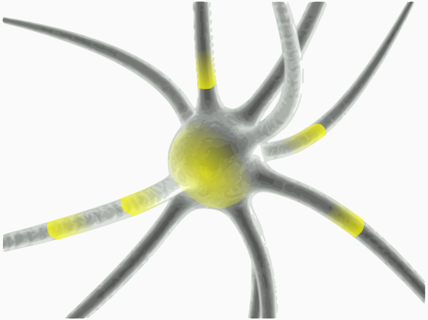
Biological neurons meet zeroth order optimisation

Wouter M. Koolen

CWI and University of Twente

Dutch Day on Optimization, November 7, 2024



Warm Thanks

Johannes Schmidt-Hieber

Menu

1. Neuroscience Motivation

2. Benchmark Task: Zeroth order Linear Regression

3. BNN meets Linear Regression

4. Reflections

Neuroscience Motivation

Motivation

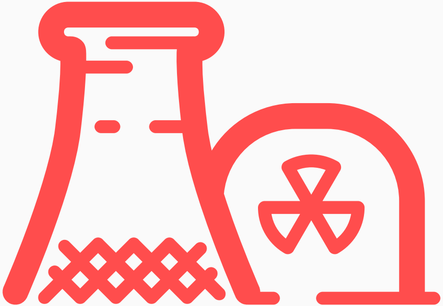
We seek to understand biological neural networks.

E.g. the brain.

And how they learn

Today in particular: connections with optimisation

Why is this interesting

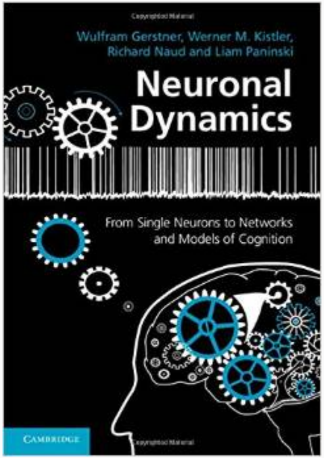


Artificial NN

VS

Biological NN

Literature



Gerstner, Kistler, Naud, and Paninski, 2014
Chapter 19: Synaptic Plasticity and Learning

arXiv:2301.11777v2 [cs.LG] 23 Mar 2023

Interpreting learning in biological neural networks as zero-order optimization method

Johannes Schmidt-Hieber*

Abstract

Recently, significant progress has been made regarding the statistical understanding of artificial neural networks (ANNs). ANNs are motivated by the functioning of the brain, but differ in several crucial aspects. In particular, the locality in the updating rule of the connection parameters in biological neural networks (BNNs) makes it biologically implausible that the learning of the brain is based on gradient descent. In this work, we look at the brain as a statistical method for supervised learning. The main contribution is to relate the local updating rule of the connection parameters in BNNs to a zero-order optimization method. It is shown that the expected values of the iterates implement a modification of gradient descent.

Keywords: Biological neural networks, zero-order optimization, derivative-free methods, supervised learning.

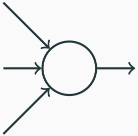
1 Introduction

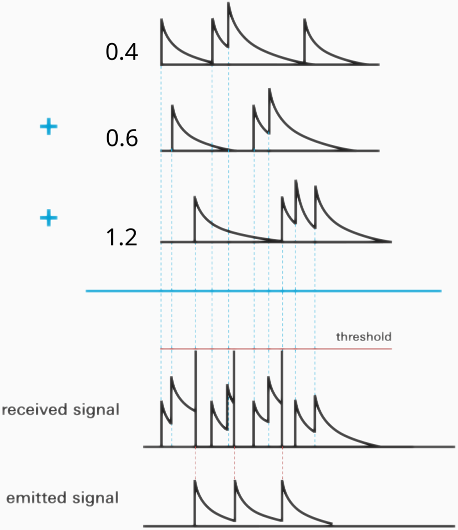
Compared to artificial neural networks (ANNs), the brain learns faster, generalizes better to new situations and consumes much less energy. A child only requires a few examples to learn to discriminate a dog from a cat. And people only need a few hours to learn how to drive a car. AI systems, however, need thousands of training samples for image recognition tasks. And the self-driving car is still under development, despite the availability of data

*University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands.
Email: a.j.schmidt-hieber@utwente.nl
This work has tremendously profited from several discussions with Wouter Koolen. The author is moreover extremely grateful for helpful suggestions and several interesting remarks that were brought up by Matus Telgarsky. The research has been supported by the NWO/STAR grant 613.009.034b and the NWO Vidi grant VI.Vidi.192.021.

Schmidt-Hieber, 2023

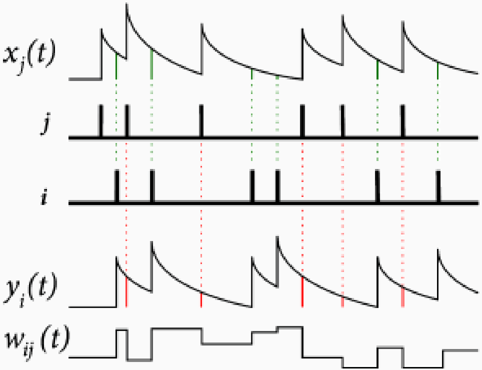
Spiking Neurons

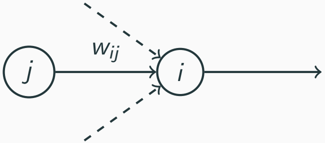




It takes multiple (20–50) concurrent incoming spikes to pass the threshold.

