Optimal Neural Network Approximation of Wasserstein Gradient Direction via Convex Optimization

Yifei Wang

Department of Electrical Engineering, Stanford University

Sep. 27th, SIAM MDS22

Joint work with Peng Chen (Gatech), Mert Pilanci (Stanford) and Wuchen Li (University of South Carolina)

Too long; didn't read

How Wasserstein Gradient Flows meet
Neural Networks and Convex Optimization?

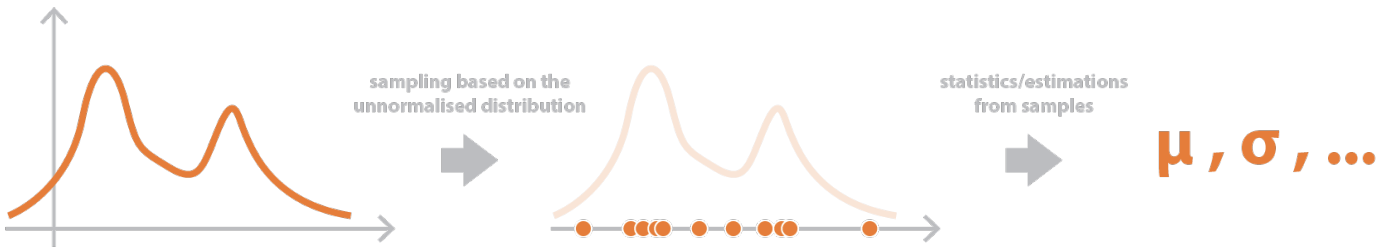
Too long; didn't read

How Wasserstein Gradient Flows meet
Neural Networks and Convex Optimization?

Bayesian inference

- A powerful tool in
  - Modeling complex data
  - Quantifying uncertainty
- Of great interests in inverse problems, information science, physics and scientific computing.
- Main problem
  - Given a prior distribution p_0 and the likelihood function $f(x)$, generate samples from the posterior distribution

$$\pi(x) = p_0(x)f(x).$$



Wasserstein gradient flow

- Effective in drawing samples from a posterior distribution

$$\begin{aligned}\partial_t \rho_t =& \Delta \rho_t - \nabla \cdot (\rho_t \nabla \log \pi) \\ =& \nabla \cdot (\rho_t (\nabla \log \rho_t - \nabla \log \pi)).\end{aligned}$$

- Corresponds to the continuous-time limit of the classical Langevin Monte Carlo Markov Chain (MCMC) algorithm.
- The Wasserstein gradient direction also provides a deterministic update of the particle system,
  - Wasserstein gradient descent (WGD) with kernel density estimation (KDE)[1].
  - Stein variational gradient descent (SVGD)[2].
  - Neural variational gradient descent[3].

[1] Liu, C. et al., Understanding and accelerating particle-based variational inference, ICML 2019.

[2] Liu, Q. and Wang, D., Stein variational gradient descent: A general purpose bayesian inference algorithm, Neurips 2016.

[3] di Langosco et al., Neural variational gradient descent, 2021.

Too long; didn't read

How Wasserstein Gradient Flows meet
Neural Networks and Convex Optimization?

Neural networks

- Exhibit tremendous optimization and generalization performance in learning complicated functions from data.
- Wide applications in Bayesian inverse problems[1].
- Arbitrarily complicated functions can be learned by a two-layer neural network with non-linear activations and a sufficient number of neurons[2].
- Functions represented by neural networks can approximate the Wasserstein gradient direction.

[1] Rezende, D. and Mohamed, S. Variational inference with normalizing flows, ICML 2015.

[2] The universal approximation theorem of neural networks

Too long; didn't read

How Wasserstein Gradient Flows meet
Neural Networks and Convex Optimization?

Convex optimization formulation of neural networks

- Directly training the neural network to miminize the loss may get the neural network stuck at local minima or saddle points and it often leads to biased sample distribution from the posterior.
- In a line of works, the regularized training problem of two-layer neural networks with ReLU[1][2]/polynomial[3] activation can be formulated as a convex program.
- The optimal solution of the convex program renders a global optimum of the nonconvex training problem.

[1] Pilanci, M. and Ergen, T. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks, ICML 2020.

[2] Sahiner, A. et al., Vector-output relu neural network problems are compositive programs: Convex analysis of two layer networks and polynomial-time algorithms, ICLR 2021.

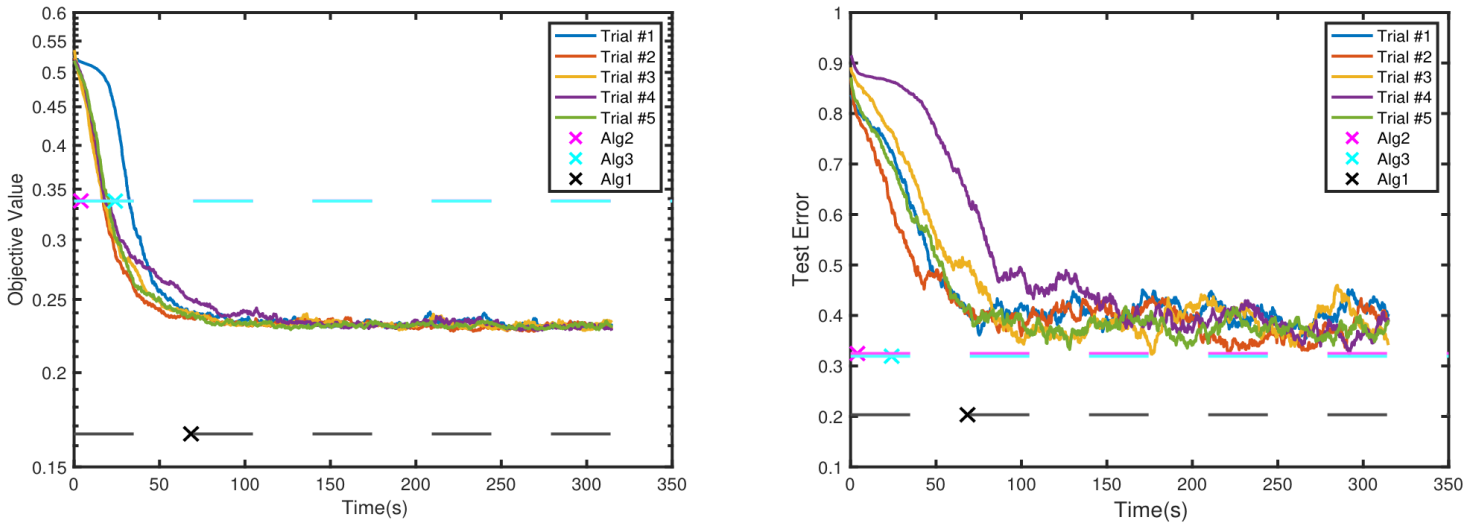
[3] Bartan, B. and Pilanci, M. Neural spectrahedra and semidefinite lifts: Global convex optimization of polynomial activation neural networks in fully polynomial-time, 2021.

