A visual (and mathematical) proof that neural nets can compute any function

Jasper Seehofer, Hannes Laslo

Eberhard Karls Universität Tübingen

09.07.2020

Table of contents

[0]

1 Overview

2 Visual Approach
- Caveats
- Activation functions
- Construction
- The interactive way
- $\mathbb{R}^n \to \mathbb{R}^m$
- Non-monotonic activation functions
- Minimize error

3 Mathematical Approach
- Mathematical expression of a neural network
- What to prove ?
- Theorem I - Approximation of functions
- Lemma I - Sigmoidal functions are discriminatory

4 Back to Neural Networks

5 Appendix

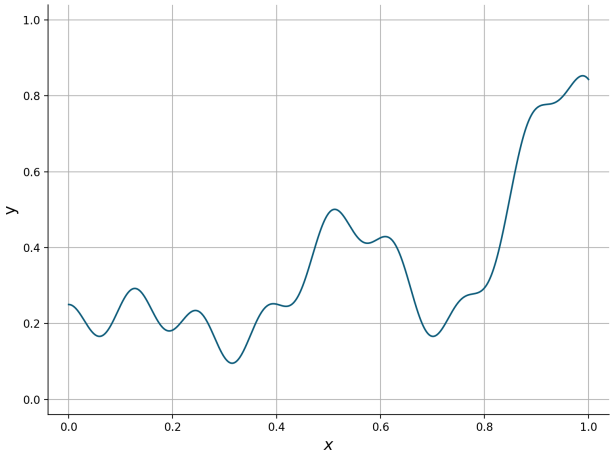
Presentations

Presentations

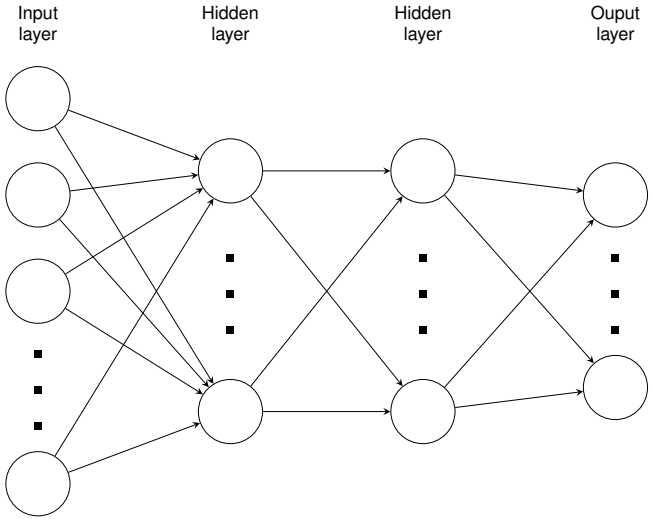
- Using neural nets to recognize handwritten digits
- How the backpropagation algorithm works
- Improving the way neural networks learn
- A visual (and mathematical) proof that neural nets can compute any function
- Why are deep neural networks hard to train ?
- Deep learning

What it is about

An arbitrary function :



The MNIST dataset :



Caveat 1

"Neural nets can compute any function"

For a given continuous function

$$f : \mathbb{R}^n \to \mathbb{R}, x \mapsto f(x) \tag{1}$$

with $n \in \mathbb{N}$ one can always find for a desired accuracy $\varepsilon > 0$ a neural network with the output

$$g : \mathbb{R}^n \to \mathbb{R}, x \mapsto g(x) \tag{2}$$

that satisfies

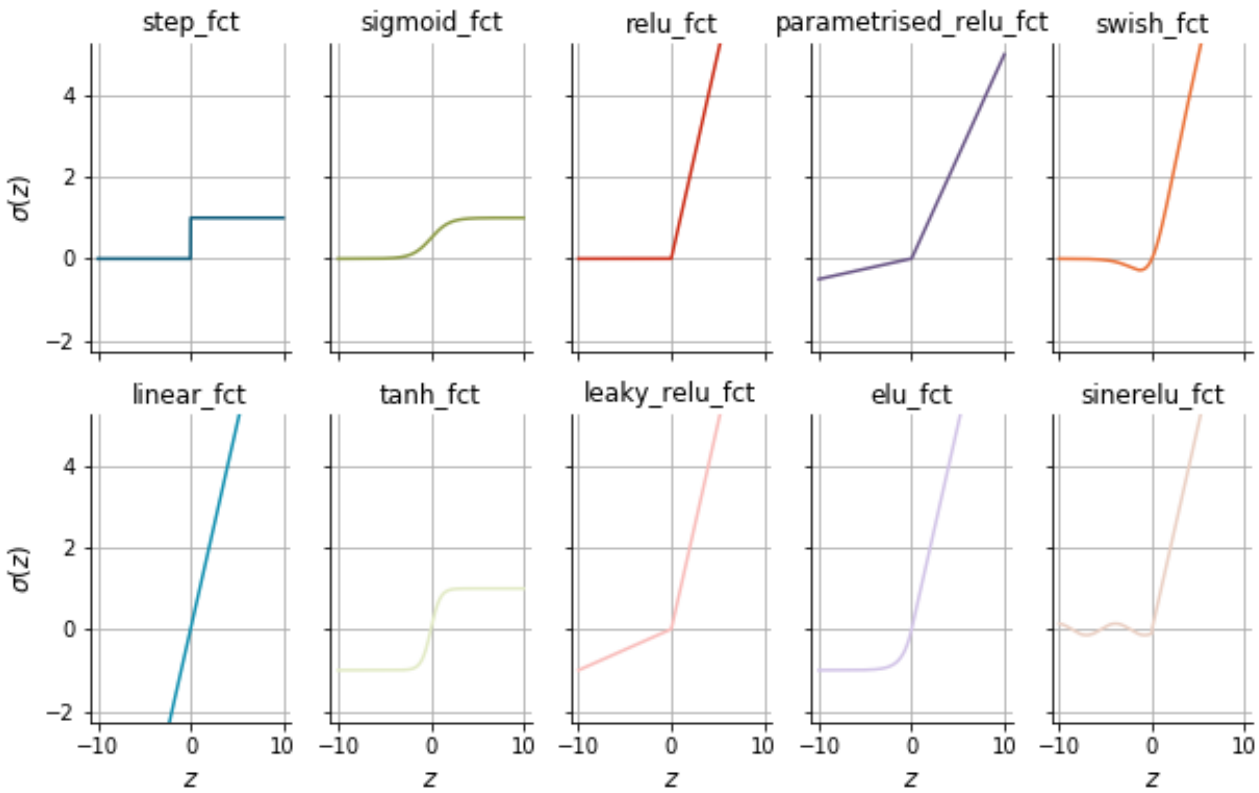
$$|f(x) - g(x)| < \varepsilon \tag{3}$$

for all $x \in \mathbb{R}^n$.

Caveat 2

A function f which can be approximated by a given continuous output of a neural network g upto a desired accuracy is an element of the class of continuous functions from $\mathbb{R}^n$ to $\mathbb{R}$.

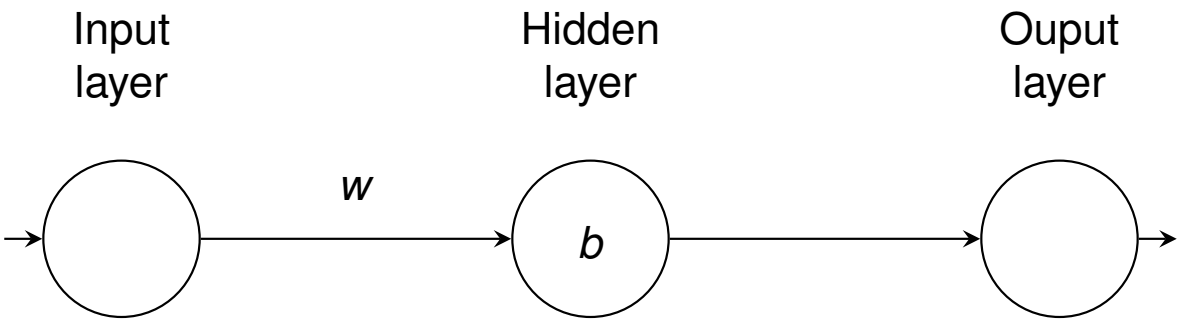
Activaltion functions



All of them are functions of the input vector $\boldsymbol{x}$ via

$$z = \boldsymbol{w} \cdot \boldsymbol{x} + b. \tag{4}$$

One hidden layer with one neuron



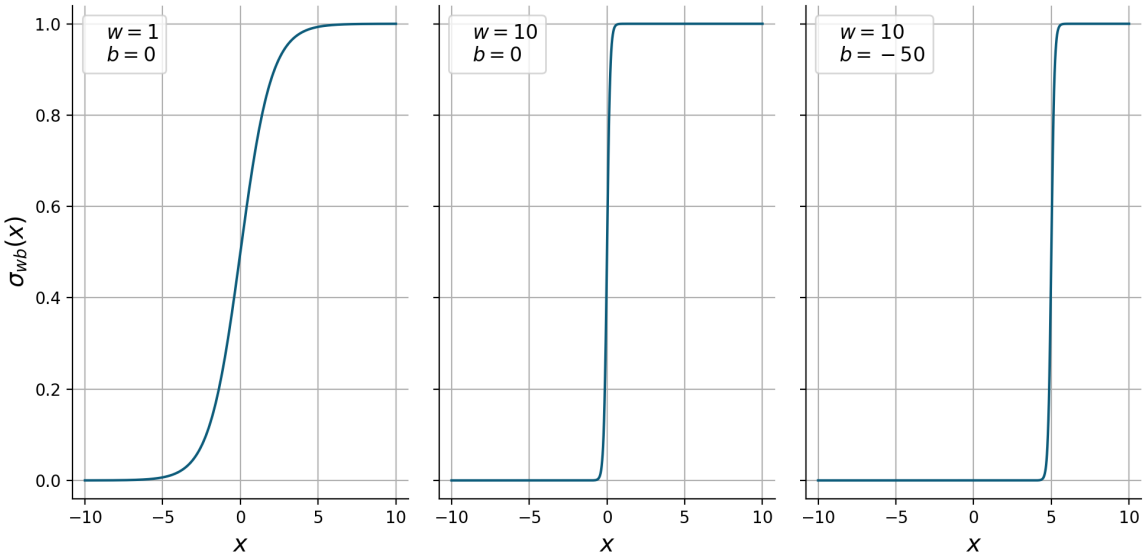
With the weight w, the bias b and the activation function

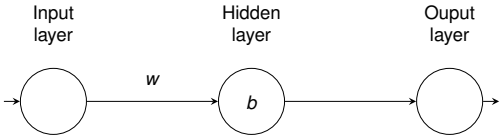
$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad \Rightarrow \qquad \sigma_{wb}(x) = \frac{1}{1+e^{-(wx+b)}} \tag{5}$$

Behaviour of the sigmoid function

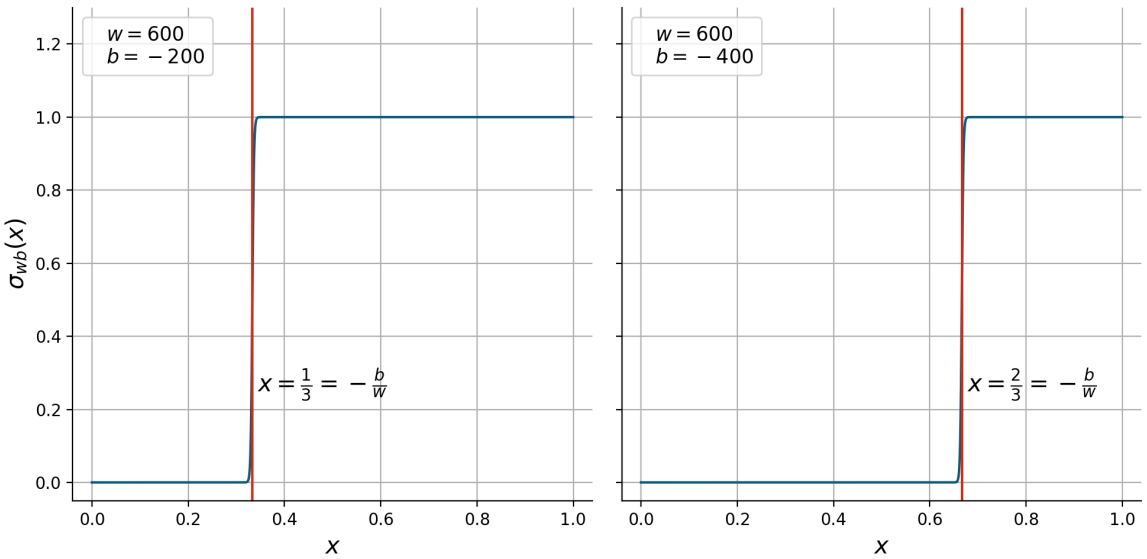
Sigmoid function :

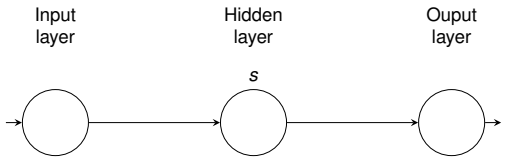
$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad \Rightarrow \qquad \sigma_{wb}(x) = \frac{1}{1+e^{-(wx+b)}} \tag{6}$$





Definition of s

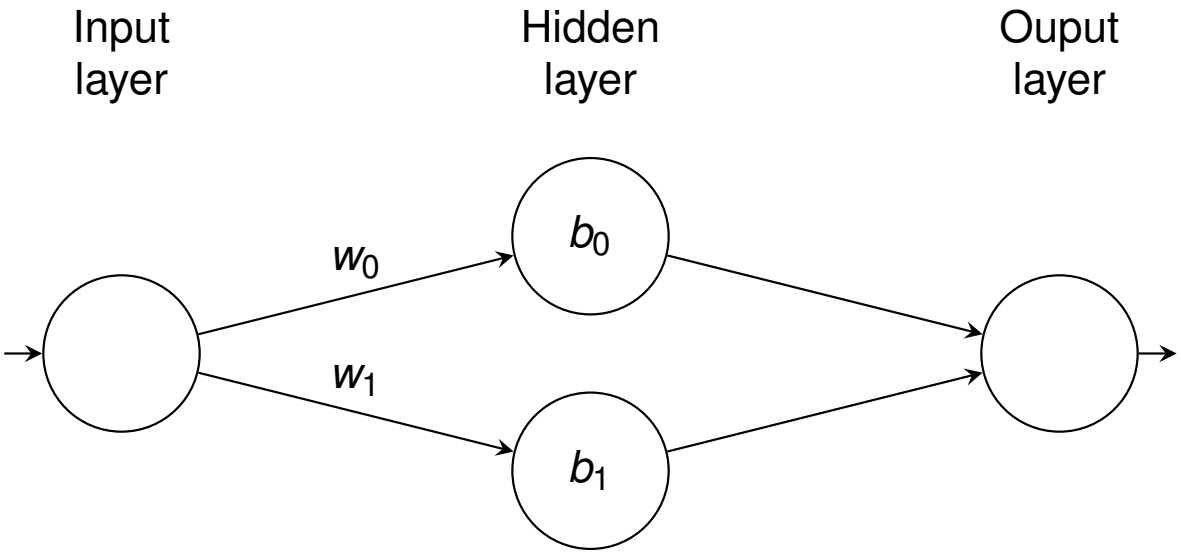




We define the parameter s as :

