Universal approximation theorem of Neural Networks

Ştefan Cobeli

Technische Universität Berlin

July 24, 2017

Overview

1. Activation functions
2. Universal Approximation Theorem
   - Three-layer networks
   - Two-layer networks
3. Final remarks

Activation functions

Function	Formula	Derivative
sigmoid	$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$	$f^{'}(x) = f(x)(1 - f(x))$
tanh	$f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f^{'}(x) = 1 - f(x)^2$
linear	$f(x) = x$	$f^{'}(x) = 1$
threshold (step)	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$f^{'}(x) = \begin{cases} 0 & \text{if } x \neq 0 \\ ? & \text{if } x = 0 \end{cases}$

Table: Different activation functions

Activation functions

Link between the activation functions

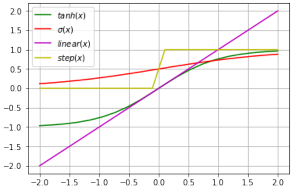
(i) $\tanh(\frac{a}{2}) = 2\sigma(a) - 1$. Therefore a network using tanh activation function has the same capabilities as one using the sigmoid (just different weights and biases).

(ii) The linear function can be obtained from σ by making the input weights small and afterwards scaling them as needed.

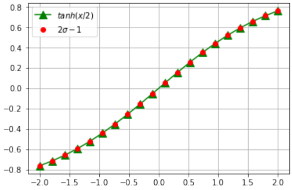
(iii) The sigmoid function can approximate arbitrarily accurately a step function just by making the weights and biases large.

- Thereof, σ has the same theoretical capabilities with all the other activation functions. We will make use of the others just by practical reasons, bearing in mind that they are similar .

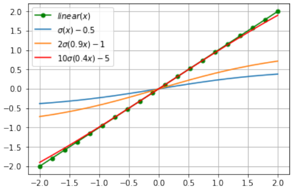
Activation functions



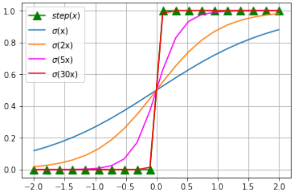
(a) tanh vs lin vs step vs σ.



(b) from σ to *tanh*.



(c) from σ to *linear*.



(d) from σ to *step*.

Figure: Comparison between tanh, linear, step and σ.

Universal Approximation Theorem

Statement

Neural Networks poses universal approximation capabilities.
Let $\varphi(\cdot)$be a nonconstant, bounded, and monotonically-increasing continuous function. Let K be a compact subset $\mathbb{R}^m$. Then the functions of form:

$$F(x) = \sum_{i=1}^{N} v_i \varphi \left(w_i^T x + b_i \right)$$

where $N \in \mathbb{N}$, $v_i, b_i \in \mathbb{R}$ real constants and $w_i \in \mathbb{R}^m$ real vectors, are dense in the space of continuous functions on K. [Wiki-UAT]

Three-layer networks

Goal

Give an intuition on how can any smooth surface in 3D space be approximate by the output of a 3-layer network with sigmoid outputs.

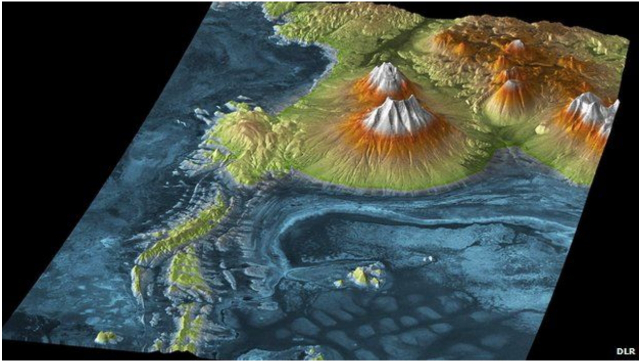


Figure: General 3D smooth surface form.
Salar de Uyuni: The largest salt flats (blue), Bolivia. [BBC-3D-Map]

Three-layer networks

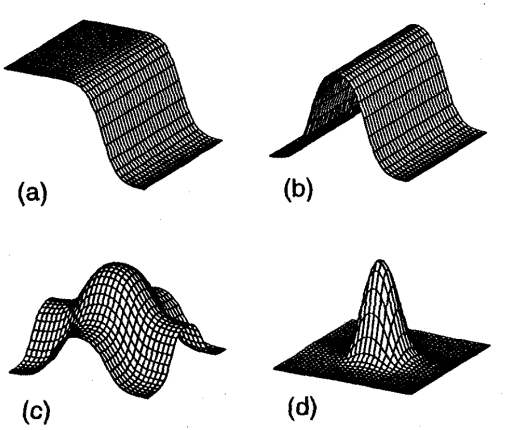


Figure: (a): The output of a single sigmoid unit.
(b): Sum of two sigmoid outputs result in a ridge form.
(c): Sum of multiple ridges.
(d): Normalization of the bumps from (c) with σ. [Bishop, 1995]

