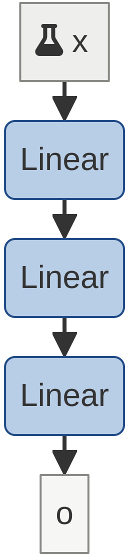
Non-Linearities

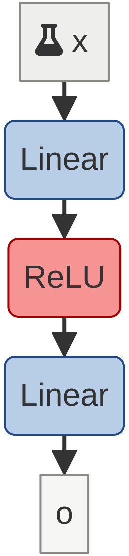
Recap: A Simple Example

Linear models

A linear model cannot distinguish paws from background

Deep Networks



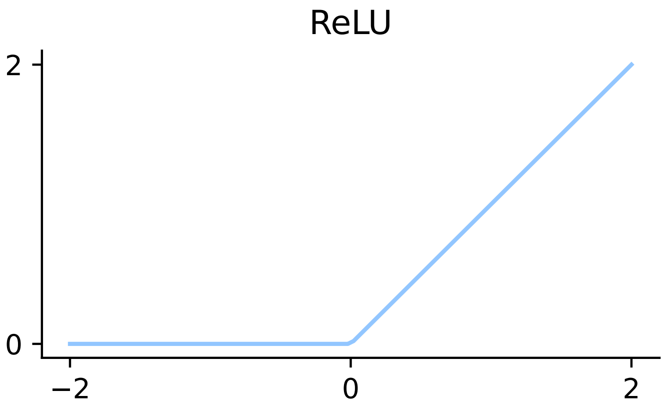


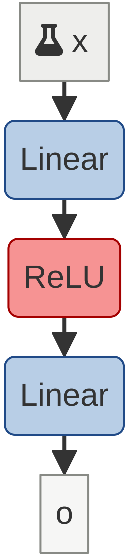
Non-Linearities

Rectified Linear Unit (ReLU)

$$\mathrm{ReLU}(x) = \max(x, 0)$$

Non-linear and differentiable almost everywhere





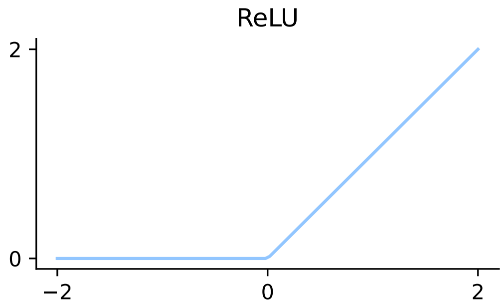
A Simple Example - Why?

Intuition

- first layer learns the color categories of the paw (white, black, grey)
- second layer classifies color as paw or not

How?

$$\mathrm{ReLU}(x) = \max(x, 0)$$



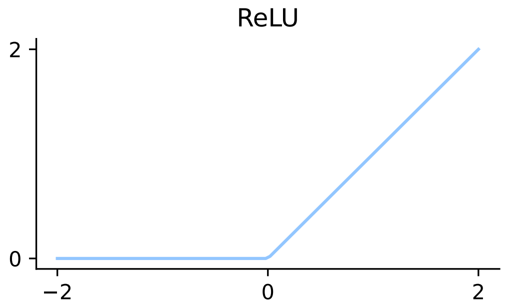
A Simple Example - Why?

Intuition

- first layer learns the color categories of the paw (white, black, grey)
- second layer classifies color as paw or not

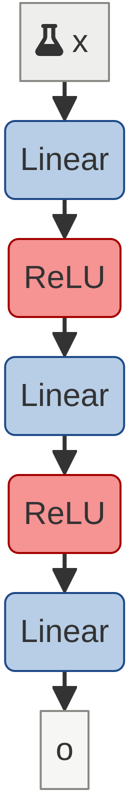
$$f(x) = \mathrm{ReLU}\left(x - \frac{1}{2}\right) + \mathrm{ReLU}\left(\frac{1}{2} - x\right) - \frac{1}{4}$$
$$= \left|x - \frac{1}{2}\right| - \frac{1}{4}$$

$$\mathrm{ReLU}(x) = \max(x, 0)$$



Deep Networks

Alternates linear and non-linear layers



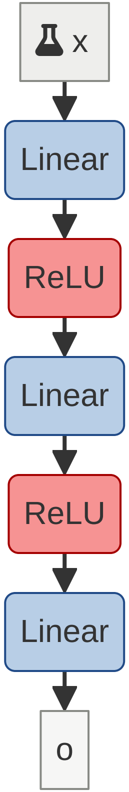
Deep Networks

Model: $f_\theta : \mathbb{R}^n \to \mathbb{R}^c$

Parameters: $\theta = (\mathbf{W}_1, \mathbf{b}_1, ..., \mathbf{W}_N, \mathbf{b}_N)$

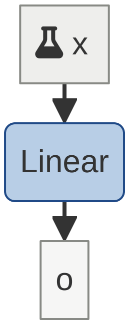
Computation:

