A Language and Smoothed Semantics for Convergent Stochastic Gradient Descent

Dominik Wagner Luke Ong

UNIVERSITY OF OXFORD

LAFI 2022

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{z}\sim q}\left[f(\boldsymbol{\theta}, \mathbf{z})\right]}$$

no dependence on $\boldsymbol{\theta}$

expressed in PL with conditionals,
may *not* be *differentiable/continuous*

Example: maximisation of ELBO for reparametrised models in *variational inference*

$$\mathrm{ELBO}(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{z}\sim q}\left[\log p(\phi_{\boldsymbol{\theta}}(\mathbf{z})) - \log q_{\theta}(\phi_{\boldsymbol{\theta}}(\mathbf{z}))\right]$$

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{z}\sim q}\left[f(\boldsymbol{\theta}, \mathbf{z})\right]}$$

no dependence on $\boldsymbol{\theta}$

expressed in PL with conditionals,
may *not* be *differentiable/continuous*

Aim: find *stationary* point, i.e. $\boldsymbol{\theta}$ s.t. $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{z}\sim q}[f(\boldsymbol{\theta}, \mathbf{z})] = 0$

Stochastic Gradient Descent (SGD)

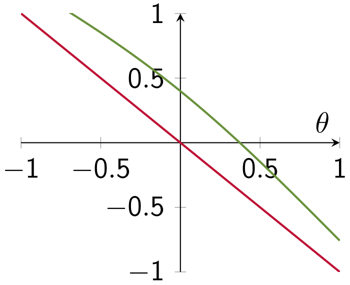
$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_k \cdot \underbrace{\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_k, \mathbf{z}_k)}_{\text{repramatrisation gradient estimator}} \qquad \mathbf{z}_k \sim q$$

*Reparametrisation gradient estimator for non-differentiable models is **biased!***

[Lee et al., NeurIPS 2018]

$$f(\theta, z) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } z + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{1})}\left[\nabla_\theta \mathbf{f}(\theta, \mathbf{z})\right] = -\theta \neq -\theta + \mathcal{N}(-\theta \mid 0, 1) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{z \sim \mathcal{N}(0,1)}\left[f(\theta, z)\right]$$



Vanishing gradient estimator does not imply stationarity!

Contributions

Provable convergence **to stationary points (and unbiased gradient estimators) for** ***typable*** **programs.**

Approach:

- *Smoothen* (discontinuous) function using sigmoid with accuracy coefficient
- Optimise expectation, enhancing accuracy in each step

This talk:

- *Reparametrisation* programming language
- Type system and smoothed semantics
- *Convergence* of *Diagonalisation* Stochastic Gradient Descent, a new variant of SGD

Reparametrisation Programming Language

simply typed λ-calculus with $\mathbb{R}$, $+$, $\cdot$ and *conditionals*

$+$ *sampling* from standard normal
transformed by *diffeomorphic* polynomials

$$\begin{aligned} M ::= \quad & \cdots \\ & \mid \mathbf{if}\ M < 0\ \mathbf{then}\ M\ \mathbf{else}\ M \\ & \mid \ \underline{\phi}_{\boldsymbol{\theta}}(M, \ldots, M, \mathbf{sample}) \end{aligned}$$

diffeomorphic polynomial

Example: sample from $\mathcal{N}(\mu, \sigma)$ using $\phi_{\mu,\sigma}(\mathbf{sample})$, where $\phi_{\mu,\sigma}(z) := \sigma \cdot z + \mu$

$$\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \left[\mathbf{f_M}(\boldsymbol{\theta}, \mathbf{z})\right]$$

where f_M is the *value*-function of a term $M : R$ with parameters $\boldsymbol{\theta} : R$.

(Integrability) $\mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[|f_M(\boldsymbol{\theta}, \mathbf{z})|] < \infty$ for all $\boldsymbol{\theta} \in \mathbb{R}^n$.

Type System

ensure guards do not directly depend on parameters

(only after transformation)

$$\textbf{if}\ \theta < 0\ \textbf{then}\ 0\ \textbf{else}\ 1 \quad ✗$$

$$(\lambda x.\, \textbf{if}\ x < 0\ \textbf{then}\ 0\ \textbf{else}\ 1)\ \theta \quad ✗$$

$$(\lambda x.\, \textbf{if}\ x < 0\ \textbf{then}\ 0\ \textbf{else}\ 1)\ (\underline{\phi}_\theta(\textbf{sample})) \quad ✓$$

$$(\lambda x.\, \underline{-0.5} \cdot \theta^2 + (\textbf{if}\ x < 0\ \textbf{then}\ \underline{0}\ \textbf{else}\ \underline{1}))\ (\underline{\phi}_\theta(\textbf{sample})) \quad ✓$$

Reparametrisation-aware symbolic execution

variant of [Mak et al., ESOP 2021]

- Collect constraints due to *branching*
- Replace $\underline{\phi}_{\boldsymbol{\theta}}(P_1, \ldots, P_\ell, \textbf{sample})$ with fresh sampling variable α_j and keep track of *transformations*

$\emptyset \mid \emptyset \vdash_{\boldsymbol{\theta}} M : R$

polynomials (*branching*)

