mathcal{N}_w)$ be the number of parameters of N_w

For every initialization $w_0 \in \mathbb{R}^{d(\mathcal{N}_w)}$, there exists a unique C^1-function $w : [0, \infty) \to \mathbb{R}^{d(\mathcal{N}_w)}$ satisfying

$$w(0) = w_0, \qquad \dot{w}(t) = -\nabla \mathcal{L}(w(t)).$$

For every such trajectory w, exactly one of the following holds:

(a) $\lim_{t \to \infty} w(t)$ exists and is a critical point of $\mathcal{L}$,

(b) $\|w(t)\| \to \infty$ and $\lim_{t \to \infty} \mathcal{L}(w(t))$ is an asymptotic critical value of $\mathcal{L}$.