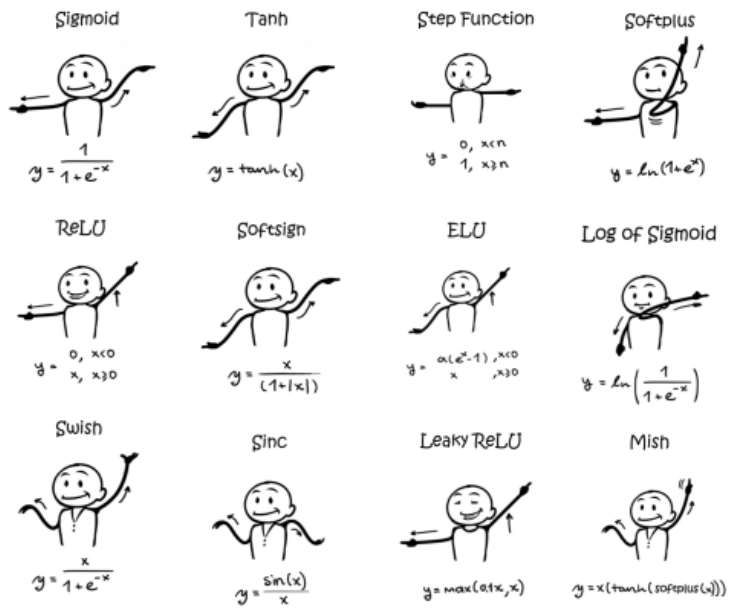
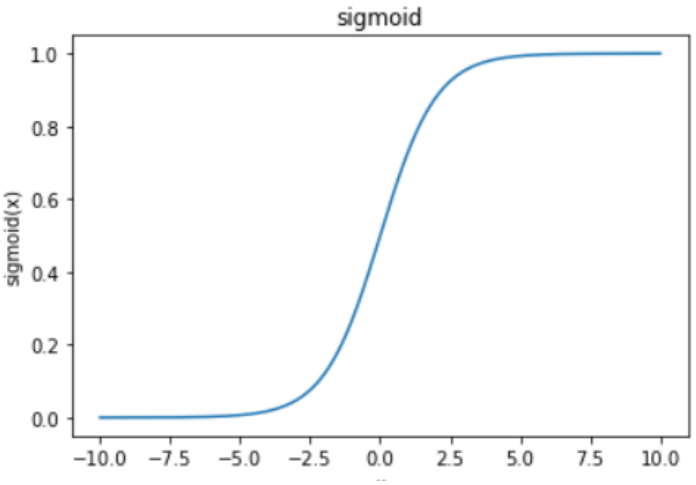
Activation Function



1. **Sigmoid, tanh**



Sigmoid

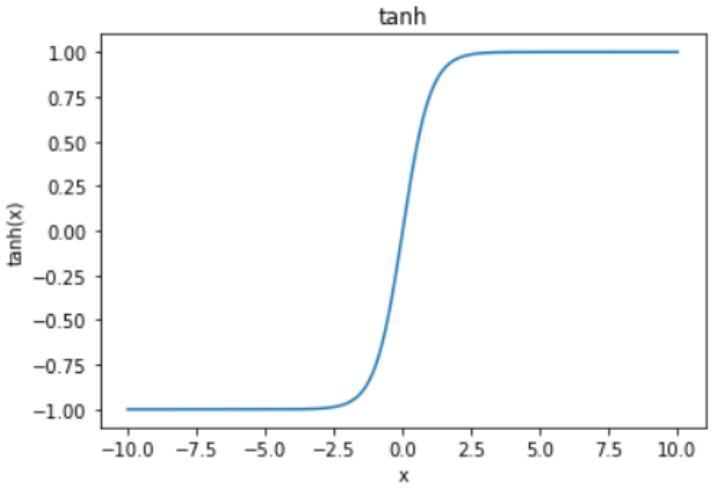
$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Range: (0,1)

Linear (Exponential)

Problem: **Vanishing Gradient**, Not Zero Centered -> **Slow Convergence (Zig-Zag)**

Recent: Output Layer(Binary Class)



Hyperbolic tangent $\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

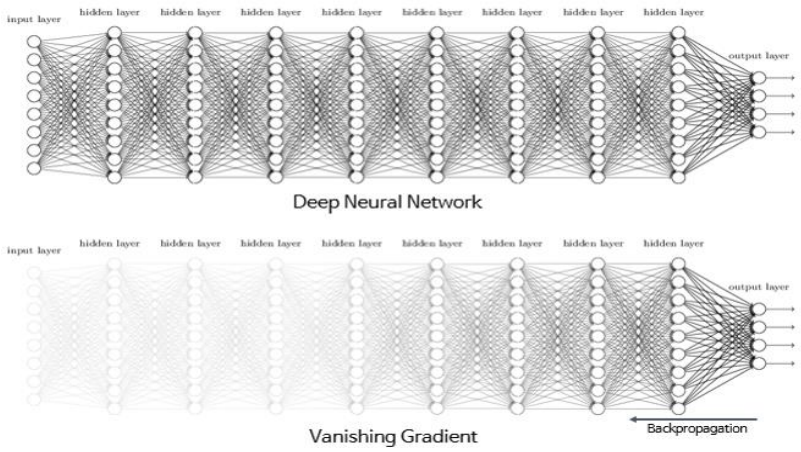
Range:(-1,1)

Linear (Exponential)**, Zero Centered**!

