Week 5
Different Types of Activation Units

Team Placeholder

Universal function approximation

- Non-linear function enables approximation to any function.
  NN with on-linear activations can represent any function!

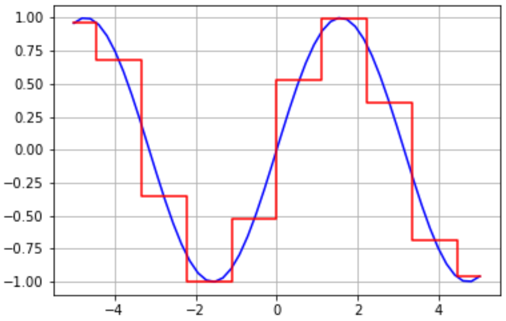
- Cascading linear layers is equivalent to using only one layer in the meaning of representational capacity.
  NN with linear activations can represent linear functions only!

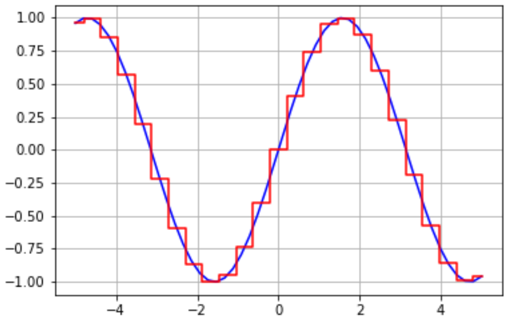
Universal function approximation

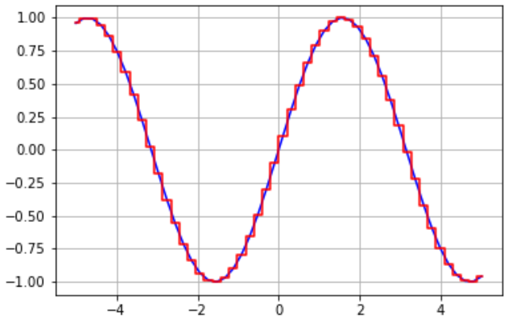
- **Non-linear function enables approximation to any function**