$$s := -\frac{b}{w} \tag{7}$$

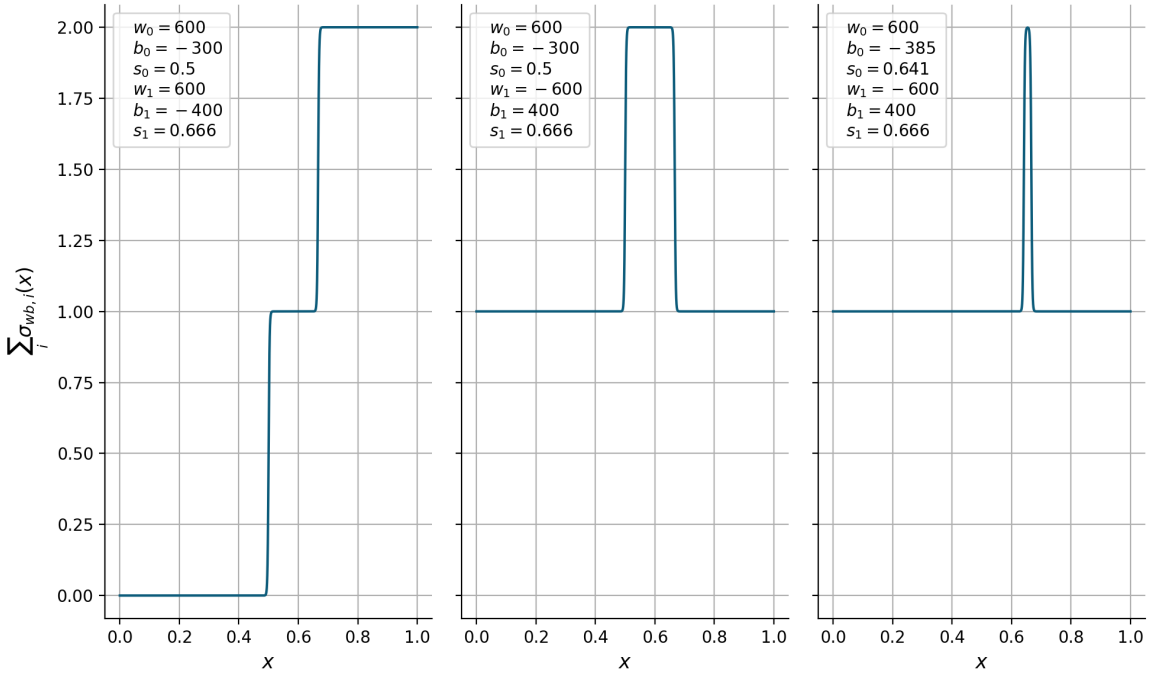
One hidden layer with two neurons

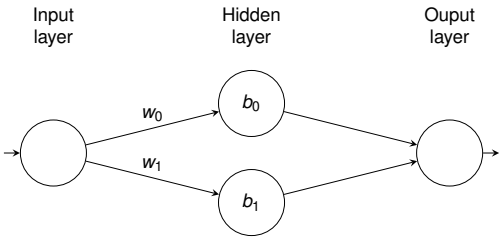


Sum of the two sigmoid functions

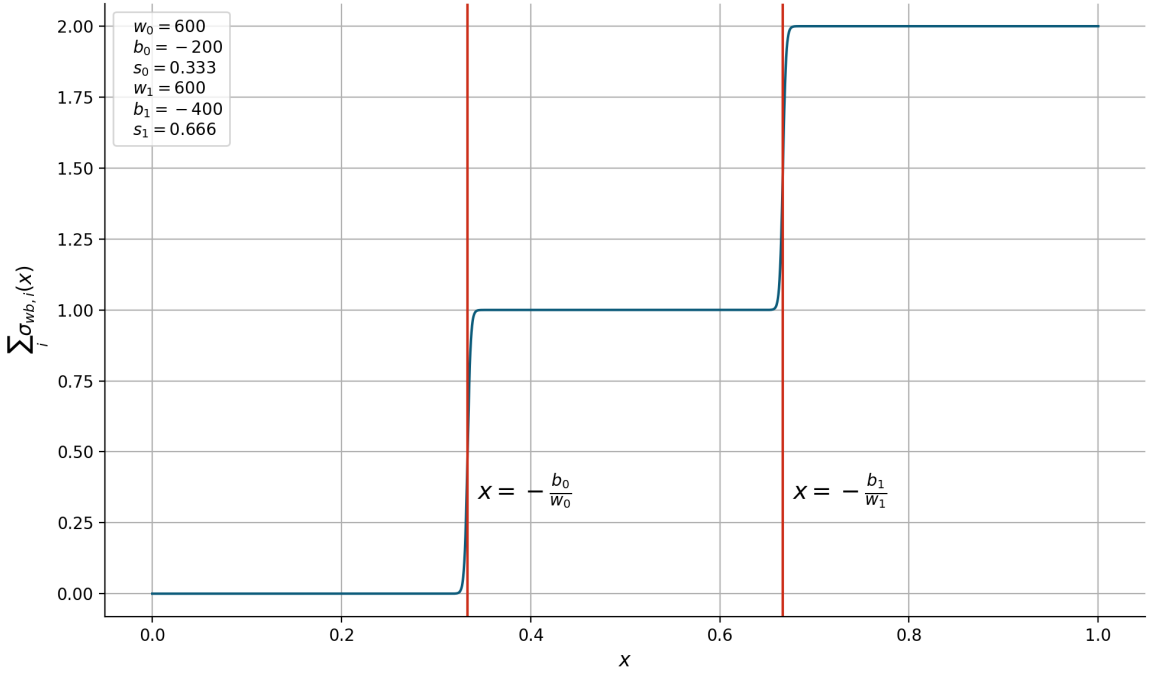
Sum of two sigmoid functions :

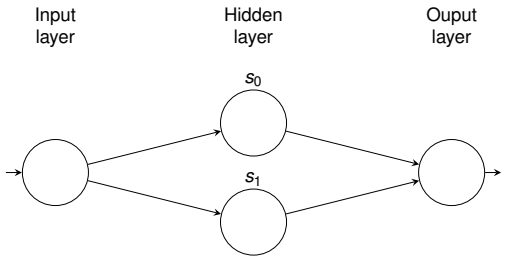
$$\sigma_{w_0 b_0}(x) + \sigma_{w_1 b_1}(x) = \frac{1}{1 + e^{-(w_0 x + b_0)}} + \frac{1}{1 + e^{-(w_1 x + b_1)}} \tag{8}$$



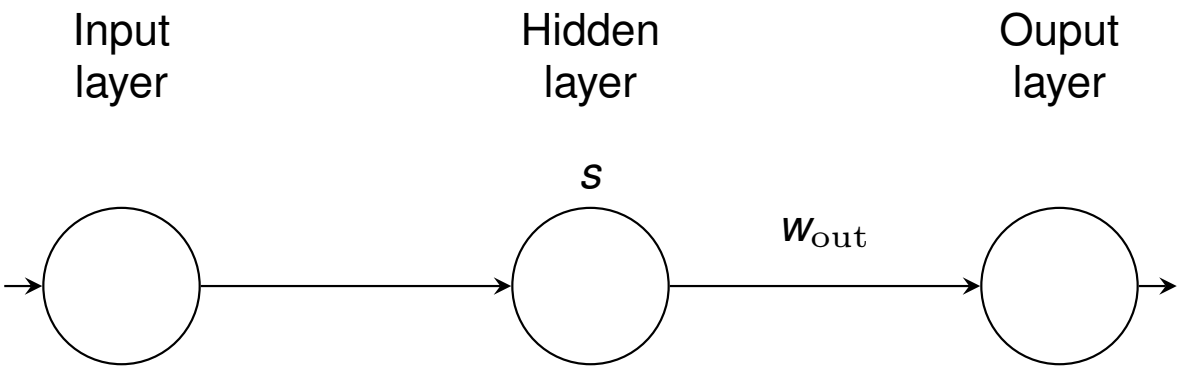


Identify s for one hidden layer with two neurons





One hidden layer with one neuron and weighted output



With the outputs or activations a of the neurons from the hidden layer we get for the activation of the neuron in the output layer

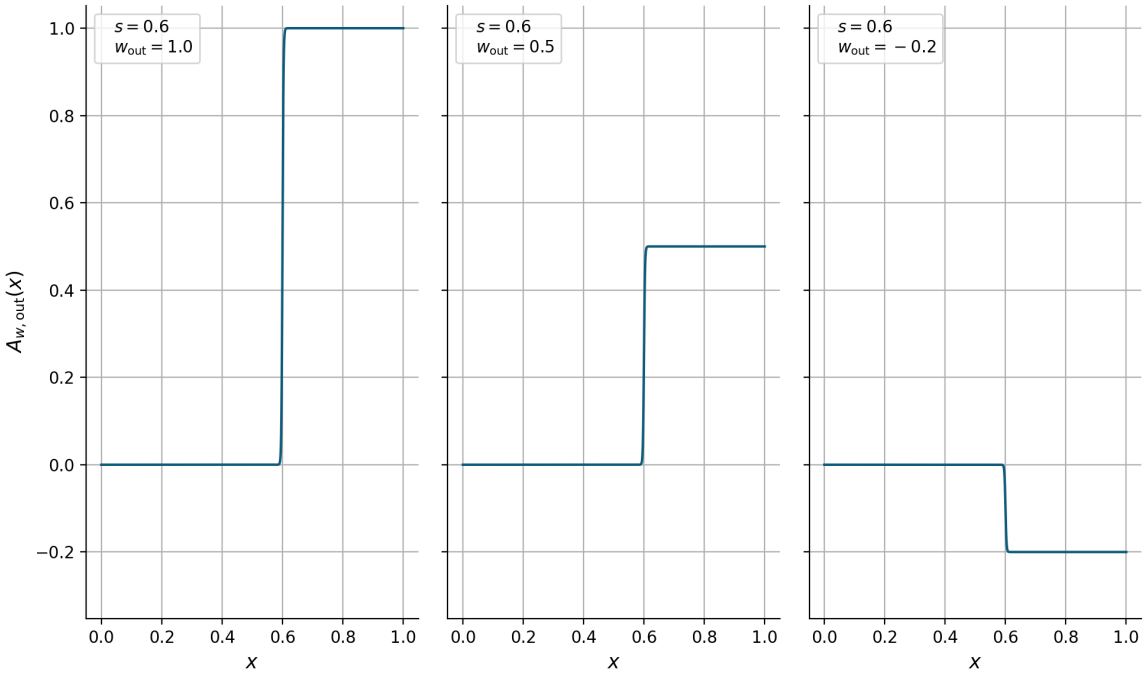
$$A_{w,\mathrm{out}}(x) := \sum_i w_{\mathrm{out},i} a_i(x) \tag{9}$$

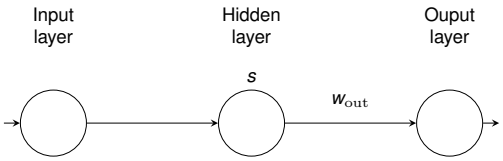
by assuming it to be a linear neuron and neglecting the its bias b_{out}.

Weighted output of one hidden layer with one neuron

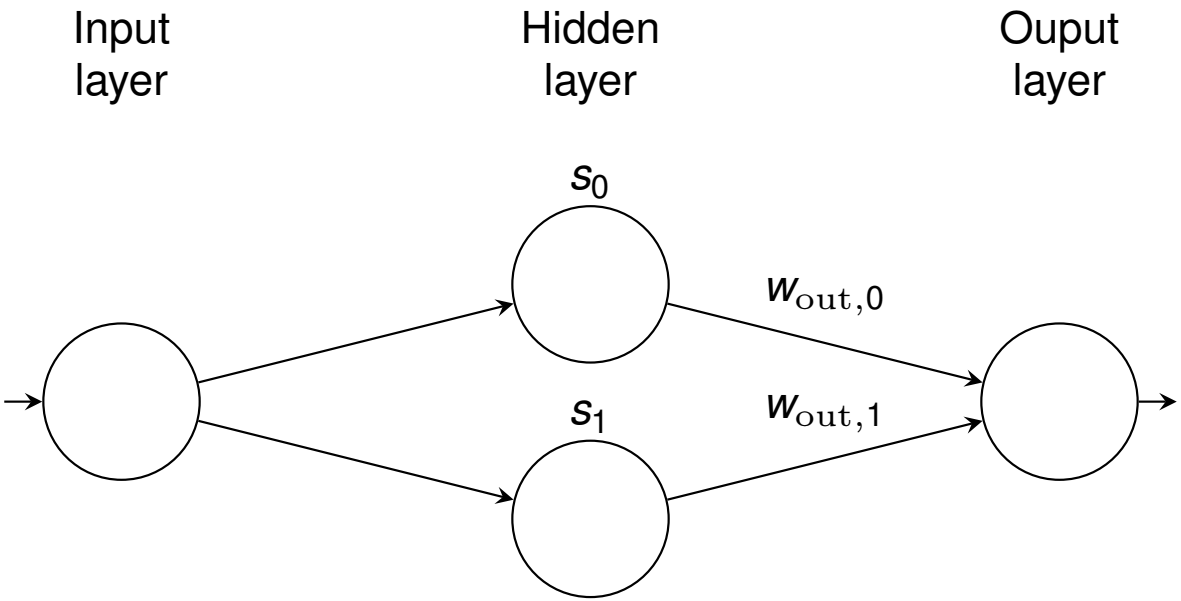
The activation of the neuron in the output layer by assuming it to be a linear neuron and neglecting the its bias b_{out}.