Our contribution

- Formulate the Wasserstein gradient direction as an optimal solution of a variational problem.
- Analyze the convex dual problem of the training problem and study its semi-definite program (SDP) relaxation by analyzing the geometry of dual constraints.
- Present practical implementation and analyze the choice of the regularization parameter.

Why convex optimization formulations of NNs?

- All globally optimal NNs can be found by solving the convex program[1].
- Globally optimal NNs have great generalization property.



(a) Training objective value

(b) Test error

Figure: Two-layer CNN training on a subset of CIFAR-10 ($n = 195$ and filter size $4 \times 4 \times 3$). Alg1: convex program. Alg2 and Alg3: approximations.

[1] Wang, Y., Lacotte, J., Pilanci, M., The Hidden Convex Optimization Landscape of Two-Layer ReLU Neural Networks: an Exact Characterization of the Optimal Solutions, International Conference on Learning Representations, ICLR 2022 Oral.

Regularized training problem

- Data: $X \in \mathbb{R}^{n \times d}$ label: $y \in \mathbb{R}^n$.
- Consider the ℓ_2-regularized training problem with squared loss:

$$p_{\text{noncvx}} := \min_{W_1 \in \mathbb{R}^{d \times m}, w_2 \in \mathbb{R}^m} \left\{ \frac{1}{2} \left\| (XW_1)_+ w_2 - y \right\|^2 + \frac{\beta}{2} (\|W_1\|_F^2 + \|W_2\|_F^2) \right\} .$$

- Easy to extend to various convex loss functions, e.g., logistic, hinge.

Convex optimization formulation

- In recent work[1], an optimal neural network can be constructed based on a solution of the convex program

$$p_{\text{convex}} := \min_{(u_i, u_i')_{i=1}^p} \left\{ \frac{1}{2} \left\| \sum_{i=1}^p D_i X (u_i - u_i') - y \right\|_2^2 + \beta \sum_{i=1}^p (\|u_i\|_2 + \|u_i'\|_2) \right\},$$
$$\text{s.t.} \quad (2D_i - I_n) X u_i \geq 0, (2D_i - I_n) X u_i' \geq 0, i \in [p].$$

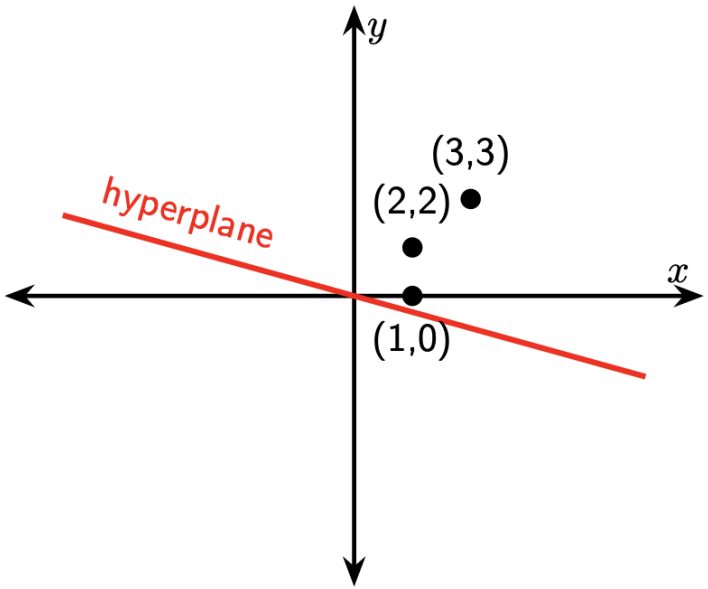
where $D_1, \ldots, D_p$ are the enumeration of all possible hyperplane arrangements

$$\{\text{diag}(\mathbf{1}(Xu \geqslant 0)) | u \in \mathbb{R}^d\}.$$

[1] Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. ICML 2020.

Hyperplane arrangements

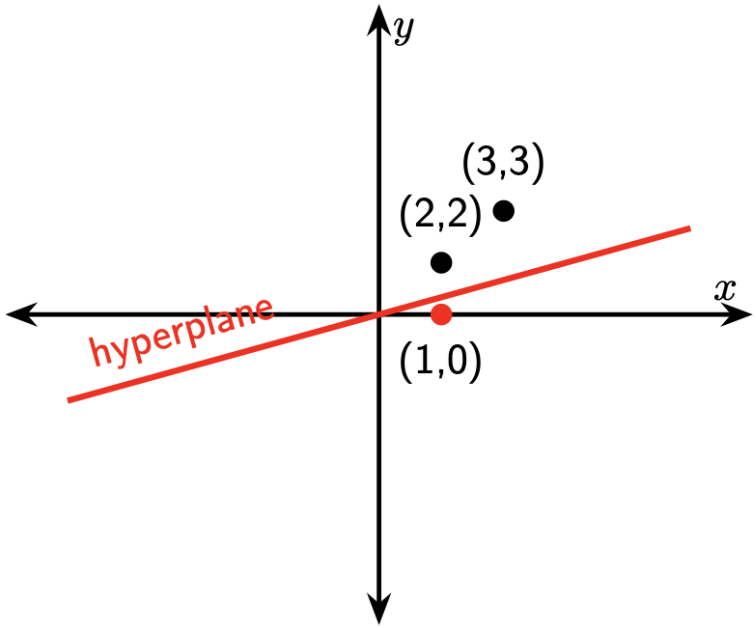
- $n = 3$ samples in $\mathbb{R}^d$, $d = 2$. $X = \begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 2 & 2 \\ 1 & 0 \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$.



