Title: Closed-Form Interpretation of Neural Network Classifiers with Symbolic Regression Gradients

Speakers: Sebastian Wetzel

Series: Machine Learning Initiative

Date: January 19, 2024 - 2:30 PM

URL: https://pirsa.org/24010081

Abstract: I introduce a unified framework for interpreting neural network classifiers tailored toward automated scientific discovery. In contrast to neural network-based regression, for classification, it is in general impossible to find a one-to-one mapping from the neural network to a symbolic equation even if the neural network itself bases its classification on a quantity that can be written as a closed-form equation. In this paper, I embed a trained neural network into an equivalence class of classifying functions that base their decisions on the same quantity. I interpret neural networks by finding an intersection between this equivalence class and human-readable equations defined by the search space of symbolic regression. The approach is not limited to classifiers or full neural networks and can be applied to arbitrary neurons in hidden layers or latent spaces or to simplify the process of interpreting neural network regressors.

Zoom link

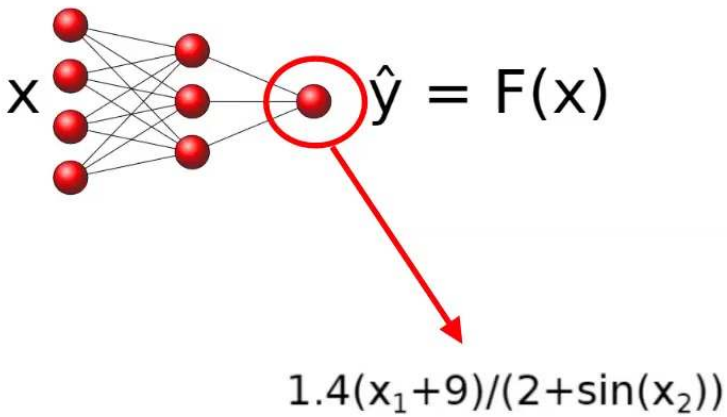
Closed-Form Interpretation of Neural Network Classifiers with Symbolic Regression Gradients

Perimeter Institute, Jan 19, 2024

Sebastian Johann Wetzel

PERIMETER INSTITUTE PIQUIL HOMES+ UNIVERSITY OF WATERLOO CONCORDIA CUM VERITATE

Overview



- Not limited to Classification
- Applicable to Neurons in Hidden Layers
- Can Help Simplify Symbolic Regression Problems

Overview

✗ Explaining the Words in the Title

✗ Theoretical Framework
- Equivalence Class
- Interpretation Algorithm

✗ Experiments
- Recover Eqs from Neural Network Classifiers
- Why Symbolic Classification is not Interpretation

Closed-Form Interpretation of Neural Network Classifiers with Symbolic Regression Gradients, Wetzel, arXiv:2401.04978

Closed-Form Equations

v·t·e	Arithmetic expressions	Polynomial expressions	Algebraic expressions	Closed-form expressions	Analytic expressions	Mathematical expressions
Constant	Yes	Yes	Yes	Yes	Yes	Yes
Elementary arithmetic operation	Yes	Addition, subtraction, and multiplication only	Yes	Yes	Yes	Yes
Finite sum	Yes	Yes	Yes	Yes	Yes	Yes
Finite product	Yes	Yes	Yes	Yes	Yes	Yes
Finite continued fraction	Yes	No	Yes	Yes	Yes	Yes
Variable	No	Yes	Yes	Yes	Yes	Yes
Integer exponent	No	Yes	Yes	Yes	Yes	Yes
Integer nth root	No	No	Yes	Yes	Yes	Yes
Rational exponent	No	No	Yes	Yes	Yes	Yes
Integer factorial	No	No	Yes	Yes	Yes	Yes
Irrational exponent	No	No	No	Yes	Yes	Yes
Exponential function	No	No	No	Yes	Yes	Yes
Logarithm	No	No	No	Yes	Yes	Yes
Trigonometric function	No	No	No	Yes	Yes	Yes
Inverse trigonometric function	No	No	No	Yes	Yes	Yes
Hyperbolic function	No	No	No	Yes	Yes	Yes
Inverse hyperbolic function	No	No	No	Yes	Yes	Yes
Root of a polynomial that is not an algebraic solution	No	No	No	No	Yes	Yes
Gamma function and factorial of a non-integer	No	No	No	No	Yes	Yes
Bessel function	No	No	No	No	Yes	Yes
Special function	No	No	No	No	Yes	Yes
Infinite sum (series) (including power series)	No	No	No	No	Convergent only	Yes
Infinite product	No	No	No	No	Convergent only	Yes
Infinite continued fraction	No	No	No	No	Convergent only	Yes
Limit	No	No	No	No	No	Yes
Derivative	No	No	No	No	No	Yes
Integral	No	No	No	No	No	Yes