Updating the Weights: Hebbian / Spike Timing-Dependent Plasticity





Let w_{ij} denote the weight from a sending (presynaptic) neuron j to a receiving (post-synaptic) neuron i. Then

$$\Delta w_{ij} = C w_{ij} e^{-c|\Delta t|} \text{ at } t_{post} \text{ for } t_{pre} < t_{post}$$
$$\Delta w_{ij} = -C w_{ij} e^{-c|\Delta t|} \text{ at } t_{pre} \text{ for } t_{pre} > t_{post}$$

where $\Delta t := |t_{post} - t_{pre}|$.

Simplifying the weight update

Upon a presynaptic spike at time τ, with previous and next postsynaptic spikes at T_- and T_+,

$$w_{ij} \leftarrow w_{ij} + w_{ij} C(-e^{-c(\tau - T_-)} + e^{-c(T_+ - \tau)}).$$

Simplifying the weight update

Upon a presynaptic spike at time τ, with previous and next postsynaptic spikes at T_- and T_+,

$$w_{ij} \leftarrow w_{ij} + w_{ij} C(-e^{-c(\tau - T_-)} + e^{-c(T_+ - \tau)}).$$

For **supervised learning**, updates modulated by actual loss $L = L(\mathbf{w})$ minus anticipated loss $\bar{L}$

$$w_{ij} \leftarrow w_{ij} + w_{ij}\alpha(L - \bar{L})(e^{-c(\tau - T_-)} - e^{-c(T_+ - \tau)}).$$

Simplifying the weight update

Upon a presynaptic spike at time τ, with previous and next postsynaptic spikes at T_- and T_+,

$$w_{ij} \leftarrow w_{ij} + w_{ij} C(-e^{-c(\tau - T_-)} + e^{-c(T_+ - \tau)}).$$

For **supervised learning**, updates modulated by actual loss $L = L(\mathbf{w})$ minus anticipated loss $\bar{L}$

$$w_{ij} \leftarrow w_{ij} + w_{ij}\alpha(L - \bar{L})(e^{-c(\tau - T_-)} - e^{-c(T_+ - \tau)}).$$

Spike times τ are roughly **uniform** in window $[T_+ - T_-]$, which we assume is of fixed length $2A$. So with U independent uniform from $[\pm A]$:

$$w_{ij} \leftarrow w_{ij} + w_{ij}\alpha(L - \bar{L})(e^{-c(A+U)} - e^{-c(A-U)}).$$

Simplifying the weight update

Upon a presynaptic spike at time τ, with previous and next postsynaptic spikes at T_- and T_+,

$$w_{ij} \leftarrow w_{ij} + w_{ij} C(-e^{-c(\tau - T_-)} + e^{-c(T_+ - \tau)}).$$

For **supervised learning**, updates modulated by actual loss $L = L(\mathbf{w})$ minus anticipated loss $\bar{L}$

$$w_{ij} \leftarrow w_{ij} + w_{ij}\alpha(L - \bar{L})(e^{-c(\tau - T_-)} - e^{-c(T_+ - \tau)}).$$

Spike times τ are roughly **uniform** in window $[T_+ - T_-]$, which we assume is of fixed length $2A$. So with U independent uniform from $[\pm A]$:

$$w_{ij} \leftarrow w_{ij} + w_{ij}\alpha(L - \bar{L})(e^{-c(A+U)} - e^{-c(A-U)}).$$

Go to **logarithmic scale**, $\theta_{ij} := \ln w_{ij}$, and absorb constants

$$\begin{aligned}\theta_{ij} &\leftarrow \theta_{ij} + \ln\Big(1 + \alpha(L - \bar{L})(e^{-U} - e^{-U})\Big)\\ &\approx \theta_{ij} + \alpha(L - \bar{L})(e^{-U} - e^{-U})\end{aligned}$$

Fixing the details

We have arrived at the update rule from (Schmidt-Hieber, 2023). Here in vector form.

After round k with data $\mathbf{X}_k, Y_k$, update to

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \alpha_k \big(L(\boldsymbol{\theta}_{k-1} + \mathbf{U}_k, \mathbf{X}_k, Y_k) - \overline{L}_k \big) \big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k} \big).$$

Fixing the details

We have arrived at the update rule from (Schmidt-Hieber, 2023). Here in vector form.

After round k with data $\mathbf{X}_k, Y_k$, update to

$$\boldsymbol{\theta}_k \;=\; \boldsymbol{\theta}_{k-1} + \alpha_k \big(L(\boldsymbol{\theta}_{k-1} + \mathbf{U}_k, \mathbf{X}_k, Y_k) - \overline{L}_k \big) \big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k} \big).$$

Remains to choose the anticipated loss $\overline{L}_k$

- Zero?
- Loss of last round?
- Average loss so far?
- Treat as separate prediction task?
- Loss at $\boldsymbol{\theta}_{k-1} - \mathbf{U}_k$ (in theory a very good idea)
- ...

