Linearly approximating Neural Networks to formally verify its properties

Final goal

Represent a neural network as a logical formula in order to formally verify and explain certain properties

One way to try and desmistify the *black box* nature of neural networks

Understanding the problem

- Lukasiewicz Logic
- McNaughton's Theorem
- Modulo Satisfability
- Neural Networks
- Segmented Regression

Lukasiewicz Logic

Extension of Classical Propositional Logic

Tries to capture the concept of "half truths"

Variables can be evaluated to any number in [0, 1]

$$v(\alpha \to \beta) \underset{def}{=} min(1, 1 - v(\alpha) + v(\beta))$$

$$v(\neg\alpha) \underset{def}{=} 1 - v(\alpha)$$

Understanding the problem

- Lukasiewicz Logic
- McNaughton's Theorem
- Modulo Satisfability
- Neural Networks
- Segmented Regression

McNaughton's Theorem

Definition 9. *A McNaughton function is a function $f : [0,1]^n \to [0,1]$ such that*

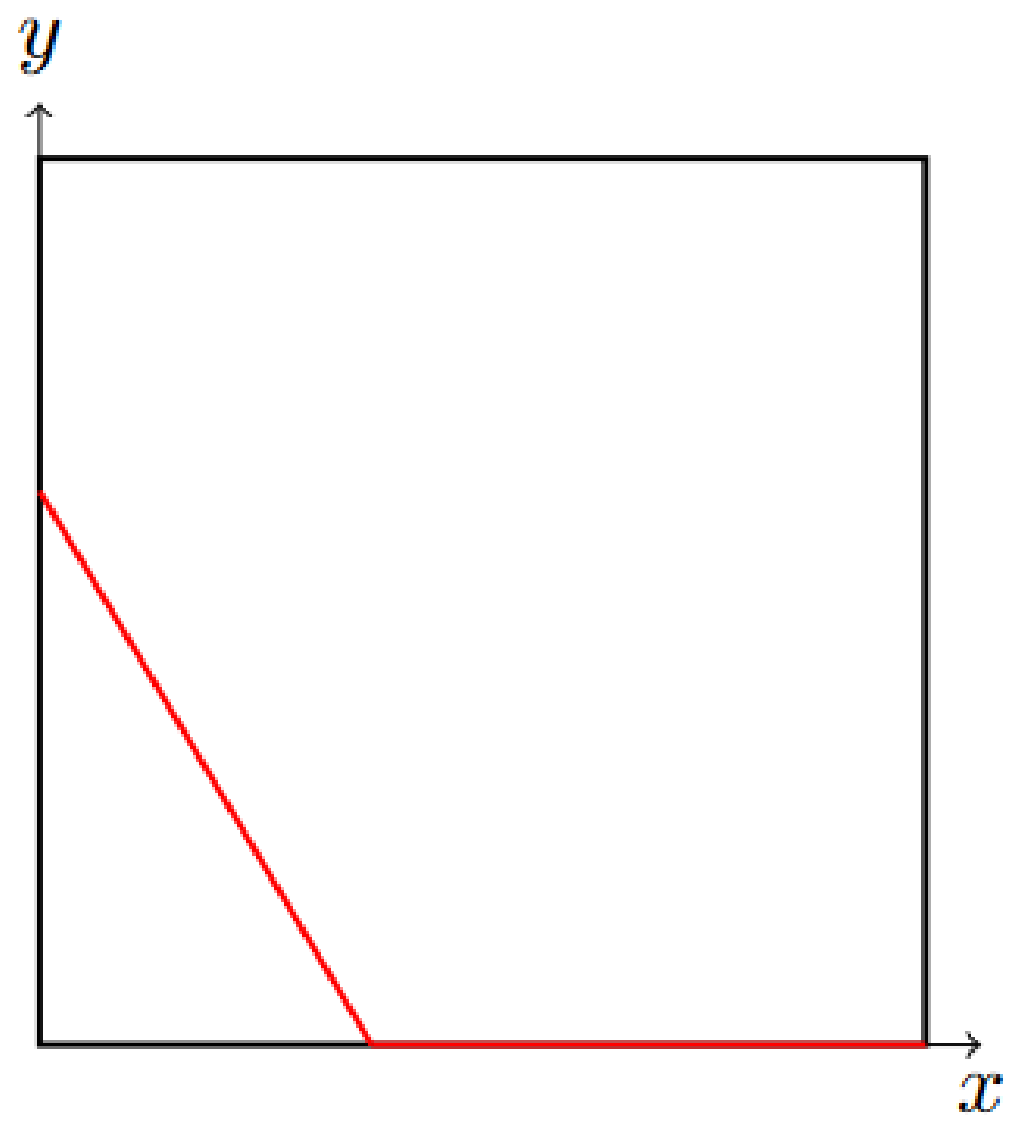
$$f(x_1, x_2, ..., x_n) = min(max(0, b + m_1x_1 + m_2x_2 + ... + m_nx_n), 1)$$

where b and m_i are integers.

Theorem 1. *For any function $f : [0,1]^n \to [0,1]$*

$$f(x_1, x_2, ..., x_n) = min(max(0, b + m_1x_1 + m_2x_2 + ... + m_nx_n), 1),$$

there is a logical formula $S(p_1, p_2, , ..., p_n)$ in Lukasiewicz Logic, such that $v(S) = f$, where p_i are sentential variables such that $v(p_i) = x_i$.