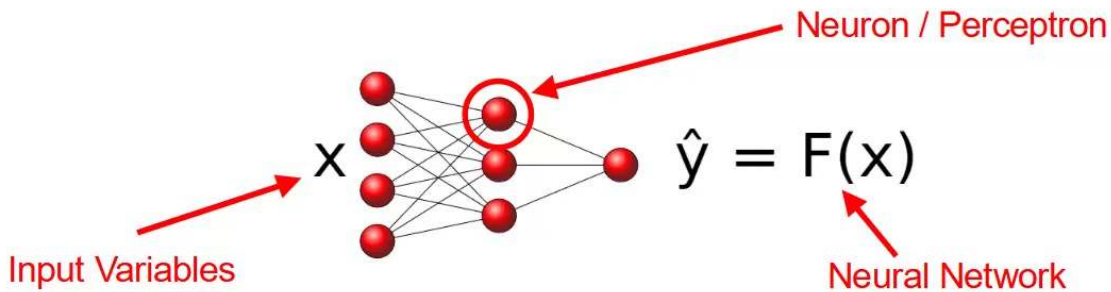
wikipedia.org

Interpretation

Interpreting an artificial neural network means formulating A mapping betweenan abstract mechanism or encoding into a domain that a human can understand.

- ✗ Mechanistic vs. Functional
- ✗ Local vs. Global
- ✗ Verify vs. Discover
- ✗ Low-Level vs. High-Level Features

Artificial Neural Network



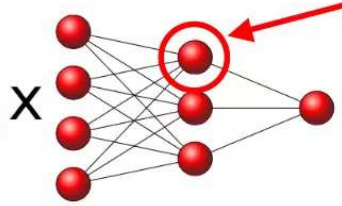
$$\hat{\mathbf{y}} = F(\mathbf{x}) = (L^{(l)} \circ \ldots \circ L^{(0)})(\mathbf{x})$$

$$F(\mathbf{x}) = \text{sigmoid}(f(\mathbf{x}))$$

Activation Function for Binary Classification

Latent Model

Artificial Neural Network



$$\hat{\mathbf{y}} = F(\mathbf{x}) = (L^{(l)} \circ \ldots \circ L^{(0)})(\mathbf{x})$$

Weight Matrix

Layers

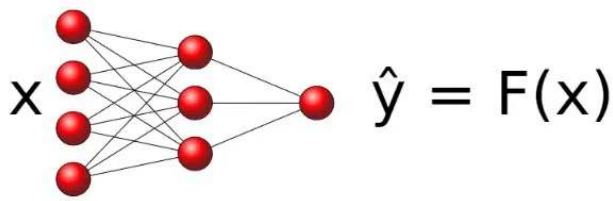
Bias Vector

$$\mathbf{a}^{(l)} = L^{l}(\mathbf{a}^{(l-1)}) = \sigma^{(l)}(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})$$

$$\mathbf{a}^{(0)} = \mathbf{x} \in \mathbb{R}^{n}$$

Nonlinear/Nonpolynomial Activation Function

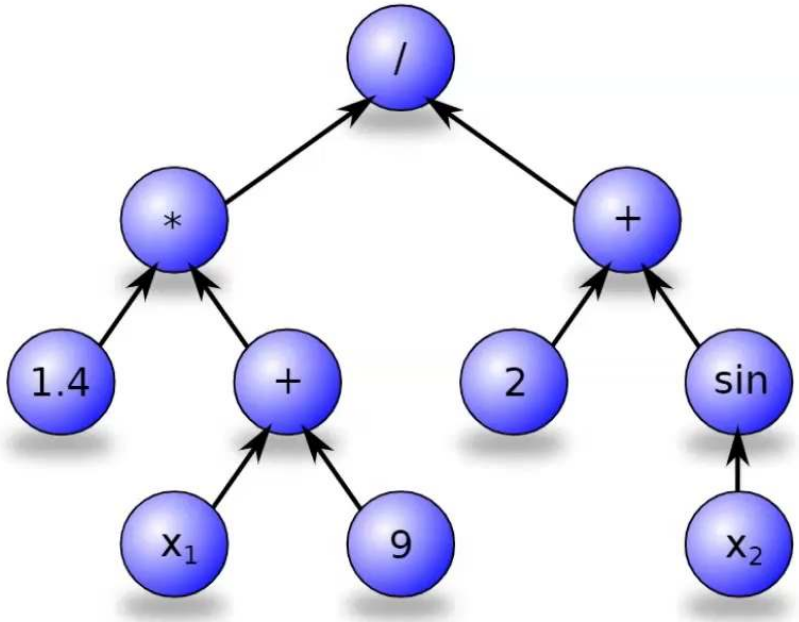
Artificial Neural Network



$$\hat{\mathbf{y}} = F(\mathbf{x}) = (L^{(l)} \circ \ldots \circ L^{(0)})(\mathbf{x})$$

Training: Use Gradient Descent/Backpropagation to adjust the weights and biases of the neural network to minimize an objective function $\mathcal{L}(F(X), Y)$ on a data set (X, Y) to approximate $F(X) \approx Y$ on the data manifold $X \subset D \subset \mathbb{R}^n$