$$A_{w,\mathrm{out}}(x) := \sum_i w_{\mathrm{out},i} a_i(x) \tag{10}$$

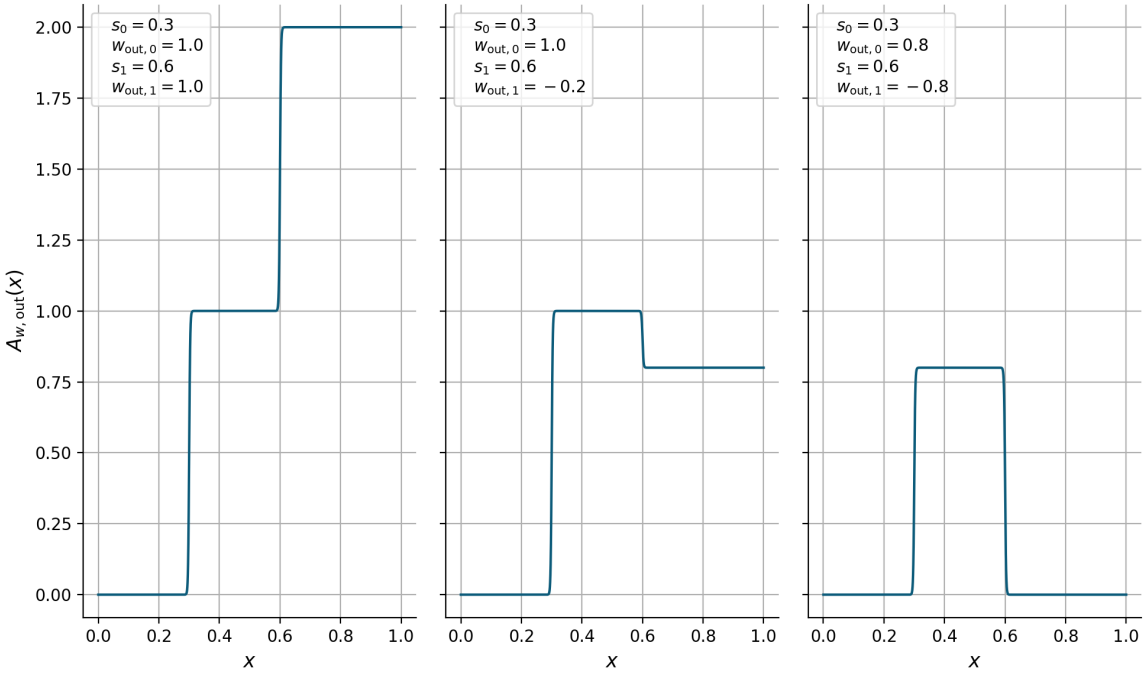


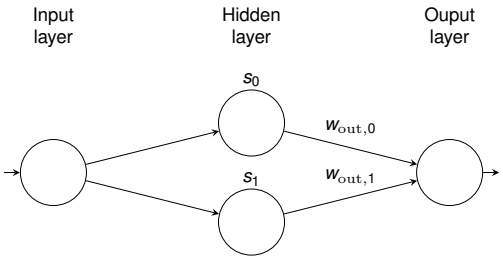


One hidden layer with two neurons and weighted output



Weighted output of one hidden layer with two neurons



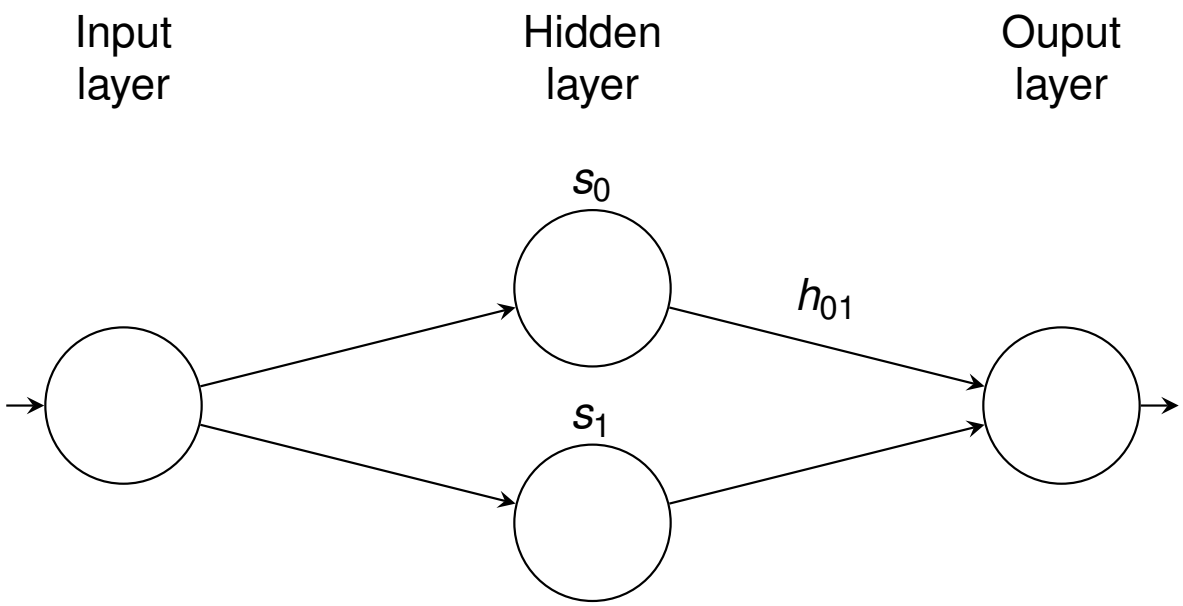


Definition of h

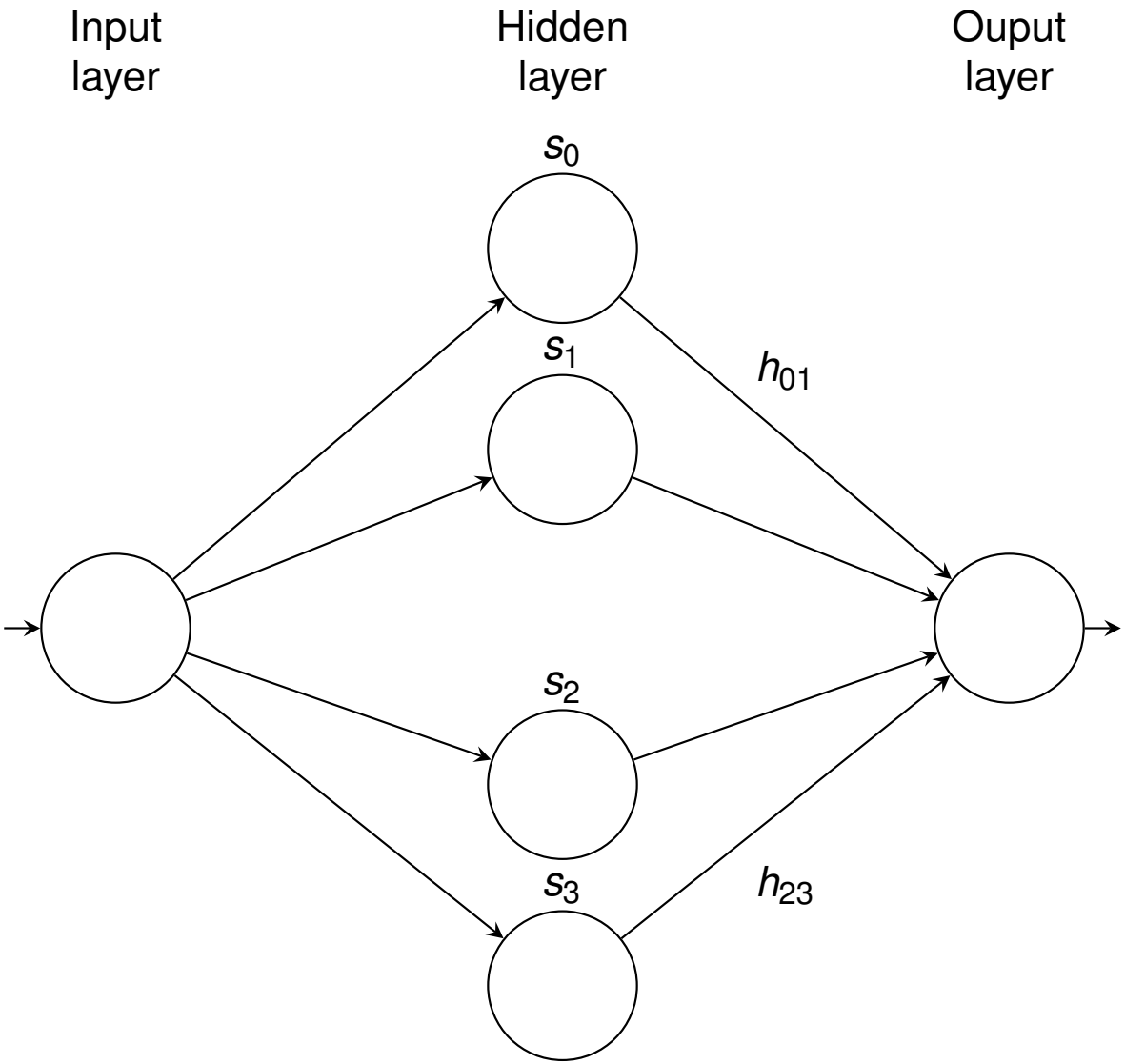
In the case $w_{\mathrm{out},i} = -w_{\mathrm{out},j}$ and $s_i < s_j$ we can define the height of the resulting towers as

$$h_{ij} := w_{\mathrm{out},i} = -w_{\mathrm{out},j}. \tag{11}$$

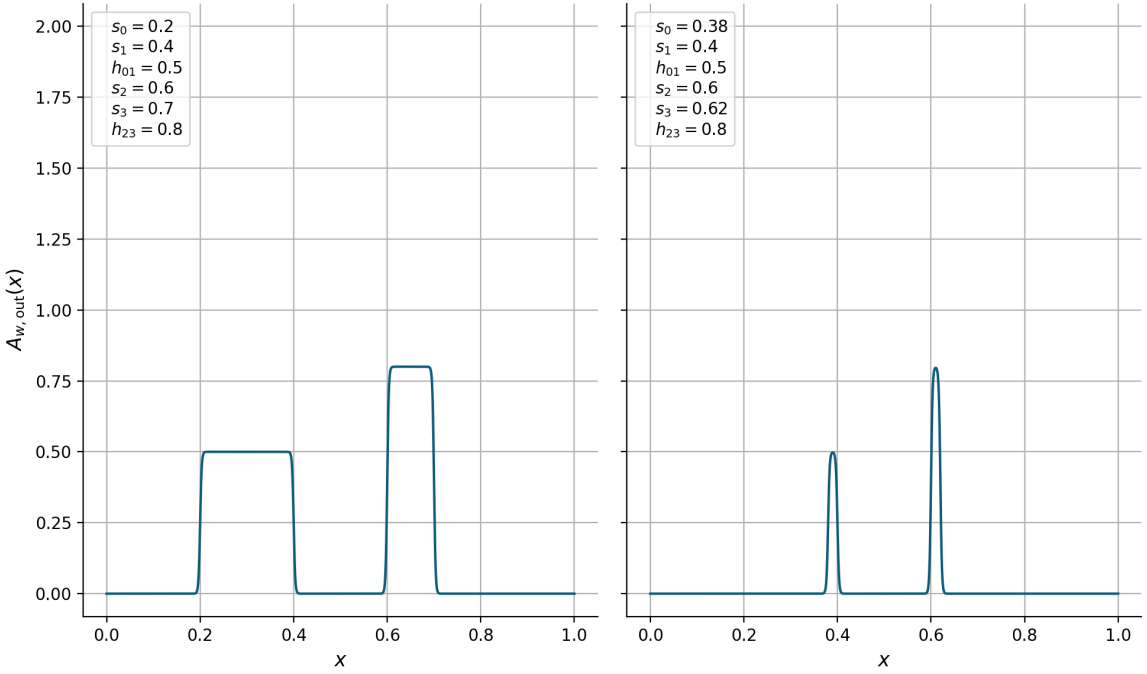
With that we get for our neural net :