Known Result

Rational McNaughton functions can approximate any continuous function

One problem

Rational McNaughton functions can approximate any continuous function

Rational McNaughton functions can approximate any continuous function

But McNaughton's theorem speaks only of functions with integer coefficients

Understanding the problem

- Lukasiewicz Logic
- McNaughton's Theorem
- Modulo Satisfability
- Neural Networks
- Segmented Regression

Modulo Satisfability

Modulo satisfability is way to represent rational McNaughton functions with Lukasiewicz Logic

$$\langle \varphi, \Phi \rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}} \right\} \right\rangle$$

Define a valuation function for the formula φ as a function that satisfies all formulas in a set Φ

$$\langle \varphi, \Phi \rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg(d-1)Z_{\frac{1}{d}} \right\} \right\rangle$$

$$v(\Phi) = v(Z_{\frac{1}{d}} \leftrightarrow \neg(d-1)Z_{\frac{1}{d}})$$

$$\boxed{\begin{gathered} v(\alpha \leftrightarrow \beta) = 1 - |v(\alpha) - v(\beta)| \\ v(\neg\alpha) = 1 - v(\alpha) \\ v(\Phi) = 1 \end{gathered}}$$

$$\langle \varphi, \Phi \rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg(d-1)Z_{\frac{1}{d}} \right\} \right\rangle$$

$$v(\Phi) = v(Z_{\frac{1}{d}} \leftrightarrow \neg(d-1)Z_{\frac{1}{d}})$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - (d-1)v(\neg Z_{\frac{1}{d}})|$$

$$\boxed{\begin{array}{c} v(\alpha \leftrightarrow \beta) = 1 - |v(\alpha) - v(\beta)| \\ v(\neg\alpha) = 1 - v(\alpha) \\ v(\Phi) = 1 \end{array}}$$

$$\langle \varphi, \Phi \rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}} \right\} \right\rangle$$

$$v(\Phi) = v(Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}})$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - (d-1) v(\neg Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - [1 - (d-1) v(Z_{\frac{1}{d}})]|$$

$$\boxed{\begin{gathered} v(\alpha \leftrightarrow \beta) = 1 - |v(\alpha) - v(\beta)| \\ v(\neg\alpha) = 1 - v(\alpha) \\ v(\Phi) = 1 \end{gathered}}$$

$$\langle \varphi, \Phi \rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}} \right\} \right\rangle$$

$$v(\Phi) = v(Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}})$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - (d-1) v(\neg Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - [1 - (d-1) v(Z_{\frac{1}{d}})]|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + (d-1) v(Z_{\frac{1}{d}})|$$

$$\boxed{\begin{gathered} v(\alpha \leftrightarrow \beta) = 1 - |v(\alpha) - v(\beta)| \\ v(\neg\alpha) = 1 - v(\alpha) \\ v(\Phi) = 1 \end{gathered}}$$

$$\langle \varphi, \Phi \rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg (d-1) Z_{\frac{1}{d}} \right\} \right\rangle$$

$$v(\Phi) = v(Z_{\frac{1}{d}} \leftrightarrow \neg (d-1) Z_{\frac{1}{d}})$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - (d-1) v(\neg Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - [1 - (d-1) v(Z_{\frac{1}{d}})]|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + (d-1) v(Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + d v(Z_{\frac{1}{d}}) - v(Z_{\frac{1}{d}})|$$

$$\boxed{\begin{gathered} v(\alpha \leftrightarrow \beta) = 1 - |v(\alpha) - v(\beta)| \\ v(\neg \alpha) = 1 - v(\alpha) \\ v(\Phi) = 1 \end{gathered}}$$

$$\langle\varphi,\Phi\rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg(d-1)Z_{\frac{1}{d}} \right\} \right\rangle$$

$$v(\Phi) = v(Z_{\frac{1}{d}} \leftrightarrow \neg(d-1)Z_{\frac{1}{d}})$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - (d-1)v(\neg Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - [1-(d-1)v(Z_{\frac{1}{d}})]|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + (d-1)v(Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + dv(Z_{\frac{1}{d}}) - v(Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |-1 + dv(Z_{\frac{1}{d}})| \Longrightarrow -1 + dv(Z_{\frac{1}{d}}) = 0$$

$$\boxed{\begin{gathered} v(\alpha \leftrightarrow \beta) = 1 - |v(\alpha) - v(\beta)| \\ v(\neg\alpha) = 1 - v(\alpha) \\ v(\Phi) = 1 \end{gathered}}$$

$$\langle \varphi, \Phi \rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}} \right\} \right\rangle$$

$$v(\Phi) = v(Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}})$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - (d-1) v(\neg Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - [1 - (d-1) v(Z_{\frac{1}{d}})]|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + (d-1) v(Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + d v(Z_{\frac{1}{d}}) - v(Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |-1 + d v(Z_{\frac{1}{d}})| \Longrightarrow -1 + d v(Z_{\frac{1}{d}}) = 0$$

$$-1 + d v(Z_{\frac{1}{d}}) = 0 \Longleftrightarrow d v(Z_{\frac{1}{d}}) = 1$$

$$\boxed{\begin{gathered} v(\alpha \leftrightarrow \beta) = 1 - |v(\alpha) - v(\beta)| \\ v(\neg \alpha) = 1 - v(\alpha) \\ v(\Phi) = 1 \end{gathered}}$$

$$\langle \varphi, \Phi \rangle = \left\langle Z_{\frac{1}{d}}, \left\{ Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}} \right\} \right\rangle$$

$$v(\Phi) = v(Z_{\frac{1}{d}} \leftrightarrow \neg(d-1) Z_{\frac{1}{d}})$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - (d-1) v(\neg Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - [1 - (d-1) v(Z_{\frac{1}{d}})]|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + (d-1) v(Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |v(Z_{\frac{1}{d}}) - 1 + dv(Z_{\frac{1}{d}}) - v(Z_{\frac{1}{d}})|$$