Symbolic Search

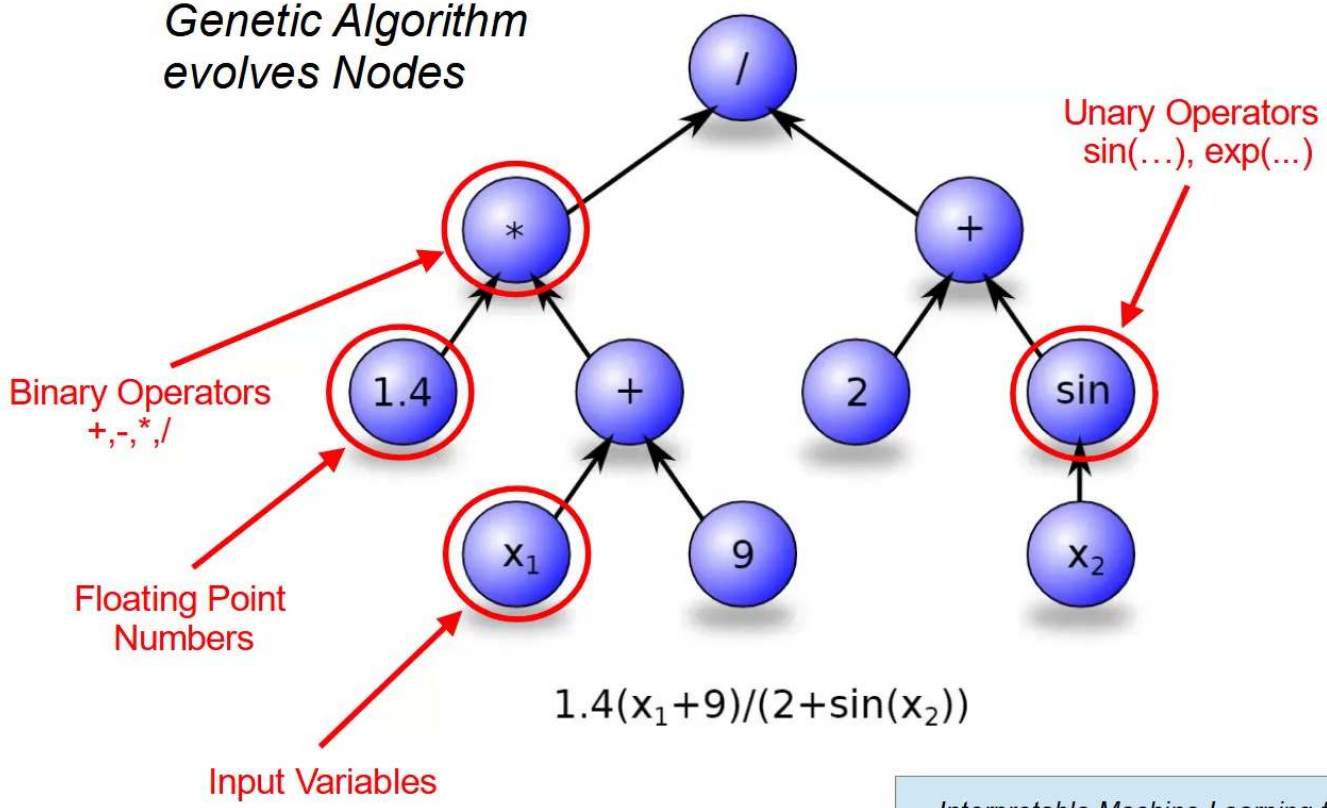


$$1.4(x_1+9)/(2+\sin(x_2))$$

Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl
Cranmer, arXiv:2305.01582

Symbolic Search

Genetic Algorithm evolves Nodes

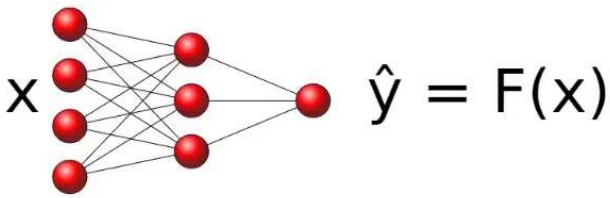


$1.4(x_1+9)/(2+\sin(x_2))$

Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl
Cranmer, arXiv:2305.01582

Idea

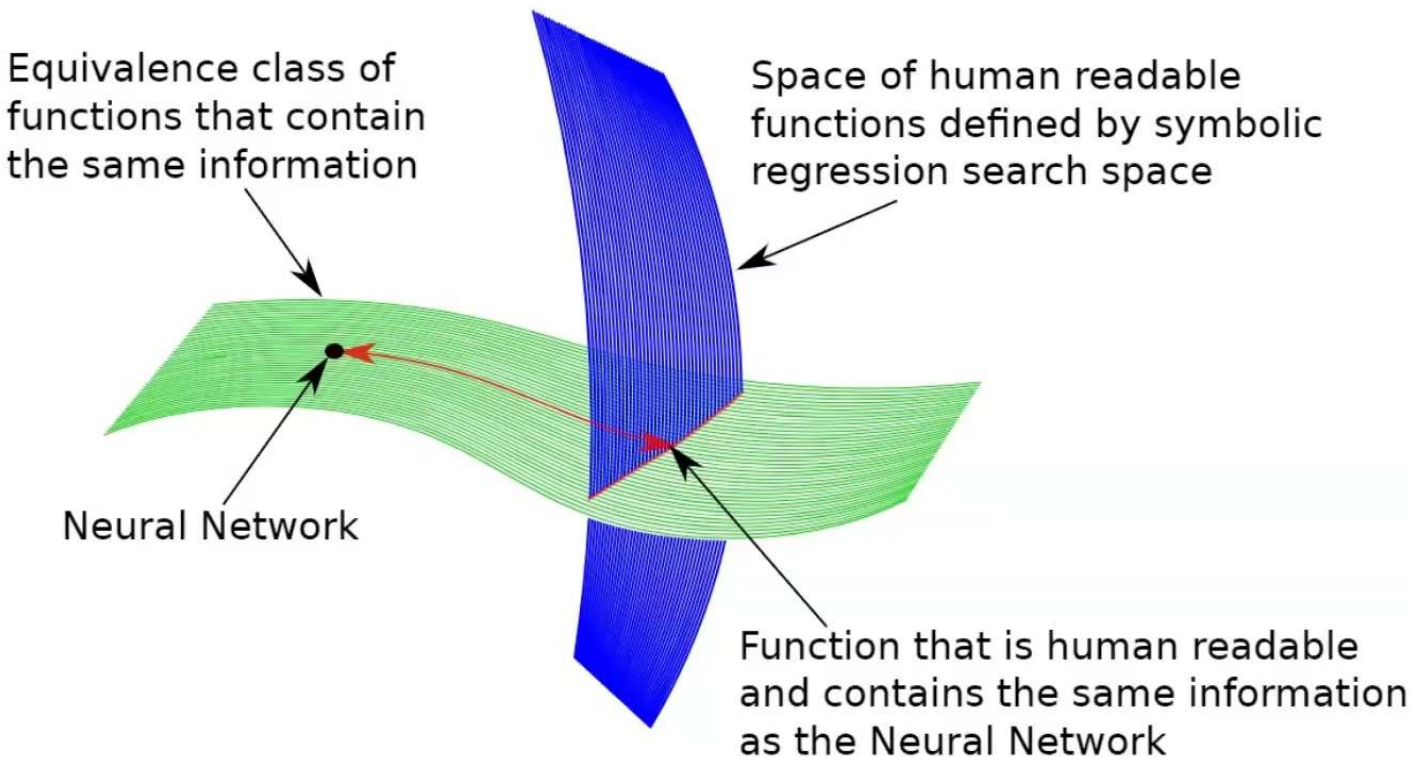
Artificial Neural Network Learns Concepts in a Highly Elusive and Convoluted Manner



If a Neural Network Learns a Concept that can be written in Human-Readable Form, is it possible to reveal the closed form concept while ignoring any uninterpretable transformations?

$$F(x_1, ..., x_n) = \underbrace{\text{sigmoid}}_{\text{activation function}} \left(\underbrace{\phi}_{\text{uninterpretable transformation}} \left(\underbrace{g(x_1, ..., x_n)}_{\text{closed form decision function}} \right) \right)$$

