Proximal Mean Field Learning in Shallow Neural Networks

Alexis M.H. Teter

Department of Applied Mathematics University of California, Santa Cruz

Joint work with



Abhishek Halder Iowa State University



Iman Nodozi University of California, Santa Cruz

SIAM Minisymposium: Recent advances in optimization for DNNs 2024 SIAM Annual Meeting, Online, July 18, 2024

Structure of shallow neural network



Risk

Population Risk

$$R(f) := \mathbb{E}_{(oldsymbol{x},y) \sim \gamma}[(oldsymbol{y} - f(oldsymbol{x},oldsymbol{ar{ heta}}))^2]$$



Large dimensional, non convex optimization problem

$$\min_{oldsymbol{ heta} \in \mathbb{R}^{p imes n_{ ext{H}}}} R(f)$$

Mean field limit

$$f(oldsymbol{x},oldsymbol{ heta}) \coloneqq rac{1}{n_H}\sum_{i=1}^{n_H}\Phi(oldsymbol{x},oldsymbol{ heta}_i)$$

Let $n_H o\infty$
 $f_{ ext{MeanField}}\coloneqq \int_{\mathbb{R}^p}\Phi(oldsymbol{x},oldsymbol{ heta})d\mu(oldsymbol{ heta}) = \mathbb{E}_{oldsymbol{ heta}}[\Phi(oldsymbol{x},oldsymbol{ heta})]$
where $d\mu(oldsymbol{ heta}) =
ho(oldsymbol{ heta})doldsymbol{ heta}$



	1				0 9	2	6	5	-	4		-	-				L
•	1	9.0		9				Ĩ			*	t	H		*		
					ä	Â		j	*	8	0		8	ιĝ.		-	
eo	×	*.			۲	1		•			•	-		*		P.e	
					Ö				å		d:	*		*		j.	ſ
		*	A			10			3			¥ .	8	9		j	
		1	*			°.	.0	he			-00			ė		8	
2	Ĭ	i i i i i i i i i i i i i i i i i i i	2	G					Q	E.			0	() ()	Ö		
		8	0.6					0								0	
													Ø	B	0	3	í
	ē.			0					101					48	2	۲	-
	-		6	0	08			0	0	0	P				-	0	
		<u>.</u>				P							Į.		11		
		- X	-								-No-	*		3.4	*	-	
	6							Å	-	*					e	T	È
A		A	e de la cel		8			9		9		-			1	×	-
			0		.				•						G		
		. 8					ê,		-	<u>ê</u>		-					-
-	-	0		1	8 6		-	4	*	8	-	-		1 1			-

$$d\mu =
ho \ dx$$

Images generated with ChatGPT by Iman Nodozi

Wasserstein metric



$$W_2^2(\pi_1,\pi_2):= \inf_{\pi\in\Pi(\pi_1,\pi_2)} \int_{\mathcal{Z}_1 imes \mathcal{Z}_2} \!\!\! \|m{z}_1 - m{z}_2\|_2^2 \, d\pi(m{z}_1,m{z}_2)$$

Image generated with ChatGPT by Iman Nodozi



 $= rginf_{ ext{decision variable}} \left\{ rac{1}{2} ext{dist}^2 (ext{decision variable}, ext{input}) + ext{time step} imes ext{functional(decision variable})
ight\}$

Gradient Flows

Gradient Flow

$$rac{\partial
ho}{\partial t} = -
abla^{W_2} F(
ho) := -
abla \cdot \left(
ho
abla rac{\delta F}{\partial
ho}
ight) \quad extbf{where} \quad
ho(oldsymbol{ heta}, 0) =
ho_0(oldsymbol{ heta})$$

Recursion

$$ho_k=
ho(\cdot,t=kh)=rgmin_{
ho\in\mathcal{P}_2(\mathbb{R}^p)}igg(rac{1}{2}W_2^2(
ho,
ho_{k-1})+hF(
ho)igg)=\mathrm{prox}_{hF}^{W_2}(
ho_{k-1})$$

 $= \underset{ ext{decision variable}}{ ext{arg inf}} \left\{ rac{1}{2} ext{dist}^2 (ext{decision variable}, ext{input}) + ext{time step} imes ext{functional(decision variable})
ight\}$

