Towards Sobolev Pruning

Training and pruning surrogate models with sensitivity information

Neil Kichler    STCE, RWTH Aachen    June 4, 2024
Joint work with:

Sher Afghan (STCE, RWTH Aachen)  Uwe Naumann (STCE, RWTH Aachen)
Motivation

Goal: Find a surrogate model, that is
- accurate
- efficient
- robust.

Typically:
- Fit a (large) neural network
- Prune network into a surrogate model

Shortcoming:
- Surrogate model does not consider sensitivities and uncertainties.
  → Derivative information may differ drastically.

Towards Sobolev Pruning | Neil Kichler
Motivation

Goal: Find a surrogate model, that is
- accurate
- efficient
- robust.

Typically:
- Fit a (large) neural network
- Prune network into a surrogate model
Motivation

Goal: Find a surrogate model, that is
  • accurate
  • efficient
  • robust.

Typically:
  • Fit a (large) neural network
  • Prune network into a surrogate model

Shortcoming:
  • Surrogate model does not consider sensitivities and uncertainties.
    \(\Rightarrow\) Derivative information may differ drastically.
Motivation

Sensitivity information is often an afterthought, but:

• Captures vital information in many applications
• Crucial in optimization (e.g., Newton’s method)
• Helps to learn robust & accurate surrogate models

Motif

Incorporate sensitivity information throughout the learning and pruning process.
Outline

1. Sobolev Training
2. Pruning
3. Case Study
4. Results
5. Conclusion
Setup

In the domain of surrogate modelling with neural networks, let:

- \( \mathcal{N}(\vartheta) \): The surrogate model as a neural network.
  - \( \vartheta \): The parameters of the neural network.
  - \( f_{\vartheta} \): The learned function.

- \( S \): The reference model sampler.

- \( \mathcal{L} \): The loss function, e.g. \( \mathcal{L} = ||\cdot||_2^2 \).

- \( (x_i, y_i) \): An (input, target) sample.
Vanilla Training

Match targets by differentiating the loss and optimize using, e.g., SGD.
Sobolev Training

Match targets and differential targets.
Sobolev Loss

Definition (Sobolev Loss)

Given input $x$, target $y$, predicted output $f_\theta(x)$, differential target $\nabla_x y$, and predicted differential $\nabla_x f_\theta(x)$, the differential loss is defined by:

$$\|y - f_\theta(x)\|_2^2 + \lambda \|\nabla_x y - \nabla_x f_\theta(x)\|_2^2,$$

where $\lambda \in \mathbb{R}_{\geq 0}$ is an added balancing factor.
Sobolev Loss: Interpretation

Srinivas and Fleuret [1] highlight that the Sobolev loss naturally arises when considering perturbations.

Perturbation perspective:

Consider perturbation of input. By Taylor expansion [1]:

\[
\mathbb{E}_{\epsilon \sim N(0, \sigma^2)} \left[ \sum_{i=1}^{m} (f(x_i + \epsilon) - f_{\theta}(x_i + \epsilon))^2 \right] = \sum_{i=1}^{m} (f(x_i) - f_{\theta}(x_i))^2 \\
+ \sigma^2 \sum_{i=1}^{m} \| \nabla_x f(x_i) - \nabla_x f_{\theta}(x_i) \|_2^2 + \mathcal{O}(\sigma^4).
\]
Sobolev Training

**Algorithm** Sobolev Training [2].

**Require:** The following inputs must all be initialized.
- Surrogate model $\mathcal{N}(\vartheta)$ with function $f_\vartheta$ and parameters $\vartheta$
- Reference model $S$
- Optimizer $G$

```latex
\textbf{while} $\vartheta$ not converged \textbf{do}
\begin{align*}
\{(x_i, y_i, \nabla_x y_i)\}_{i=1}^m &\sim S \quad \triangleright \text{Sample training data} \\
\hat{g} &\leftarrow \frac{1}{m} \nabla_\vartheta \sum_{i=1}^m \mathcal{L}(f_\vartheta(x_i), y_i) + \lambda \mathcal{L}(\nabla_x f_\vartheta(x_i), \nabla_x y_i) \quad \triangleright \text{Update parameters} \\
\vartheta &\leftarrow G(\vartheta, \hat{g})
\end{align*}
\textbf{end while}

\textbf{return} $\mathcal{N}$
```
How large should the surrogate model be?
How large should the surrogate model be?
How small can the surrogate model get?
Pruning

Goal
Prune the surrogate model to increase the computational efficiency while retaining accuracy.

Dense Pruning.

Sparse Pruning.
Pruning

Goal
Prune the surrogate model to increase the computational efficiency while retaining accuracy.