Idea


Equivalence class of functions that contain the same information
Space of human readable functions defined by symbolic regression search space
Neural Network
Function that is human readable and contains the same information as the Neural Network

Equivalence Class

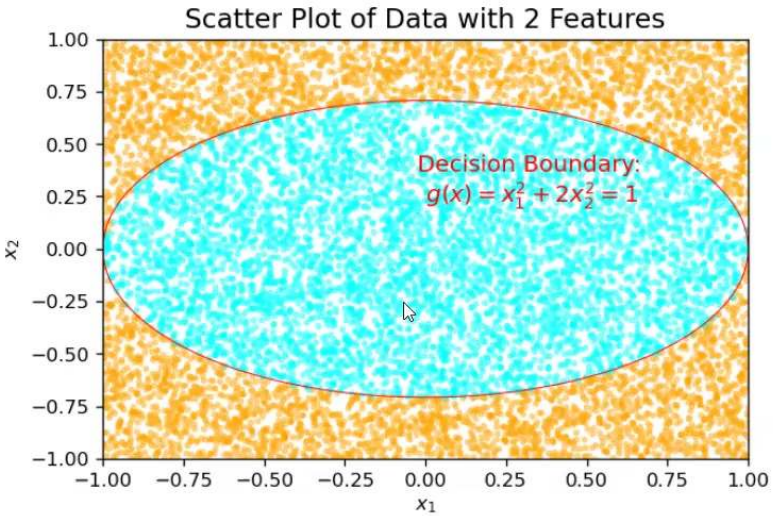
A function $f(\mathbf{x})$ contains the full information about a certain Quantity $g(\mathbf{x})$ if $g(\mathbf{x})$ can be faithfully reconstructed from $f(\mathbf{x})$. Conversely, if $f(\mathbf{x})$ only contains information from $g(\mathbf{x})$ it is possible to reconstruct $f(\mathbf{x})$ from the knowledge of $g(\mathbf{x})$. In mathematical terms that means that there exists an invertible function such that $f(\mathbf{x}) = \phi(g(\mathbf{x}))$