$$v(\Phi) = 1 - |-1 + dv(Z_{\frac{1}{d}})| \Longrightarrow -1 + dv(Z_{\frac{1}{d}}) = 0$$

$$-1 + dv(Z_{\frac{1}{d}}) = 0 \Longleftrightarrow dv(Z_{\frac{1}{d}}) = 1$$

$$v(Z_{\frac{1}{d}}) = \frac{1}{d}$$

$$v(\alpha \leftrightarrow \beta) = 1 - |v(\alpha) - v(\beta)|$$

$$v(\neg \alpha) = 1 - v(\alpha)$$

$$v(\Phi) = 1$$

Understanding the problem

Lukasiewicz Logic

McNaughton's Theorem

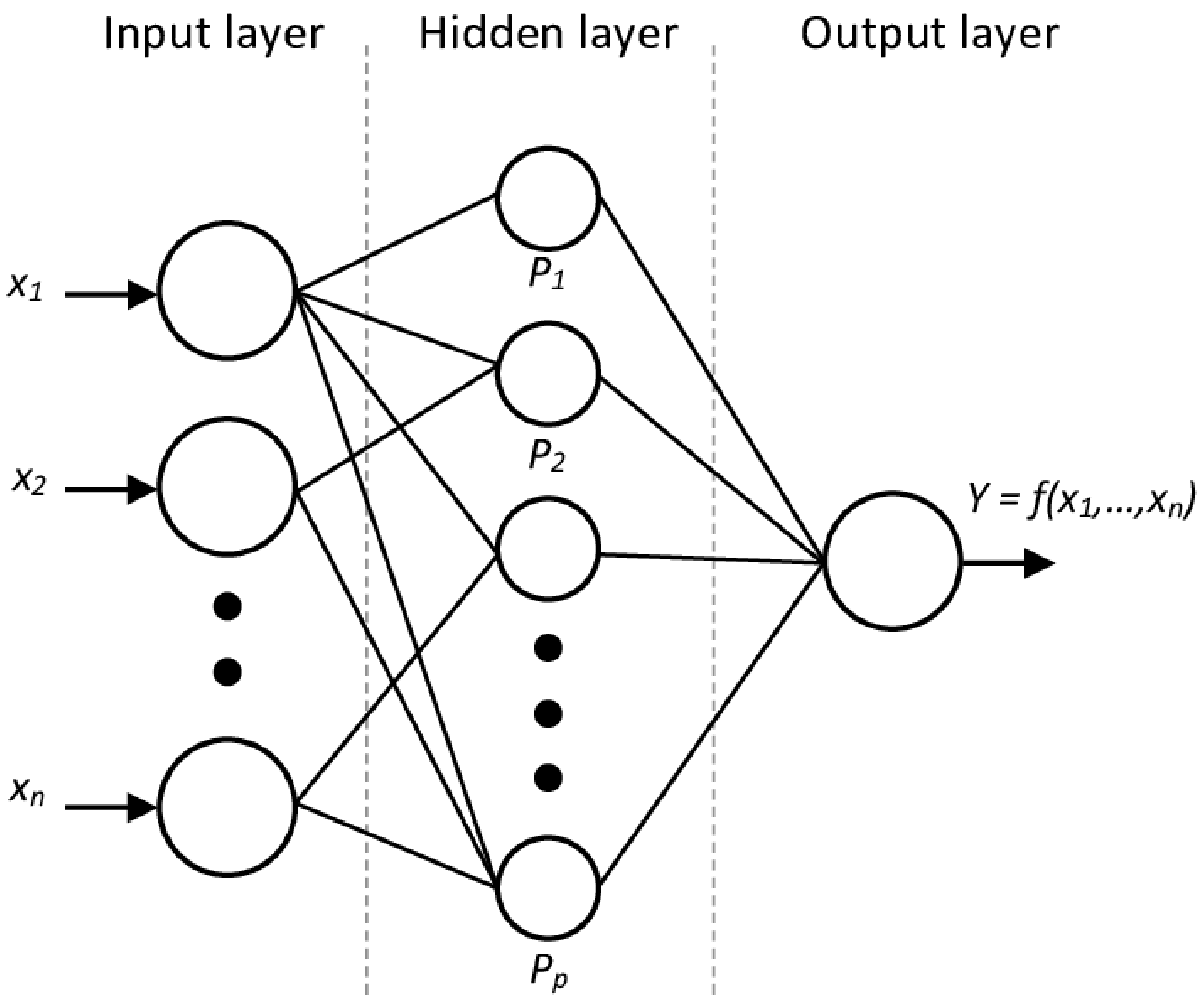
Modulo Satisfability

Neural Networks

Segmented Regression

Neural Networks