$$D_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, D_1 X = \begin{bmatrix} 3 & 3 \\ 2 & 2 \\ 1 & 0 \end{bmatrix}.$$

Hyperplane arrangements

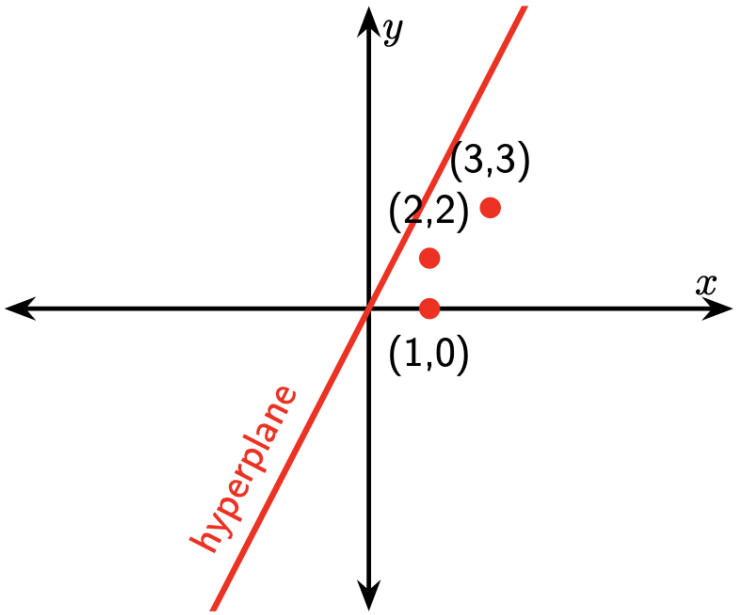
- $n = 3$ samples in $\mathbb{R}^d$, $d = 2$. $X = \begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 2 & 2 \\ 1 & 0 \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$.



$$D_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, D_2 X = \begin{bmatrix} 3 & 3 \\ 2 & 2 \\ 0 & 0 \end{bmatrix}.$$

Hyperplane arrangements

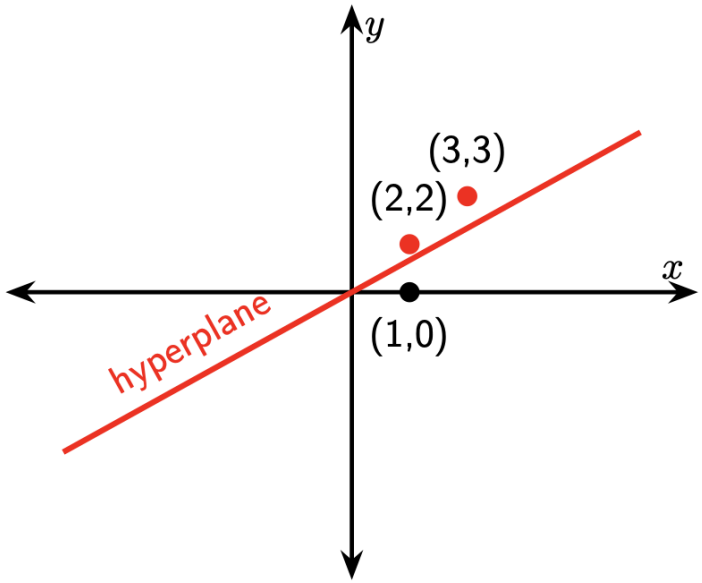
- $n = 3$ samples in $\mathbb{R}^d$, $d = 2$. $X = \begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 2 & 2 \\ 1 & 0 \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$.



$$D_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, D_3 X = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Hyperplane arrangements

- $n = 3$ samples in $\mathbb{R}^d$, $d = 2$. $X = \begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 2 & 2 \\ 1 & 0 \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$.



$$D_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, D_4 X = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Upperbound on the number of hyperplane arrangement patterns

- For $X \in \mathbb{R}^{N \times d}$, $p = \#\{\mathbf{1}(Xw \geqslant 0) | w \in \mathbb{R}^d\}$ is bounded by

$$p \leq 2r \left(\frac{e(N-1)}{r} \right)^r,$$

  where r is the rank of X.[1]
- For CNNs, the number of hyperplane arrangement patterns reduces to

$$O(r^3 (n/r)^{3r}),$$

  where r is the filter size, e.g., $r = 9$ for a 3×3 filter.

[1] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE transactions on electronic computers. 1965.

