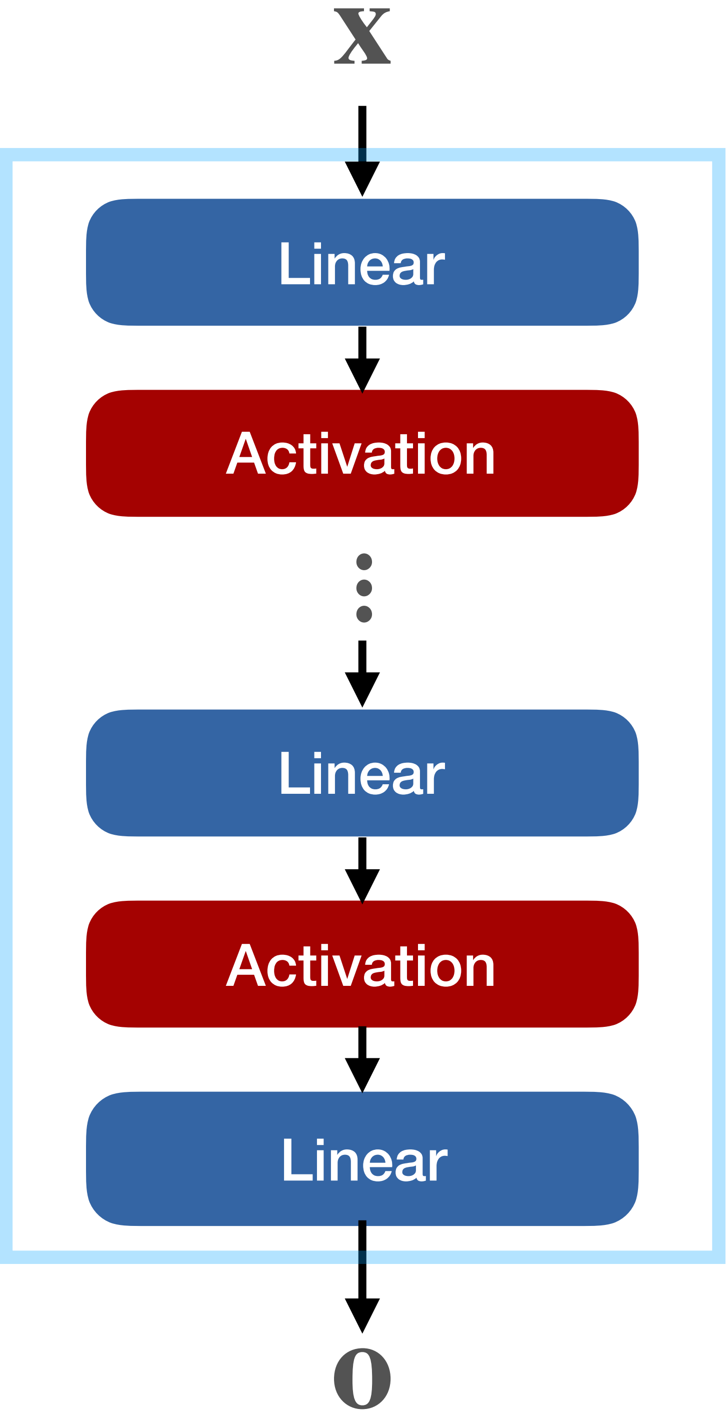
Output representations

Inputs and outputs of networks

$f_\theta : X \rightarrow O$

- Input:
  - Tensor $\mathbf{x}$
- Output:
  - Tensor $\mathbf{o}$



$\mathbf{x}$

Linear

Activation

⋮

Linear

Activation

Linear

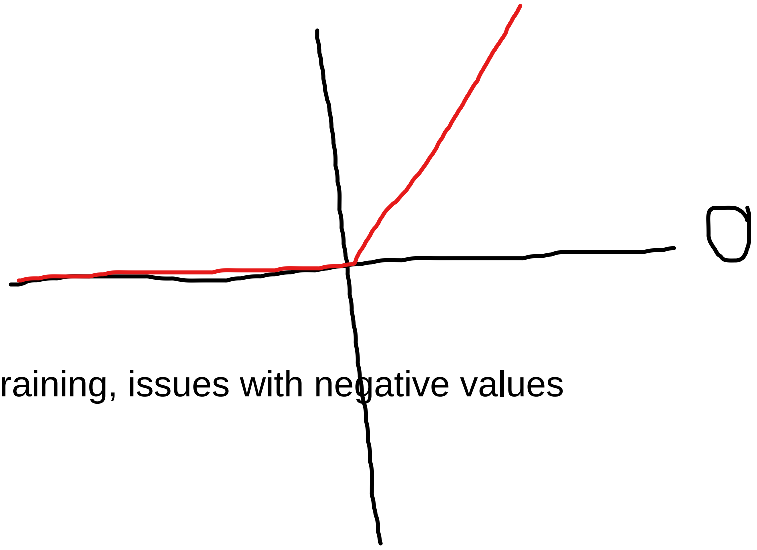
$\mathbf{o}$

Regression

- vanilla tensor $\hat{\mathbf{y}} = \mathbf{o}$

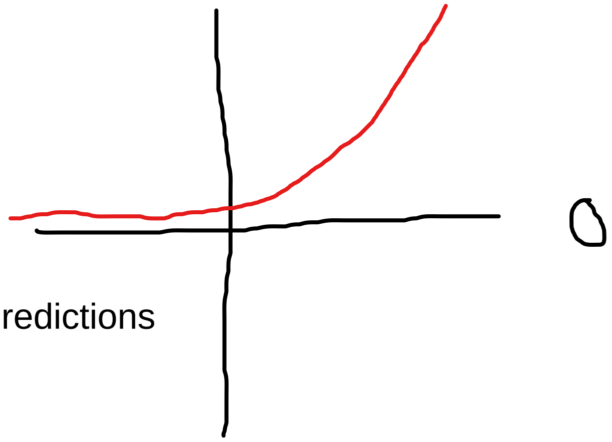
Positive regression

- Option 1: ReLU
  - $\hat{\mathbf{y}} = \max(\mathbf{o}, 0)$



Don't use ReLU for regression training, issues with negative values

- Option 2: Soft ReLU
  - $\hat{\mathbf{y}} = \log(1 + e^{\mathbf{o}})$



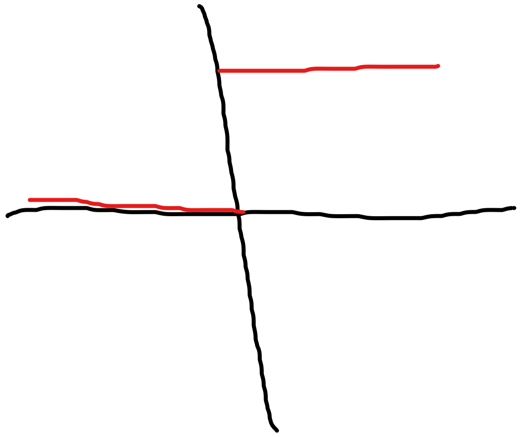
This function can recover from negative predictions

Binary Classification

- Option 1: Thresholding