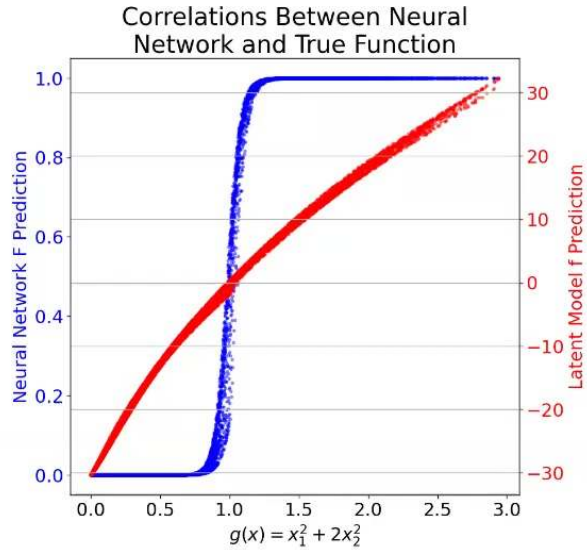
$$\begin{tikzcd} f(\mathbf{x}) \arrow[r, bend left, "\phi"] & g(\mathbf{x}) \arrow[l, bend left, "\phi^{-1}"] \end{tikzcd}$$

- *In the case of a one-dimensional input this is trivial*

Equivalence Class

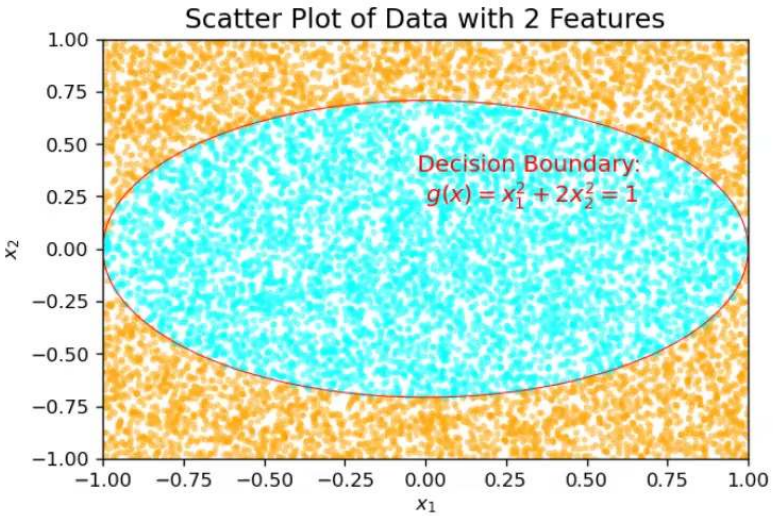
Example: Train a Neural Network Classifier to solve a binary Classification problem with decision boundary $x_1^2 + 2x_2^2 = 1$

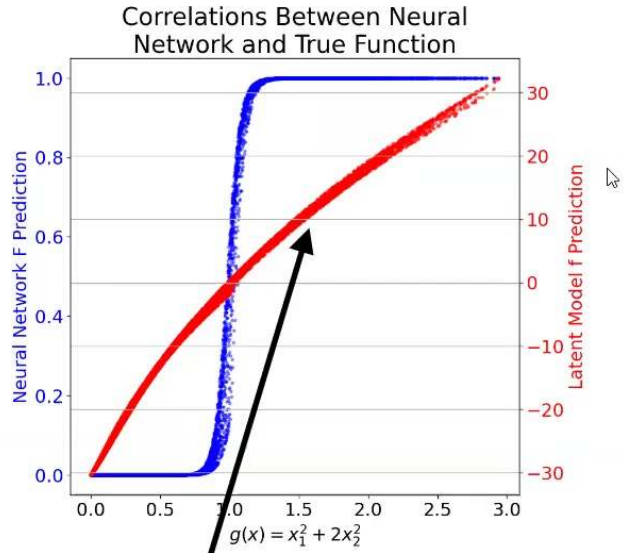




Equivalence Class

Example: Train a Neural Network Classifier to solve a binary Classification problem with decision boundary $x_1^2 + 2x_2^2 = 1$





There is ϕ !

Equivalence Class

Define the equivalence class of functions $f(\mathbf{x})$ that contain the same information as $g(\mathbf{x})$

$$\tilde{H}_g = \left\{ f(\mathbf{x}) \in C^1(D \subset \mathbb{R}^n, \mathbb{R}) \mid \exists \text{ invertible } \phi \in C^1(\mathbb{R}, \mathbb{R}) \ : \ f(\mathbf{x}) = \phi(g(\mathbf{x})) \right\}$$