Simplified/concrete/tractable form

Use loss with independent realisation $\mathbf{U}'$ of noise as the anticipated loss $\bar{L}_k$.

$$\boldsymbol{\theta}_k \;=\; \boldsymbol{\theta}_{k-1} + \alpha_k \Big(L(\boldsymbol{\theta}_{k-1} + \mathbf{U}_k, \mathbf{X}_k, Y_k) - L(\boldsymbol{\theta}_{k-1} + \mathbf{U}'_k, \mathbf{X}_k, Y_k) \Big) \big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k} \big).$$

What is this scheme?

Let's stare at it:

$$\boldsymbol{\theta}_k \;=\; \boldsymbol{\theta}_{k-1} + \alpha_k \Big(L(\boldsymbol{\theta}_{k-1} + \mathbf{U}_k, \mathbf{X}_k, Y_k) - L(\boldsymbol{\theta}_{k-1} + \mathbf{U}'_k, \mathbf{X}_k, Y_k) \Big) \big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k} \big).$$

- Zeroth order: evaluates loss $L(\boldsymbol{\theta}_{k-1} + \mathbf{U}_k, \mathbf{X}_k, Y_k)$, **no derivatives**
- Two-point scheme: for each data item $\mathbf{X}_k$, Y_k evaluate loss of *two* parameters $\boldsymbol{\theta}_{k-1} + \mathbf{U}_k$ and $\boldsymbol{\theta}_{k-1} + \mathbf{U}'_k$

Benchmark Task: Zeroth order Linear Regression

Linear Regression

Model $(\boldsymbol{\theta}^\star, \mathbb{P}, \sigma^2)$: well-specified linear regression with random design.

Linear Regression

Model $(\boldsymbol{\theta}^\star, \mathbb{P}, \sigma^2)$: well-specified linear regression with random design.

unknown **true regression coefficient** $\boldsymbol{\theta}^\star \in \mathbb{R}^d$

unknown **covariate distribution** $\mathbb{P}$ on $\mathbb{R}^d$ and

known **noise level** $\sigma > 0$.

Linear Regression

Model $(\boldsymbol{\theta}^\star, \mathbb{P}, \sigma^2)$: well-specified linear regression with random design.

unknown **true regression coefficient** $\boldsymbol{\theta}^\star \in \mathbb{R}^d$
unknown **covariate distribution** $\mathbb{P}$ on $\mathbb{R}^d$ and
known **noise level** $\sigma > 0$.

- Covariates $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are drawn i.i.d. from $\mathbb{P}$.

Linear Regression

Model $(\boldsymbol{\theta}^\star, \mathbb{P}, \sigma^2)$: well-specified linear regression with random design.

unknown **true regression coefficient** $\boldsymbol{\theta}^\star \in \mathbb{R}^d$
unknown **covariate distribution** $\mathbb{P}$ on $\mathbb{R}^d$ and
known **noise level** $\sigma > 0$.

- Covariates $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are drawn i.i.d. from $\mathbb{P}$.
- Response variables are $Y_k := \mathbf{X}_k^\intercal \boldsymbol{\theta}^\star + \epsilon_k$ with independent Gaussian noise $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$.

Linear Regression

Model $(\boldsymbol{\theta}^\star, \mathbb{P}, \sigma^2)$: well-specified linear regression with random design.

unknown **true regression coefficient** $\boldsymbol{\theta}^\star \in \mathbb{R}^d$
unknown **covariate distribution** $\mathbb{P}$ on $\mathbb{R}^d$ and
known **noise level** $\sigma > 0$.

- Covariates $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are drawn i.i.d. from $\mathbb{P}$.
- Response variables are $Y_k := \mathbf{X}_k^\intercal \boldsymbol{\theta}^\star + \epsilon_k$ with independent Gaussian noise $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$.
- Loss of parameter $\boldsymbol{\theta}$ on data item $\mathbf{X}, Y$ is the **square loss**

$$L(\boldsymbol{\theta}, \mathbf{X}, Y) \;:=\; (\mathbf{X}^\intercal \boldsymbol{\theta} - Y)^2$$

Linear Regression

Model $(\boldsymbol{\theta}^\star, \mathbb{P}, \sigma^2)$: well-specified linear regression with random design.

unknown **true regression coefficient** $\boldsymbol{\theta}^\star \in \mathbb{R}^d$
unknown **covariate distribution** $\mathbb{P}$ on $\mathbb{R}^d$ and
known **noise level** $\sigma > 0$.

- Covariates $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are drawn i.i.d. from $\mathbb{P}$.
- Response variables are $Y_k := \mathbf{X}_k^\intercal \boldsymbol{\theta}^\star + \epsilon_k$ with independent Gaussian noise $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$.
- Loss of parameter $\boldsymbol{\theta}$ on data item $\mathbf{X}, Y$ is the **square loss**

$$L(\boldsymbol{\theta}, \mathbf{X}, Y) \;:=\; (\mathbf{X}^\intercal \boldsymbol{\theta} - Y)^2$$

- Risk (expected loss) of parameter $\boldsymbol{\theta}$ is

$$\mathbb{E}\big[L(\boldsymbol{\theta}, \mathbf{X}, Y)\big] \;=\; \mathbb{E}\left[(\mathbf{X}^\intercal(\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon)^2\right] \;=\; \|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|_Q^2 + \sigma^2$$

where we write $Q := \mathbb{E}[\mathbf{X}\mathbf{X}^\intercal] \succ 0$ for the (uncentred) covariance matrix of the covariates.