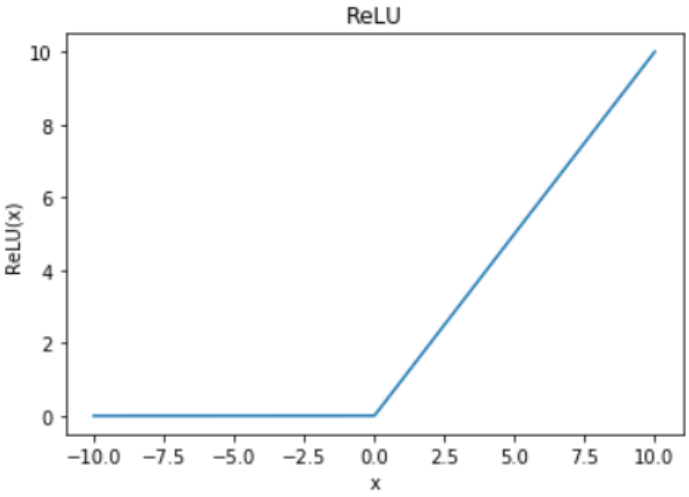
Problem: **Vanishing Gradient**

Recent: Output Layer(Binary Class), **Hidden Layer**(Better than Sigmoid but be careful!)

2. Vanishing Gradient



3. Relu



Rectifier, ReLU (Rectified Linear Unit) $\phi(z) = max(0, z)$

Range: [0, ∞]

Non-Linear

Solved Vanishing Gradient!

Learning Speed(Easy Calculation)

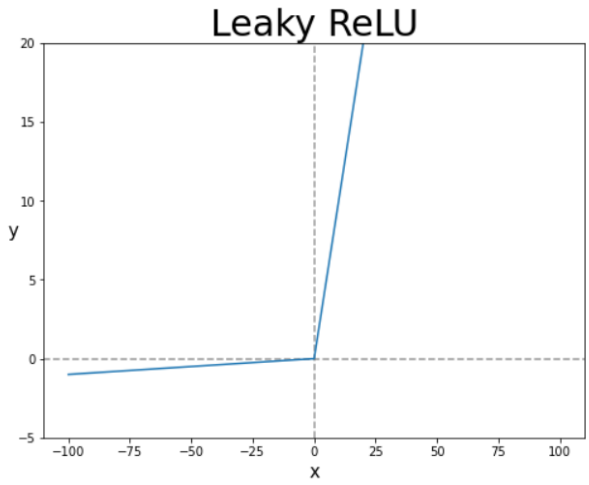
Recent: **Hidden Layer(Major)**

4. Limitation(Problem) of Relu

- Negative Input => 0 Output, 0 Gradient(**Dying Relu)**
  ※But Sometimes, dying Relu make artificial neural network more effective(similar to drop-out)
- (Only Effective in Hidden Layer)
- **Zig-Zag**(Similar to Sigmoid)
- **Cannot differentiate at zero.**

5. Relu Family

- **Leaky Relu(LRelu)& Parameter Relu(PRelu)**

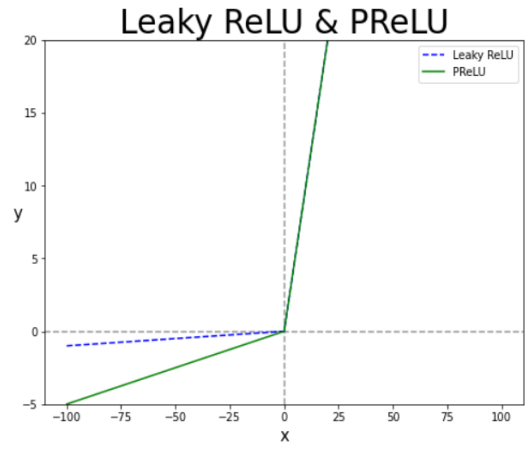


$$\text{LeakyReLU}_\alpha(x) = \max(\alpha x, x)$$

Purpose: Don't ignore negative input!