- Example: Unit step function

  Problem: The gradient of the unit step function is zero everywhere

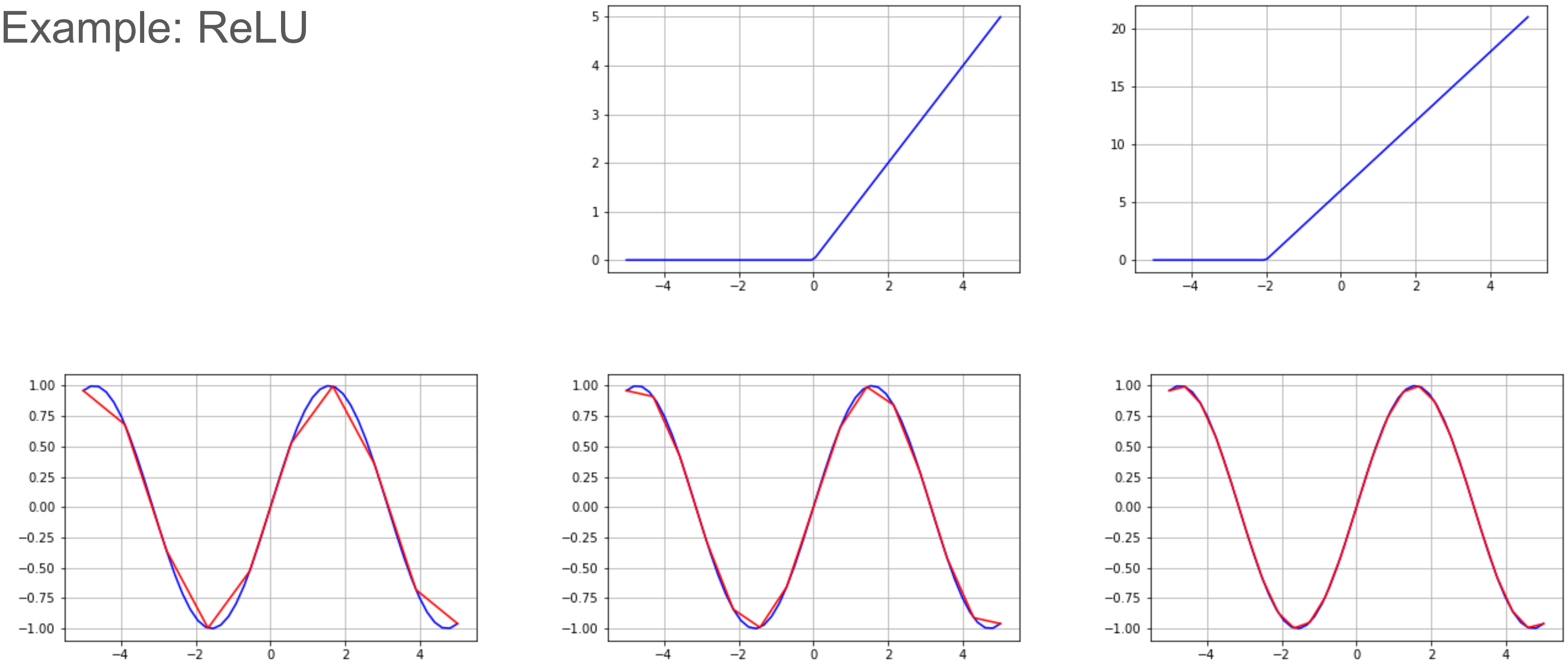






Universal function approximation

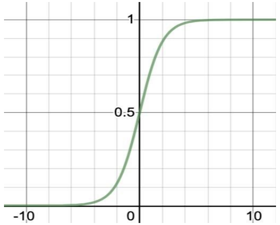
Example: ReLU



Comparison of Different Activation Units

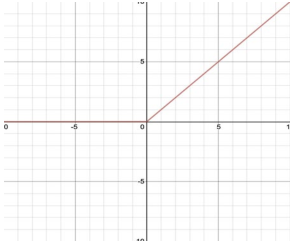
- Sigmoid

$$f(z) = \frac{1}{1 + e^{-z}}$$



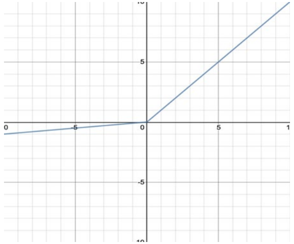
- ReLU

$$f(z) = \begin{cases} z & if \quad z \geq 0, \\ 0 & else. \end{cases}$$



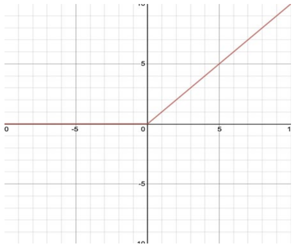
- Parametric ReLU

$$f(z) = \begin{cases} z & if \quad z \geq 0, \\ az & else. \end{cases}$$

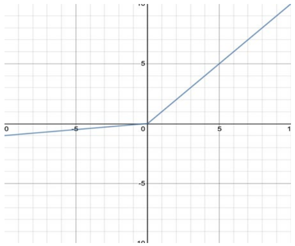


ReLU

- ReLU doesn’t saturate when z approaches infinity
    - First derivative has constant 1 when z > 0
    - Less likely to have vanishing gradient
- ReLU dies when z < 0
    - Doesn’t solve the vanishing gradient problem in the z<0 region



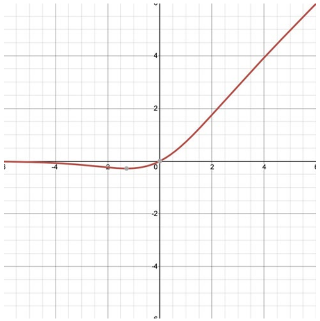
- Parametric ReLU
    - First derivative has non zero value everywhere
    - Solves the dying ReLU problem