Linear Regression

Model $(\boldsymbol{\theta}^\star, \mathbb{P}, \sigma^2)$: well-specified linear regression with random design.

unknown **true regression coefficient** $\boldsymbol{\theta}^\star \in \mathbb{R}^d$
unknown **covariate distribution** $\mathbb{P}$ on $\mathbb{R}^d$ and
known **noise level** $\sigma > 0$.

- Covariates $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are drawn i.i.d. from $\mathbb{P}$.
- Response variables are $Y_k := \mathbf{X}_k^\intercal \boldsymbol{\theta}^\star + \epsilon_k$ with independent Gaussian noise $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$.
- Loss of parameter $\boldsymbol{\theta}$ on data item $\mathbf{X}, Y$ is the **square loss**

$$L(\boldsymbol{\theta}, \mathbf{X}, Y) \;:=\; (\mathbf{X}^\intercal \boldsymbol{\theta} - Y)^2$$

- Risk (expected loss) of parameter $\boldsymbol{\theta}$ is

$$\mathbb{E}\big[L(\boldsymbol{\theta}, \mathbf{X}, Y)\big] \;=\; \mathbb{E}\left[(\mathbf{X}^\intercal(\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon)^2\right] \;=\; \|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|_Q^2 + \sigma^2$$

  where we write $Q := \mathbb{E}[\mathbf{X}\mathbf{X}^\intercal] \succ 0$ for the (uncentred) covariance matrix of the covariates.
- Excess risk of $\boldsymbol{\theta}$ over risk minimiser $\boldsymbol{\theta}^\star$ is

$$\|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|_Q^2$$

Interaction Protocol: Two-Point Zeroth-order Stochastic Optimization

For $k = 1, 2, \ldots$

1. Learner picks *two* query points $\boldsymbol{\theta}_{k-1}^{(1)}$ and $\boldsymbol{\theta}_{k-1}^{(2)}$
2. Data item $\mathbf{X}_k, Y_k$ is drawn from linear regression model behind the scenes
3. Learner observes losses $L(\boldsymbol{\theta}_{k-1}^{(1)}, \mathbf{X}_k, Y_k)$ and $L(\boldsymbol{\theta}_{k-1}^{(2)}, \mathbf{X}_k, Y_k)$ of the two query points
4. Learner recommends evaluation point $\boldsymbol{\theta}_k$

Interaction Protocol: Two-Point Zeroth-order Stochastic Optimization

For $k = 1, 2, \ldots$

1. Learner picks *two* query points $\boldsymbol{\theta}_{k-1}^{(1)}$ and $\boldsymbol{\theta}_{k-1}^{(2)}$
2. Data item $\mathbf{X}_k, Y_k$ is drawn from linear regression model behind the scenes
3. Learner observes losses $L(\boldsymbol{\theta}_{k-1}^{(1)}, \mathbf{X}_k, Y_k)$ and $L(\boldsymbol{\theta}_{k-1}^{(2)}, \mathbf{X}_k, Y_k)$ of the two query points
4. Learner recommends evaluation point $\boldsymbol{\theta}_k$

NB: Learner has **no access** to data $\mathbf{X}_k, Y_k$ or gradient $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_{k-1}^{(1)}, \mathbf{X}_k, Y_k), \ldots$

Interaction Protocol: Two-Point Zeroth-order Stochastic Optimization

For $k = 1, 2, \ldots$

1. Learner picks *two* query points $\boldsymbol{\theta}_{k-1}^{(1)}$ and $\boldsymbol{\theta}_{k-1}^{(2)}$
2. Data item $\mathbf{X}_k, Y_k$ is drawn from linear regression model behind the scenes
3. Learner observes losses $L(\boldsymbol{\theta}_{k-1}^{(1)}, \mathbf{X}_k, Y_k)$ and $L(\boldsymbol{\theta}_{k-1}^{(2)}, \mathbf{X}_k, Y_k)$ of the two query points
4. Learner recommends evaluation point $\boldsymbol{\theta}_k$

NB: Learner has **no access** to data $\mathbf{X}_k, Y_k$ or gradient $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_{k-1}^{(1)}, \mathbf{X}_k, Y_k), \ldots$

We are interested in the excess risk of the evaluation point $\boldsymbol{\theta}_k$ as a function of time k.

Interaction Protocol: Two-Point Zeroth-order Stochastic Optimization

For $k = 1, 2, \dots$

1. Learner picks *two* query points $\boldsymbol{\theta}_{k-1}^{(1)}$ and $\boldsymbol{\theta}_{k-1}^{(2)}$
2. Data item $\mathbf{X}_k, Y_k$ is drawn from linear regression model behind the scenes
3. Learner observes losses $L(\boldsymbol{\theta}_{k-1}^{(1)}, \mathbf{X}_k, Y_k)$ and $L(\boldsymbol{\theta}_{k-1}^{(2)}, \mathbf{X}_k, Y_k)$ of the two query points
4. Learner recommends evaluation point $\boldsymbol{\theta}_k$

NB: Learner has **no access** to data $\mathbf{X}_k, Y_k$ or gradient $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_{k-1}^{(1)}, \mathbf{X}_k, Y_k), \dots$

We are interested in the excess risk of the evaluation point $\boldsymbol{\theta}_k$ as a function of time k.