Sketch of the methology

- Step 1: rescale parameters to obtain the primal problem.

$$p = \min_{W_1, w_2} \frac{1}{2}\|(XW_1)_+ w_2 - y\|^2 + \beta\|w_2\|_1, \text{ s.t. } \|w_{1,i}\|_2 \leq 1, i \in [m].$$

Sketch of the methology

- Step 1: rescale parameters to obtain the primal problem.

$$p = \min_{W_1, w_2} \frac{1}{2}\|(XW_1)_+ w_2 - y\|^2 + \beta\|w_2\|_1, \text{ s.t. } \|w_{1,i}\|_2 \leq 1, i \in [m].$$

- Step 2: derive the dual problem by Lagrangian duality.

$$d = \max_{\lambda} -\frac{1}{2}\|\lambda - y\|_2^2 + \frac{1}{2}\|y\|_2^2, \text{ s.t. } \max_{w:\|w\|_2 \leq 1} |\lambda^T (Xw)_+| \leq 1$$

Sketch of the methology

- Step 1: rescale parameters to obtain the primal problem.

$$p = \min_{W_1, w_2} \frac{1}{2}\|(XW_1)_+ w_2 - y\|^2 + \beta\|w_2\|_1, \text{ s.t. } \|w_{1,i}\|_2 \leq 1, i \in [m].$$

- Step 2: derive the dual problem by Lagrangian duality.

$$d = \max_{\lambda} -\frac{1}{2}\|\lambda - y\|_2^2 + \frac{1}{2}\|y\|_2^2, \text{ s.t. } \max_{w:\|w\|_2 \leq 1} |\lambda^T (Xw)_+| \leq 1$$

- Step 3: rewrite the dual constraint in terms of hyperplane arrangements.

$$\begin{aligned} d = &\max_{\lambda} -\frac{1}{2}\|\lambda - y\|_2^2 + \frac{1}{2}\|y\|_2^2, \\ &\text{s.t. } \max_{w:\|w\|_2 \leq 1, (2D_i - I)Xw \geq 0} |\lambda^T (Xw)_+| \leq 1, i \in [p] \end{aligned}$$

Sketch of the methology

- Step 1: rescale parameters to obtain the primal problem.

$$p = \min_{W_1, w_2} \frac{1}{2}\|(XW_1)_+ w_2 - y\|^2 + \beta\|w_2\|_1, \text{ s.t. } \|w_{1,i}\|_2 \leq 1, i \in [m].$$

- Step 2: derive the dual problem by Lagrangian duality.

$$d = \max_{\lambda} -\frac{1}{2}\|\lambda - y\|_2^2 + \frac{1}{2}\|y\|_2^2, \text{ s.t. } \max_{w:\|w\|_2 \leq 1} |\lambda^T (Xw)_+| \leq 1$$

- Step 3: rewrite the dual constraint in terms of hyperplane arrangements.

$$d = \max_{\lambda} -\frac{1}{2}\|\lambda - y\|_2^2 + \frac{1}{2}\|y\|_2^2,$$
$$\text{s.t. } \max_{w:\|w\|_2 \leq 1, (2D_i - I)Xw \geq 0} |\lambda^T (Xw)_+| \leq 1, i \in [p]$$

- Step 4: derive the bi-dual problem p_{convex}!

Wasserstein gradient descent

- Consider an optimization problem in the probability space:

$$\inf_{\rho\in\mathcal{P}} \mathrm{D}_{\mathrm{KL}}(\rho\|\pi) = \int \rho(\log\rho - \log\pi)dx.$$

  π: a known probability density function of the posterior distribution.

- The Wasserstein gradient flow for the KL divergence satisfies

$$\begin{aligned}\partial_t\rho_t =& \nabla\cdot\left(\rho_t\nabla\frac{\delta}{\delta\rho_t}\mathrm{D}_{\mathrm{KL}}(\rho_t\|\pi)\right)\\ =& \nabla\cdot(\rho_t(\nabla\log\rho_t - \nabla\log\pi)).\end{aligned}$$

Sample update and discrete-time formulation

- Update in terms of samples:

$$dx_t = -(\nabla \log \rho_t(x_t) - \nabla \log \pi(x_t))dt.$$

  x_t follows the distribution of ρ_t.
- Wasserstein gradient descent (WGD) on the particle system $\{x_l^n\}$

$$x_{l+1}^n = x_l^n - \alpha_l \nabla \Phi_l(x_l^n),$$

  where $\{x_l^n\}$ are samples drawn from ρ_l and $\Phi_l : \mathbb{R}^d \to \mathbb{R}$ is a function which approximates $\log \rho_l - \log \pi$.

Variational formulation

- Let $\mathcal{H} \subseteq C^1(\mathbb{R}^d)$ be a finite dimensional function space. Consider the following variational problem

$$\inf_{\Phi \in \mathcal{H}} \frac{1}{2} \int \|\nabla \Phi - (\nabla \log \rho - \nabla \log \pi)\|_2^2 \rho dx.$$

or equivalently

$$\inf_{\Phi \in \mathcal{H}} \frac{1}{2} \int \|\nabla \Phi\|_2^2 \rho dx + \int \Delta \Phi \rho dx + \int \langle \nabla \log \pi, \nabla \Phi \rangle \rho dx.$$

- In terms of finite samples

$$\inf_{\Phi \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^N \left(\frac{1}{2} \|\nabla \Phi(x_n)\|_2^2 + \Delta \Phi(x_n) \right) + \frac{1}{N} \sum_{n=1}^N \langle \nabla \log \pi(x_n), \nabla \Phi(x_n) \rangle .$$

Two-layer neural networks

- Consider the case where $\mathcal{H}$ is a class of two-layer neural network with the activation function $\psi(z)$:

$$\mathcal{H} = \left\{ \Phi_{\boldsymbol{\theta}} \in C^1(\mathbb{R}^d) | \Phi_{\boldsymbol{\theta}}(x) = \alpha^T \psi(W^T x) \right\},$$

- The gradient and Laplacian of $\Phi \in \mathcal{H}$

$$\nabla \Phi_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{m} \alpha_i w_i \psi'(w_i^T x) = W(\psi'(W^T x) \circ \alpha),$$

$$\Delta \Phi_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{m} \alpha_i \|w_i\|_2^2 \psi''(w_i^T x).$$