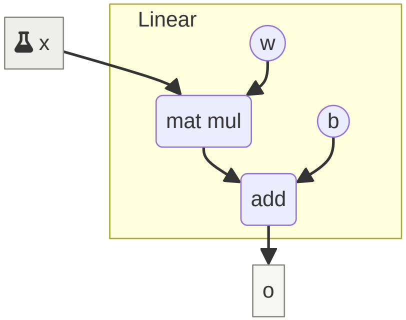
$$\begin{aligned} \mathbf{h}_1 &= \mathrm{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \\ \mathbf{h}_2 &= \mathrm{ReLU}(\mathbf{W}_2 \mathbf{h}_1 + \mathbf{b}_2) \\ &\vdots \\ \mathbf{h}_{N-1} &= \mathrm{ReLU}(\mathbf{W}_{N-1} \mathbf{h}_{N-2} + \mathbf{b}_{N-1}) \\ f_\theta(\mathbf{x}) &= \mathbf{W}_N \mathbf{h}_{N-1} + \mathbf{b}_N \end{aligned}$$

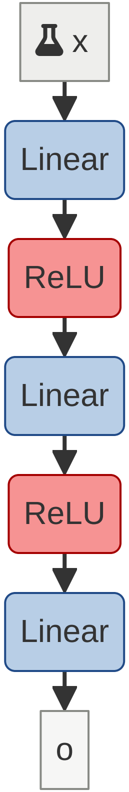


What Is a Layer?

Largest computational unit that remains unchanged throughout different architectures

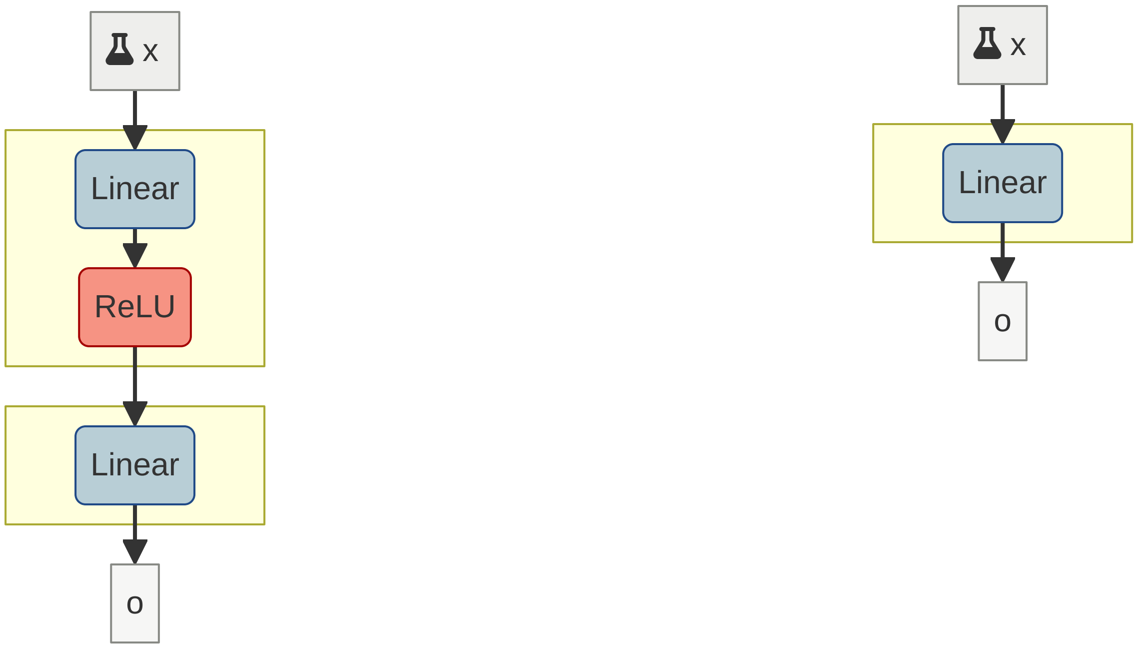






How Many Layers Does a Deep Network Have?

We only count linear layers

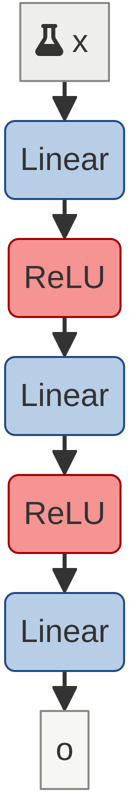


Universal Approximation Theorem

Universal Approximation Theorem

A two-layer deep network can approximate any continuous function.

- Constructing is inefficient
- Deep learning exploit structure in data to find efficient approximations



Approximation by superpositions of a sigmoidal function, Cybenko 1989 ↩️

Non-Linearities - TL;DR

Deep networks are stacks of alternating linear and non-linear layers

Deep networks belong to a class of continuous functions that can approximate **any** continuous function!