The evaluation point $\boldsymbol{\theta}_k$ is random due to random data $\mathbf{X}_1, Y_1, \dots$ (and randomised queries)
So we evaluate a strategy for Learner by its expected excess risk after k rounds

$$\mathbb{E}_{(\boldsymbol{\theta}_0^{(1)}, \boldsymbol{\theta}_0^{(2)}, \mathbf{X}_1, Y_1)\dots(\boldsymbol{\theta}_{k-1}^{(1)}, \boldsymbol{\theta}_{k-1}^{(2)}, \mathbf{X}_k, Y_k)} \left[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^\star\|_Q^2 \right]$$

Impact of the Query model for Linear Regression

If we query at $\boldsymbol{\theta}$, we see the scalar loss

$$L = (\mathbf{X}^\intercal \boldsymbol{\theta} - Y)^2 = (\mathbf{X}^\intercal (\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon)^2$$

Impact of the Query model for Linear Regression

If we query at $\boldsymbol{\theta}$, we see the scalar loss

$$L \;=\; (\mathbf{X}^\intercal \boldsymbol{\theta} - Y)^2 \;=\; (\mathbf{X}^\intercal (\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon)^2$$

If further $\mathbf{X} \sim \mathbb{P} = \mathcal{N}(0, I)$ for simplicity, we have

$$\mathbf{X}^\intercal (\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon \quad \sim \; \mathcal{N}\left(0, \|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|^2 + \sigma^2\right)$$

so that the loss is scaled chi-squared

$$L \;= (\mathbf{X}^\intercal (\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon)^2 \;\sim\; \left(\|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|^2 + \sigma^2\right) \chi_1^2$$

Impact of the Query model for Linear Regression

If we query at $\boldsymbol{\theta}$, we see the scalar loss

$$L \;=\; (\mathbf{X}^\intercal \boldsymbol{\theta} - Y)^2 \;=\; (\mathbf{X}^\intercal (\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon)^2$$

If further $\mathbf{X} \sim \mathbb{P} = \mathcal{N}(0, I)$ for simplicity, we have

$$\mathbf{X}^\intercal (\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon \quad \sim \mathcal{N}\left(0, \|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|^2 + \sigma^2\right)$$

so that the loss is scaled chi-squared

$$L \;= (\mathbf{X}^\intercal (\boldsymbol{\theta} - \boldsymbol{\theta}^\star) - \epsilon)^2 \;\sim\; \left(\|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|^2 + \sigma^2\right) \chi_1^2$$

Multiplicative noise. Very different from additive noise $L \sim \left(\|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|^2 + \sigma^2\right) + \mathcal{N}(0, \text{const})$.

How hard is this task?

Minimax lower bound for any two-point scheme $\mathcal{V}_k, \widehat{\boldsymbol{\theta}}$.

Theorem

If $d \geq 3$ and $k \geq d^2$, then,

$$\inf_{\mathcal{V}_k, \widehat{\boldsymbol{\theta}}} \sup_{\boldsymbol{\theta}^\star \in B_R(0)} \mathbb{E}_{\boldsymbol{\theta}^\star, \mathcal{V}_k} \left[\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|^2 \right] \geq \frac{1}{162}\left(1 - \frac{1}{\sqrt{2}}\right)\left(R^2 \wedge \frac{d^2}{k}\sigma^2\right).$$

How hard is this task?

Minimax lower bound for any two-point scheme $\mathcal{V}_k, \widehat{\boldsymbol{\theta}}$.

Theorem

If $d \geq 3$ and $k \geq d^2$, then,

$$\inf_{\mathcal{V}_k, \widehat{\boldsymbol{\theta}}} \sup_{\boldsymbol{\theta}^\star \in B_R(0)} \mathbb{E}_{\boldsymbol{\theta}^\star, \mathcal{V}_k} \left[\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|^2 \right] \geq \frac{1}{162}\left(1 - \frac{1}{\sqrt{2}}\right)\left(R^2 \wedge \frac{d^2}{k}\sigma^2\right).$$

Minimax excess risk lower bound for non-adaptive two-point schemes

Theorem

If $d \geq 6$, then for any $k = 1, 2, \ldots$

$$\inf_{\mathcal{V}_k \in \mathcal{M}_k, \widehat{\boldsymbol{\theta}}} \sup_{\boldsymbol{\theta}^\star \in B_R(0)} \mathbb{E}_{\boldsymbol{\theta}^\star, \mathcal{V}_k} \left[\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|^2 \right] \geq 2^{-18}\left(R^2 \wedge \frac{d^2}{k}(R^2 \vee \sigma^2)\right).$$

BNN meets Linear Regression

Slogan

For our combination of loss and update, (almost) everything is fully explicit **linear/quadratic**.