Regularized variational problem

- Focus on the squared ReLU activation $\psi(z) = (z)_+^2 = (\max\{z, 0\})^2$
- This leads to the following training problem

$$\begin{aligned}
\min_{W,\alpha} \frac{1}{2N} \sum_{n=1}^{N} \left\| \sum_{i=1}^{m} \alpha_i w_i \psi'(w_i^T x_n) \right\|_2^2 \\
+ \frac{1}{N} \sum_{n=1}^{N} \left\langle \sum_{i=1}^{m} \alpha_i w_i \psi'(w_i^T x_n), \nabla \log \pi(x_n) \right\rangle \\
+ \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{m} \alpha_i \|w_i\|_2^2 \psi''(w_i^T x_n) + \frac{\beta}{2} \sum_{i=1}^{m} (\|w_i\|_2^3 + |\alpha_i|^3),
\end{aligned}$$

Proposition (Primal problem)

The regularized variational problem is equivalent to

$$\min_{W,\alpha,Z} \frac{1}{2}\|Z\|_F^2 + \sum_{n=1}^{N}\sum_{i=1}^{m} \alpha_i \|w_i\|_2^2 \psi''(w_i^T x_n) + \mathrm{tr}(Y^T Z) + \tilde{\beta}\|\alpha\|_1,$$

$$\text{s.t. } z_n = \sum_{i=1}^{m} \alpha_i w_i \psi'(x_n^T w_i), n \in [N], \|w_i\|_2 \le 1, i \in [m].$$

where $\tilde{\beta} = 3 \cdot 2^{-5/3} N\beta$ *and* $Y = \begin{bmatrix} \nabla \log \pi(x_1)^T \\ \vdots \\ \nabla \log \pi(x_N)^T \end{bmatrix} \in \mathbb{R}^{N \times d}$.

Proposition (Dual problem)

The dual problem of the regularized variational problem is

$$\max_{\Lambda \in \mathbb{R}^{N \times d}} \quad -\frac{1}{2}\|\Lambda + Y\|_F^2,$$

$$\text{s.t.} \quad \max_{w:\|w\|_2 \le 1} \left| \sum_{n=1}^{N} \|w\|_2^2 \psi''(x_n^T w) - \lambda_n^T w \psi'(x_n^T w) \right| \le \tilde{\beta},$$

which provides a lower-bound on the primal problem.

Analysis of the dual constraint

- Analyze the dual constraint

$$\max_{w:\|w\|_2\leq 1}\left|\sum_{n=1}^{N}\|w\|_2^2\mathbb{I}(w^Tx_n\geq 0)-\lambda_n^Tw(x_n^Tw)_+\right|\leq\tilde{\beta}/2.$$

- Let $X=[x_1,\ldots,x_N]^T\in\mathbb{R}^{N\times d}$. Denote the set of all possible hyper-plane arrangements corresponding to the rows of X as

$$\{D=\mathrm{diag}(\mathbb{I}(Xw\geq 0))|w\in\mathbb{R}^d\}.$$

Relaxed Dual problem

- The relaxed dual problem is the SDP:

$$\begin{aligned}
&\max_{\Lambda\in\mathbb{R}^{N\times d},r^{(j,-)},r^{(j,+)}\in\mathbb{R}^{n+1}} \quad -\frac{1}{2}\|\Lambda+Y\|_F^2,\\
&\text{s.t. } \tilde{A}_j(\Lambda)+\tilde{B}_j+\sum_{n=0}^{N} r_n^{(j,-)}H_n^{(j)}+\tilde{\beta}e_{d+1}e_{d+1}^T\succeq 0,\\
&\qquad -\tilde{A}_j(\Lambda)-\tilde{B}_j+\sum_{n=0}^{N} r_n^{(j,+)}H_n^{(j)}+\tilde{\beta}e_{d+1}e_{d+1}^T\succeq 0,\\
&\qquad r^{(j,-)}\geq 0, r^{(j,+)}\geq 0, j\in[p].
\end{aligned}$$

- $\tilde{A}_j:\mathbb{R}^{N\times d}\to\mathbb{R}^{(d+1)\times(d+1)}$ is a linear mapping .
- $H_n^{(j)}\in\mathbb{R}^{(d+1)\times(d+1)}$ is generated by the data and hyperplane arrangement.

Proposition (Relaxed bi-dual problem)

The dual of the relaxed dual problem is as follows

$$\min_{Z,S^{(j,+)},S^{(j,-)}} \frac{1}{2}\|Z+Y\|_F^2 - \frac{1}{2}\|Y\|_F^2 + \sum_{j=1}^{p} \mathrm{tr}(\tilde{B}_j(S^{(j,+)} - S^{(j,-)}))$$

$$+ \tilde{\beta}\sum_{j=1}^{p} \mathrm{tr}\left((S^{(j,+)} + S^{(j,-)})e_{d+1}e_{d+1}^T\right),$$

$$\text{s.t. } Z = \sum_{j=1}^{p} \tilde{A}_j^*(S^{(j,-)} - S^{(j,+)}),$$

$$\mathrm{tr}(S^{(j,-)}H_n^{(j)}) \le 0, \mathrm{tr}(S^{(j,+)}H_n^{(j)}) \le 0, n = 0, \ldots, N,$$

$$S^{(j,-)} \succeq 0, S^{(j,-)} \succeq 0, j \in [p].$$

Here $\tilde{A}_j^*$ is the adjoint operator of the linear operator $\tilde{A}_j$.

Theorem

Suppose that (Z, W, α) *is feasible to the primal problem. Then, there exist matrices* $\{S^{(j,+)}, S^{(j,-)}\}_{j=1}^{p}$ *constructed from* (W, α) *such that* $(Z, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^{p})$ *is feasible to the relaxed bi-dual problem. Moreover, the objective value of the relaxed bi-dual problem at* $(Z, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^{p})$ *is the same as objective value of the primal problem at* (Z, W, α).

Optimality analysis

- Optimality of the relaxed bi-dual problem

$$J(\tilde{Z}^*, \{\tilde{S}^{(j,+)}, \tilde{S}^{(j,-)}\}_{j=1}^p) \leq J(Z^*, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p).$$

- At (Z^*, W^*, α^*) we obtain the optimal approximation of $\nabla \log \rho - \nabla \log \pi$ at $x_1, \ldots, x_N$ in the family of two-layer squared-ReLU networks.
- Smaller or equal objective value of the relaxed bi-dual problem can be achieved at $(\tilde{Z}^*, \{\tilde{S}^{(j,+)}, \tilde{S}^{(j,-)}\}_{j=1}^p)$.