Risk functional

$$R(f_{ ext{Mean Field}}(oldsymbol{x},
ho)) = \mathbb{E}_{(oldsymbol{x},y)}igg(y - \int_{\mathbb{R}^p} \Phi(oldsymbol{x}, heta)
ho(oldsymbol{ heta}) doldsymbol{ heta}igg)^2igg)$$

$$egin{aligned} R(f_{ ext{Mean Field}}(oldsymbol{x},
ho)) &= F_0 + \int_{\mathbb{R}^p} V(oldsymbol{ heta})
ho(oldsymbol{ heta}) doldsymbol{ heta} + \int_{\mathbb{R}^{2p}} U(oldsymbol{ heta},oldsymbol{ heta})
ho(oldsymbol{ heta}) doldsymbol{ heta} doldsymbol{ heta} \ Drift ext{ potential} & ext{Interaction potential} \ F_0 &:= \mathbb{E}_{(oldsymbol{x},y)}[y^2] \ V(oldsymbol{ heta}) &:= \mathbb{E}_{(oldsymbol{x},y)}[-2y \ \Phi(oldsymbol{x},oldsymbol{ heta})] \ U(oldsymbol{ heta},oldsymbol{ heta}) &:= \mathbb{E}_{(oldsymbol{x},y)}[\Phi(oldsymbol{x},oldsymbol{ heta})] \end{aligned}$$

Supervised learning in mean field limit

Supervised learning problem

$$\min_{
ho} \; F(
ho) := \min_{
ho} \; R(f_{ ext{Mean Field}}(oldsymbol{x},
ho))$$

With convex regularizer

Proximal recursions

$$egin{aligned} &\mathcal{P}\mathbf{roximal\ recursion}\ arrho_k = \mathrm{prox}_{hF_eta}^{W_2}(arrho_{k-1}) \coloneqq rgin_{arrho \in \mathcal{P}_2(\mathbb{R}^p)} igg\{rac{1}{2}(W_2(arrho, arrho_{k-1}))^2 + h\ F_eta(arrho)igg\}\ & ext{where}\ arrho_{k-1}(\cdot) \coloneqq arrho(\cdot, t_{k-1})\ &arrho_0 \equiv
ho_0 \end{aligned}$$

Approximate bilinear term as...

$$\int_{\mathbb{R}^{2p}} U(oldsymbol{ heta}, ilde{oldsymbol{ heta}}) arrho(oldsymbol{ heta}) doldsymbol{ heta} d ilde{oldsymbol{ heta}} pprox \int_{\mathbb{R}^{2p}} U(oldsymbol{ heta}, ilde{oldsymbol{ heta}}) arrho(oldsymbol{ heta}) doldsymbol{ heta} d ilde{oldsymbol{ heta}}$$

(Benamou et al., 2016, Sec. 4)

Proximal recursions

$$egin{aligned} &\mathcal{P}\mathbf{roximal\ recursion}\ arrho_k = \mathrm{prox}_{hF_eta}^{W_2}(arrho_{k-1}) \coloneqq rgin_{arrho \in \mathcal{P}_2(\mathbb{R}^p)} igg\{rac{1}{2}(W_2(arrho, arrho_{k-1}))^2 + h \, \hat{F}_eta(arrho) igg\}\ & ext{where}\ arrho_{k-1}(\cdot) \coloneqq arrho(\cdot, t_{k-1})\ &arrho_0 \equiv
ho_0 \end{aligned}$$

Approximation of regularized risk functional

$$\hat{F}_eta(arrho,arrho_{k-1}):=\int_{\mathbb{R}^p}igg(F_0+V(oldsymbol{ heta})+igg(oldsymbol{ heta})+eta^{-1}\logarrho(oldsymbol{ heta})igg)arrho(oldsymbol{ heta})doldsymbol{ heta}igg)+eta^{-1}\logarrho(oldsymbol{ heta})arrho(oldsymbol{ heta})doldsymbol{ heta}$$

Proximal recursions

Thm. 1:

As h
ightarrow 0 , proximal updates converge to solution to PDE IVP.