Connecting BNN to 2P-0O-StochOpt

We query at

$$\boldsymbol{\theta}_{k-1}^{(1)} \;=\; \boldsymbol{\theta}_{k-1} + \mathbf{U}_k \qquad\qquad \boldsymbol{\theta}_{k-1}^{(2)} \;=\; \boldsymbol{\theta}_{k-1} + \mathbf{U}'_k$$

Connecting BNN to 2P-0O-StochOpt

We query at

$$\boldsymbol{\theta}_{k-1}^{(1)} = \boldsymbol{\theta}_{k-1} + \mathbf{U}_k \qquad\qquad \boldsymbol{\theta}_{k-1}^{(2)} = \boldsymbol{\theta}_{k-1} + \mathbf{U}'_k$$

and update using

$$\begin{aligned}
\boldsymbol{\theta}_k &= \boldsymbol{\theta}_{k-1} + \alpha_k \Big((\mathbf{X}_k^\intercal(\boldsymbol{\theta}_{k-1} + \mathbf{U}_k) - Y_k)^2 - (\mathbf{X}_k^\intercal(\boldsymbol{\theta}_{k-1} + \mathbf{U}'_k) - Y_k)^2 \Big) \big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k} \big) \\
&= \boldsymbol{\theta}_{k-1} + \alpha_k \Big((\mathbf{X}_k^\intercal(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}^\star + \mathbf{U}_k) - \epsilon_k)^2 - (\mathbf{X}_k^\intercal(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}^\star + \mathbf{U}'_k) - \epsilon_k)^2 \Big) \big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k} \big).
\end{aligned}$$

Connecting BNN to 2P-0O-StochOpt

We query at

$$\boldsymbol{\theta}_{k-1}^{(1)} = \boldsymbol{\theta}_{k-1} + \mathbf{U}_k \qquad \boldsymbol{\theta}_{k-1}^{(2)} = \boldsymbol{\theta}_{k-1} + \mathbf{U}'_k$$

and update using

$$\begin{aligned}\boldsymbol{\theta}_k &= \boldsymbol{\theta}_{k-1} + \alpha_k\Big((\mathbf{X}_k^\intercal(\boldsymbol{\theta}_{k-1} + \mathbf{U}_k) - Y_k)^2 - (\mathbf{X}_k^\intercal(\boldsymbol{\theta}_{k-1} + \mathbf{U}'_k) - Y_k)^2\Big)\big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\big) \\ &= \boldsymbol{\theta}_{k-1} + \alpha_k\Big((\mathbf{X}_k^\intercal(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}^\star + \mathbf{U}_k) - \epsilon_k)^2 - (\mathbf{X}_k^\intercal(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}^\star + \mathbf{U}'_k) - \epsilon_k)^2\Big)\big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\big).\end{aligned}$$

So with $\boldsymbol{\delta}_k := \boldsymbol{\theta}_k - \boldsymbol{\theta}^\star$, we get the recurrence

$$\begin{aligned}\boldsymbol{\delta}_k &= \boldsymbol{\delta}_{k-1} + \alpha_k\Big((\mathbf{X}_k^\intercal(\boldsymbol{\delta}_{k-1} + \mathbf{U}_k) - \epsilon_k)^2 - (\mathbf{X}_k^\intercal(\boldsymbol{\delta}_{k-1} + \mathbf{U}'_k) - \epsilon_k)^2\Big)\big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\big). \\ &= \boldsymbol{\delta}_{k-1} + \alpha_k\Big(2(\mathbf{X}_k^\intercal\boldsymbol{\delta}_{k-1} - \epsilon_k)\mathbf{X}_k^\intercal(\mathbf{U}'_k - \mathbf{U}_k) + (\mathbf{X}_k^\intercal\mathbf{U}_k)^2 - (\mathbf{X}_k^\intercal\mathbf{U}'_k)^2\Big)\big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\big). \\ &= \Big(I + 2\alpha_k\big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\big)(\mathbf{U}'_k - \mathbf{U}_k)^\intercal\mathbf{X}_k\mathbf{X}_k^\intercal\Big)\boldsymbol{\delta}_{k-1} \\ &\quad + \alpha_k\Big(-2\epsilon_k\mathbf{X}_k^\intercal(\mathbf{U}'_k - \mathbf{U}_k) + (\mathbf{X}_k^\intercal\mathbf{U}_k)^2 - (\mathbf{X}_k^\intercal\mathbf{U}'_k)^2\Big)\big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\big)\end{aligned}$$

Does it even make sense on average?

We expressed our update rule in Stochastic Approximation form

$$\boldsymbol{\delta}_k \;=\; (I - \alpha_k \mathbf{A}_k)\boldsymbol{\delta}_{k-1} + \alpha_k \mathbf{b}_k$$

for i.i.d. random matrix $\mathbf{A}_k$ and vector $\mathbf{b}_k$ given by

$$\begin{aligned}
\mathbf{A}_k &:= -2\left(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\right)(\mathbf{U}'_k - \mathbf{U}_k)^\intercal \mathbf{X}_k \mathbf{X}_k^\intercal, \\
\mathbf{b}_k &:= \left(-2\epsilon_k \mathbf{X}_k^\intercal(\mathbf{U}'_k - \mathbf{U}_k) + (\mathbf{X}_k^\intercal \mathbf{U}_k)^2 - (\mathbf{X}_k^\intercal \mathbf{U}'_k)^2\right)\left(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\right).
\end{aligned}$$

Does it even make sense on average?