This could be seen as the latent model

This could be seen as a learned concept

Let us calculate the gradient with respect to the input:

$$\nabla f(\mathbf{x}) = \phi'(g(\mathbf{x}))\nabla g(\mathbf{x})$$

We observe that the Gradients are parallel

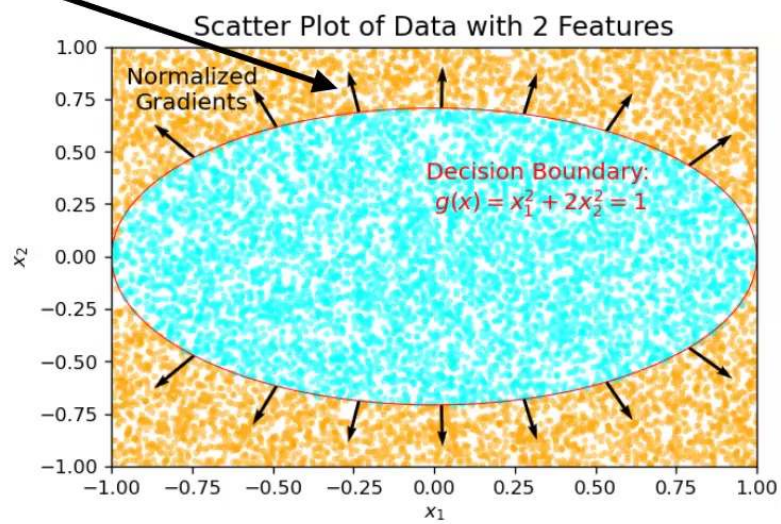
$$H_g = \left\{ f(\mathbf{x}) \in C^1(D \subset \mathbb{R}^n, \mathbb{R}) \mid \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} = \frac{\nabla g(\mathbf{x})}{\|\nabla g(\mathbf{x})\|} \vee \nabla f(\mathbf{x}) = \nabla g(\mathbf{x}) = 0, \ \forall \mathbf{x} \in D \right\}$$

It is possible to prove that both equivalence classes are the same $H_g = \tilde{H}_g$ under reasonable assumptions

Interpretation Algorithm

In order to find a function that contains the same information as the neural network and is human readable, we calculate the intersection between H_g and the symbolic search space.

We fit a symbolic regression model to replicate the normalized gradients of a latent neural network



Interpretation Algorithm

Algorithm 1: Train Neural Network for Binary Classification

Data: Labelled data set $D = (X_{train}, Y_{train})$

Input: Neural Network Hyperparameters

Initialize neural network classifier with sigmoid activation at output F

Train F on D

$f \leftarrow$ remove sigmoid activation of output neuron of F

Output: Trained model F, latent model f

Interpretation Algorithm

Algorithm 2: Obtain Gradients of Latent Model

Data: Labelled data set $D = (X_{train}, Y_{train})$
Unlabelled data set $D_u = (X_u)$ ▷ Artificial data not used for training

Input: Trained model F and latent model f
Selection threshold δ

$\tilde{X} \leftarrow (X_{train}, X_u)$ ▷ Add artificial unlabelled data

$\tilde{X} \leftarrow \tilde{X}$ **where** $F(\tilde{X}) \in [\delta, 1-\delta]$ ▷ Select data close to decision boundary

$G_f \leftarrow [\nabla f(\mathbf{x})$ for $\mathbf{x}$ in $\tilde{X}]$ ▷ Gradients of latent model wrt. input

$G_f \leftarrow [$**if** $\nabla f(\mathbf{x}) \neq 0 : \nabla f(\mathbf{x}) / \|\nabla f(\mathbf{x})\|$

else $\nabla f(\mathbf{x})$ for $\nabla f(\mathbf{x})$ in $G_f]$ ▷ Normalize Gradients

Output: $(\tilde{X}, G_f)$

Interpretation Algorithm

Algorithm 3: Symbolic Search

Data: Gradient data set $(\tilde{X}, G_f)$

Input: Symbolic Regression Hyperparameters
Set of unary and binary operations.

initialize symbolic regression model T

evolve T **with** (

$G_T \leftarrow [\nabla T(\mathbf{x})$ for $\mathbf{x}$ in $\tilde{X}]$ ▷ Gradients of symbolic model

$G_T \leftarrow [$**if** $\nabla T(\mathbf{x}) \neq 0 : \nabla T(\mathbf{x}) / \|\nabla T(\mathbf{x})\|$

else $\nabla T(\mathbf{x})$ for $\nabla T(\mathbf{x})$ in $G_T]$) ▷ Normalize Gradients