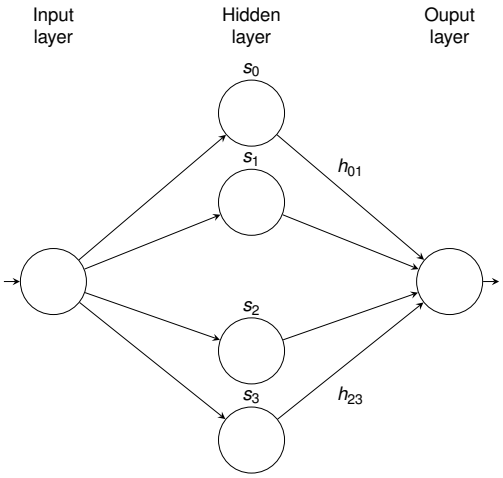


One hidden layer with four neurons

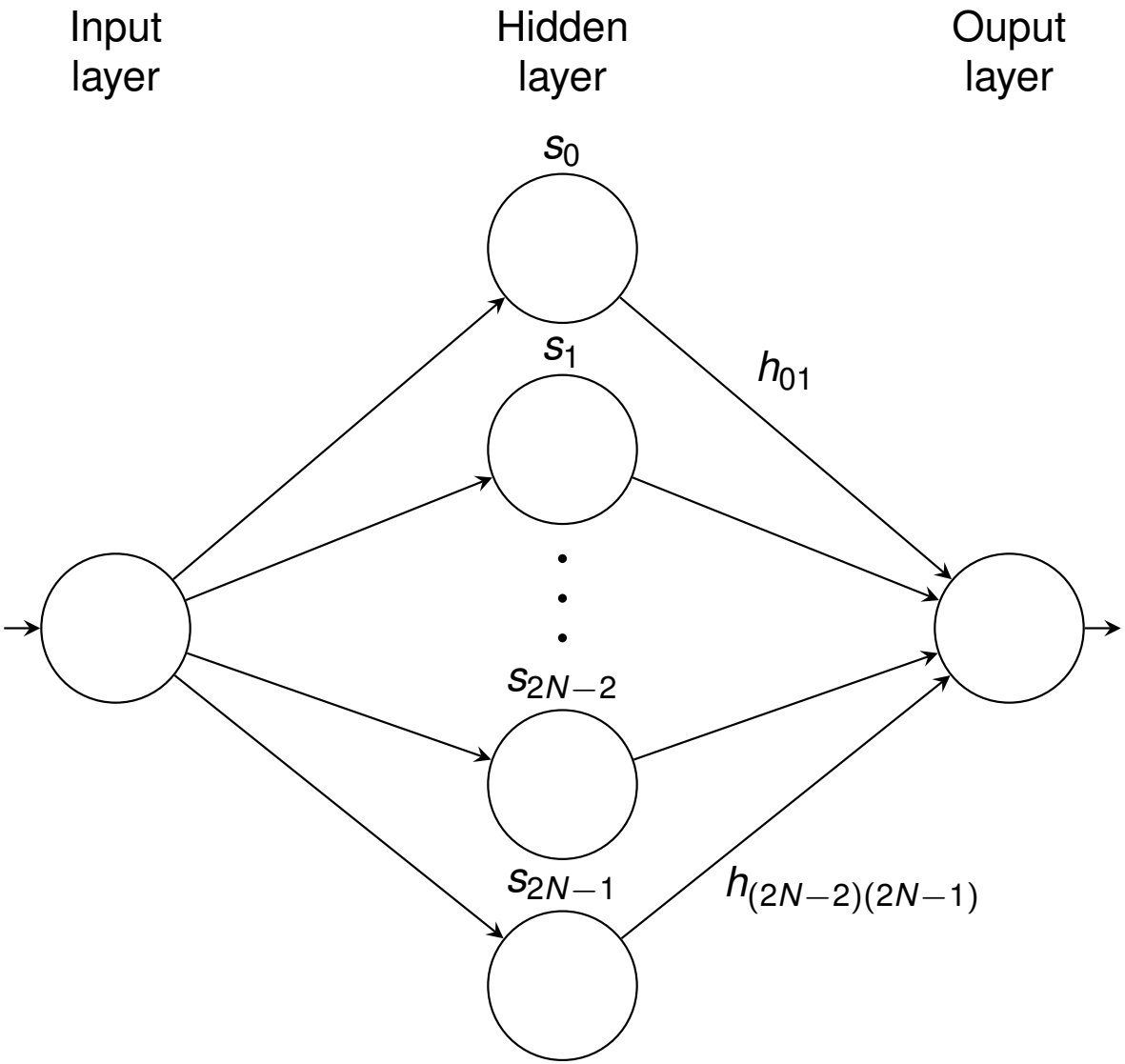


One hidden layer with four neurons

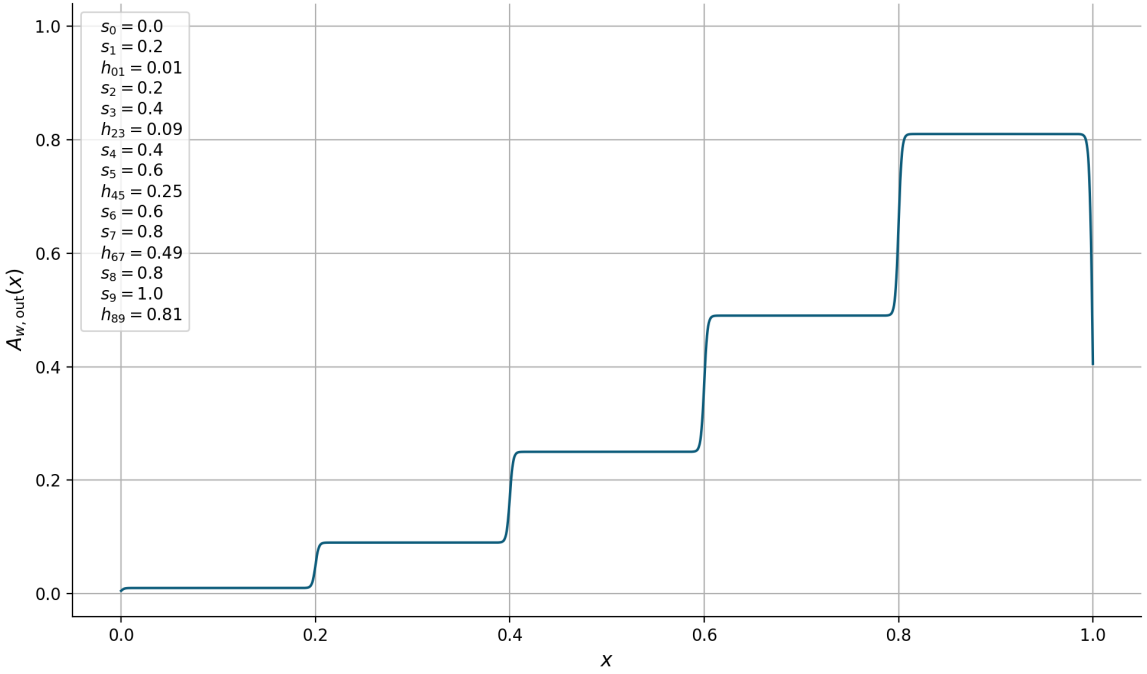


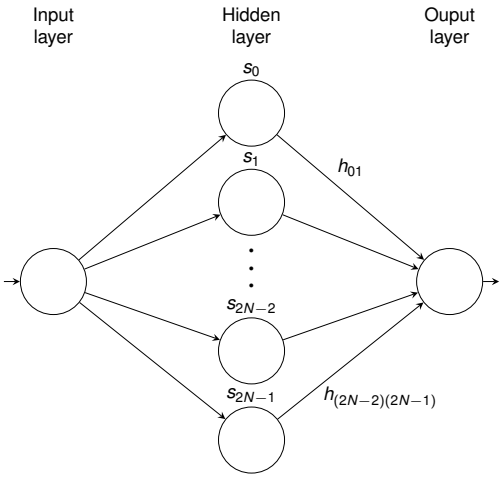


One hidden layer with $2N$ neurons

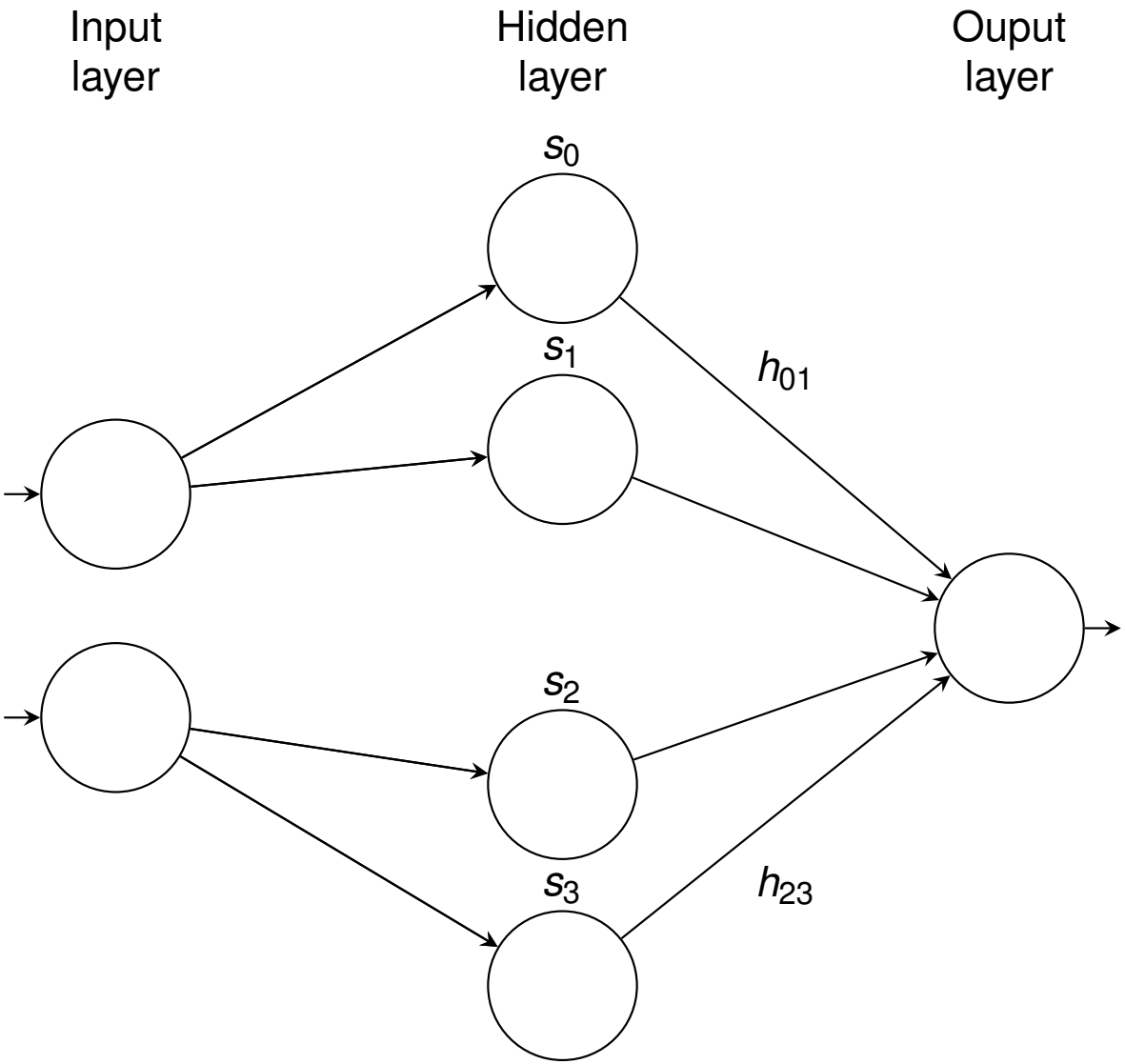


Weighted output of one hidden layer with ten neurons

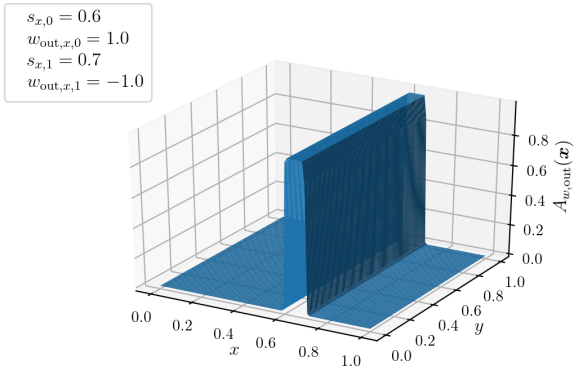


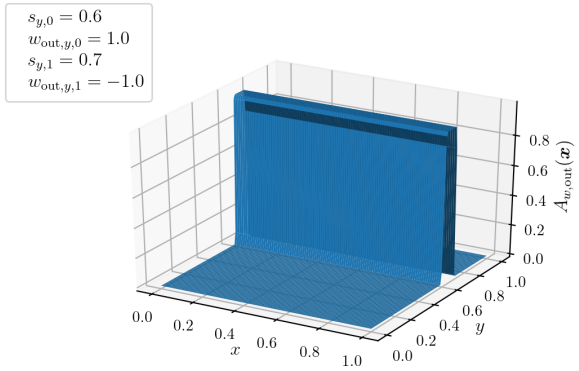


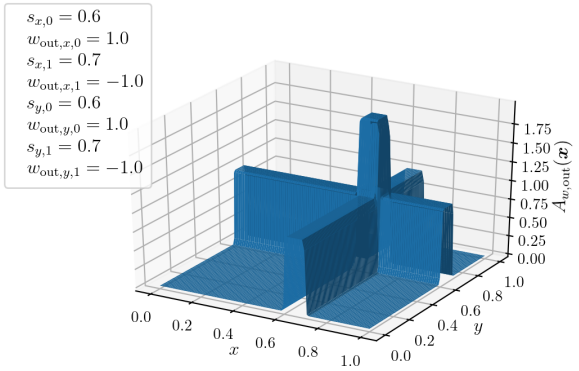
Two dimensional input

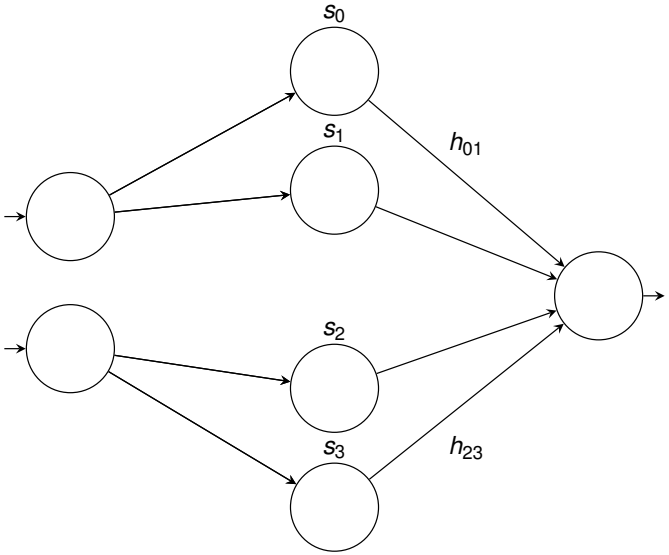


One dimensional towers in two dimensions



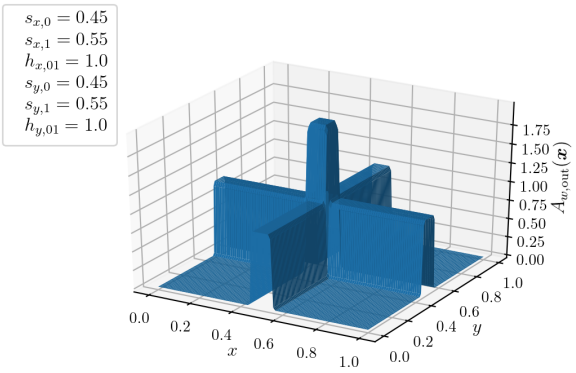


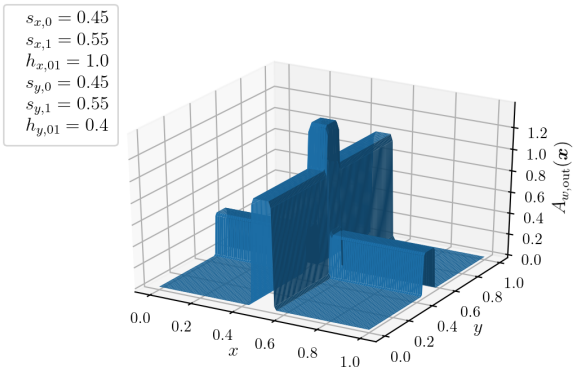


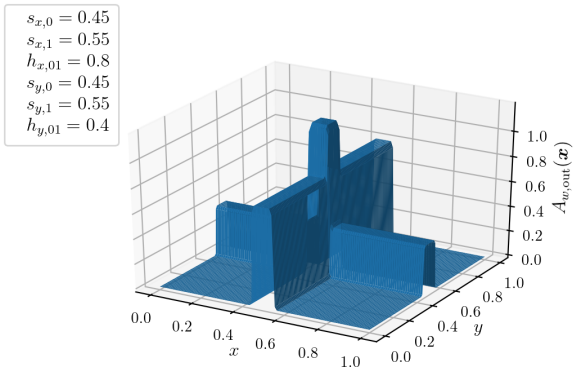


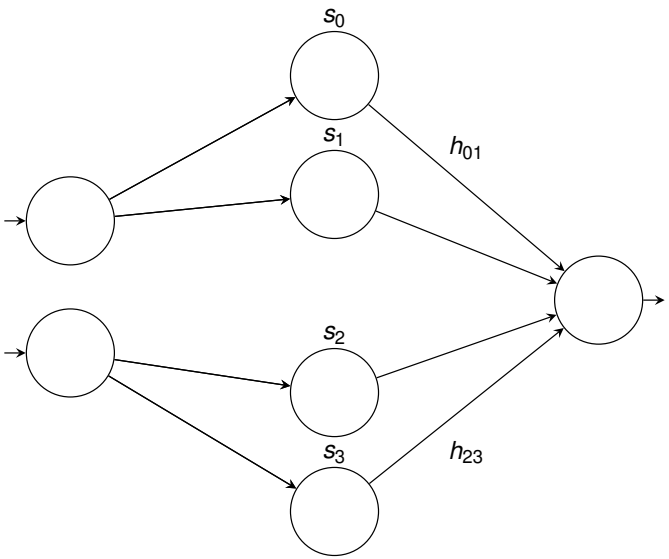
Vary heights two dimensions

Vary the *h* values :









Until now we have always looked at the linear function

$$A_{w,\mathrm{out}}(\boldsymbol{x}) = \sum_i w_{\mathrm{out},i} a_i(\boldsymbol{x}) \tag{12}$$

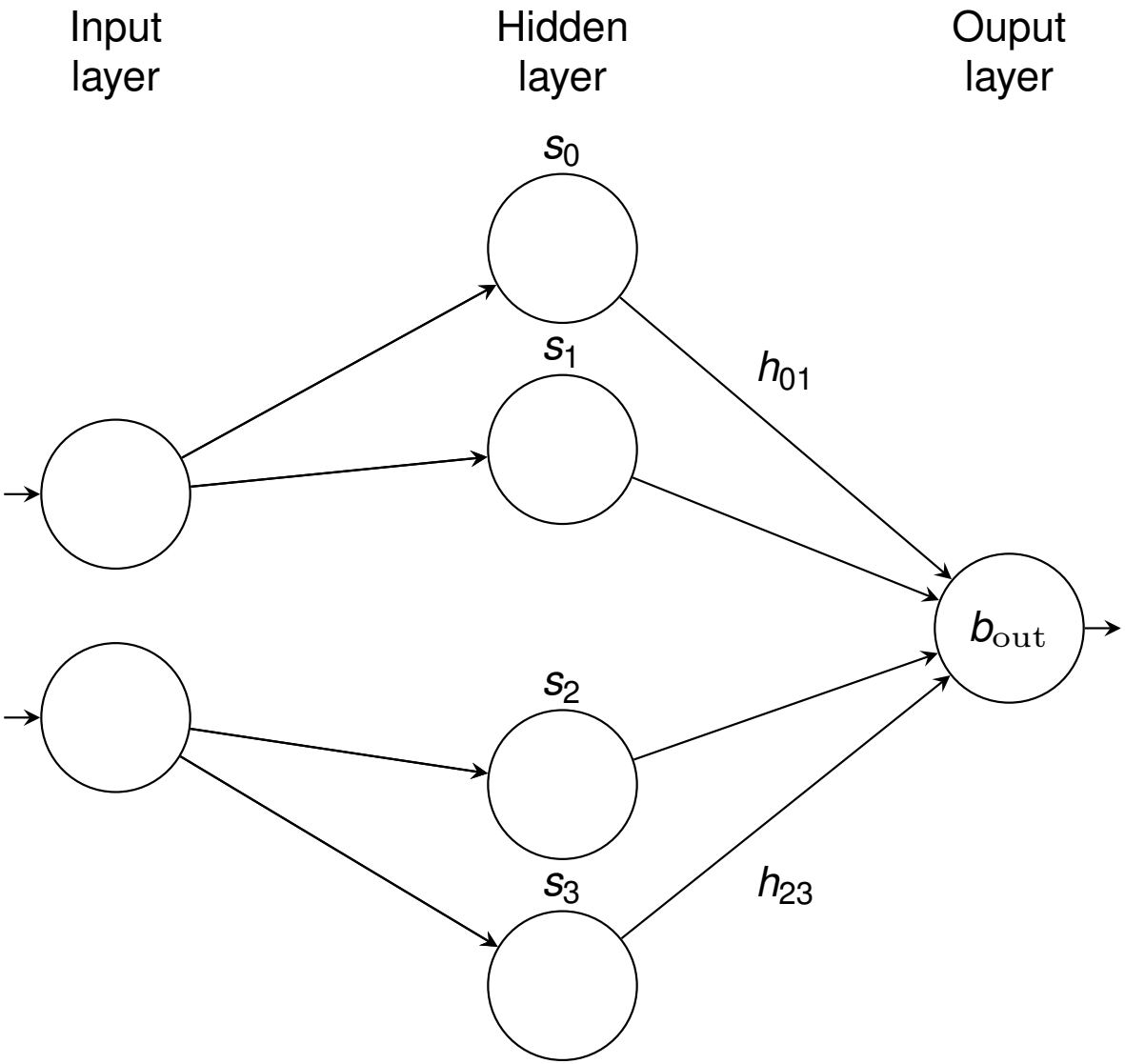
without taking a possible bias b_{out} into account.
By taking the bias b_{out} into account and still assuming that the neuron in the output layer is a linear neuron we get a activation of this neuron of

$$A_{wb,\mathrm{out}}(\boldsymbol{x}) := \sum_i w_{\mathrm{out},i} a_i(\boldsymbol{x}) + b_{\mathrm{out}}. \tag{13}$$

The output of the neuron in the output layer then becomes

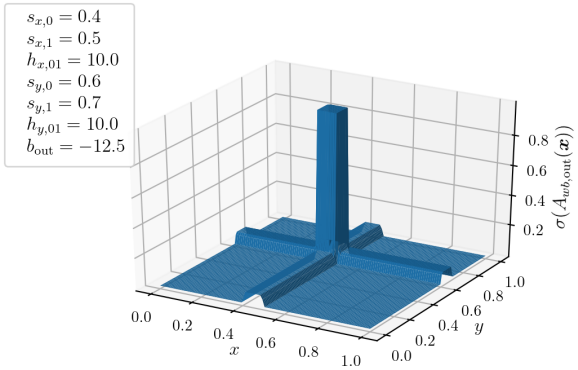
$$O = \sigma(A_{wb,\mathrm{out}}(\boldsymbol{x})). \tag{14}$$

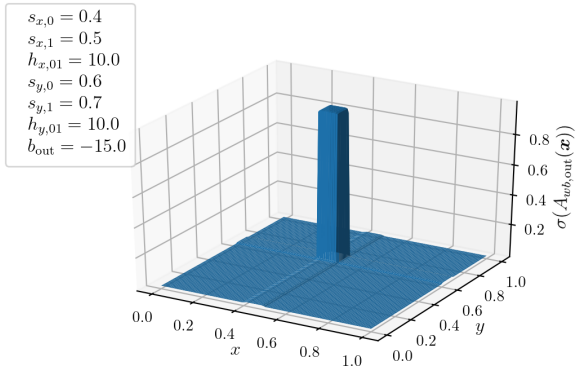
Two dimensional input with b_{out}

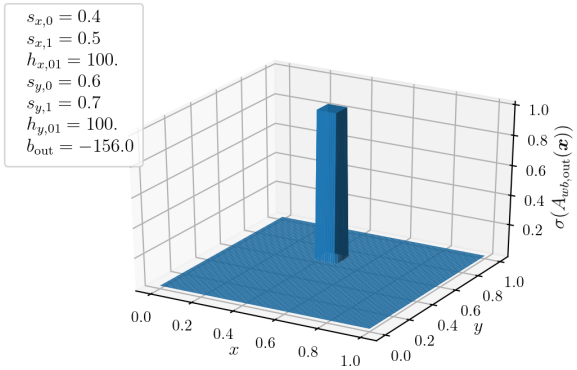


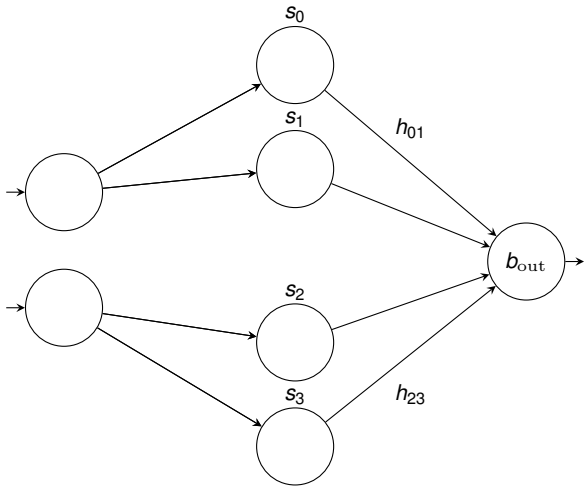
One tower in two dimensions

Vary the b_{out} values :









Best towers for

$$\frac{h_{x,01}}{b_{\text{out}}} \approx -\frac{2}{3} \approx \frac{h_{y,01}}{b_{\text{out}}}. \tag{15}$$

The parameter w'_{out}

How to combine those towers ?
We defined previously

$$A_{wb,\text{out}}(\boldsymbol{x}) := \sum_i w_{\text{out},i} a_i(\boldsymbol{x}) + b_{\text{out}} \tag{16}$$

