Output Representations

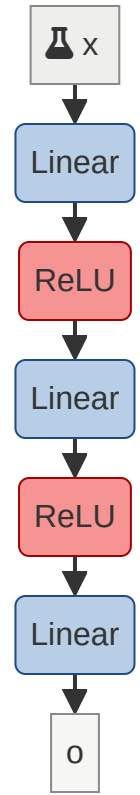
Recap: Deep Networks

Universal Approximation Theorem

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

We might not always want continuous (real-valued) outputs

- How can we convert the real value to what we want?



Inputs and Outputs of Networks

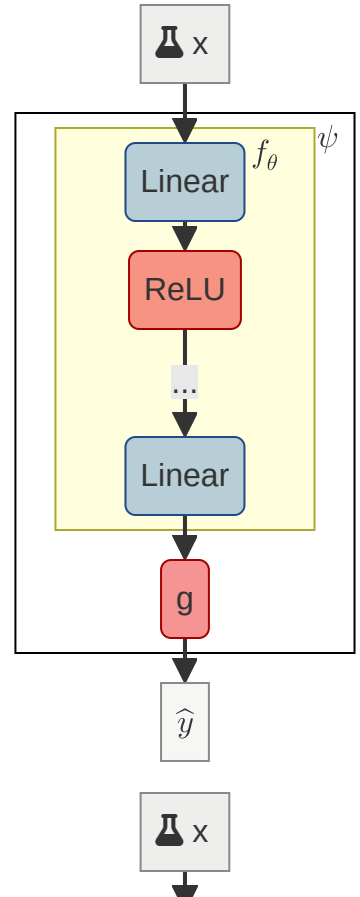
Input: $\mathbf{x} \in \mathbb{R}^n$

Output: $\mathbf{o} = f_\theta(\mathbf{x})$

- f_θ: deep network

Output transformations: g

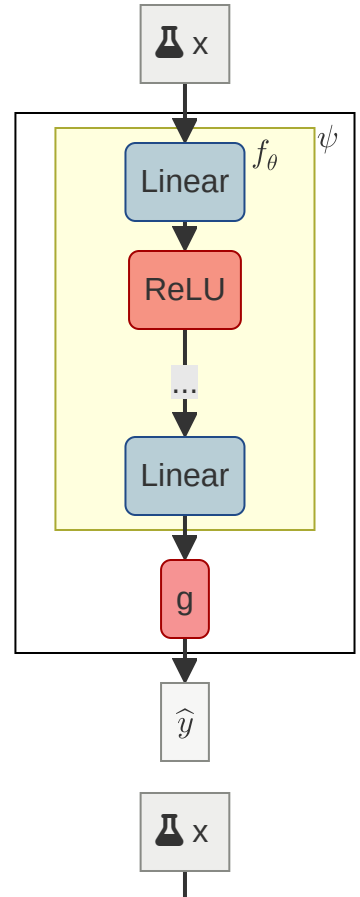
$$\psi : f_\theta \circ g$$



Regression

Regression $\psi : \mathbb{R}^n \to \mathbb{R}$

- Identity mapping: $g(\mathbf{o}) = \mathbf{o}$



Positive Regression

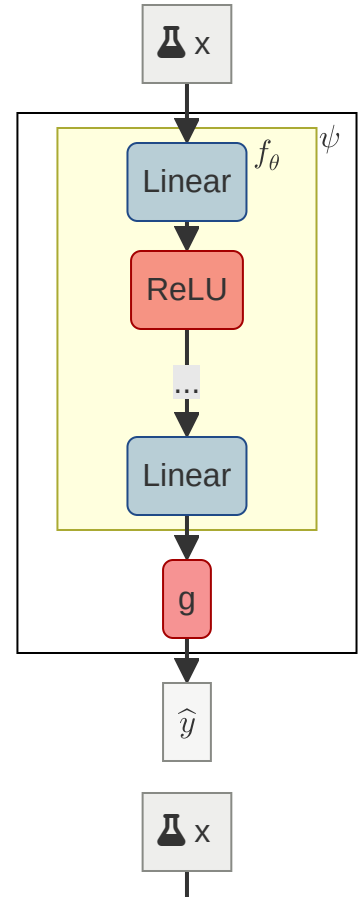
Positive regression $\psi : \mathbb{R}^n \to \mathbb{R}_+$