Practical implementation

- Computationally costly to enumerate all possible p matrices $D_1, \ldots, D_p$ to represent the constraints in the relaxed dual problem.
- Randomly sample M i.i.d. random vectors $u_1, \ldots, u_M \sim \mathcal{N}(0, I_d)$ and generate a subset $\hat{\mathcal{S}}$ of $\mathcal{S}$ as follows:

$$\hat{\mathcal{S}} = \{\mathrm{diag}(\mathbb{I}(Xu_j \geq 0) | j \in [M]\}.$$

- Optimize the randomly sub-sampled version of the relaxed dual problem based on the subset $\hat{\mathcal{S}}$ with $|\hat{\mathcal{S}}| = \hat{p}$.
- Applying the standard interior point method leads to the computational time up to

$$O((\max\{N, d^2\}\hat{p})^6).$$

Dimension reduction

- For high-dimensional problems, i.e., d is large, the computational cost of solving SDP can be large.
- In this case, we apply the dimension-reduction techniques[1][2] to reduce the parameter dimension d to a data-informed intrinsic dimension $\hat{d}$, which is often very low, i.e., $\hat{d} \ll d$.
- This can dramatically decrease the computational time.

$$O((\max\{N, \hat{d}^2\}\hat{p})^6).$$

[1] Chen, P. and Ghattas, O.. Projected stein variational gradient descent. Neurips 2020.

[2] Wang, Y., Chen, P. and Li, W.. Projected wasserstein gradient descent for high-dimensional bayesian inference.

Adjustment of regularization parameter

- Dual problem

$$\max_{\Lambda \in \mathbb{R}^{N\times d}} \quad -\frac{1}{2}\|\Lambda + Y\|_F^2,$$
$$\text{s.t.} \quad \max_{w:\|w\|_2\le 1} \left| \sum_{n=1}^{N} \|w\|_2^2 \psi''(x_n^T w) - \lambda_n^T w \psi'(x_n^T w) \right| \le \tilde{\beta},$$

- If the regularization parameter is too large, then we will have $-\Lambda - Y = 0$, which makes the particle system unchanged.
- To ensure that $\tilde{\beta}$ is not too large, we decay $\tilde{\beta}$ by a factor $\gamma_1 \in (0, 1)$.
- If $\tilde{\beta}$ is too small resulting the relaxed dual problem infeasible, we increase $\tilde{\beta}$ by multiplying γ_2^{-1}, where $\gamma_2 \in (0, 1)$.

Algorithm 1 Convex Neural Wasserstein descent

Require: initial positions $\{x_0^n\}_{n=1}^N$, step size α_l, initial regularization parameter $\tilde{\beta}_0$, $\gamma_1, \gamma_2 \in (0,1)$.

while not converge **do**
  Form X_l and Y_l based on $\{x_l^n\}_{n=1}^N$ and $\{\nabla \log \pi(x_l^n)\}_{n=1}^N$.
  Solve Λ_l from the relaxed dual problem with $\tilde{\beta} = \tilde{\beta}_l$.
  if the relaxed dual problem with $\tilde{\beta} = \tilde{\beta}_l$ is infeasible **then**
    Set $X_{l+1} = X_l$ for $n \in [N]$ and set $\tilde{\beta}_{l+1} = \gamma_2^{-1}\tilde{\beta}_l$.
  else
    Update $X_{l+1} = X_l + \alpha_l(\Lambda_l + Y_l)$ for $n \in [N]$ and set $\tilde{\beta}_{l+1} = \gamma_1\tilde{\beta}_l$.
  end if
end while

Toy example

$$\pi(x) = \exp\left(-\frac{1}{2}\left(F(x) - y\right)^2 - \frac{1}{2}\|x\|_2^2\right),$$
$$F(x) = \log\left((x_1 - 1)^2 + 100(x_2 - x_1^2)^2\right).$$

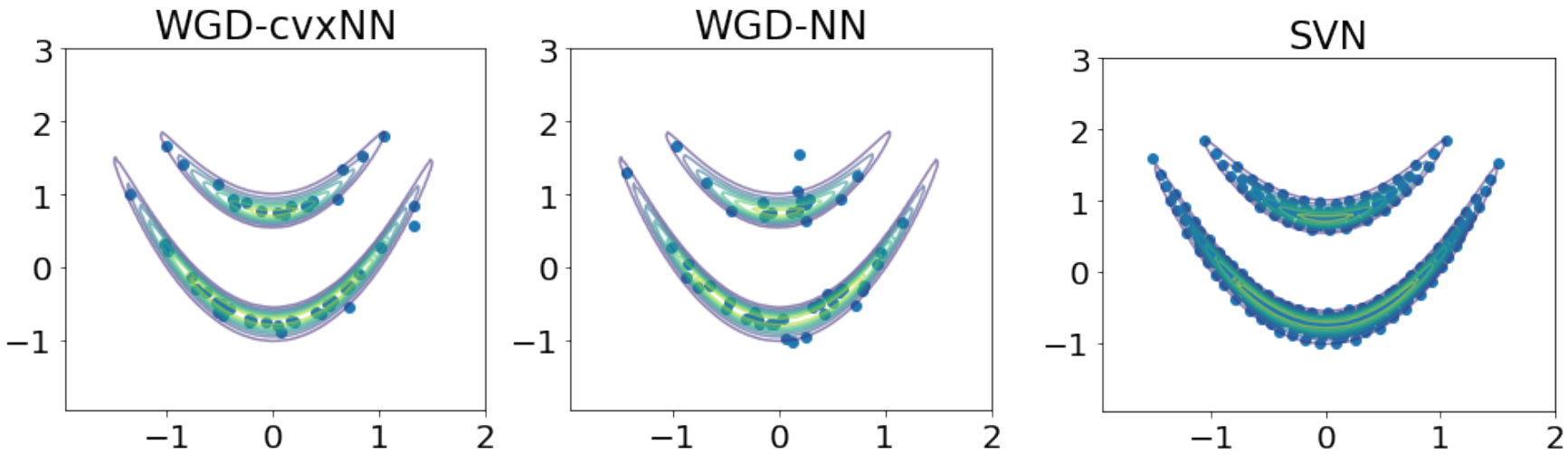


Figure: Posterior density and sample distributions by WGD-cvxNN and WGD-NN at the final step of 100 iterations, compared to the reference SVN samples (right).