Can approximate any continuous function by generating a continuous function

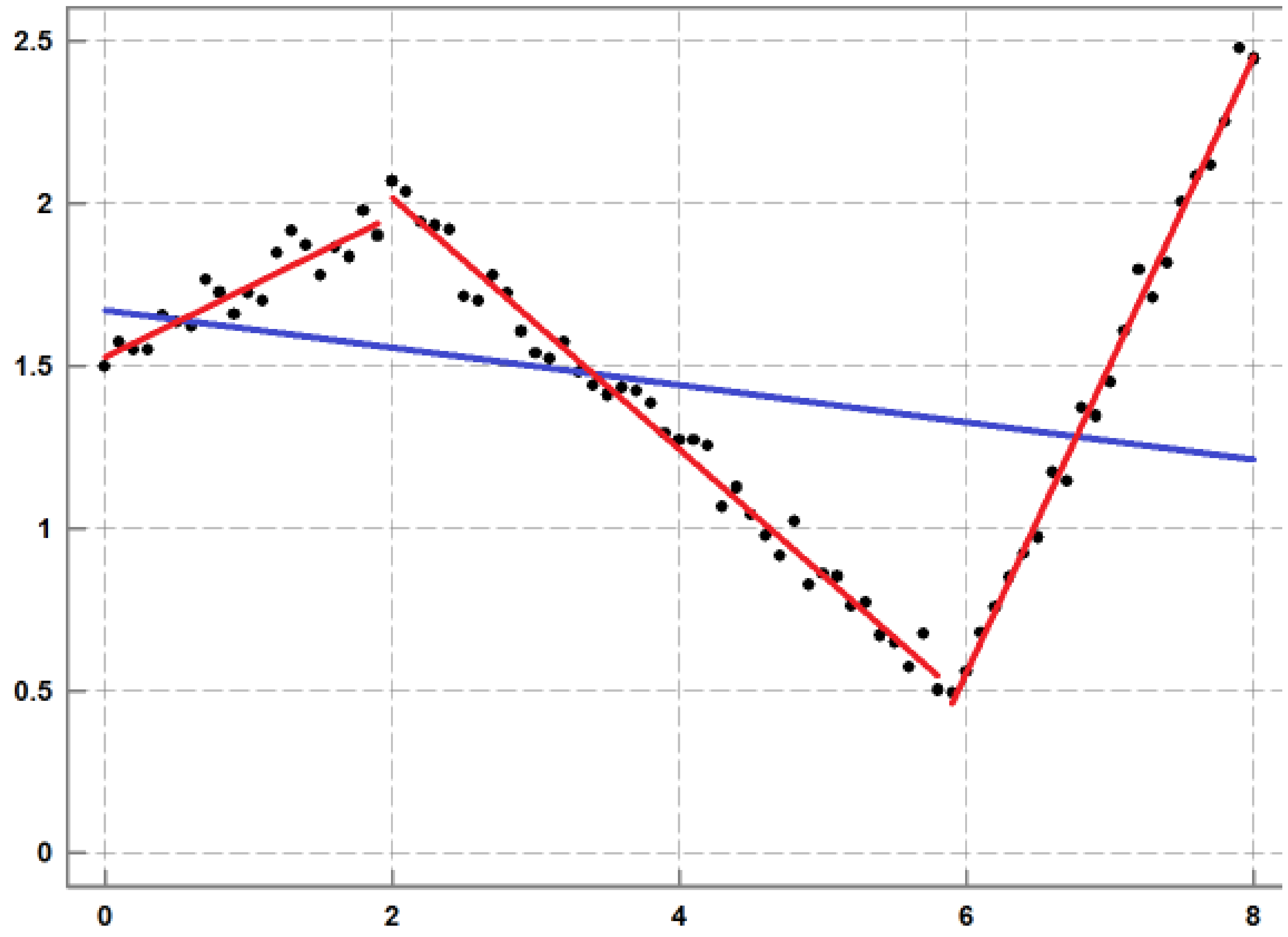


Understanding the problem

- Lukasiewicz Logic
- McNaughton's Theorem
- Modulo Satisfability
- Neural Networks
- Segmented Regression

Segmented Regression

Use an algorithm based on Dynammic programming to obtain the hyperplanes that best fit subsets of the input points



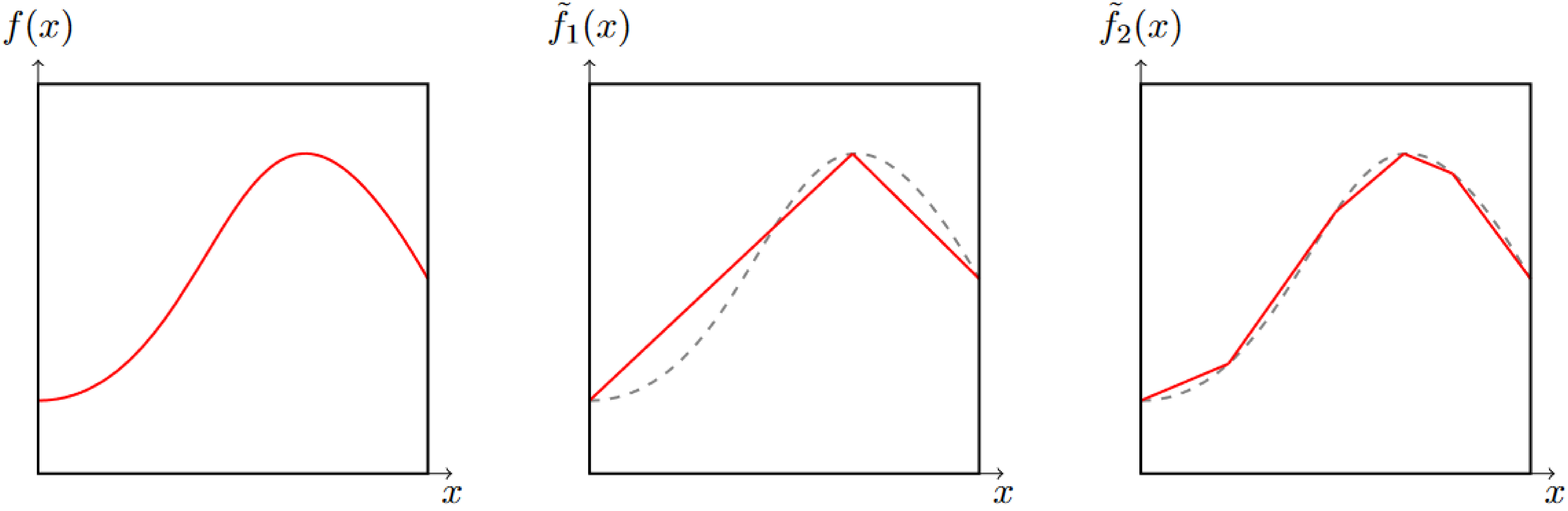


Figure 3.1: *Continuous one-variable function approximated by rational McNaughton functions*