Option 1: ReLU

- $\hat{\mathbf{y}} = g(\mathbf{o}) = \max(\mathbf{o}, 0)$

Option 2: Soft ReLU

- $\hat{\mathbf{y}} = g(\mathbf{o}) = \log(1 + e^{\mathbf{o}})$



Binary Classification

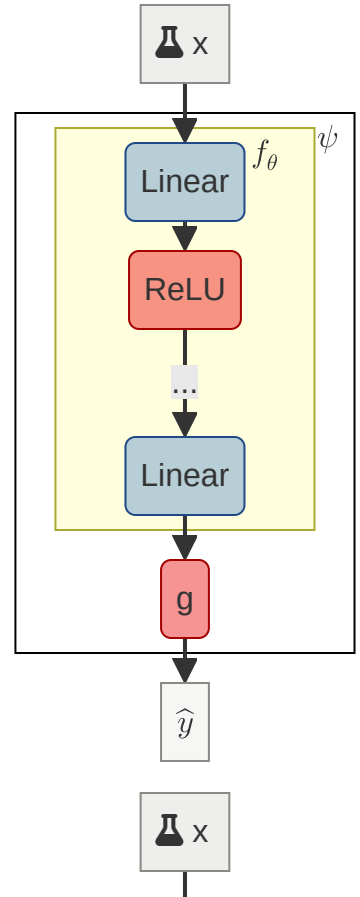
Binary classification $\psi : \mathbb{R}^n \to [0, 1]$

Option 1: Thresholding

- $\hat{\mathbf{y}} = g(\mathbf{o}) = 1\{\mathbf{o} > 0\}$

Option 2: Logistic Regression

- $\hat{\mathbf{y}} = \sigma(\mathbf{o}) = \frac{1}{1+e^{-\mathbf{o}}}$



General Classification

Multi-class classification $\psi : \mathbb{R}^n \rightarrow [1 \dots C]$

Option 1: argmax

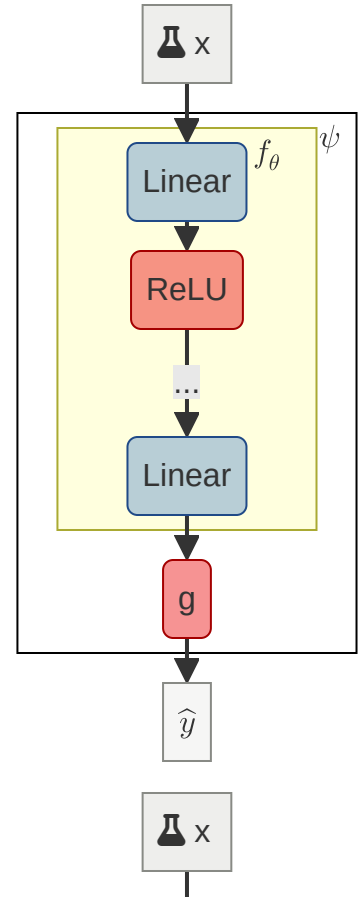
- $\hat{y} = \arg\max(\mathbf{o})$

Option 2: one-hot

- $\hat{\mathbf{y}} = [0, ..., 1, ..., 0]^\top$
- $\hat{\mathbf{y}}_i = 1 \text{ if } \mathbf{o}_i \geq \mathbf{o}_j \forall j$

Option 3: softmax

- $p(y) = \text{softmax}(\mathbf{o})$



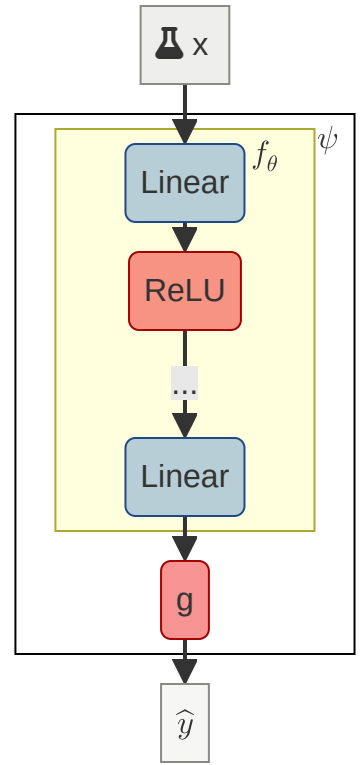
Output Representations in Practice

Do **not** add to model

- Most output transformations are not differentiable (or hard to differentiate)
- Model cannot train with them

Model output

Always output raw values



Output Representations - TL;DR

Deep networks always output real values

Output transformations convert them into what you want

Train the network *without* output transformations!