We expressed our update rule in Stochastic Approximation form

$$\boldsymbol{\delta}_k \;=\; (I - \alpha_k \mathbf{A}_k)\boldsymbol{\delta}_{k-1} + \alpha_k \mathbf{b}_k$$

for i.i.d. random matrix $\mathbf{A}_k$ and vector $\mathbf{b}_k$ given by

$$\begin{aligned} \mathbf{A}_k \;&:=\; -2\big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\big)(\mathbf{U}'_k - \mathbf{U}_k)^\intercal \mathbf{X}_k \mathbf{X}_k^\intercal, \\ \mathbf{b}_k \;&:=\; \Big(-2\epsilon_k \mathbf{X}_k^\intercal(\mathbf{U}'_k - \mathbf{U}_k) + (\mathbf{X}_k^\intercal \mathbf{U}_k)^2 - (\mathbf{X}_k^\intercal \mathbf{U}'_k)^2\Big)\big(e^{-\mathbf{U}_k} - e^{\mathbf{U}_k}\big). \end{aligned}$$

We have $\mathbb{E}[\mathbf{b}] = 0$ and $\mathbb{E}[\mathbf{A}] = \eta Q$ with constant $\eta := 2\,\mathbb{E}[\big(e^{-U} - e^{U}\big)U]$ depending on the scale A of noise $\mathbf{U}$.

In expectation, our update gives

$$\mathbb{E}_k[\boldsymbol{\delta}_k] \;=\; \Big(I - \alpha_k \eta Q\Big)\boldsymbol{\delta}_{k-1}$$

That is **exactly** gradient descent on the **risk** $\|\boldsymbol{\delta}\|_Q^2 + \sigma^2$, with learning rate $\frac{1}{2}\alpha_k \eta$.

Case closed?

So the average iterate $\mathbb{E}[\boldsymbol{\theta}_k] \to \boldsymbol{\theta}^\star$ converges to the risk minimiser. Exponentially fast.

Case closed?

So the average iterate $\mathbb{E}[\boldsymbol{\theta}_k] \to \boldsymbol{\theta}^\star$ converges to the risk minimiser. Exponentially fast.

The metric of interest is excess risk $\|\boldsymbol{\delta}\|_Q^2$. Variance matters!

Case closed?

So the average iterate $\mathbb{E}[\boldsymbol{\theta}_k] \to \boldsymbol{\theta}^\star$ converges to the risk minimiser. Exponentially fast.

The metric of interest is excess risk $\|\boldsymbol{\delta}\|_Q^2$. Variance matters!

So let's work on the expected excess risk after k rounds (whp bounds also interesting):

$$\Xi_k \;:=\; \mathbb{E}\left[\|\boldsymbol{\delta}_k\|_Q^2\right] \qquad \text{where} \qquad Q \;=\; \mathbb{E}[\mathbf{X}\mathbf{X}^\intercal]$$

Can we get a recurrence for Ξ_k? Yes!

Recurrence for excess risk

Recall our update rule is of the form

$$\boldsymbol{\delta}_k = (I - \alpha_k \mathbf{A}_k)\boldsymbol{\delta}_{k-1} + \alpha_k \mathbf{b}_k$$

for i.i.d. random matrix $\mathbf{A}_k$ and vector $\mathbf{b}_k$, with $\mathbb{E}[\mathbf{A}] = \eta Q$, and $\mathbb{E}[\mathbf{b}] = \mathbb{E}[\mathbf{A}^\intercal Q \mathbf{b}] = 0$.

Recurrence for excess risk

Recall our update rule is of the form

$$\boldsymbol{\delta}_k \;=\; (I - \alpha_k \mathbf{A}_k)\boldsymbol{\delta}_{k-1} + \alpha_k \mathbf{b}_k$$

for i.i.d. random matrix $\mathbf{A}_k$ and vector $\mathbf{b}_k$, with $\mathbb{E}[\mathbf{A}] = \eta Q$, and $\mathbb{E}[\mathbf{b}] = \mathbb{E}[\mathbf{A}^\intercal Q \mathbf{b}] = 0$.

So the excess risk satisfies

$$\begin{aligned}
\Xi_k &= \mathbb{E}_k\left[\boldsymbol{\delta}_k^\intercal Q \boldsymbol{\delta}_k\right] \\
&= \mathbb{E}_k\left[((I - \alpha_k \mathbf{A}_k)\boldsymbol{\delta}_{k-1} + \alpha_k \mathbf{b}_k)^\intercal Q((I - \alpha_k \mathbf{A}_k)\boldsymbol{\delta}_{k-1} + \alpha_k \mathbf{b}_k)\right] \\
&= \boldsymbol{\delta}_{k-1}^\intercal \,\mathbb{E}_k\left[(I - \alpha_k \mathbf{A}_k)^\intercal Q (I - \alpha_k \mathbf{A}_k)\right] \boldsymbol{\delta}_{k-1} + \alpha_k^2\, \mathbb{E}_k\left[\mathbf{b}_k^\intercal Q \mathbf{b}_k\right] \\
&= \boldsymbol{\delta}_{k-1}^\intercal \left\{(I - \alpha_k \eta Q)^\intercal Q (I - \alpha_k \eta Q) + \alpha_k^2\, \mathbb{E}_k\left[(\mathbf{A}_k - \eta Q)^\intercal Q (\mathbf{A}_k - \eta Q)\right]\right\} \boldsymbol{\delta}_{k-1} + \alpha_k^2\, \mathbb{E}_k\left[\mathbf{b}_k^\intercal Q \mathbf{b}_k\right] \\
&\leq \left((1 - \alpha_k \eta \lambda_{\mathsf{min}}(Q))^2 + \alpha_k^2 \beta\right) \Xi_{k-1} + \alpha_k^2 \gamma
\end{aligned}$$