Swish and Mish

- Latest state of the art activation functions
- Swish

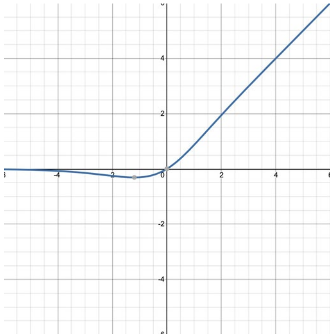
$$f(z) = \frac{z}{1 + e^{-\beta z}}$$



1. Unbounded above, bounded below
2. Non-monotonicity
3. Differentiable everywhere

- Mish -- Improve upon swish

$$f(z) = z \tanh\left(\ln\left(1 + e^{z}\right)\right)$$



1. First derivative is preconditioned

Gradient Vanishing/Exploding

- Recall back-propagation

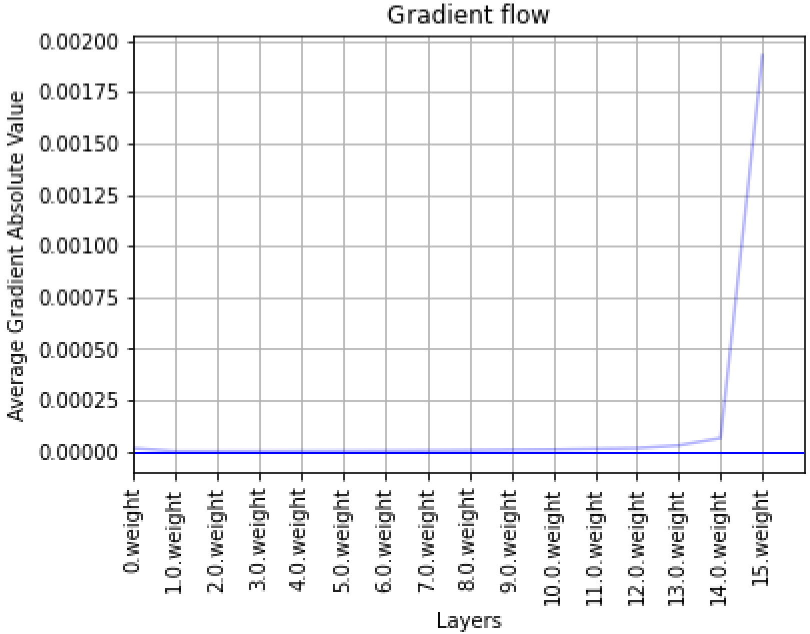
$$\nabla_{\mathbf{W}^{(l)}}\mathcal{L} = \delta^{(l)}\mathbf{a}^{(l-1)T}$$

$$\begin{aligned}\delta^{(l)} &= \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(l)T}} \\ &= \mathrm{diag}(\sigma'(\mathbf{z}^{(l)}))\mathbf{W}^{(l+1)T}\mathrm{diag}(\sigma'(\mathbf{z}^{(l+1)}))\cdots\mathbf{W}^{(L)T}\mathrm{diag}(\sigma'(\mathbf{z}^{(L)}))\frac{\partial \mathcal{L}}{\partial \mathbf{y}^T}\end{aligned}$$

- Gradients are **proportional to** the **multiplication of derivatives of activation functions** and **weight matrices** in the following layers

Gradient Vanishing Example

- A 16-layer MLP with sigmoid activations.
  Absolute value of elements in gradients w.r.t weights drops to **0** quickly during the back-propagation!
  The gradients vanished.



Gradient Vanishing/Exploding

- Happens when
  - the there are too many cascaded layers (0.9^**100** ≈ 0 or 1.1^**100** ≈ ∞)
  - the model is poorly initialized (**0.1**^10 ≈ 0 or **10.0**^10 ≈ ∞)
  - nonlinear functions are inappropriate (**0.1**^10 ≈ 0 or **10.0**^10 ≈ ∞)
  - activations / inputs are inappropriate (**0.1**^10 ≈ 0 or **10.0**^10 ≈ ∞)
  - …
- Solve it from the source
  - use fewer layers
  - design suitable initialization methods (Xavier, Kaiming, etc)
  - change activation units
  - batch normalization
  - ...