to minimize $\mathrm{MSE}(G_f, G_T)$

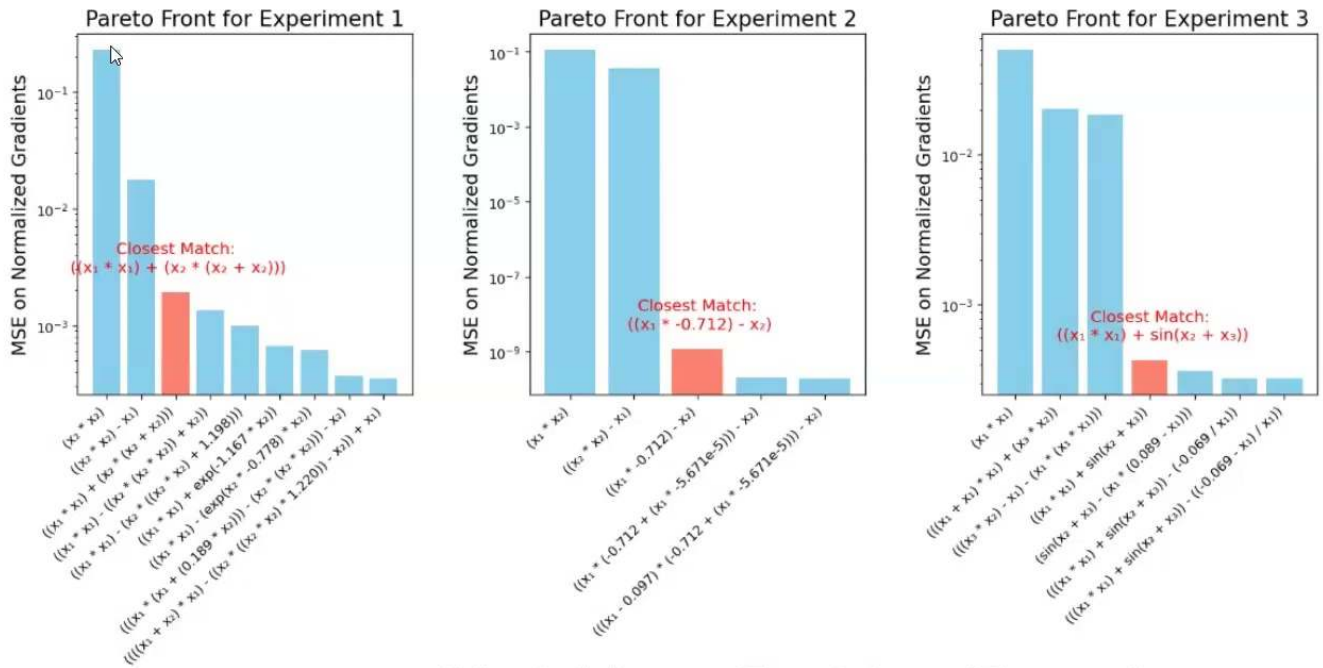
Output: Symbolic regression model T

Interpretation

We train a Neural Network Classifier to solve 7 binary classification problems and interpret the learned concepts by applying our interpretation framework

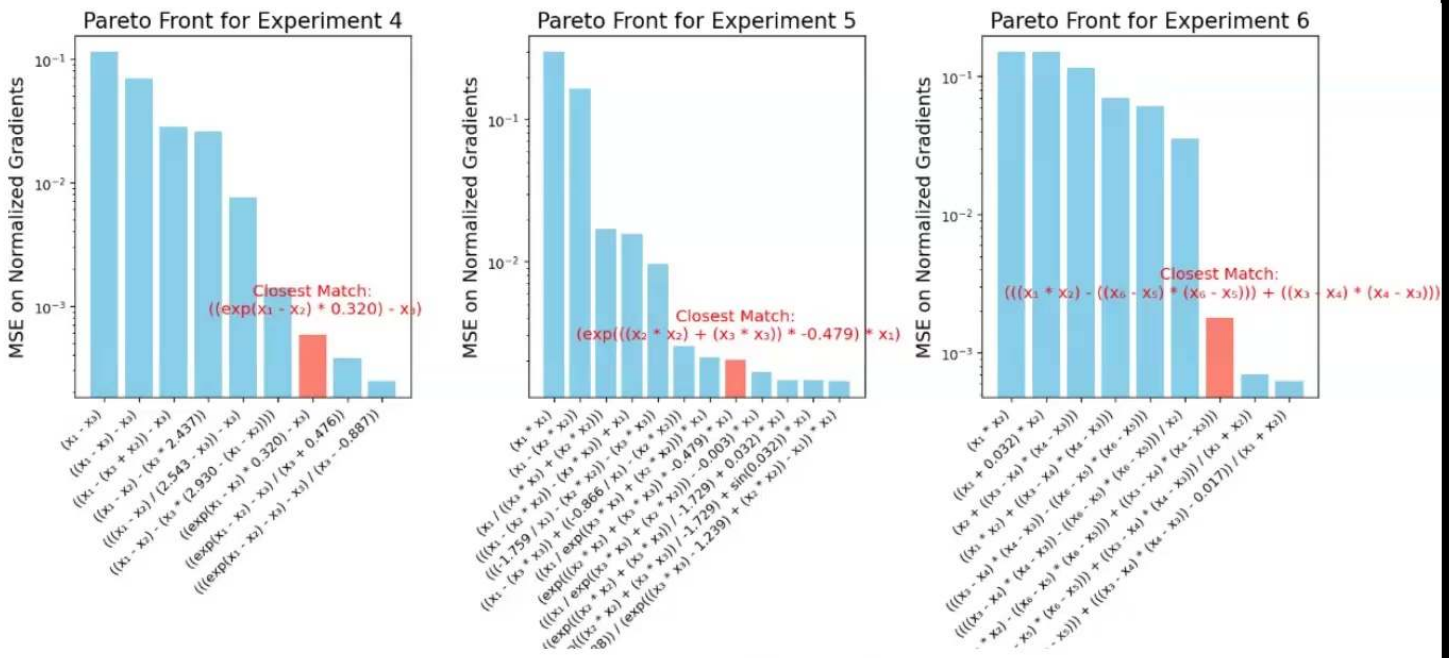
	Variables	Decision Formula
Experiment 1	2	$x_1^2 + 2x_2^2 > 1$
Experiment 2	2	$x_1^2 + 3x_1x_2 + 2x_2^2 > 5$
Experiment 3	3	$x_1^2 + \sin(x_2 + x_3) > 1$
Experiment 4	3	$\exp(x_1 - x_2) - \pi x_3 > 1$
Experiment 5	3	$x_1 \exp\left(-\frac{1}{2}(x_2^2 + x_3^2)\right) > \frac{1}{5}$
Experiment 6	6	$\frac{x_1x_2}{(x_3-x_4)^2+(x_5-x_6)^2} > 1$
Experiment 7	4	$x_1x_2 > 1 \wedge x_3x_4 > 1$ vs $x_1x_2 < 1 \wedge x_3x_4 < 1$