abbreviating $\beta := \lambda_{\mathsf{max}}\left(\mathbb{E}_k\left[Q^{-1/2}(\mathbf{A}_k - \eta Q)^\intercal Q (\mathbf{A}_k - \eta Q) Q^{-1/2}\right]\right)$ and $\gamma := \mathbb{E}_k\left[\mathbf{b}_k^\intercal Q \mathbf{b}_k\right]$.

Inspecting where we are

Our state of progress so far is

$$\Xi_k \;\leq\; \Big((1-\alpha_k\eta\lambda_{\mathsf{min}}(Q))^2+\alpha_k^2\beta\Big)\Xi_{k-1}+\alpha_k^2\gamma$$

for fixed η, $\lambda_{\mathsf{min}}(Q)$, β and γ. The question is how to tune α_k. This is now a scalar problem.

Inspecting where we are

Our state of progress so far is

$$\Xi_k \leq \Big((1-\alpha_k\eta\lambda_{\min}(Q))^2 + \alpha_k^2\beta\Big)\Xi_{k-1} + \alpha_k^2\gamma$$

for fixed η, $\lambda_{\min}(Q)$, β and γ. The question is how to tune α_k. This is now a scalar problem.

Cancelling derivative reveals this bound is optimised in α_k at

$$\alpha_k^* = \frac{\eta\lambda_{\min}(Q)}{\eta^2\lambda_{\min}(Q)^2 + \beta + \frac{\gamma}{\Xi_{k-1}}}$$

and at that point we obtain

$$\Xi_k \leq \left(\frac{\beta + \frac{\gamma}{\Xi_{k-1}}}{\eta^2\lambda_{\min}(Q)^2 + \beta + \frac{\gamma}{\Xi_{k-1}}}\right)\Xi_{k-1}$$

Cute ODE upper bound

We can write our recurrence so far as a difference equation

$$\frac{\Xi_k - \Xi_{k-1}}{\Xi_{k-1}} \leq -\frac{\eta^2 \lambda_{\min}(Q)^2}{\eta^2 \lambda_{\min}(Q)^2 + \beta + \frac{\gamma}{\Xi_{k-1}}}$$

and solve the corresponding **differential equation** with equality to find

$$\frac{\Xi_k}{\Xi_1} \leq \frac{y}{W\left(ye^{y+xk}\right)} \quad \text{with} \quad x := \frac{\eta^2 \lambda_{\min}(Q)^2}{\eta^2 \lambda_{\min}(Q)^2 + \beta} \quad \text{and} \quad y := \frac{\gamma / \Xi_1}{\eta^2 \lambda_{\min}(Q)^2 + \beta}$$

so that all in all the excess risk decays as $\Xi_k \cong \Xi_1 / k$ and the **learning rate** as $\alpha_k^* \cong 1/k$.

More precisely in terms of relevant problem-dependent constants

We arrive at excess risk bound

Theorem

$$\Xi_k \;\leq\; \frac{121\kappa d^2}{2\lambda_{\min}(Q)}\frac{48\sigma^2 M_2 + 107A^2 d M_4}{k+C}$$

where $\kappa = \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)}$ *is the condition number of* Q*, and* M_p *bounds the* i*th moment of each entry of the covariate vector* $\mathbf{X} \sim \mathbb{P}$.

If $A^2 d$ is at most of order σ^2, this is d^2/k. Matching lower bounds.

Reflections

To think about

- Is the optimal tuning $\alpha_k \cong 1/k$ biologically realistic?
- Learning rate α_k needs to decay. What decides a *new task* in the brain?
- Optimal tuning for α_k depends on zoo of unknowns. How are these estimated?
- Brutal tuning $\alpha_k = \frac{c}{C+k}$ may result in risk rising to e^{const} before $1/k$ decay kicks in.
- Is the noise rate A biologically small compared to $\sigma/\sqrt{d}$?
- Realism in the model
  - More than one neuron
  - Depth, architecture
  - Other tasks and losses

Conclusion

We saw a simple model for spiking neurons inspired by biology.

We saw a concrete rendering of resulting update rule.

We interpreted it as a zeroth-order two-point iterative scheme.

We evaluated this scheme on a linear regression task.

We derived a rate for the excess risk, and proved that it matches lower bounds.

Let's talk!

References i

Gerstner, W., W. M. Kistler, R. Naud, and L. Paninski (2014). **Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition.** Cambridge University Press.

Schmidt-Hieber, J. (2023). **"Interpreting learning in biological neural networks as zero-order optimization method".** In: *arXiv preprint*, arXiv:2301.11777.