Method: Multiply alpha(normally 0.01) on negative input

Limit: Linear on negative -> **Cannot use on complicate classification**, Poor Performance(worse than Sigmoid or Tanh)

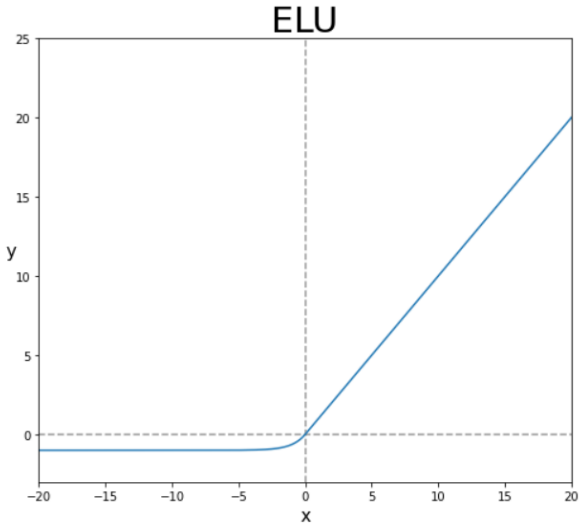


Purpose: Use alpha as weight!

Method: Update alpha by backpropagation

Limit: Same as LRelu, Overfitting Risk

- **Exponential Linear unit(ELU)**



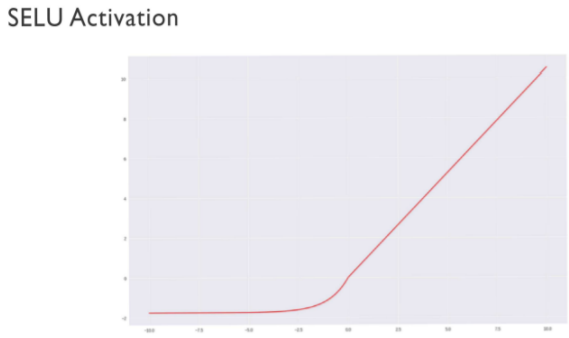
$$f(x) = \begin{cases} x & (x > 0) \\ \alpha(e^x - 1) & (x \leq 0) \end{cases}$$

Purpose: **Make differentiable at Zero!** & **Print negative output!**

Method: Use Exponential on negative input

Limit: No significant increase on performance, low learning speed(exponential calculation)

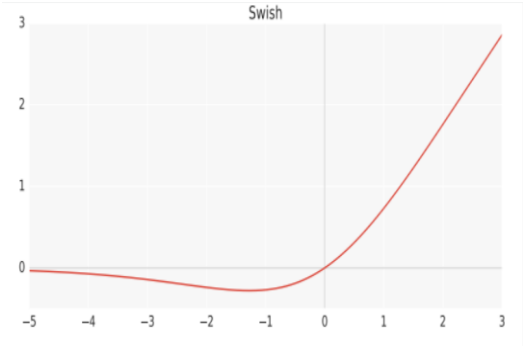
- **Scaled ELU(SELU)**



$$\operatorname{selu}(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leqslant 0 \end{cases}$$

Purpose: ELU + Self-Normalizing(Similar to PRelu)

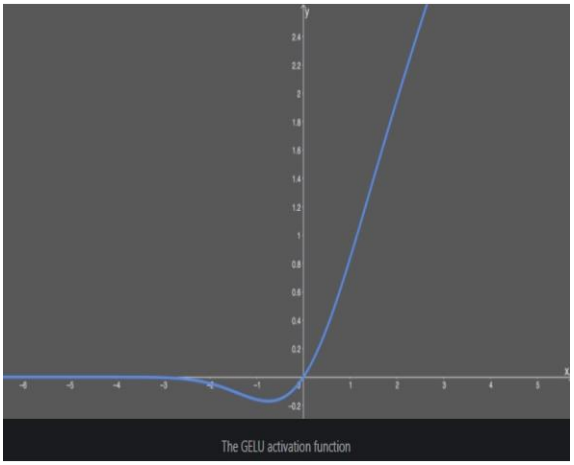
- **SWISH**



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

Purpose: Multiply x on sigmoid

- **GELU**



$$\mathrm{GELU}(x) = 0.5x\left(1+\tanh\left(\sqrt{2/\pi}(x+0.044715x^3)\right)\right)$$

Purpose:

Recent: Best above all!

그림 출처

https://enjoyso.tistory.com/119 (Activation Function)

http://rasbt.github.io/mlxtend/user_guide/general_concepts/activation-functions/ (Activation Function)

https://excelsior-cjh.tistory.com/177 (Vanishing Gradient)

참고 자료

https://gooopy.tistory.com/55 (Relu)

https://gooopy.tistory.com/56?category=824281 (Relu Family)

https://gooopy.tistory.com/52?category=824281 (Sigmoid)

https://gooopy.tistory.com/54?category=824281 (Hyperbolic Tangent)

https://brunch.co.kr/@kdh7575070/27 (Dying Relu)

https://medium.com/@neuralnets/swish-activation-function-by-google-53e1ea86f820 (SWISH)