Toy example

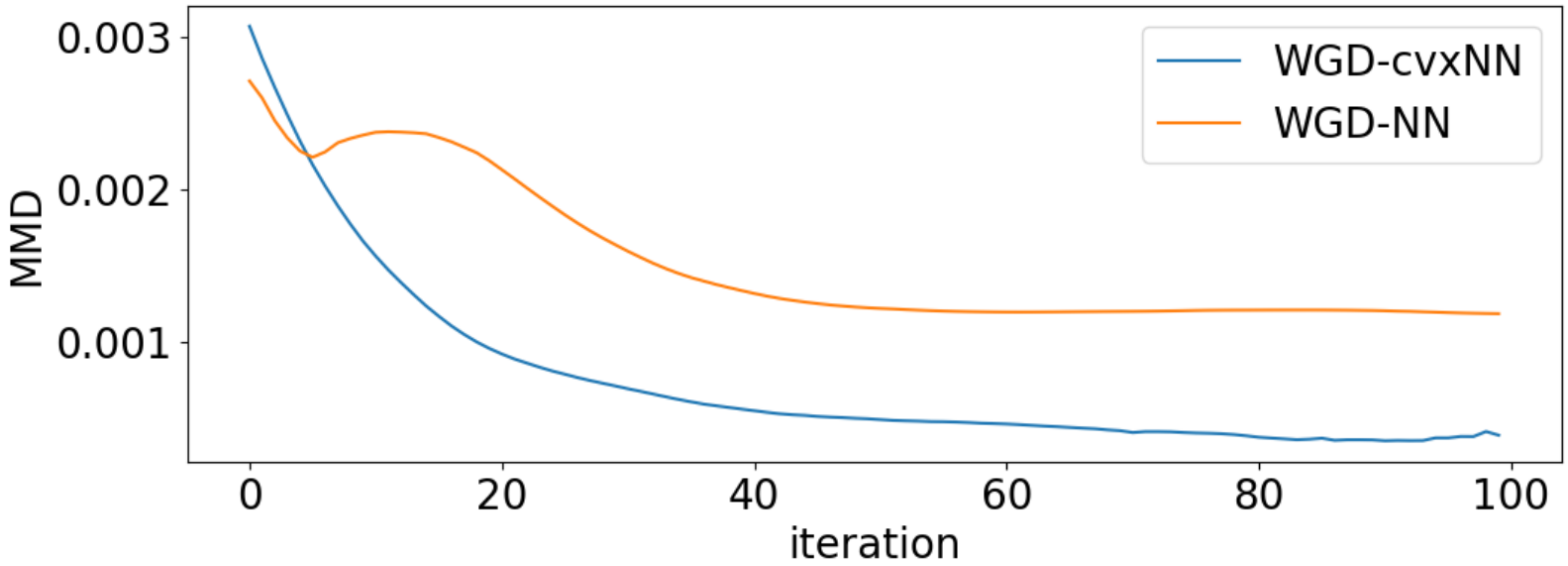


Figure: MMD of WGD-cvxNN and WGD-NN samples compared to the reference SVN samples at each iteration.

PDE-constrained nonlinear Bayesian inference

- Partial differential equation (PDE) with application to subsurface (Darcy) flow in a physical domain $D = (0,1)^2$,

$$\begin{aligned} \mathbf{v} + e^x \nabla u &= 0 \quad \text{in } D, \\ \nabla \cdot \mathbf{v} &= h \quad \text{in } D, \end{aligned}$$

- u is pressure, $\mathbf{v}$ is velocity, h is force, e^x is a random (permeability) field equipped with a Gaussian prior $x \sim \mathcal{N}(x_0, C)$ with covariance operator $C = (-\delta\Delta + \gamma I)^{-\alpha}$.
- Use a finite element method with piecewise linear elements for the discretization of the problem, resulting in $d = 81$ dimensions for the discrete parameter.