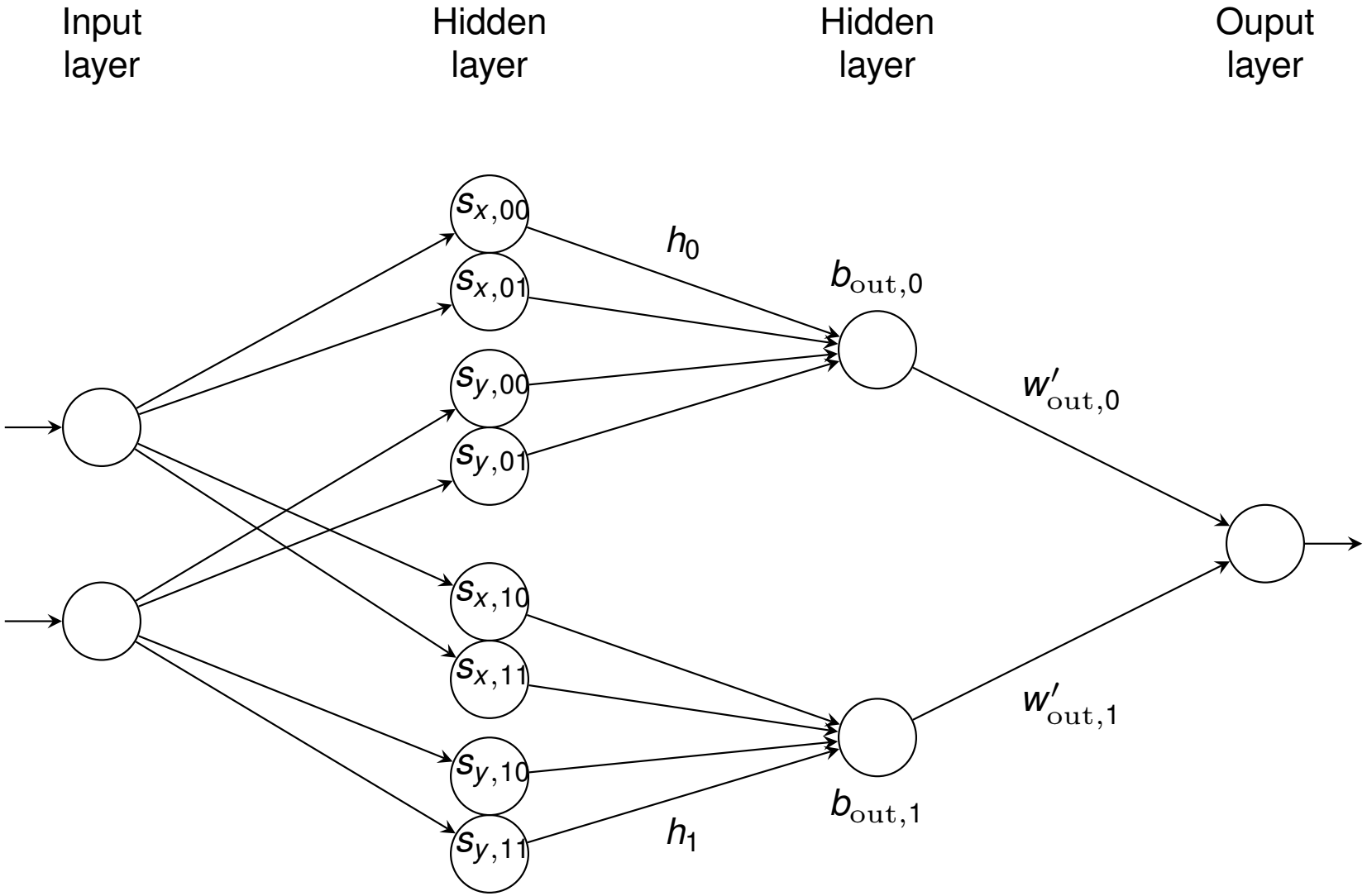
to describe the weighted output of the neuron in the output layer by assuming it to be a linear neuron.
By introducing another layer which gets as input the outputs of multiple of such activations of not linear but sigmoid neurons and its activation is our desired approximation function $g(\boldsymbol{x})$ we get

$$g(\boldsymbol{x}) = \sigma \circ \sum_i w'_{\text{out},i} \sigma(A_{wb,\text{out},i}(\boldsymbol{x})) \tag{17}$$

$$\sigma^{-1} \circ g(\boldsymbol{x}) = \sum_i w'_{\text{out},i} \sigma(A_{wb,\text{out},i}(\boldsymbol{x})) \tag{18}$$

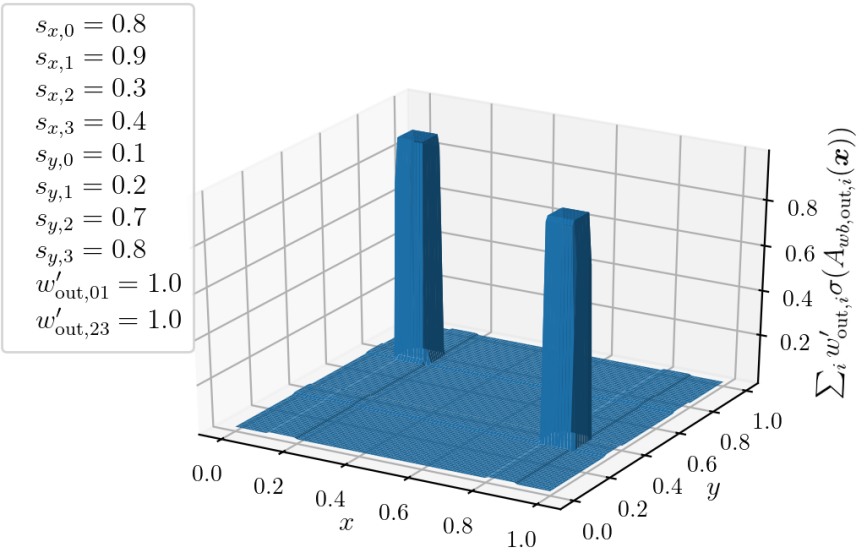
with the bias of the neuron in the new output layer $b'_{\text{out}} = 0$.

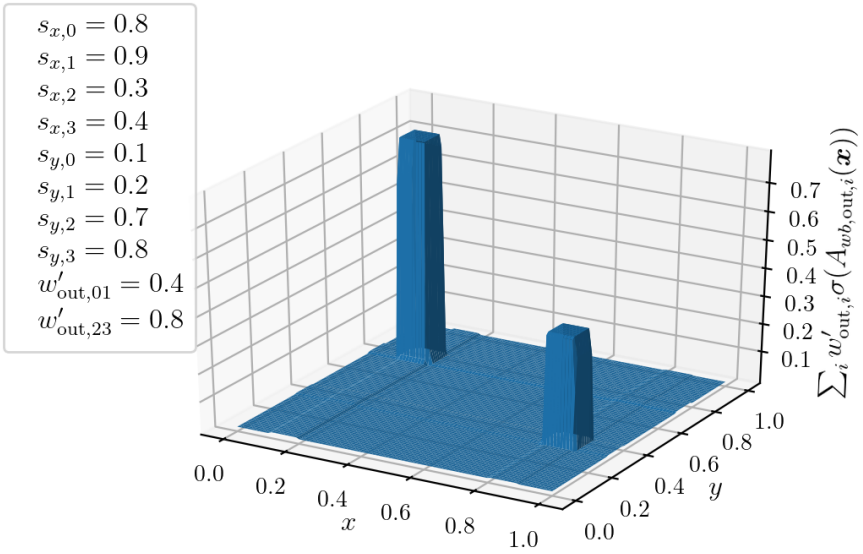
Two hidden layers with w'_{out}

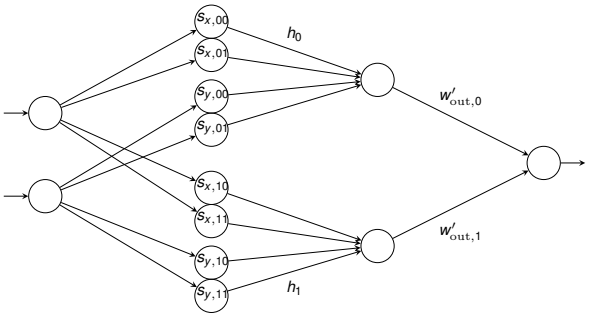


Two weighted towers in two dimensions

Vary the w'_{out} values with $\frac{h_0}{b_{\text{out},0}} = \frac{h_1}{b_{\text{out},1}} \approx -\frac{2}{3}$:







Web-application

Now in an interactive way in a web-application built with Streamlit .

$\mathbb{R}^n \to \mathbb{R}^m$

We only considered the case of approximating real-valued functions like

$$f : \mathbb{R}^n \to \mathbb{R}, \boldsymbol{x} \mapsto f(\boldsymbol{x}) \tag{19}$$

with $n \in \mathbb{N}$.
What about vector-valued functions like

$$\tilde{f} : \mathbb{R}^n \to \mathbb{R}^m, \boldsymbol{x} \mapsto \tilde{f}(\boldsymbol{x}) \tag{20}$$

with $n, m \in \mathbb{N}$ and $m > 1$?
We can simply approximate each ith component of the function $\tilde{f}$ with m different networks and combine them all together like

$$\tilde{f} = \begin{pmatrix} \tilde{f}_0 \\ \vdots \\ \tilde{f}_{m-1} \end{pmatrix} \tag{21}$$

Non-monotonic activation function

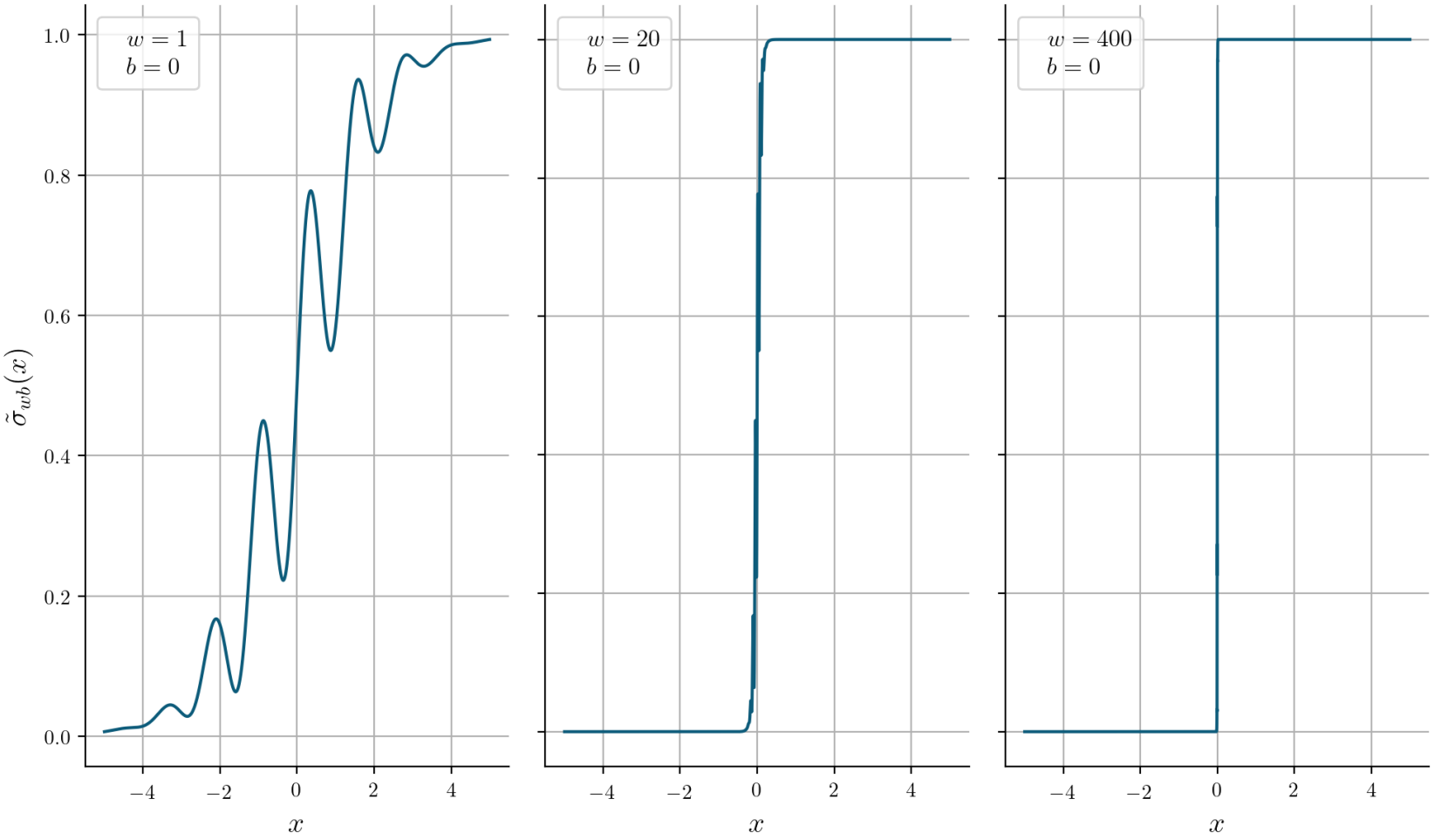
Does our activation function have to be monotonic ?
Let's have a look at the activation function

$$\tilde{\sigma}(z) = \frac{1}{1+e^{-z}} + 0.2\exp\left(-\frac{z^2}{4}\right)\sin(5z)\,. \tag{22}$$

With $z = wx + b$ we get

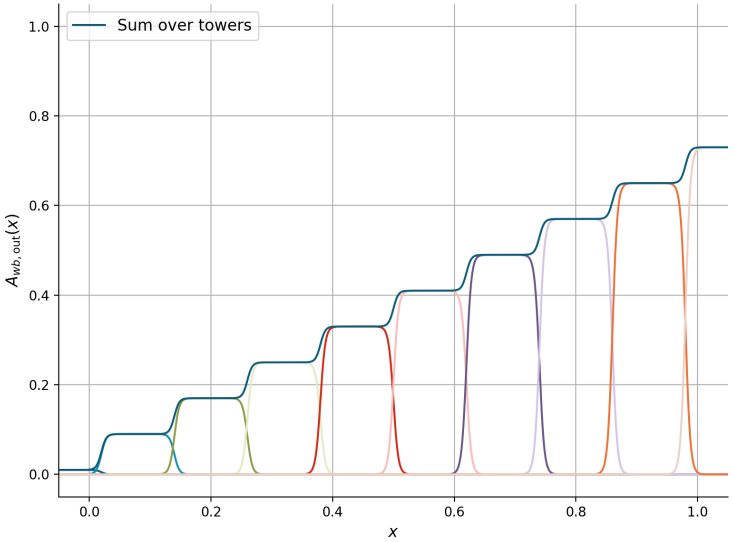
$$\tilde{\sigma}_{wb}(x) = \frac{1}{1+\exp(-(wx+b))} + 0.2\exp\left(-\frac{(wx+b)^2}{4}\right)\sin(5(wx+b))\,. \tag{23}$$

Non-monotonic activation function

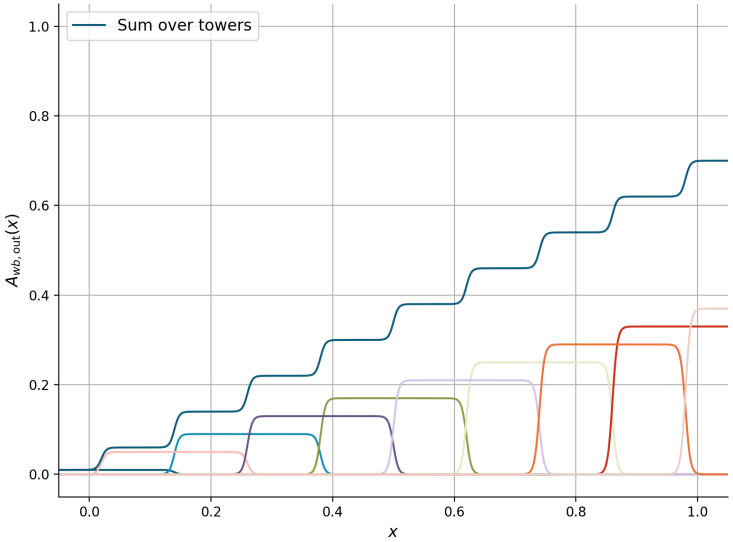


Minimize error

One approximation of $\sigma^{-1} \circ f(x)$:

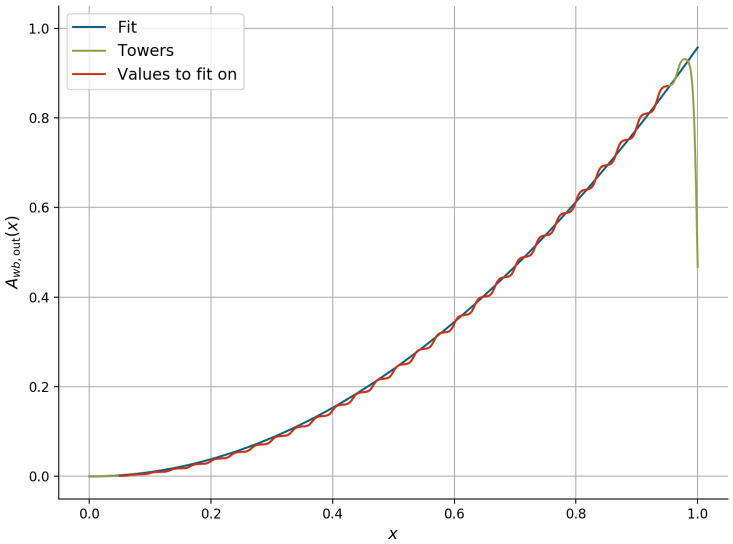


Two approximations of $\sigma^{-1} \circ f(x)/2$:



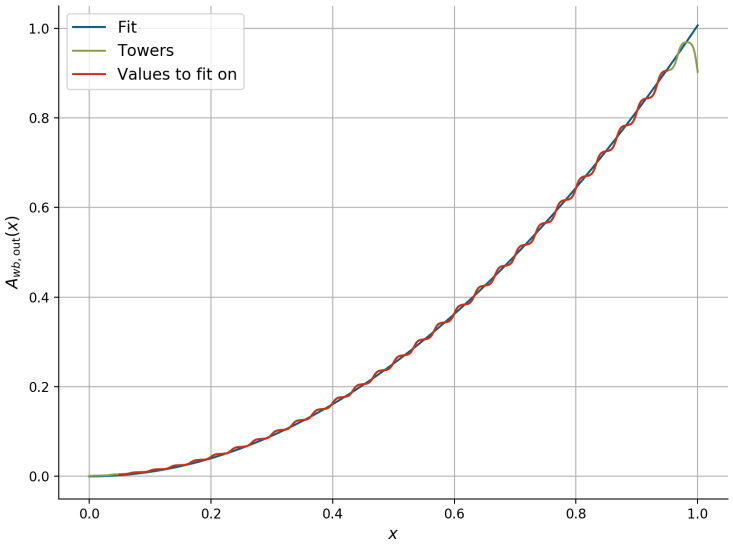
Minimize error example

One approximation with thirty towers.



$$\sigma_{\mathrm{std}} \approx 1.65 \cdot 10^{-4} \tag{24}$$

Six approximations with five towers each.



$$\sigma_{\mathrm{std}} \approx 1.31 \cdot 10^{-4} \tag{25}$$

Mathematical Approach

— A Brief Overview. [0]

1 Overview

2 Visual Approach
- Caveats
- Activation functions
- Construction
- The interactive way
- $\mathbb{R}^n \to \mathbb{R}^m$
- Non-monotonic activation functions
- Minimize error

3 Mathematical Approach
- Mathematical expression of a neural network
- What to prove ?
- Theorem I - Approximation of functions
- Lemma I - Sigmoidal functions are discriminatory

4 Back to Neural Networks

5 Appendix

Abstraction of a given problem

Neural Network	$\longrightarrow$	Mathematical Expression
Problem with n input parameters	$\longrightarrow$	Input-vector $\tilde{\boldsymbol{x}} \in \mathbb{R}^n$
Neural Network yields a scalar	$\longrightarrow$	$g\colon \mathbb{R}^n \longrightarrow \mathbb{R}$
Generalized input-type	$\longrightarrow$	$\tilde{\boldsymbol{x}}_i > 0 \;\wedge\; \boldsymbol{x} := \frac{\tilde{\boldsymbol{x}}}{\lVert\tilde{\boldsymbol{x}}\rVert_{\max}} \in [0,1]^n =: \boldsymbol{\mathcal{I}}_n$

Motivation

Every problem is some $f\colon \boldsymbol{\mathcal{I}}_n \longrightarrow \mathbb{R}$, i.e. $f \in \mathcal{C}(\boldsymbol{\mathcal{I}}_n)$.