ProxLearn Algorithm

Euler-Maruyama

$$egin{aligned} oldsymbol{ heta}_k^i &= oldsymbol{ heta}_{k-1}^i - h
abla ig(Vig(oldsymbol{ heta}_{k-1}^iig) + \omegaig(oldsymbol{ heta}_{k-1}^iig)ig) + \sqrt{2eta^{-1}}ig(oldsymbol{\eta}_k^i - oldsymbol{\eta}_{k-1}^iig) \ & ext{where} \ \ \omega(\cdot) &:= \int U(\cdot,oldsymbol{ heta})arrhoig(oldsymbol{ heta}) \mathrm{d}oldsymbol{ heta} \ & ext{and} \ \ oldsymbol{\eta}_{k-1}^i &:= oldsymbol{\eta}^i(t=(k-1)h) \end{aligned}$$

Euler-Maruyama

$$egin{aligned} egin{aligned} egin{aligned} eta_k^i &= m{ heta}_{k-1}^i - h
abla ig(Vig(m{ heta}_{k-1}^iig) + \omegaig(m{ heta}_{k-1}^iig)ig) + \sqrt{2eta^{-1}}ig(m{\eta}_k^i - m{\eta}_{k-1}^iig) & \ \end{aligned}$$
 where $\omega(\cdot) &:= \int U(\cdot,m{ heta}) arrho(m{ heta}) \mathrm{d}m{ heta}$ and $m{\eta}_{k-1}^i &:= m{\eta}^i(t=(k-1)h)$

Algorithm 2 Euler-Maruyama Algorithm

1: procedure EULERMARUYAMA $(h, \beta, \Theta_{k-1}, X, y, \varrho_{k-1})$ $\boldsymbol{P}_{k-1} \leftarrow \boldsymbol{\Phi}(\boldsymbol{\Theta}_{k-1}, \boldsymbol{X})$ \triangleright Lines 2-4 construct the argument of the gradient in (35) 2: $oldsymbol{U}_{k-1} \leftarrow 1/n_{ ext{data}}oldsymbol{P}_{k-1}oldsymbol{P}_{k-1}^ op$ 3: $\boldsymbol{u}_{k-1} \leftarrow \boldsymbol{U}_{k-1} \boldsymbol{\varrho}_{k-1}$ 4: $oldsymbol{v}_{k-1} \leftarrow -2/n_{ ext{data}}oldsymbol{P}_{k-1}oldsymbol{y}$ 5: $\boldsymbol{D} \leftarrow \operatorname{Backward} (\boldsymbol{u}_{k-1} + \boldsymbol{v}_{k-1})$ \triangleright Approximate the gradient of (35) using PyTorch library 6: BACKWARD (Paszke et al., 2017) $\boldsymbol{G} \leftarrow \sqrt{2h/\beta} \times \mathrm{randn}_{N \times p}$ 7: $\boldsymbol{\Theta}_k \leftarrow \boldsymbol{\Theta}_{k-1} + h \times \boldsymbol{D} + \boldsymbol{G}$ \triangleright Complete the location update via (35) 8: 9: end procedure

Proximal recursion (semi-implicit variant)

$$arrho_k = \mathrm{prox}_{hF_eta}^{W_2}(arrho_{k-1}) := rginf_{arrho \in \mathcal{P}_2(\mathbb{R}^p)} igg\{ rac{1}{2} (W_2(arrho, arrho_{k-1}))^2 + h \ \hat{F}_eta(arrho) igg\}$$

Discrete version of proximal recursion

$$oldsymbol{arrho}_k = rgmin_{oldsymbol{arrho}} \left\{ \min_{oldsymbol{M} \in \Pi(oldsymbol{arrho}_{k-1},oldsymbol{arrho})} rac{1}{2} \langle oldsymbol{C}_k,oldsymbol{M}
angle + h ig\langle oldsymbol{v}_{k-1} + oldsymbol{U}_{k-1} oldsymbol{arrho}_{k-1} + eta^{-1}\logoldsymbol{arrho},oldsymbol{arrho}
angle
ight\}$$
where $\Pi(oldsymbol{arrho}_{k-1},oldsymbol{arrho}) := \{oldsymbol{M} \in \mathbb{R}^{N imes N} \mid oldsymbol{M} \geq oldsymbol{0} ext{ (elementwise)}, oldsymbol{M} oldsymbol{1} = oldsymbol{arrho}_{k-1}, oldsymbol{M}^{ op} oldsymbol{1} = oldsymbol{arrho}
angle
angle
ight\}$