Why not start with Sparse Training?
Dynamic Sparse Training (e.g., SET [3], RIGL [4]) works, but:
• are not designed for modern architectures $\rightarrow$ requires masking.
• still requires reasonable starting size guess.
• worse performance, in practice, compared to dense2sparse training.
Pruning

Typically

Magnitude Pruning: $|w|$
Salience Pruning: $|\frac{\partial C}{\partial w} w|$
Pruning

Typically

Magnitude Pruning: $|w|$
Salience Pruning: $|\frac{\partial C}{\partial w}|$

... 

Downsides?

• must iterate over training data.
• sensitivity information?
• no global perspective.
Pruning

Can we get a global perspective on weight importance?

Interval arithmetic!
Interval Arithmetic

Fundamentals:

- Considers variables to be inside a fixed range, a trust region.
- Replace $x \in \mathbb{R}$ with $[x] \in \mathbb{IR}$.
- $[x] = [x, \bar{x}]$ if $\{x \in \mathbb{R} \mid x \leq x \leq \bar{x}\}$.
- $f([x]) := \{f(x) \mid x \in [x]\}$.

Fundamental Theorem of Interval Arithmetic

A function $f$ over an interval input box $X = ([x]_0, \cdots, [x]_n)$ is guaranteed to enclose the range of $f$ over those inputs, i.e., $\text{range}(f) \subseteq f(X)$ (Moore et al.).
Interval Adjoint

Algorithmic Differentiation (AD) [6] naturally applies to interval arithmetic [7].

**AD on Interval Arithmetic**

- Apply AD as usual.
- Replace all operations of the source transformed code with the interval arithmetic equivalent.
Interval Adjoint Significance Analysis

Significance Measure

\[ S_{[y]}([n]_{l,i}) = \text{width}([n]_{l,i}) \cdot \max(|\nabla_{[n]_{l,i}}[y]|), \]

where:

- \([y]\): interval output
- \(\text{width}([x]) = \bar{x} - \underline{x}\)
- \(l\): hidden layer index
- \(i\): node index in given layer
General framework

Towards Sobolev Pruning | Neil Kichler
Consider the SDE:

$$dS_t = a(S, t) \, dt + b(S, t) \, dW_t,$$

where:

- $dW_t$ is a Wiener process
- $a : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$, the drift
- $b : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$, the volatility
Case Study: Option pricing

Consider the SDE:

$$dS_t = a(S, t) \, dt + b(S, t) \, dW_t,$$

Interested in the price:

$$V = \mathbb{E}[\nu(S_T, K)],$$

with:

- $S_T$ the price at maturity $T$
- $K$ the strike price
- $\nu(S_T, K)$ the payoff function
The Bachelier model can be described as a SDE:

\[ dS_t = \mu S_t dt + \sigma dW_t, \]

where:

- \( t \), the time index.
- \( \mu \), the constant drift for the interest rate.
- \( \sigma \), the constant volatility.
- \( S_t \), the underlying asset price at time \( t \).
- \( dW_t \), a Wiener process, i.e. Brownian motion.
Gaussian Basket

Definition (Basket)

A Basket $\mathbf{S}_t \in \mathbb{R}^m$ of $m$ securities $S_t^{[0]}, \ldots, S_t^{[m]}$ has price:

$$S_t = \sum_{i=0}^{m} \omega_i S_t^{[i]}, \quad \sum_{i=0}^{m} \omega_i = 1,$$

where $\omega_i$ is the weight associated with the $i$th security.

$\Rightarrow$ Directly applicable to the Bachelier model.
Surrogate Objective

Given

\( S_0 \): initial spot price from basket

Find

\( V \): option price (Value)

\( \frac{\partial V}{\partial S_0} \): 1st-order price sensitivity (Delta)

\( \frac{\partial^2 V}{\partial S_0^2} \): 2nd-order price sensitivity (Gamma)

⇒ Solved using Least-Squares Monte Carlo.
Least-Squares Monte Carlo

Least-Squares Monte Carlo method is equivalent to optimizing

\[ \vartheta^* = \arg \min_{\vartheta} \mathbb{E}_{(\theta, z) \sim \Theta_{\text{in}} \times Z} \left[ \| \nu(g(\theta, z)) - f_{\vartheta}(\theta) \|_2^2 \right], \]

where:

- \( f_{\vartheta} \) is the fitted curve with coefficients \( \vartheta \)
- random input parameters \( \theta \sim \Theta_{\text{in}} \) (here: \( \theta = \{ S_0 \} \))
- random path noise samples \( z \sim Z \)
- payoff function \( \nu \).