$\downarrow$

Can g represent any $f \in \mathcal{C}(\boldsymbol{\mathcal{I}}_n)$?

Space of functions

Neural Network	$\longrightarrow$	Mathematical Expression
Neurons : weight $\boldsymbol{w}$, bias b and activation function σ	$\longrightarrow$	Define $\mathcal{G}$, space of functions, $g(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \sigma(\boldsymbol{w}_i^T \boldsymbol{x} + b_i)$
In 1-D $\sigma(wx + b) \xrightarrow{\text{large } w} \theta(-\frac{b}{w})$ θ Heaviside-function	$\longrightarrow$	Define space of **simple functions** $\mathcal{S}$, $s(\boldsymbol{x}) = \sum_{i=1}^{N'} c_i \chi_{A_i}(\boldsymbol{x})$

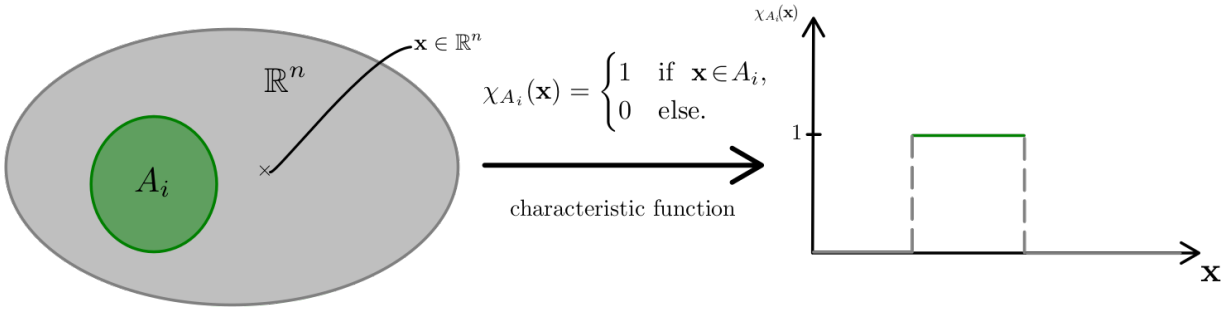


FIGURE – Characteristic function $\chi_{A_i}(\boldsymbol{x})$

Motivation

Given a function $f \in \mathcal{C}(\boldsymbol{\mathcal{I}}_n)$ — How good can we approximate it with some $g \in \mathcal{G}$?

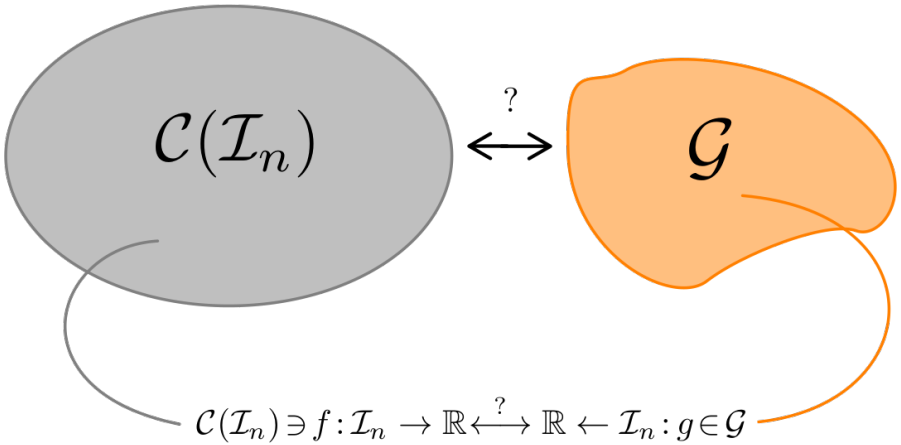


FIGURE – Function f we want to approximate with a function g.

$\longrightarrow$ It is a **topological** problem :

Question

Given $f \in \mathcal{C}(\boldsymbol{\mathcal{I}}_n)$ and $\varepsilon > 0$ arbitrary small. Do we find a function $g \in \mathcal{G}$ with

$$|g(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon \quad , \forall \boldsymbol{x} \in \boldsymbol{\mathcal{I}}_n \tag{26}$$

or topologically speaking, is $\mathcal{G}$ dense in $\mathcal{C}(\boldsymbol{\mathcal{I}}_n)$?

Is $\mathcal{G}$ dense in $\mathcal{C}(\mathcal{I}_n)$?

How do $\mathcal{G}$ and $\mathcal{C}(\mathcal{I}_n)$ topologically relate to each other ?

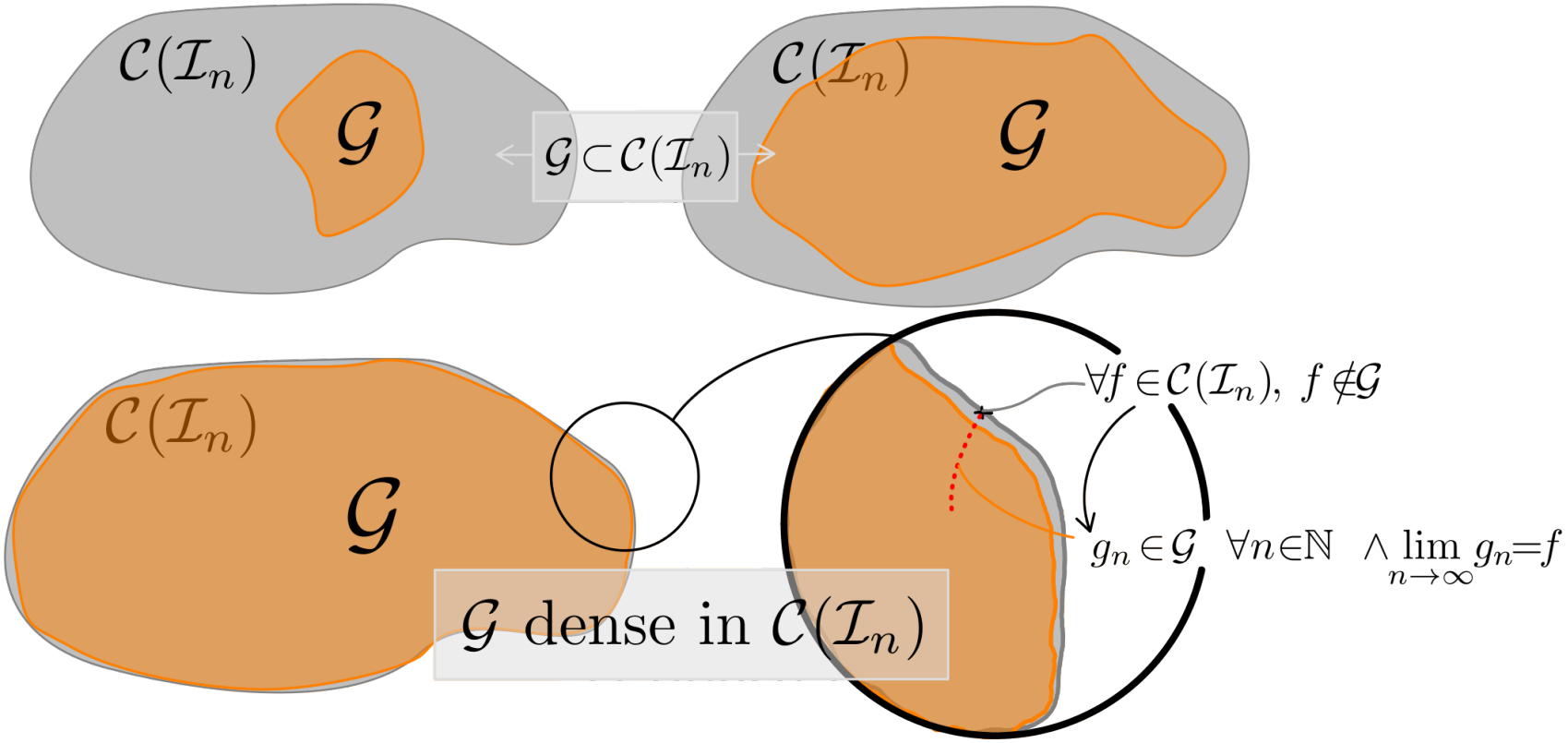


FIGURE – Topological relation of $\mathcal{G}$ and $\mathcal{C}(\mathcal{I}_n)$.

To prove

$\longrightarrow$ **Indeed we will show that $\mathcal{G}$ is dense in $\mathcal{C}(\mathcal{I}_n)$.**

To prove

$\mathcal{G}$ is dense in $\mathcal{C}(\mathcal{I}_n)$ if $\forall f \in \mathcal{C}(\mathcal{I}_n), f \notin \mathcal{G}$ and $\varepsilon > 0$, $\exists$ a function $g \in \mathcal{G}$ with,

$$\begin{aligned} &|g(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon \quad , \forall \boldsymbol{x} \in \mathcal{I}_n \\ &\Leftrightarrow \overline{\mathcal{G}} = \mathcal{C}(\mathcal{I}_n) \quad , \overline{\mathcal{G}} \text{ closure of } \mathcal{G}. \end{aligned} \tag{27}$$

Further Definitions

Definition (Sigmoidal Functions)

A continuous function $\sigma \in \mathcal{C}(\mathcal{I}_n)$ is called **sigmoidal** if,

$$\sigma(z) \to \begin{cases} 1 & \text{if } z \to \infty \\ 0 & \text{if } z \to -\infty \end{cases} \tag{28}$$

Definition (Discriminatory)

A function σ is called **discriminatory** if for a measure $\mu \in \mathcal{M}(\mathcal{I}_n)$,

$$\int_{\mathcal{I}_n} \sigma(\boldsymbol{w}^T \boldsymbol{x} + b) \mathrm{d}\mu(\boldsymbol{x}) = 0 \quad \forall \boldsymbol{w} \in \mathbb{R}^n, b \in \mathbb{R}$$
$$\Rightarrow \mu = 0 \tag{29}$$

$\mathcal{C}(\mathcal{I}_n)$	Continuous functions
$\mathcal{G} \subset \mathcal{C}(\mathcal{I}_n)$	Functions of form $g(\boldsymbol{x})$
$\mathcal{S}$	Simple functions $s(\boldsymbol{x})$
sigmoidal	$\sigma = \begin{cases} 1 & \text{if } z \to \infty \\ 0 & \text{if } z \to -\infty \end{cases}$
discriminatory	$0 = \int_{\mathcal{I}_n} f \, \mathrm{d}\mu \Rightarrow \mu = 0$

Theorem I

Let σ be any **discriminatory** continuous function.Then $\mathcal{G}$, i.e. the space functions of the form

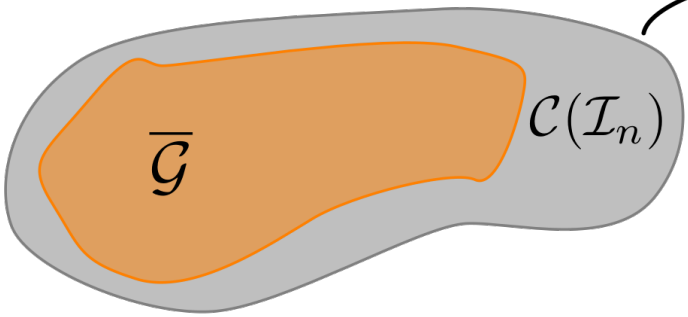
$$g(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \sigma(\boldsymbol{w_i}^T \boldsymbol{x} + b_i) \quad \alpha_i, b_i \in \mathbb{R} \;\; \boldsymbol{w}_i \in \mathbb{R}^n, \tag{30}$$

is dense in $\mathcal{C}(\boldsymbol{\mathcal{I}}_n)$, i.e. given any function $f \in \mathcal{C}(\boldsymbol{\mathcal{I}}_n)$ and $\mathbb{R} \ni \varepsilon > 0 \Rightarrow \exists \;\; g \in \mathcal{G}$ for which,

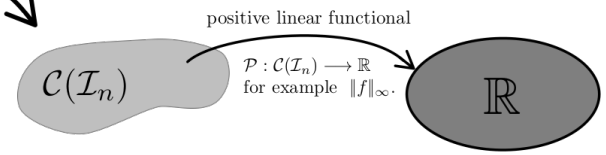
$$|g(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon \quad \forall \boldsymbol{x} \in \boldsymbol{\mathcal{I}}_n.$$

Proof - in a visual way

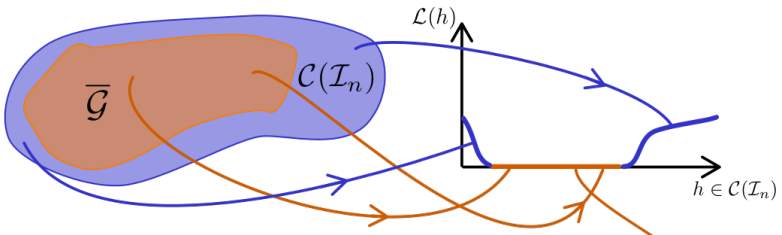
Assumption $\mathcal{G}$ is not dense in $\mathcal{C}(\mathcal{I}_n)$



Hahn-Banach Theorem



$\Downarrow$



$$\mathcal{L}: \mathcal{C}(\mathcal{I}_n) \longrightarrow \mathbb{R}, h \mapsto \begin{cases} \neq 0 & h \in \mathcal{C}(\mathcal{I}_n) \backslash \overline{\mathcal{G}}, \\ = 0 & h \in \overline{\mathcal{G}}. \end{cases}$$

Riesz-Representation Theorem

$$\Rightarrow \mathcal{L}(h) = \int_{\mathcal{I}_n} h(\mathbf{x}) \mathrm{d}\mu(\mathbf{x}) \quad , \forall h \in \mathcal{C}(\mathcal{I}_n)$$

for $h = \sigma(\mathbf{x})$ (**discriminatory**)

all $h \in \overline{\mathcal{G}} = \mathcal{C}(\mathcal{I}_n)$

$$\mathcal{L}(\sigma) = \int_{\mathcal{I}_n} \sigma(\mathbf{x}) \mathrm{d}\mu(\mathbf{x}) = 0 \implies \mu = 0 \implies \boxed{\mathcal{L}(h) = 0} \; \forall h \in \mathcal{C}(\mathcal{I}_n)$$

Conclusion of Theorem I

$\mathcal{G}$ is dense in $\mathcal{C}(\mathcal{I}_n)$ where

$$\mathcal{G} = \{g(\boldsymbol{x}) | \alpha_j, b_j \in \mathbb{R} \wedge \boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^n\}$$

with

$$g(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \sigma(\boldsymbol{w}^T \boldsymbol{x} + b_j)$$

if σ is *discriminatory*.

Are all sigmoidal functions discriminatory ?

Lemma I

If σ is a continuous bounded *sigmoidal* function, then σ is *discriminatory*.