Regularized discrete version of proximal recursion

$$oldsymbol{arrho}_k = rgmin_{oldsymbol{arrho}} \left\{ \min_{oldsymbol{M} \in \Pi(oldsymbol{arrho}_{k-1},oldsymbol{arrho})} rac{1}{2} \langle oldsymbol{C}_k,oldsymbol{M}
angle + oldsymbol{\epsilon} \langle oldsymbol{M}, \logoldsymbol{M}
angle + hig\langle oldsymbol{v}_{k-1} + oldsymbol{U}_{k-1} + oldsymbol{arrho}_{k-1} + oldsymbol{eta}^{-1} \logoldsymbol{arrho},oldsymbol{arrho}
ight
angle
ight\}$$

where $\Pi(\boldsymbol{\varrho}_{k-1}, \boldsymbol{\varrho}) := \{ \boldsymbol{M} \in \mathbb{R}^{N \times N} \mid \boldsymbol{M} \geq \boldsymbol{0} \text{ (elementwise)}, \boldsymbol{M} \boldsymbol{1} = \boldsymbol{\varrho}_{k-1}, \ \boldsymbol{M}^{\top} \boldsymbol{1} = \boldsymbol{\varrho} \}$

Regularized discrete version of proximal recursion

$$oldsymbol{arrho}_k = rgmin_{oldsymbol{arrho}} \left\{ \min_{oldsymbol{M} \in \Pi(oldsymbol{arrho}_{k-1},oldsymbol{arrho})} rac{1}{2} \langle oldsymbol{C}_k,oldsymbol{M}
angle + \epsilon \langle oldsymbol{M}, \log oldsymbol{M}
angle + h igl\langle oldsymbol{v}_{k-1} + oldsymbol{U}_{k-1} oldsymbol{arrho}_{k-1} + eta^{-1} \log oldsymbol{arrho}, oldsymbol{arrho}
ight
angle
ight\}$$

where $\Pi(\boldsymbol{\varrho}_{k-1}, \boldsymbol{\varrho}) := \{ \boldsymbol{M} \in \mathbb{R}^{N \times N} \mid \boldsymbol{M} \geq \boldsymbol{0} \text{ (elementwise)}, \boldsymbol{M} \boldsymbol{1} = \boldsymbol{\varrho}_{k-1}, \ \boldsymbol{M}^{\top} \boldsymbol{1} = \boldsymbol{\varrho} \}$

Use Lagrange dual problem with Lagrange multipliers λ_0 and λ_1

$$egin{aligned} \mathbf{Let:}\ oldsymbol{z} &:= \exp(oldsymbol{\lambda}_1 h/\epsilon)\ oldsymbol{q} &:= \exp(oldsymbol{\lambda}_0 h/\epsilon)\ oldsymbol{\Gamma}_k &:= \exp(-oldsymbol{C}_k/2\epsilon)\ oldsymbol{\xi}_{k-1} &:= \exp(-eta oldsymbol{v}_{k-1} - eta oldsymbol{U}_{k-1} oldsymbol{arphi}_{k-1} - oldsymbol{1}) \end{aligned}$$