PDE-constrained nonlinear Bayesian inference

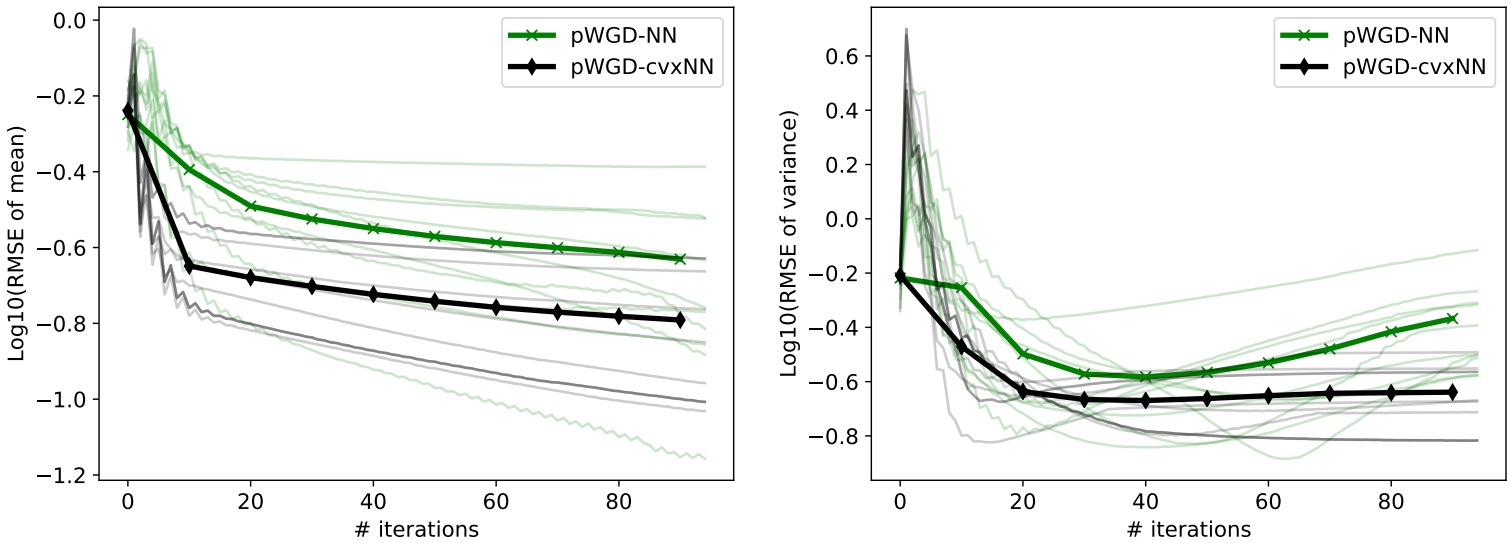


Figure: Ten trials and the RMSE of the sample mean (top) and sample variance (bottom) by pWGD-NN and pWGD-cvxNN at different iterations. The referenced sample mean and sample variance are generated by DILI-MCMC algorithm[1] with 10000 effective samples.

[1] Cui, T., Law, K. J., and Marzouk, Y. M., Dimension-independent likelihood-informed mcmc. Journal of Computational Physics 2016.

Bayesian inference for COVID-19

- Apply Bayesian inference[1] to learn the dynamics of the transmission and severity of COVID-19 from the recorded data for New York state.
- Model the transmission reduction effect of social distancing
- Observation data: number of hospitalized cases
- Target: infer social distancing parameter, a time-dependent stochastic process that is equipped with a Tanh-Gaussian prior
- 96 dimensions after discretization.

[1] Chen, P. and Ghattas, O. Projected stein variational gradient descent. Neurips 2020.

Bayesian inference for COVID-19

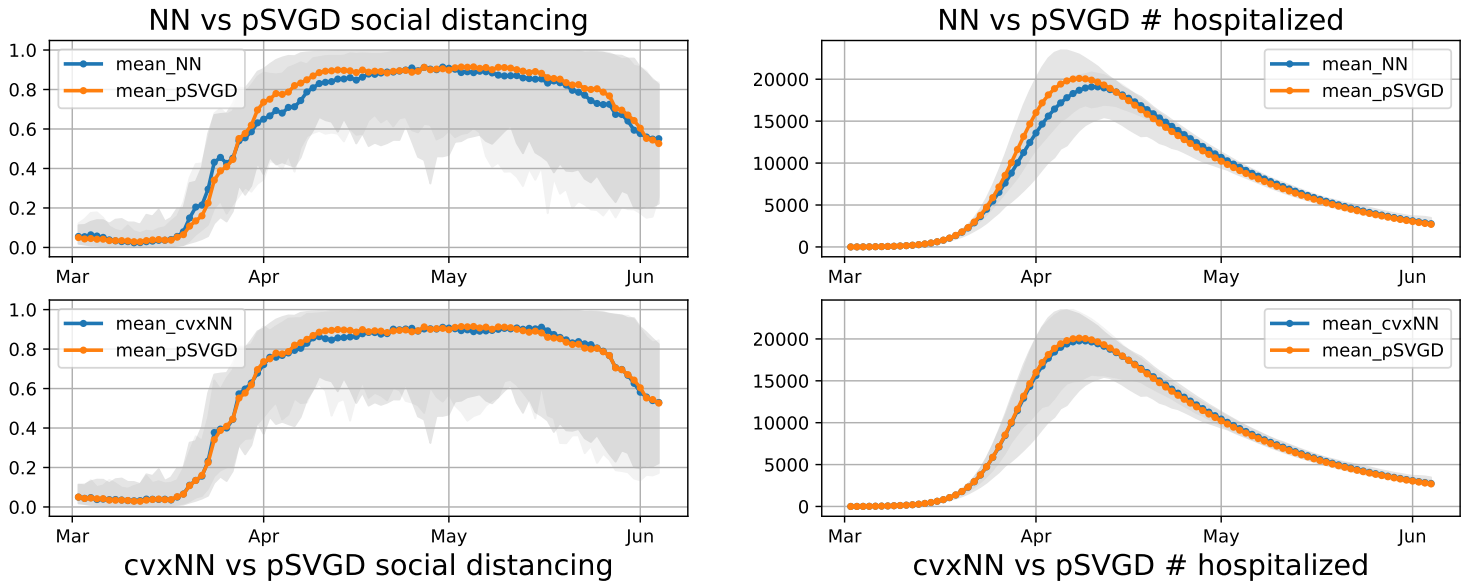


Figure: Comparison of pWGD-cvxNN and pWGD-NN to the reference by pSVGD for Bayesian inference of the social distancing parameter (top) from the data of the hospitalized cases (bottom) with sample mean and 90% credible interval.

Conclusion and future directions

- Consider the approximation of the Wasserstein gradient direction by the gradient of functions in the family of two-layer neural networks.
- Propose a convex SDP relaxation of the dual of the variational primal problem.

Conclusion and future directions

- Consider the approximation of the Wasserstein gradient direction by the gradient of functions in the family of two-layer neural networks.
- Propose a convex SDP relaxation of the dual of the variational primal problem.
- Theoretical guarantee for the subsampling procedure of hyperplane arrangements (On working!)
- Design specialized solver for induced SDPs.
- Application of convex optimization formulation of NN to the computation/approximation of generalized Wasserstein flows, e.g., normalizing flows, mean field games.