Proof - in a visual way

Proof of Lemma I

Σ

$\sigma(\mathbf{w}^T\mathbf{x}+b) \equiv \sigma(z)$

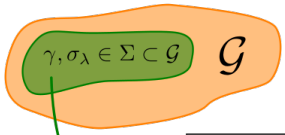
$\sigma_\lambda \equiv \sigma(\lambda z+\varphi) \quad \lambda,\varphi \in \mathbb{R}$

$$\sigma(z) = \begin{cases} 1 & \text{for } z \to \infty, \\ 0 & \text{for } z \to -\infty. \end{cases}$$

Convergence for $\lambda \to \infty$

$$\gamma(z,\varphi) \equiv \lim_{\lambda\to\infty} \sigma(\lambda z+\varphi) = \begin{cases} 1 & \text{if } z>0, \\ 0 & \text{if } z<0, \\ \sigma(\varphi) & \text{if } z=0. \end{cases}$$

Σ



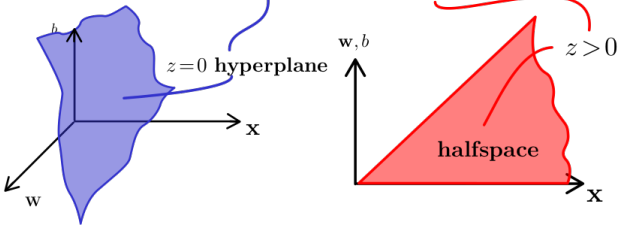
Lebesgue Bounded Convergence Theorem

— σ_λ **bounded, measurable and converges pointwise**

$$\Rightarrow \quad 0 = \lim_{\lambda\to\infty} \int_{\mathcal{I}_n} \sigma_\lambda \mathrm{d}\mu = \int_{\mathcal{I}_n} \gamma \mathrm{d}\mu$$

$$= \sigma(\varphi)\mu(\Pi_{\mathbf{w},b}) + \mu(\mathcal{H}_{\mathbf{w},b})$$

$$\boxed{\sigma(\varphi)\mu(\Pi_{\mathbf{w},b}) + \mu(\mathcal{H}_{\mathbf{w},b}) = 0}$$



Lemma I - Sigmoidal functions are discriminatory

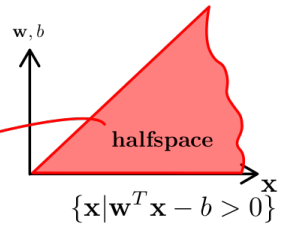
$$\boxed{\sigma(\varphi)\mu(\Pi_{\mathbf{w},b}) + \mu(\mathcal{H}_{\mathbf{w},b}) = 0}$$

$\forall \varphi, b, \mathbf{w}$

construct $\mathcal{F}$ and fix w.

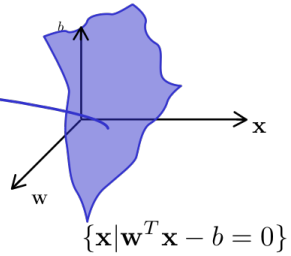
$$\mathcal{F}(h) = \int_{\mathcal{I}_n} h(\mathbf{w}^T\mathbf{x})\mathrm{d}\mu$$

h bounded, measurable $\Rightarrow \mathcal{F} \in L^\infty(\mathbb{R})$

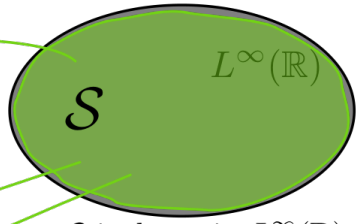


$$\text{for } \tilde{h} \equiv \begin{cases} 1 & \text{if } \mathbf{w}^T\mathbf{x} \geq b, \\ 0 & \text{else.} \end{cases} \Rightarrow \boxed{\mathcal{F}(\tilde{h}) = \mu(\Pi_{\mathbf{w},-b}) + \mu(\mathcal{H}_{\mathbf{w},-b}) = 0}$$

b arbitrary, also for sums of χ_{A_i}



$$\mathcal{F}(\tilde{h}) = 0 \quad \forall \tilde{h} \in \mathcal{S}$$



$$\boxed{\Rightarrow \mathcal{F} = 0 \quad \forall h \in L^\infty}$$

$\mathcal{S}$ is dense in $L^\infty(\mathbb{R})$

$$\mathcal{F}(\cos \mathbf{v}^T\mathbf{x} + i \sin \mathbf{v}^T\mathbf{x}) = \int_{\mathcal{I}_n} e^{i\mathbf{v}^T\mathbf{x}}\mathrm{d}\mu(\mathbf{x}) = 0 \quad \Leftrightarrow \boxed{\mu = 0}$$

$\square$

Conclusion

Any *sigmoidal* function σ is *discriminatory*.

$\longrightarrow$ That is why *sigmoidal* functions are interesting !

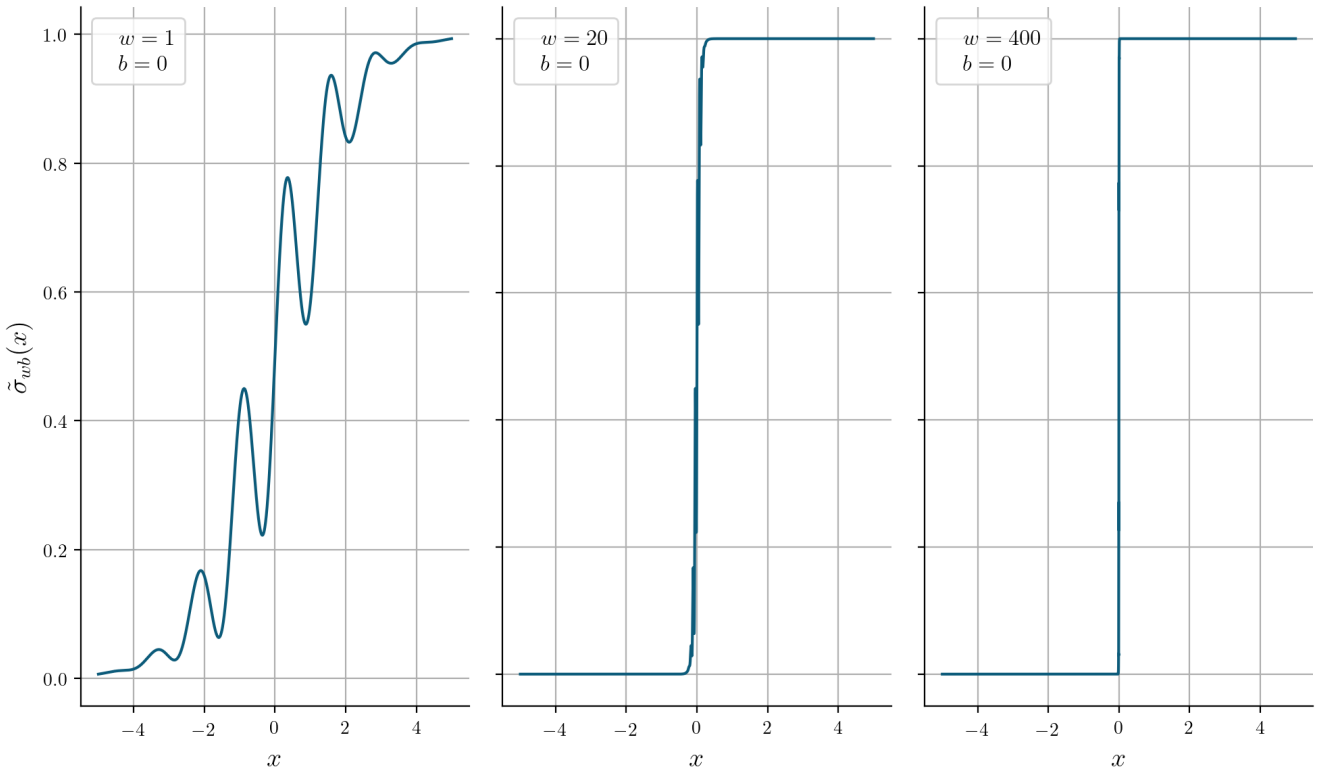


FIGURE – Proven for continuous bounded sigmoidal function σ.

Solving any function

What we have proven :

I. $\mathcal{G}$ is dense in $\mathcal{C}(\mathcal{I}_n)$, hence for any $f \in \mathcal{C}(\mathcal{I}_n)$ and $\varepsilon > 0 \ \exists \ g(\sigma(\boldsymbol{x}))$ with

$$|g(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon \quad \forall \boldsymbol{x} \in \mathbb{R}^n,$$

if σ is *discriminatory*.

II. Any bounded continuous *sigmoidal* function σ is *discriminatory*.

Computing any Function

A neuronal network with a *sigmoidal* activation-function and sufficiently many neurons can compute any function.

$$+$$

Any problem is a function

One can solve any problem in terms of computing a function.

Problem of Machine Learning solved —?

Back to Machine Learning

We dont know the function, we want to approximate.

Problems

- **How many Neurons needed for good approximation ?**
  $\rightarrow$ should still be efficient & accurate.
- **Finite number of testdata & errors in testdata**
  $\rightarrow$ *wrong* function is being approximated.

The Idea

Find the algorithm (learning system, e.g. *Back-Propagation*) that will approximate the *correct* or rather *correctest* function to a given Problem, with the knowledge that any function can be approximated !

Thank you for your attention !

Are there any questions ?

Activation functions

Step function :

$$\sigma(z) = \begin{cases} 0 & : z < 0 \\ 1 & : z \geq 0 \end{cases} \tag{31}$$

Sigmoid function :

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{32}$$

ReLU function :

$$\sigma(z) = \begin{cases} 0 & : z < 0 \\ z & : z \geq 0 \end{cases} \tag{33}$$

Parametrised ReLU function :

$$\sigma(z) = \begin{cases} \alpha_1 z & : z < 0 \\ \alpha_2 z & : z \geq 0 \end{cases} \tag{34}$$

Swish function :

$$\sigma(z) = \frac{z}{1 + e^{-z}} \tag{35}$$

Linear function :

$$\sigma(z) = \alpha z \tag{36}$$

Tanh function :

$$\sigma(z) = \frac{2}{1 + e^{-2z}} - 1 \tag{37}$$

Leaky ReLU function :

$$\sigma(z) = \begin{cases} 0.1z & : z < 0 \\ z & : z \geq 0 \end{cases} \tag{38}$$

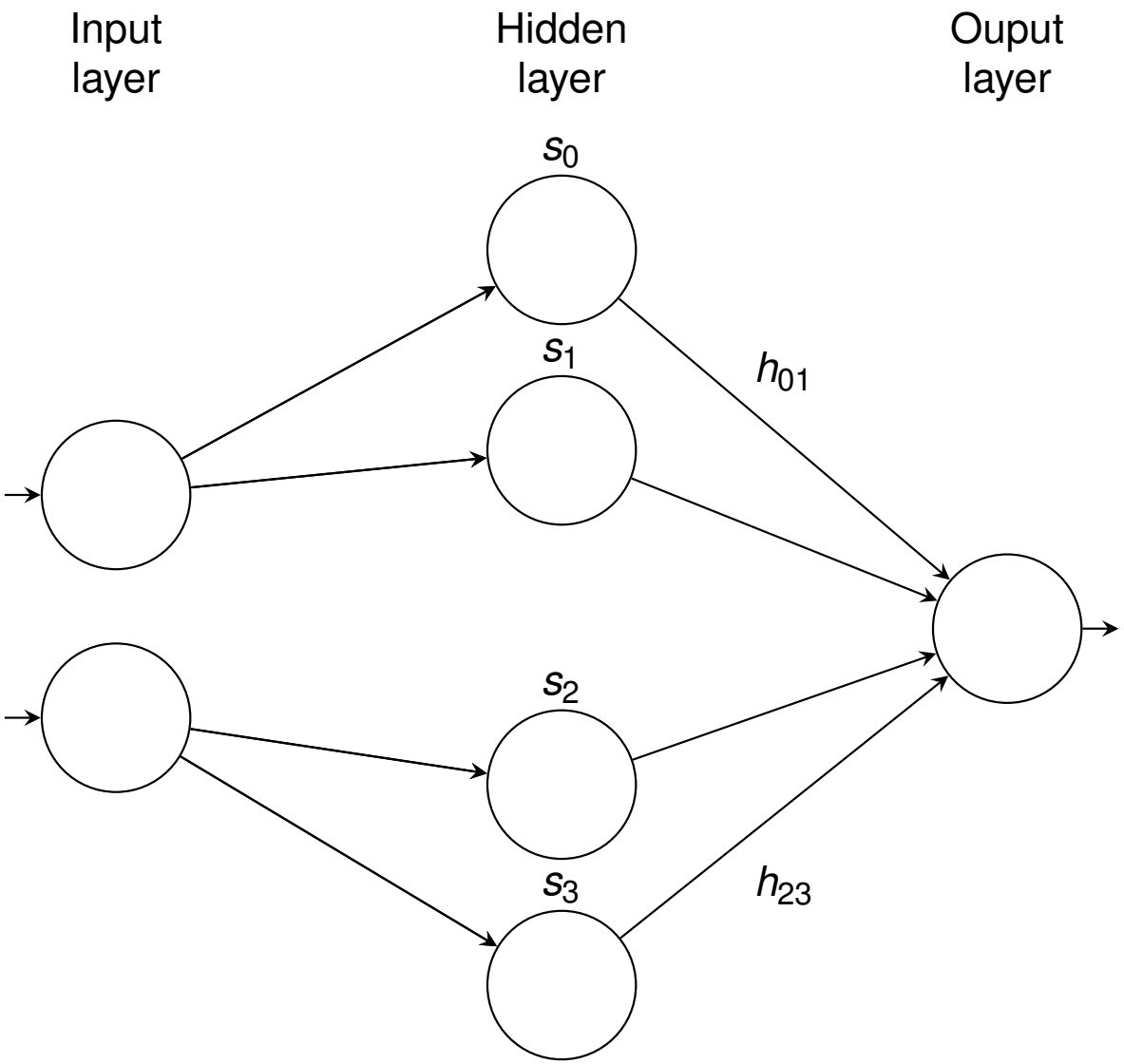
ELU function :

$$\sigma(z) = \begin{cases} \alpha(e^z - 1) & : z < 0 \\ z & : z \geq 0 \end{cases} \tag{39}$$

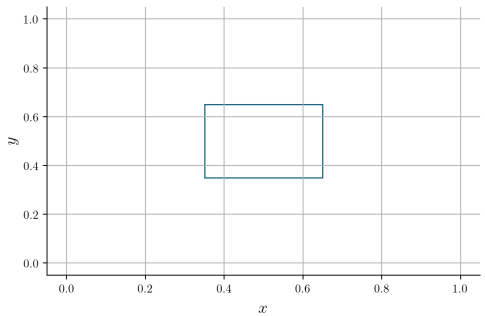
Sine ReLU function :

$$\sigma(z) = \begin{cases} \alpha(\sin(z) - \cos(z)) & : z < 0 \\ z & : z \geq 0 \end{cases} \tag{40}$$

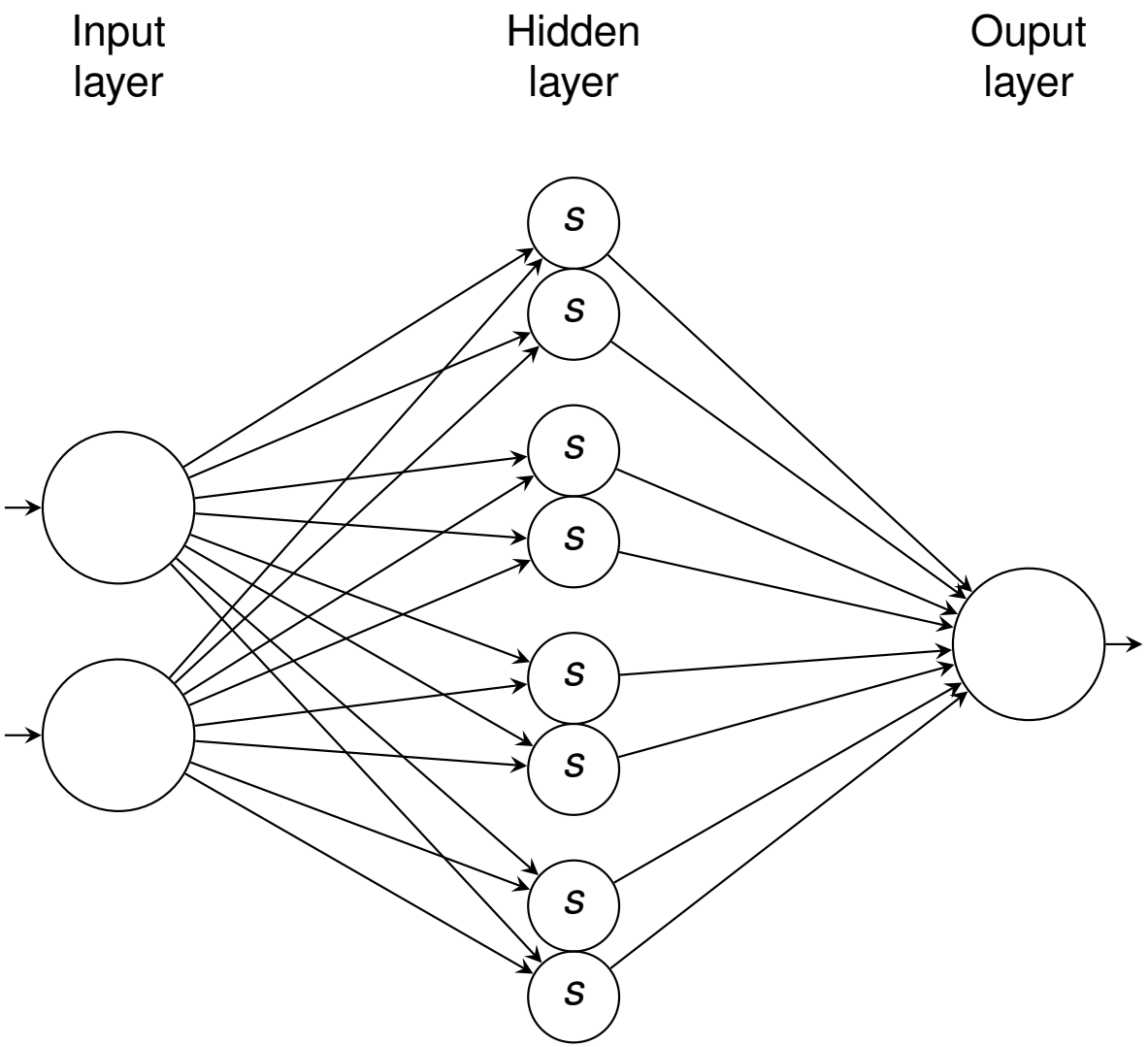
Two dimensional input - rectangle



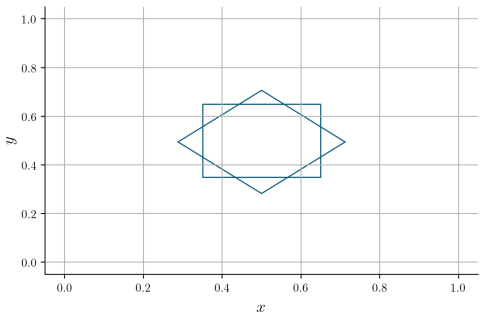
Corresponding area :



