Activation Functions

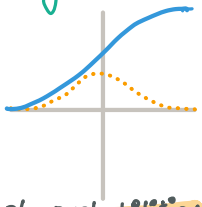
- non-linear
- unbounded
- continuously differentiable

- Output layer
- Hidden layer

Sigmoid <1990's

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

$f'(x) = f(x)(1-f(x))$



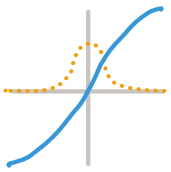
Use
* Logistic Regression
* binary classification

* o/p probabilities b/w 0 and 1
* differentiable
* derivative b/w -3 and +3 vanishing grad
* exponential computation is costly
* saturates quickly

tanh 1990-2000

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

$f'(x) = 1-f(x)^2$



* can be used in hidden layer (RNN)

* zero centered b/w -1 and +1 enabling faster learning, differentiable
* gradient steeper than sigmoid
* exponential is costly
* vanishing gradient, saturates quickly

Softmax 2000

$f(x) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$

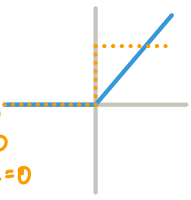
* extension of sigmoid
* sum of probabilities in o/p equals to 1
* Perfect for multi-nominal multi-class problems
* should only be used in output layer

Rectified Linear Unit (RELU) >2010

most widely used

$f(x) = \max(0, x)$

$f'(x) = \begin{cases} 0, & x<0 \\ 1, & x>0 \\ \text{Undef}, & x=0 \end{cases}$

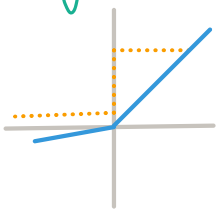


Pros
* inbuit regularization (-ve neurons turned off)
* Unboundedness accelerated learning
* computationally simple

Cons
* Dead neurons that never activates (-ve neuron turn off)
* not continuously differentiable, but works well with low learn rate and large negative bias

Improvements to ReLU

Leaky ReLU >2011

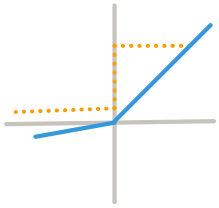


$f(x) = \max(0.01x, x)$

$f'(x) = \begin{cases} 0.01, & x<0 \\ 1, & x>0 \\ \text{Undef}, & x=0 \end{cases}$

* avoid dead neurons allowing back propogation for negative values, albeit at a slower rate
* may not be universaly better than ReLU
* better suited for tasks involving sparse gradients like GAN

Parametric ReLU >2011



$f(x) = \max(ax, x)$

$f'(x) = \begin{cases} a, & x<0 \\ 1, & x>0 \\ \text{Undef}, & x=0 \end{cases}$

* generalization of Leaky ReLU where the scalar multiple is learnt from training
* very sensitive to the scalar value 'a' and performs differently for different problems
* alternative when Leaky ReLU doesn't fix dead neuron problem

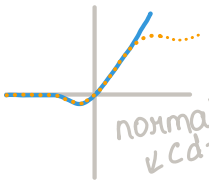
Exponential Linear Unit (ELU) >2015

$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^{x}-1), & x<0 \end{cases}$

$f'(x) = \begin{cases} 1, & x \geq 0 \\ f(x)+\alpha, & x<0 \end{cases}$

* negative value slowly smoothes unlike ReLU avoiding dead neurons
* exponential is expensive
* exploding gradient for bad initializations
* learning of alpha needed

Guassian Error Linear Unit (GELU) >2018



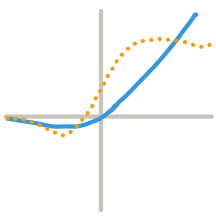
normalized ↙ cdf

$f(x) = x\,\phi(x)$

$f'(x) = \phi(x) + x\,\phi(x)$

* combines dropout (zeroing out neurons) randomly & zoneout (maintain previous values) and ReLU
* weights inputs by percentile rather than gates, leading to a smoothen version of ReLU
* Used in GPT-3 BERT and other transformers
* better than ReLU for computer vision, NLP and speech recog

Swish > 2017



$f(x) = x \cdot \text{sigmoid}(x)$

$f'(x) = \text{sigmoid}(x) + x \cdot \text{sigmoid}(x) * (1 - \text{sigmoid}(x))$

* large negative values have a derivative of 0 which helps in sparsity and small negative values are still relevant
* drop in replacement for relu. outperforms relu in image classification and machine translation

- Adiamaan Keerthi