Algorithm for segmented regression

Algorithm 2 Segmented Regression by Dynamic Programming

Input

X Data matrix of points sampled

y Data matrix of outputs of the f evaluated on X

C Cost for creating a segment

Output

Cost for creating the optimal piecewise linear function

$OPT[0] \leftarrow 0$

for $j \in \{1, ..., N\}$ **do**

  for $i \in \{1, ..., j\}$ **do**

    $err(i, j) \leftarrow$ least square error for indices in the interval $\{i, i+1, ..., j\}$

  end for

end for

for $j \in \{1, ..., N\}$ **do**

  $OPT[j] = min_{i<j}(err(i, j) + OPT[i-1] + C)$

end for

return $OPT[n]$

Understanding the problem

- Lukasiewicz Logic
- McNaughton's Theorem
- Modulo Satisfability
- Neural Networks
- Segmented Regression

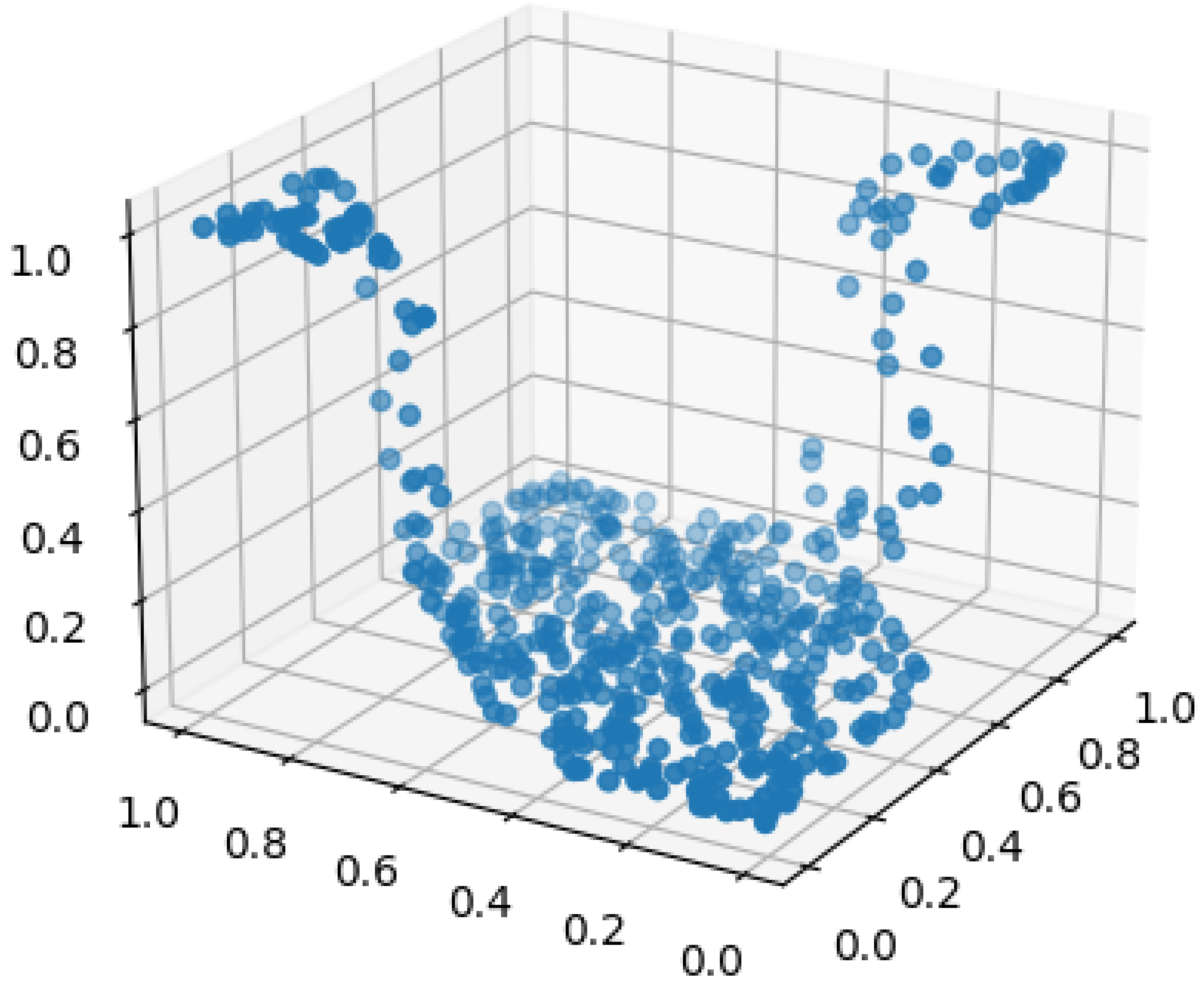
Putting it all together

Train a neural net