Two-layer networks

- Can approximate arbitrarily well any continuous mapping between two finite dimensional spaces (regression problem).
- Therefore can approximate any decision boundary between classes (classification problem).
- Multiple approaches can be made to proof this property.
- Funahashi (1989), Hecht-Nielsen (1989), Cybenko (1989), Hornik et al. (1989), Stinchecombe and White (1989), Cotter (1990), Ito (1991), Hornik (1991) and Kreinovich (1991).
- We will present the proof developed by Jones, (1990); Blum and Li, (1991).

Two-layer networks

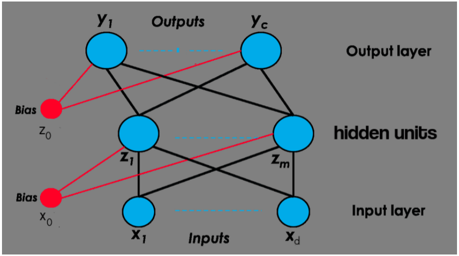


Figure: General Topology of a 2-layer network. [2-layer-NN]

Output of a 2-layer Network

$$y_k(x_1, \dots, x_d) = \tilde{g}\left(\sum_{j=0}^{m} w_{kj}^{(2)} g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right)\right)$$

Two-layer networks Universality Theorem Proof

Step 1

Consider the network has two neurons in the input layer and only one in the output one. So we want to approximate the real function $y(x_1, x_2)$. (this assumption does not restrict the generality of the proof)

Step 2

We take the Fourier decomposition of y in the variable x_2. We obtain the approximation:

$$y(x_1, x_2) \approx \sum_s A_s(x_1) cos(sx_2).$$

Step 3

Further we also decompose the coefficients A_s which are functions of x_1:

$$y(x_1, x_2) \approx \sum_s \sum_l A_{sl} cos(lx_1) cos(sx_2).$$

Two-layer networks Universality Theorem Proof

Step 4

We use the identity: $\cos\alpha\cos\beta = \frac{1}{2}\cos(\alpha+\beta) + \frac{1}{2}\cos(\alpha-\beta)$ in order to write the above form as a linear combination of cosines.

$$y(x_1, x_2) \approx \frac{1}{4}\sum_s \sum_l A_{sl} cos(z_{sl}) + cos(z_{sl}^{'}).$$

Where $z_{sl} = lx_1 + sx_2$ and $z_{sl}^{'} = lx_1 - sx_2$.

Two-layer networks Universality Theorem Proof

Step 5

We make the observation that the function $\cos(z)$ can be approximate with a sum of threshold functions as follows:

$$\cos(z) \approx f_0 + \sum_{i=0}^{N} \{f_{i+1} - f_i\} H(z - z_i).$$

Where f_i are step functions and $H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$.

Step 6

Combining steps 4 & 5 we conclude that $y(x_1, x_2)$ can be written as a linear combination of step functions with arguments linear combinations of x_1 and x_2.

Conclusion

The function $y(x_1, x_2)$ is approximable by a 2-layer network with threshold hidden units and linear output units. □

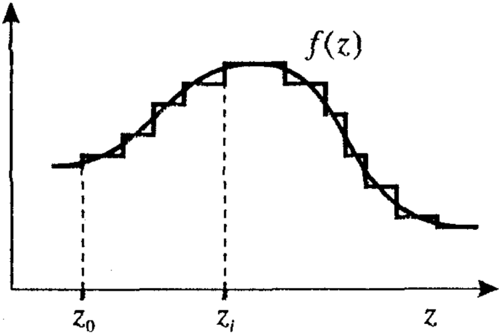


Figure: Aproximation of a continious function with step functions. [Bishop, 1995]

Two-layer networks Universality Theorem Proof

Final Observations

- This constructive approximation loses information about the derivative of the function.
- The derivative of the new obtained function is 0.
- A proof that preserves also the derivative of the function was given by Hornik et al. (1990).

Final remarks

- These were just existence proofs.
- Is there any reason to use other types of network topologies?
- Nothing about how to found the optimal weights.

References

Bishop, Christopher M. (1995)
Neural networks for pattern recognition.
Oxford university press Nov(23), 126 – 132.

Wikipedia Article on the *Universal approximation theorem*
https://en.wikipedia.org/wiki/Universal_approximation_theorem

2-layer network topology base Image
From Researchgate.net
https://goo.gl/TDEA4W

Mapping Earth's surface in 3D
BBC article (18 January 2012, section Science & Environment)
https://goo.gl/KATCAu

The End