Proximal update

$$oldsymbol{arrho}_k = oldsymbol{z} \odot oldsymbol{\Gamma}_k^ op oldsymbol{q}$$

Proximal algorithm

Algorithm 1 Proximal Algorithm

1: procedure PROXLEARN($\boldsymbol{\varrho}_{k-1}, \boldsymbol{\Theta}_{k-1}, \beta, h, \varepsilon, N, \boldsymbol{X}, \boldsymbol{y}, \delta, L$)						
2: $\boldsymbol{v}_{k-1}, \boldsymbol{U}_{k-1}, \boldsymbol{\Theta}_k \leftarrow \text{EULERMARUYAMA}(h, \beta, \boldsymbol{\Theta}_{k-1}, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\varrho}_{k-1}) \triangleright \text{Update the location of the samples}$						
3: $\boldsymbol{C}_{k}(i,j) \leftarrow \left\ \boldsymbol{\theta}_{k}^{i} - \boldsymbol{\theta}_{k-1}^{j} \right\ _{2}^{2}$						
4: $\boldsymbol{\Gamma}_k \leftarrow \exp(-\boldsymbol{C}_k/2\varepsilon)$						
5: $\boldsymbol{\xi}_{k-1} \leftarrow \exp(-\beta \boldsymbol{v}_{k-1} - \beta \boldsymbol{U}_{k-1} \boldsymbol{\varrho}_{k-1} - 1)$						
6: $\mathbf{z}_0 \leftarrow \operatorname{rand}_{N \times 1}$						
7: $\boldsymbol{z} \leftarrow [\boldsymbol{z}_0, \boldsymbol{0}_{N \times (L-1)}]$						
8: $\boldsymbol{q} \leftarrow [\boldsymbol{\varrho}_{k-1} \oslash (\boldsymbol{\Gamma}_{\boldsymbol{k}} \boldsymbol{z}_{\boldsymbol{0}}), \boldsymbol{0}_{N \times (L-1)}]$						
9: $\ell = 1$						
10: while $\ell \leq L$ do						
11: $\mathbf{z}(:, \ell+1) \leftarrow (\mathbf{\xi}_{k-1} \oslash (\mathbf{\Gamma}_{l}^{\top} \mathbf{a}(:, \ell))^{\frac{1}{1+\beta\varepsilon/h}}$						
12: $\boldsymbol{a}(\cdot,\ell+1) \leftarrow \boldsymbol{a}_{k-1} \oslash (\boldsymbol{\Gamma}_k \boldsymbol{z}(\cdot,\ell+1))$						
$\frac{q(\cdot, v + 1)}{(1 + 1)} = \frac{q(\cdot, v + 1)}{(1 + v)} = \frac{q(\cdot, v + 1)}{(1 + v)} = \frac{q(\cdot, v)}{(1 + 1)} = q(\cdot, v$						
13: If $ \mathbf{q}(., \ell + 1) - \mathbf{q}(., \ell) \le 0$ and $ \mathbf{z}(., \ell + 1) - \mathbf{z}(., \ell) \le 0$ then 14. Proof-						
14: Dreak						
15: else						
16: $\ell \leftarrow \ell + 1$						
17: end if						
18: end while						
19: return $\boldsymbol{\rho}_k \leftarrow \boldsymbol{z}(:,\ell) \odot (\boldsymbol{\Gamma}_k^\top \boldsymbol{q}(:,\ell))$						
20: end procedure						

Case Study: Binary Classification on WDBC Data

WDBC:
of features:
$$n_x = 30$$

of data points:
 $n = 569$

Source: UCI machine learning repository, 2017, Available: <u>http://archive.ics.uci.edu/ml/index.php</u>

Case Study: Binary Classification on WDBC Data

β	Unweighted	Weighted
0.03	91.17%	92.35%
0.05	92.94%	92.94%
0.07	78.23%	92.94%

β	Unweighted	Weighted	Runtime (hr)
0.03	91.18%	91.18%	1.415
0.05	91.18%	92.94%	1.533
0.07	90.59%	91.76%	1.704

Case Study: Binary Classification

Comparison to Mokrov et al (2021) & Bonet et al (2022)

Dataset	JKO-ICNN	SWGF + RealNVP	ProxLearn, Weighted	ProxLearn, Unweighted
Banana	0.550 ± 10^{-2}	0.559 ± 10^{-2}	0.551 ± 10^{-2}	$0.535 \pm 5 \cdot 10^{-2}$
Diabetes	$0.777 \pm 7 \cdot 10^{-3}$	$0.778 \pm 2 \cdot 10^{-3}$	$0.736 \pm 2 \cdot 10^{-2}$	0.731 ± 10^{-2}
Twonorm	$0.981 \pm 2 \cdot 10^{-4}$	$0.981 \pm 6 \cdot 10^{-4}$	$0.972 \pm 2 \cdot 10^{-3}$	$0.972 \pm 2 \cdot 10^{-3}$

Case Study: Multi-Class Classification

Semeion Handwritten Digit Data Set # of features: $n_x = 16 imes 16 = 256$ # of data points: n = 1593

Dua and Graff (2017) <u>http://archive.ics.uci.edu/ml</u>

Case Study: Multi-Class Classification

$$egin{aligned} oldsymbol{P}_{k-1}(j,i) &:= oldsymbol{\Phi}(oldsymbol{ heta}_{k-1}^j,oldsymbol{X}(i,:),oldsymbol{Y}(i,:))\ &:= \left\langle ext{softmax}(oldsymbol{X}(i,:)(oldsymbol{ heta}_{k-1}^j)^ op),(oldsymbol{Y}(i,:))^ op
ight
angle
ight
angle \end{aligned}$$

Learning a sinusoid

Additional Avenues of Research

Multiple hidden layer setting

(*) Infinite width limit on one hidden layer; width of other hidden layers held constant

(*) Widths of all hidden layers go to infinity

Thank You

Acknowledgement:

2112755