Approximate the network via segmented regression

Represent the function as a logical formula

Verify properties of the network

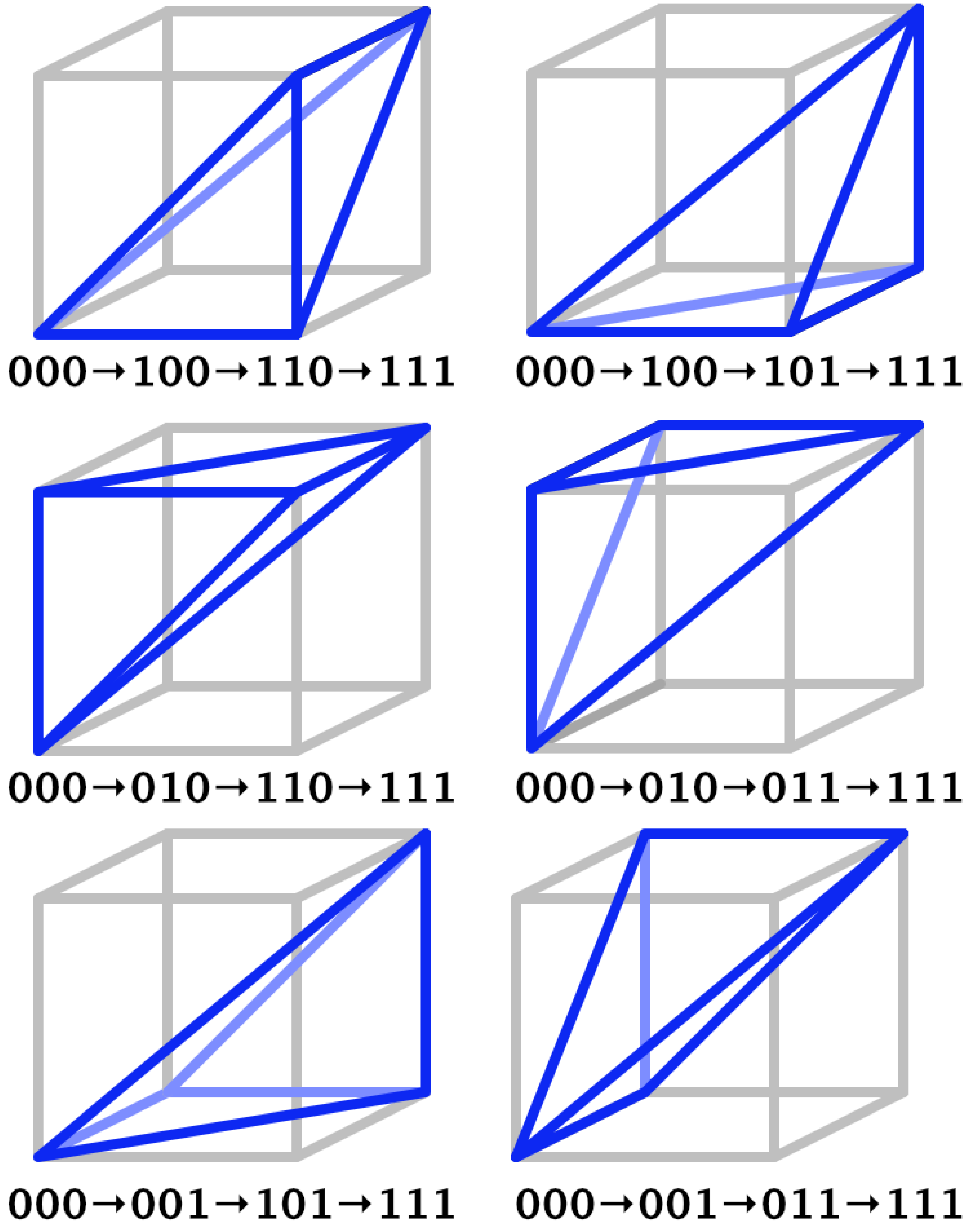


Experiments on Neural Networks

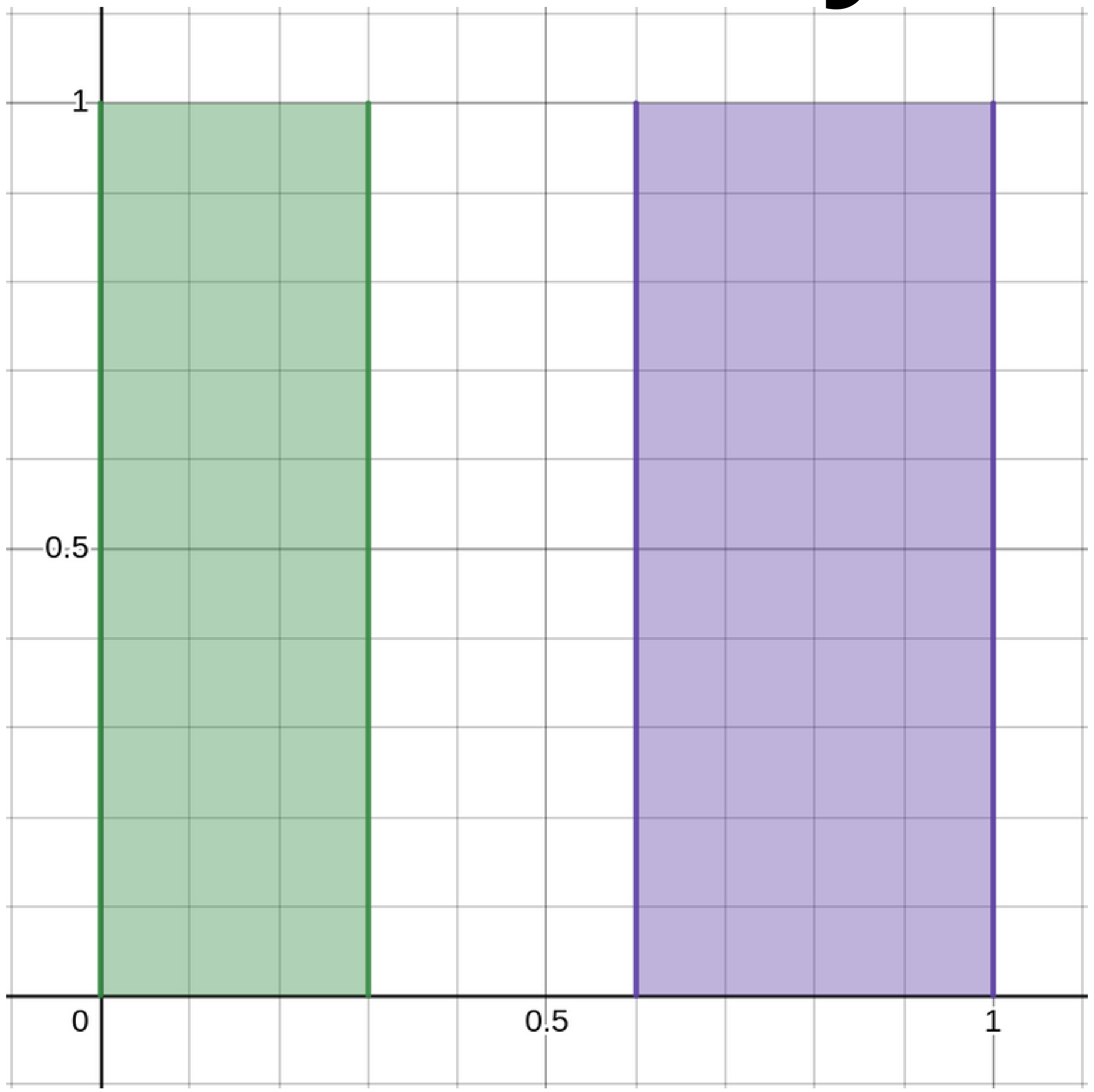
XOR Network

Digression I: Simplex Division

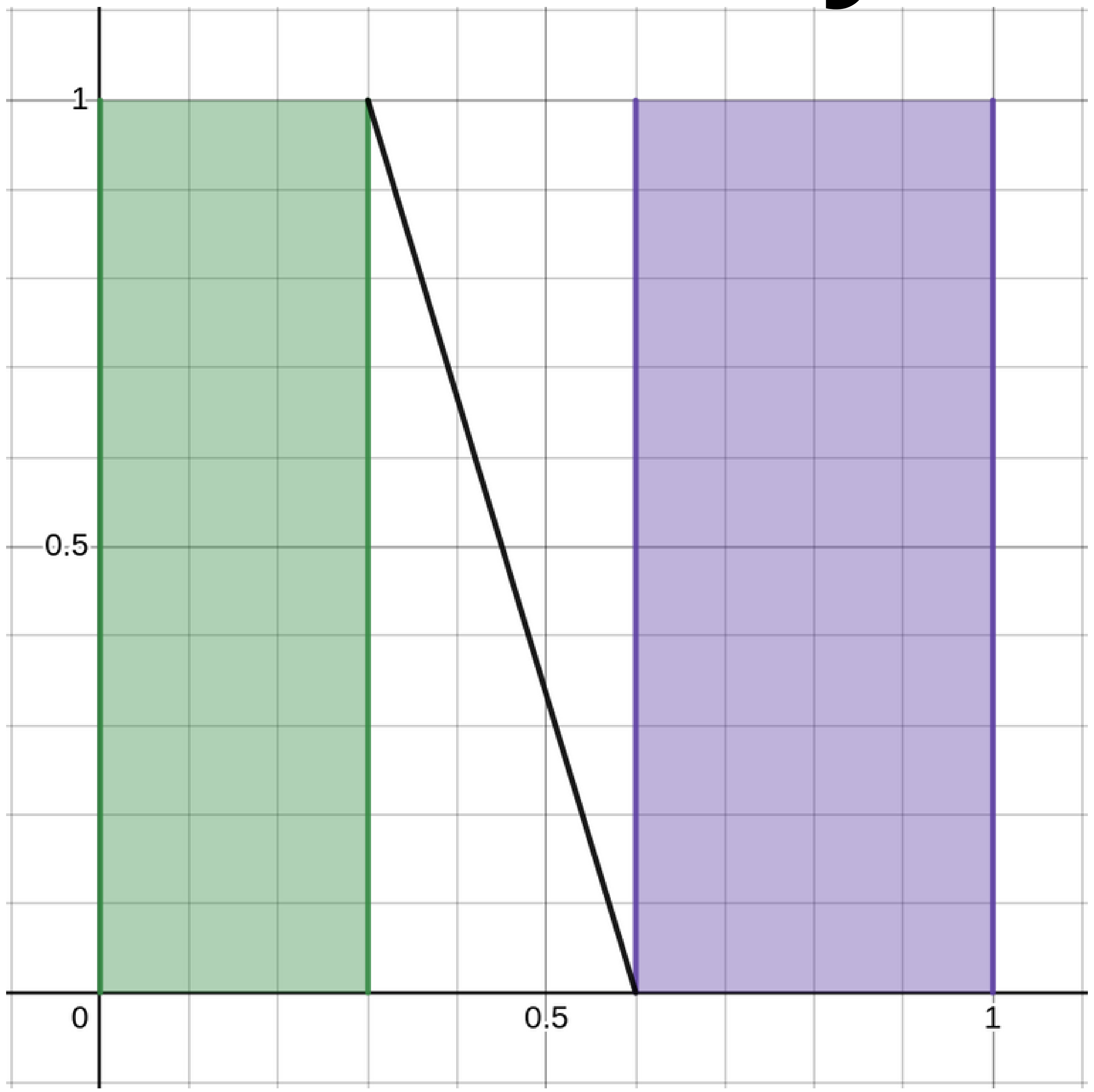
Partition the domain in mulitple pieces each one being a simplex in a way such that they cover the whole space