  - $\hat{\mathbf{y}} = \mathbf{o} > 0$

hard to train, the gradient itself of this function is going to be 0 everywhere.

Does not give you a signal on how to fit the network better



- Option 2: Logistic Regression
  - $p(1) = \sigma(\mathbf{o})$

General Classification

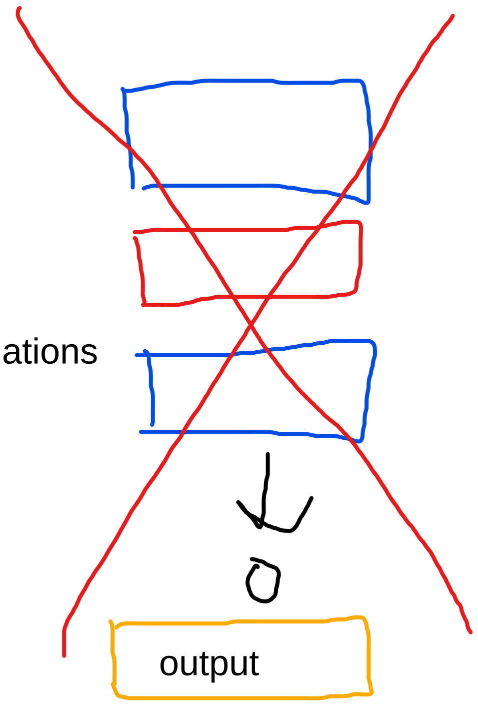
- Output more values, one per class
- Option 1: argmax
    - $\hat{y} = \text{argmax}_i \mathbf{o}_i$

can not train the network, can not compute the gradient with this function

- Option 2: softmax
    - $p(y) = \text{softmax}(\mathbf{o})_y$

Output representations in practice

- Do not add into model
- Always output raw values



Never add an output transformation
inside your network because output transformations
are hard to differentiate through.