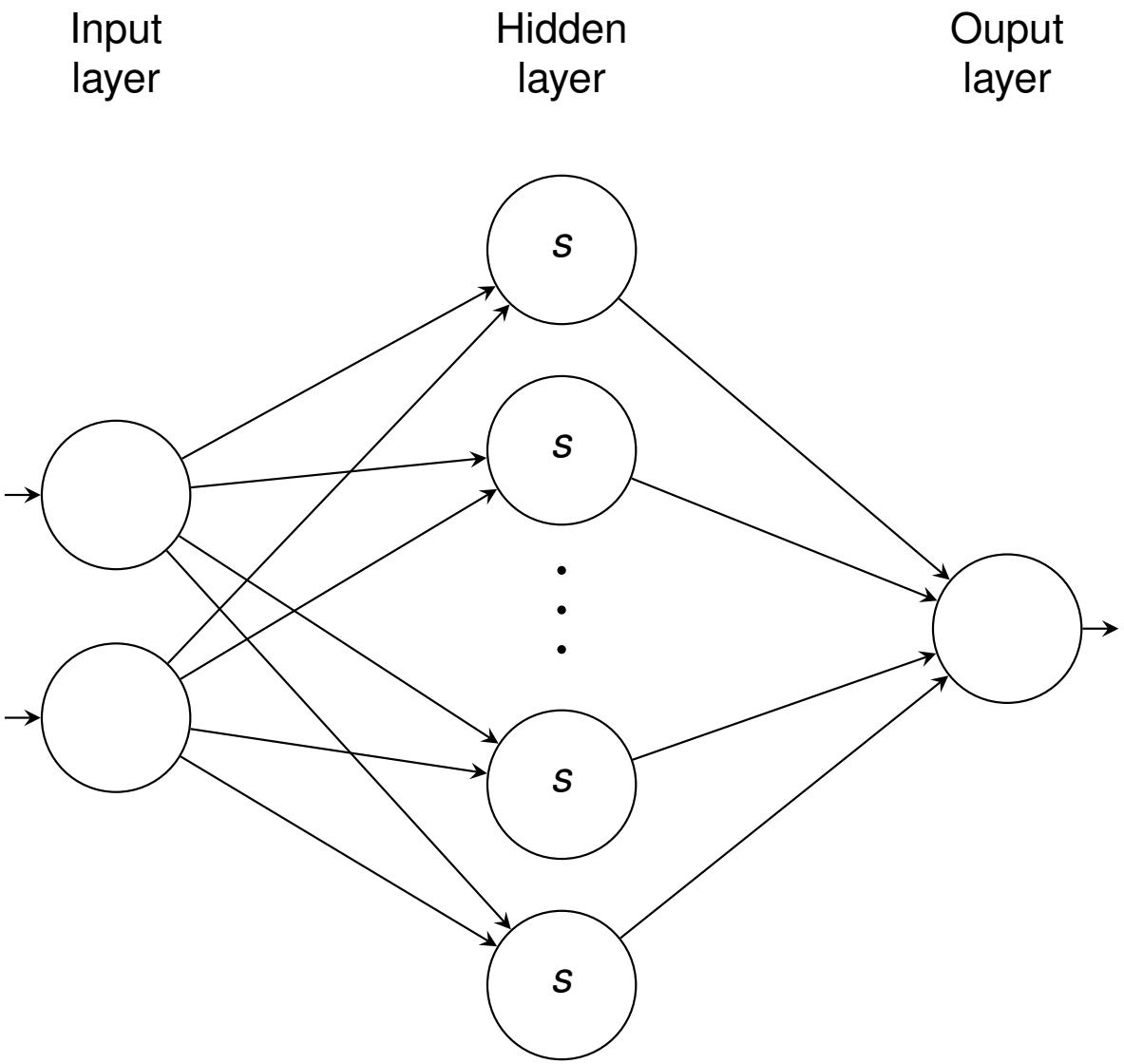
Two dimensional input - two rectangles



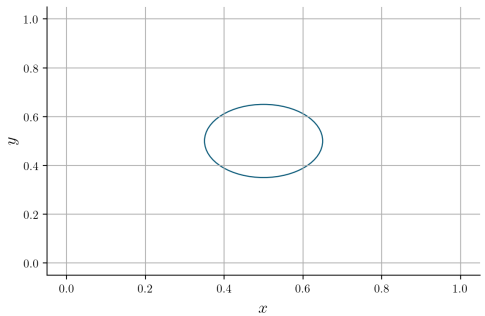
Corresponding area :



Two dimensional input - circle



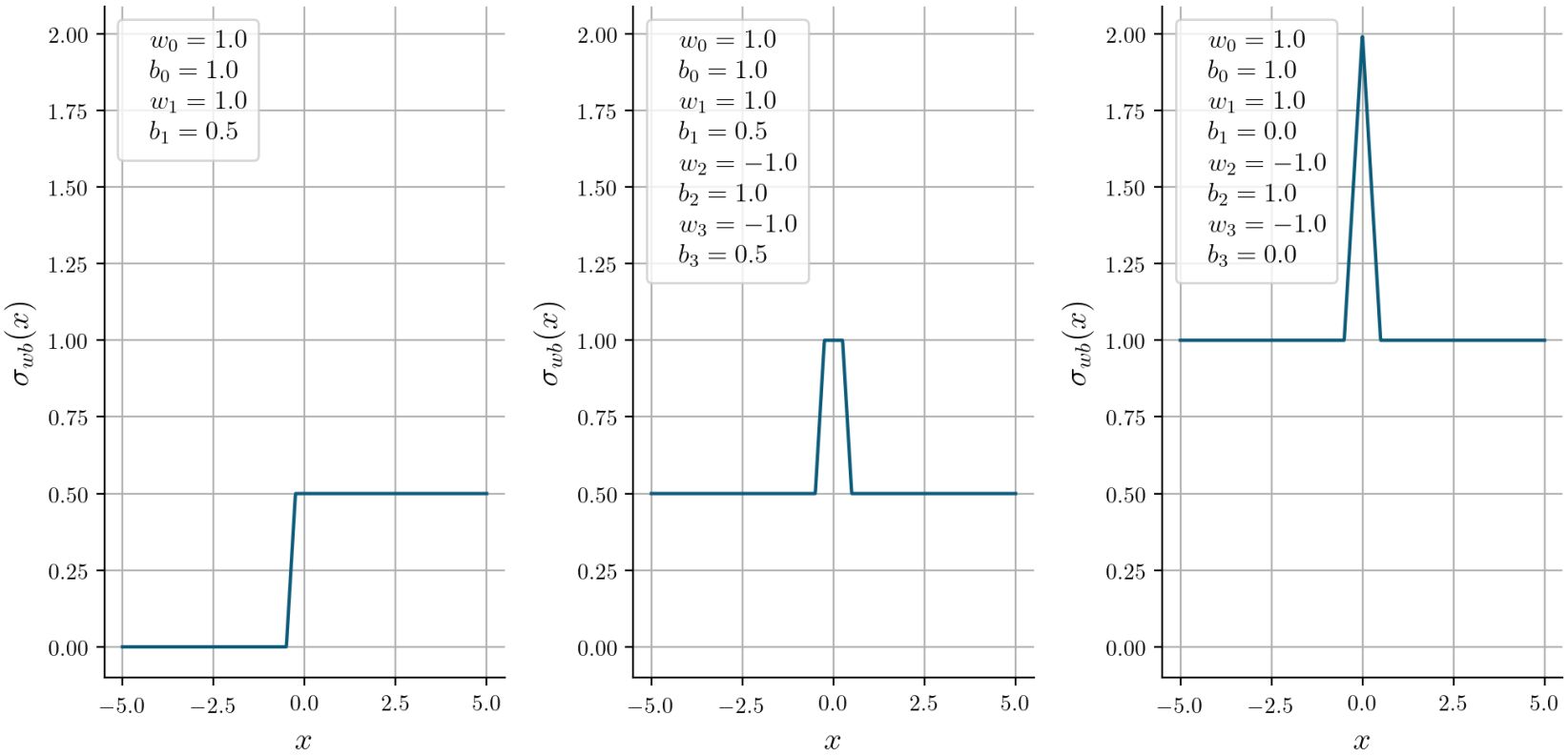
Corresponding area :



Towers built out of ReLU functions

ReLU function :

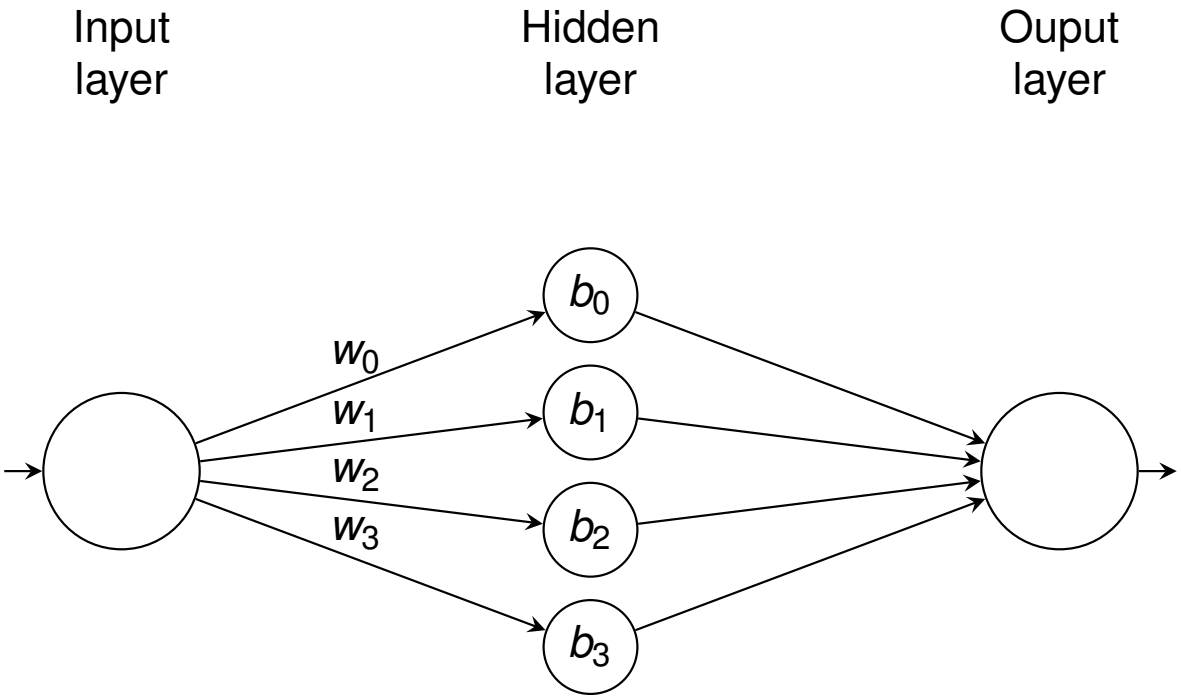
$$\sigma(z) = \begin{cases} 0 & : z < 0 \\ z & : z \geq 0 \end{cases} \tag{41}$$



Towers built out of ReLU functions

ReLU function :

$$\sigma(z) = \begin{cases} 0 & : z < 0 \\ z & : z \geq 0 \end{cases} \tag{42}$$



$\mathcal{G}$ is dense in $\mathcal{C}(\mathcal{I}_n)$

proof by contradiction

Assumption $\mathcal{G} \subset \mathcal{C}(\mathcal{I}_n)$ is not dense in $\mathcal{C}(\mathcal{I}_n)$.

$\Leftrightarrow$ $\overline{\mathcal{G}} \subset \mathcal{C}(\mathcal{I}_n)$ proper closed subset.

Hahn-Banach $\exists$ bounded linear functional $\mathcal{L}$ on $\mathcal{C}(\mathcal{I}_n)$, with

$$\mathcal{L} \neq 0 \quad \wedge \quad \mathcal{L}(\overline{\mathcal{G}}) = 0. \tag{43}$$

Riesz Repr. $\mathcal{L}$ has a unique representation

$$\mathcal{L}(h) = \int_{\mathcal{I}_n} h(\boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{x}) \tag{44}$$

with measure $\mu \in M(I_n)$ and $\forall h \in \mathcal{C}(\mathcal{I}_n)$.

In particular $h(\boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + b) \in \overline{\mathcal{G}}$,

$$\mathcal{L}(h) \overset{\text{(Gl.44)}}{=} \int_{\mathcal{I}_n} \sigma(\boldsymbol{w}^T \boldsymbol{x} + b) \mathrm{d}\mu(\boldsymbol{x}) \overset{\text{(Gl.43)}}{=} 0.$$

$\Rightarrow$ $\mu = 0$ and therefore $\mathcal{L}(h) = 0 \;\; \forall h \in \mathcal{C}(\mathcal{I}_n)$.

$\perp$ **Contradiction to the assumption** $\overline{\mathcal{G}} \subset \mathcal{C}(\mathcal{I}_n)$. $\square$

Proof - Lemma I

$\longrightarrow$ Let $\sigma(\boldsymbol{w}^T\boldsymbol{x} + b) =: \sigma(z)$ be *sigmoidal*, i.e.

rewrite $\sigma(z) \to \sigma(\lambda z + \varphi)$ with $\lambda, \varphi \in \mathbb{R}$,

$$\sigma(\lambda z + \varphi) \to \begin{cases} 1 & \text{if } z > 0 \ \wedge \ \lambda \to \infty, \\ 0 & \text{if } z < 0 \ \wedge \ \lambda \to \infty, \\ \sigma(\varphi) & \text{if } z = 0 \quad \forall \lambda. \end{cases} \tag{45}$$

define

$$\gamma(z, \varphi) := \lim_{\lambda \to \infty} \sigma(\lambda z + \varphi) = \begin{cases} 1 & \text{if } z > 0, \\ 0 & \text{if } z < 0, \\ \sigma(\varphi) & \text{if } z = 0. \end{cases} \tag{46}$$

Lebesgue Bounded Convergence $\Pi_{\boldsymbol{w},b} := \{\boldsymbol{x} | \boldsymbol{w}^T\boldsymbol{x} + b = 0\}$ and the half-space $\mathcal{H}_{\boldsymbol{w},b} := \{\boldsymbol{x} | \boldsymbol{w}^T\boldsymbol{x} + b > 0\}$.

$$\begin{aligned} 0 &= \int_{I_n} \sigma(\boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{x}) \\ &= \int_{I_n} \gamma(\boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{x}) \\ &= \sigma(\varphi)\mu\left(\Pi_{\boldsymbol{w},b}\right) + \mu\left(\mathcal{H}_{\boldsymbol{w},b}\right) \end{aligned}$$

Proof

Left to prove $\sigma(\varphi)\mu(\Pi_{\boldsymbol{w},b}) + \mu(\mathcal{H}_{\boldsymbol{w},b}) = 0 \Rightarrow \mu = 0.$

Let h bounded measurable function, F linear functional,

$$F(h) = \int_{I_n} h(\boldsymbol{w}^T\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{x}). \tag{47}$$

Let $h(\boldsymbol{w}^T\boldsymbol{x}) = \theta(b)$ Heaviside-Function

$$\Rightarrow F(h) = \int_{I_n} h(\boldsymbol{w}^T\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{x}) = \mu\left(\Pi_{\boldsymbol{w},-b}\right) + \mu\left(\mathcal{H}_{\boldsymbol{w},-b}\right) \overset{*}{=} 0$$

* obtain any simple function from sums of Heaviside-Functions. Simple Functions are dense in $\mathcal{L}^\infty(\mathbb{R})$, $F(h) is bounden linear functional \Rightarrow F = 0$.

In particular $\sin(\boldsymbol{\omega}\boldsymbol{x}), \cos(\boldsymbol{\omega}\boldsymbol{x}) \in \mathcal{L}^\infty(\mathbb{R})$,

$$\int_{I_n} (\cos(\boldsymbol{\omega}\boldsymbol{x}) + i\sin(\boldsymbol{\omega}\boldsymbol{x}))\,\mathrm{d}\mu(\boldsymbol{x}) = \int_{\mathcal{I}_n} \exp(i\boldsymbol{\omega}\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{x}) = 0$$

$\forall\omega$.

$\Rightarrow$ Fourier transform of μ is $0 \Rightarrow \mu = 0$.

$\Rightarrow$ σ is *discriminatory*. $\square$