Towards Sobolev Pruning | Neil Kichler
Regression using Neural Networks

Baseline results (normalized) of Vanilla ML using a basic Multi-Layer Perceptron (MLP).
After Interval Adjoint Significance Pruning

Results of pruned model (and vanilla ML fine-tuning).
After Sobolev fine-tuning

Results after Sobolev fine-tuning on derivative samples from learned NN.
After Sobolev fine-tuning on reference

Results after Sobolev fine-tuning on derivative samples from Bachelier reference model.
Results: Overview

$R^2$ score of surrogate models for a Bachelier modelled basket option (7 dimensions).

<table>
<thead>
<tr>
<th>Predict</th>
<th>Oversized</th>
<th>Pruned</th>
<th>Sobolev fine-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NN</td>
<td>NN</td>
<td>NN Data</td>
</tr>
<tr>
<td>Values</td>
<td>0.999545</td>
<td>0.999296</td>
<td>0.999805</td>
</tr>
<tr>
<td>Deltas</td>
<td>0.998700</td>
<td>0.996718</td>
<td>0.999479</td>
</tr>
<tr>
<td>Gammas</td>
<td>0.997033</td>
<td>0.902470</td>
<td>0.987393</td>
</tr>
</tbody>
</table>

Towards Sobolev Pruning | Neil Kichler
Limitations & Future Work

Things to keep in mind

It requires:

- AD for interval arithmetic (no built-in support in ML libraries).
- Access to intermediate local partial derivatives.
- Derivative information from reference model
  - need access to source.
  - Expensive Jacobians? Approximate by sampling vjps.

Going beyond

- Add second-order sensitivity information?
Conclusion

Add sensitivity information by:

• Pruning previously learned network with Interval Adjoint Significance Analysis.
• Using Sobolev Training to improve accuracy and retain sensitivity information.

Paper & Code: github.com/neilkichler/sobolev-pruning
Conclusion

Thank you for your attention!

Add sensitivity information by:

- Pruning previously learned network with Interval Adjoint Significance Analysis.
- Using Sobolev Training to improve accuracy and retain sensitivity information.

Paper & Code: github.com/neilkichler/sobolev-pruning
References I


References II


Backup Slides
Comparison of Loss Curves

Surrogates for Bachelier Basket Option

- Vanilla Train Loss
- Vanilla Test Loss
- Sobolev Train Loss
- Sobolev Test Loss
Second-order sensitivity information

Hessian too expensive? → sample directions.

Random directions

Draw random vectors $\mathbf{v}$, s.t. $\mathbb{E}[\mathbf{vv}^\top] = \mathbf{I}$.

$$\Rightarrow \mathbb{E}[\mathbf{Hvv}^\top] = \mathbf{H}\mathbb{E}[\mathbf{vv}^\top] = \mathbf{H}.$$  

E.g., $N(\mu = 0, \Sigma = \mathbf{I})$.

PCA directions

- Take principal components.
- Further reduction by taking k-most significant components.

Towards Sobolev Pruning | Neil Kichler
Pathwise Sensitivities

Fix some random sample path $z \sim \mathcal{Z}$ and input parameters $\theta$.

We have unbiased estimates of, e.g., pathwise deltas, if:

$$
\mathbb{E}_{z \sim \mathcal{Z}} \left[ \frac{\partial}{\partial S_0} \nu(g(\theta, z)) \right] = \frac{\partial}{\partial S_0} \mathbb{E}_{z \sim \mathcal{Z}} \left[ \nu(g(\theta, z)) \right],
$$

i.e. we can interchange the expectation with the derivative operator.

Practical Conditions for (1):

- Payoff $\nu$ must be differentiable almost everywhere.
- Payoff $\nu$ is Lipschitz continuous.
Smoothing

We perform smoothing between function $f_1 : \mathbb{R} \to \mathbb{R}$ and $f_2 : \mathbb{R} \to \mathbb{R}$ via

\[ \tilde{f} : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R} \]

deﬁned as

\[ \tilde{f}(x, p, w) = (1 - \sigma(x, p, w))f_1(x) + \sigma(x, p, w)f_2(x), \]

where

\[ \sigma(x, p, w) = \frac{1}{1 + e^{-\frac{x-p}{w}}}, \]

and $p$ is the position to change between the two functions and $w$ the width of the smoothing.
Smoothing

For $\nu = (\cdot)^+$:

Split function into

$$
\begin{cases}
0, & x < 0 \\
x, & x \geq 0
\end{cases}
$$

We obtain:

$$
\tilde{\nu}(x, w) = \frac{x}{1 + e^{-\frac{x}{w}}}
$$

Equivalent to SiLU if $p = 0$, $w = 1$.

Smoothing $(\cdot)^+$, with $p = 0$, $w = 0.05$. 

Towards Sobolev Pruning | Neil Kichler