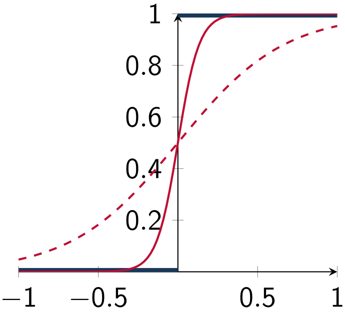
$$M \Downarrow_{\boldsymbol{\phi}}^{(\Psi_<, \Psi_\geq)} P$$

diffeomorphic polynomials (*transformations*)

polynomial term

"Standard" Semantics for accuracy coefficient $k \in \mathbb{N}$

$$f_M(\boldsymbol{\theta}, \mathbf{z}) = \sum_{M \Downarrow_{\boldsymbol{\phi}}^{(\Psi_<, \Psi_\geq)} P} f_P(\boldsymbol{\theta}, \boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{z})) \cdot \prod_{\psi \in \boldsymbol{\Psi}_<} [\psi(\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{z})) < 0] \cdot \prod_{\psi \in \boldsymbol{\Psi}_\geq} [\psi(\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{z})) \geq 0]$$



Smoothed Semantics for accuracy coefficient $k \in \mathbb{N}$

$$f_{M,k}(\boldsymbol{\theta}, \mathbf{z}) = \sum_{M \Downarrow_{\boldsymbol{\phi}}^{(\Psi_<, \Psi_\geq)} P} f_P(\boldsymbol{\theta}, \phi_{\boldsymbol{\theta}}(\mathbf{z})) \cdot \prod_{\psi \in \boldsymbol{\Psi}_<} \sigma_{\mathbf{k}}(-\psi(\phi_{\boldsymbol{\theta}}(\mathbf{z}))) \cdot \prod_{\psi \in \boldsymbol{\Psi}_\geq} \sigma_{\mathbf{k}}(\psi(\phi_{\boldsymbol{\theta}}(\mathbf{z})))$$

*Adapt (backward mode) **automatic differentiation** to compute smoothing*

(Unbiasedness) $\nabla_{\mathbf{z}}\, \mathbb{E}_{\mathbf{z}}[f_{M,k}(\boldsymbol{\theta}, \mathbf{z})] = \mathbb{E}_{\mathbf{z}}[\nabla_{\mathbf{z}}\, f_{M,k}(\boldsymbol{\theta}, \mathbf{z})]$ for all $k \in \mathbb{N}$.

Use SGD for $f_{M,k}$ for fixed $k \in \mathbb{N}$

(Uniform Convergence of Gradients) If $\boldsymbol{\Theta} \subseteq \mathbb{R}^n$ is compact then

$$\nabla_{\mathbf{z}}\, \mathbb{E}_{\mathbf{z}}[\mathbf{f}_{\mathbf{M},\mathbf{k}}(\boldsymbol{\theta}, \mathbf{z})] \xrightarrow{\text{unif}} \nabla_{\mathbf{z}}\, \mathbb{E}_{\mathbf{z}}[\mathbf{f}_{\mathbf{M}}(\boldsymbol{\theta}, \mathbf{z})] \qquad \text{as } k \to \infty \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

Key trick:

$$\frac{\partial(\sigma_k \circ \psi \circ \boldsymbol{\phi}_{.})}{\partial \theta_i}(\boldsymbol{\theta}, \mathbf{z}) = \nabla_{\mathbf{z}}(\sigma_k \circ \psi \circ \boldsymbol{\phi}_{.})(\boldsymbol{\theta}, \mathbf{z}) \cdot \mathbf{J}_{\mathbf{z}}^{-1} \boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{z}) \cdot \mathbf{J}_{\theta_i} \boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{z})$$

(enables integration by parts of $\mathbb{E}_{\mathbf{z}}[\frac{\partial f_{M,k}}{\partial \theta_i}]$)

Diagonalisation Stochastic Gradient Descent (DSGD)

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_k \cdot \nabla_{\boldsymbol{\theta}} \, \mathsf{f}_{\mathbf{M,k}}(\boldsymbol{\theta}_k, \mathbf{z}_k) \qquad \mathbf{z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

As a consequence of unbiasedness, uniform convergence (of gradients), etc.

Convergence on Typable Programs

If $\emptyset \mid \emptyset \vdash_{\boldsymbol{\theta}} M : R$ then a DSGD sequence $(\boldsymbol{\theta}_k)_{k\in\mathbb{N}}$

1. is unbounded or
2. has a *stationary* accumulation point.

Related Work

[Lee et al., NeurIPS 2018]:

- Fix (biased) reparametrisation gradient estimator for non-differentiable models by additional non-trivial *boundary* terms

✗ Only discuss efficient method for *affine* guards

✗ Not concerned with *convergence* of SGD

✗ No discussion of PL aspects

Conclusion

This work:

- ✓ Type system enforcing very mild restrictions on PL
- ✓ Smoothed semantics avoids boundary term
- ✓ Not only unbiasedness but also *convergence* of DSGD
- *Asymptotic* result, for each fixed accuracy smoothing (only) approximation

Ongoing and future work:

- Experimental evaluation
- Recursion, beyond polynomials

Conclusion

Provable convergence to stationary points (and unbiased gradient estimators) for typable programs.

Dominik Wagner Luke Ong

dominik.wagner@cs.ox.ac.uk