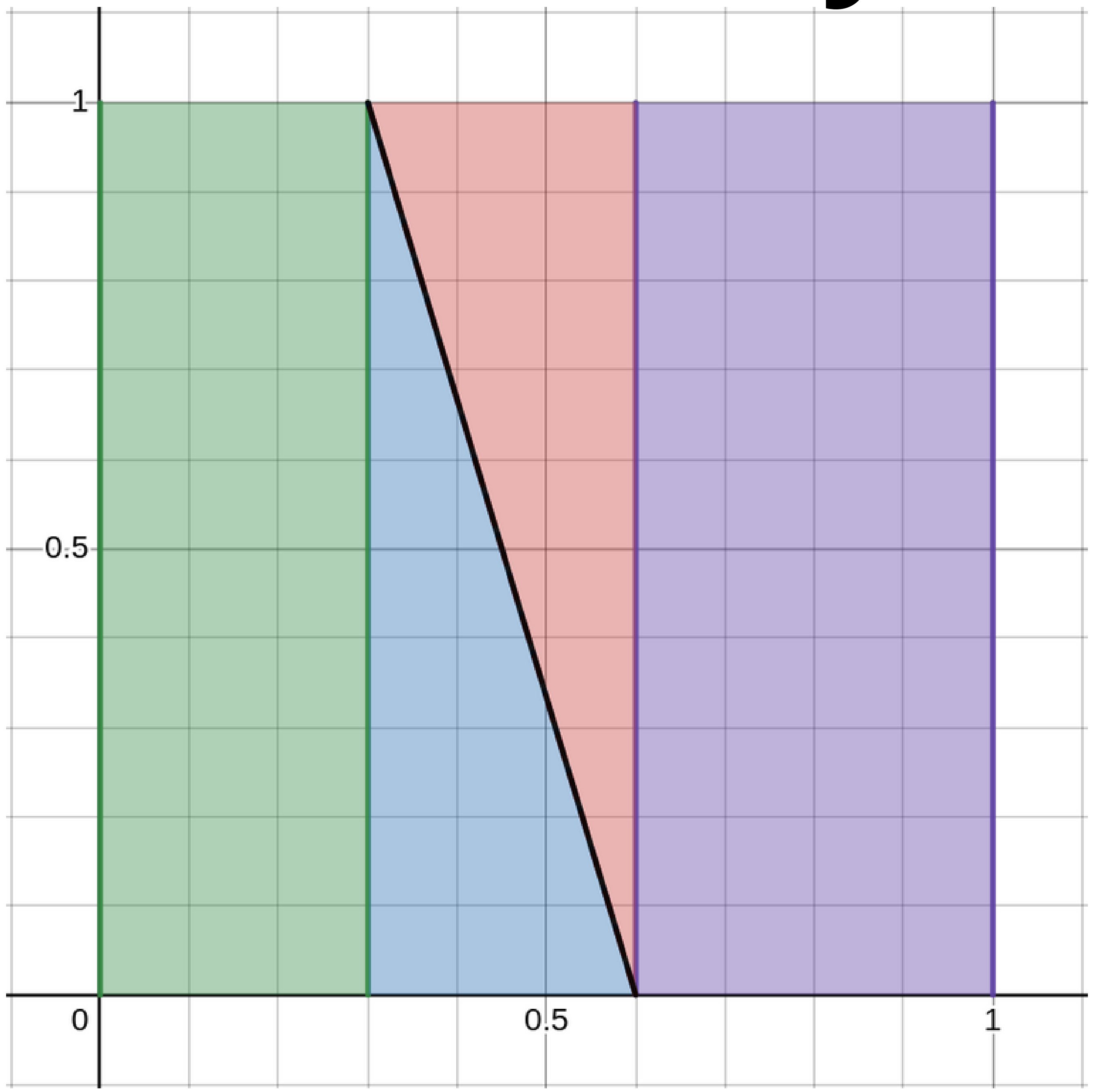
Simplex Division to force continuity



Simplex Division to force continuity



Simplex Division to force continuity



Digression II: Order matters

X1: sorting by the first coordinate x_1;
C: sorting based on values $c = x_1 - x_2$.

Accessibility

Verify if the network reaches a certain state/value

Accesibility	
Parameters	Result
$\pi = 0.1$	✓
$\pi = 0.2$	✓
$\pi = 0.3$	✓
$\pi = 0.4$	✓
$\pi = 0.5$	✓
$\pi = 0.6$	✓
$\pi = 0.7$	✓
$\pi = 0.8$	✓
$\pi = 0.9$	✓

Robustness

Verify how much the value of a neural network if affected by a "small" perturbation

Robustness	
Parameters	Result
$\pi = 0.75, \varepsilon = 0.01$	✓
$\pi = 0.75, \varepsilon = 0.1$	✓
$\pi = 0.75, \varepsilon = 0.2$	✓
$\pi = 0.75, \varepsilon = 0.25$	✓
$\pi = 0.75, \varepsilon = 0.3$	✗
$\pi = 0.75, \varepsilon = 0.35$	✗
$\pi = 0.75, \varepsilon = 0.4$	✗
$\pi = 0.75, \varepsilon = 0.5$	✗

Conclusions and future work

- I. Discontinuity
- II. Order matters
- III. Other ideas