Interpretation



	Variables	Decision Formula
Experiment 1	2	$x_1^2 + 2x_2^2 > 1$
Experiment 2	2	$x_1^2 + 3x_1x_2 + 2x_2^2 > 5$
Experiment 3	3	$x_1^2 + \sin(x_2 + x_3) > 1$

Interpretation

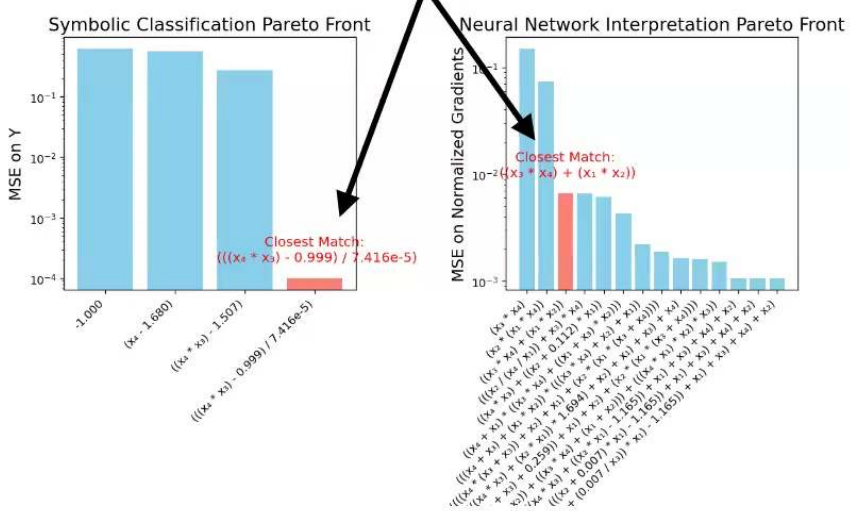


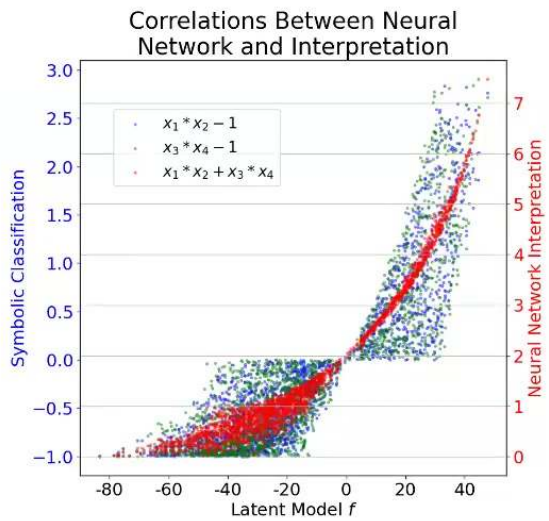
	Variables	Decision Formula
Experiment 4	3	$\exp(x_1 - x_2) - \pi x_3 > 1$
Experiment 5	3	$x_1 \exp\left(-\frac{1}{2}(x_2^2 + x_3^2)\right) > \frac{1}{5}$
Experiment 6	6	$\frac{x_1 x_2}{(x_3 - x_4)^2 + (x_5 - x_6)^2} > 1$

Interpretation

We compare our interpretation framework to direct symbolic classification on a problem with multiple concepts that can be learned for a successful classification.

Neural Network lears combined concept while Symbolic Classification learns only one





	Variables	Decision Formula
Experiment 7	4	$x_1x_2 > 1 \wedge x_3x_4 > 1$ vs $x_1x_2 < 1 \wedge x_3x_4 < 1$

Conclusion

- ✗ It is possible to find closed-form interpretations of Neural Network Classifiers
- ✗ Same goes for neurons in hidden layers or latent spaces
- ✗ If you want to interpret your neural network, let me know

Closed-Form Interpretation of Neural Network Classifiers with Symbolic Regression Gradients, Wetzel, arXiv:2401.04978