Activation Functions

(Tanh, ReLU, Swish, Mish)

Habib Mbow
Fenosoa Randrianjatovo
Heritiana Daniel Andriasolofo
Santatriniaina Avotra Randrianambinina

African Master of Machine Intelligence (AMMI)
at
African Institute for Mathematical Sciences (AIMS)

April 29, 2022

Contents

Generality on supervised machine learning

In supervised machine learning, we are given a sample data $\{(X_i, y_i)\}_{i=1}^{N}$.

Goal

Find a function F such that

$$y = F(X)$$

for all (X, y) from the population of the given sample data.

Generality on supervised machine learning

In supervised machine learning, we are given a sample data $\{(X_i, y_i)\}_{i=1}^{N}$.

Goal

Find a function F such that

$$y = F(X)$$

for all (X, y) from the population of the given sample data.

Approach

Approximating F by a function f using the sample data points, i.e.

$$f \approx F.$$

The function f is called **predictor function**.

Examples of approximation method

Linear functions are easy to parameterized, and that makes it easy to learn.

Definition (Linear Regression)

The approach of choosing the predictor function to be a *linear* function is called **Linear Regression**.

Examples of approximation method

Linear functions are easy to parameterized, and that makes it easy to learn.

Definition (Linear Regression)

The approach of choosing the predictor function to be a *linear* function is called **Linear Regression**.

One can also write the predictor function as a composition of functions i.e.

$$f = f_k \circ f_{k-1} \circ \cdots \circ f_1,$$

for some functions f_i.

This is the approach we use in Deep Learning ($k \geq 2$).

Neural Network & Activation Functions

Now we will use the approach of writing f as a composition of functions as follows

$$f = (g_k \circ f_k) \circ (g_{k-1} \circ f_{k-1}) \circ \cdots \circ (g_1 \circ f_1).$$

Now we will use the approach of writing f as a composition of functions as follows

$$f = (g_k \circ f_k) \circ (g_{k-1} \circ f_{k-1}) \circ \cdots \circ (g_1 \circ f_1).$$

Neural Network - Activation Function

In neural network, we add the following assumptions :

- f_i's are linear functions,
- g_i's are function that we set ourselves known as **activation functions**.

Neural Network & Activation Functions

Now we will use the approach of writing f as a composition of functions as follows

$$f = (g_k \circ f_k) \circ (g_{k-1} \circ f_{k-1}) \circ \cdots \circ (g_1 \circ f_1).$$

Neural Network - Activation Function

In neural network, we add the following assumptions :

- f_i's are linear functions,
- g_i's are function that we set ourselves known as **activation functions**.

For the approximation, we only adjust the weights of f_i's.

Gradient Descent

Loss function

It is a distance function between f and F.

Gradient Descent

Loss function

It is a distance function between f and F.

A common algorithm to adjust weights of the predictor to get small loss is the Gradient Descent algorithm.

Gradient Descent

Let w be a weight of the predictor. To minimize the loss function ℓ w.r.t w, we can adjust w by

$$w \leftarrow w - lr \cdot \frac{\partial \ell}{\partial w}, \quad \text{for some } lr > 0.$$

Sigmoid

Sigmoid function is one of the most popular activation function due to that fact that it can be interpreted as probability.

Definition (Sigmoid)

The sigmoid function and its derivative are given by:

$$\begin{aligned} \sigma(x) &= \frac{1}{1+e^{-x}} \\ \Longrightarrow \sigma'(x) &= \sigma(x)(1-\sigma(x)). \end{aligned}$$

The sigmoid is a non zero-centered function. Its range lies between 0 to 1.

Hyperbolic Tangent

The tanh activation function has been conceived to overcome disadvantage of non-zero centered as in sigmoid.

Definition (Hyperbolic Tangent)

The tanh function and its derivate are defined by:

$$
\begin{aligned}
\tanh(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}} \\
\Longrightarrow \tanh'(x) &= 1 - \tanh^2(x).
\end{aligned}
$$

Hyperbolic Tangent

The tanh activation function has been conceived to overcome disadvantage of non-zero centered as in sigmoid.

Definition (Hyperbolic Tangent)

The tanh function and its derivate are defined by:

$$\begin{aligned} \tanh(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}} \\ \Longrightarrow \tanh'(x) &= 1 - \tanh^2(x). \end{aligned}$$

It can be seen as deformed sigmoid.

Property

We have

$$\tanh(x) = 2\sigma(2x) - 1.$$

Graphs of Hyperbolic Tangent and its derivative

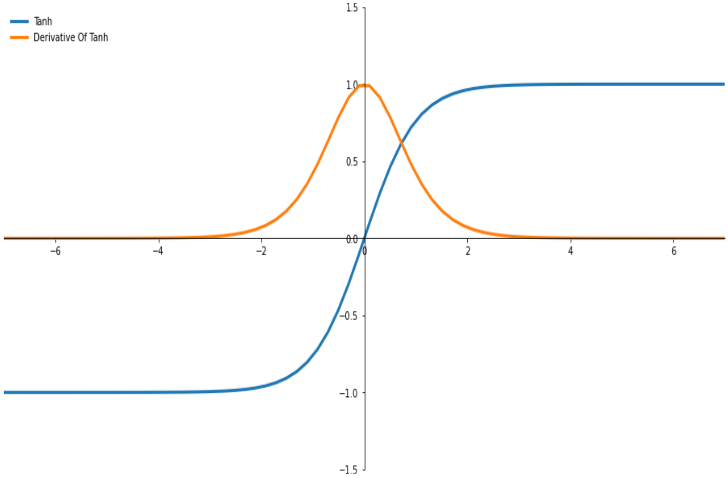

Figure: Hyperbolic tangent vs its derivative

Advantages - Disadvantages

Advantages

- Tanh is **continuously differentiable** and provides a smooth gradient, i.e., fast convergence.

Advantages - Disadvantages

Advantages

- Tanh is **continuously differentiable** and provides a smooth gradient, i.e., fast convergence.
- Tanh is a **zero-centered** function and its range lies between -1 to 1

Advantages - Disadvantages

Advantages

- Tanh is **continuously differentiable** and provides a smooth gradient, i.e., fast convergence.
- Tanh is a **zero-centered** function and its range lies between -1 to 1

Disadvantages

- Saturation leads to vanishing gradient

Advantages - Disadvantages

Advantages

- Tanh is **continuously differentiable** and provides a smooth gradient, i.e., fast convergence.
- Tanh is a **zero-centered** function and its range lies between -1 to 1

Disadvantages

- Saturation leads to vanishing gradient
- Computationally expensive

Rectified Linear Unit - ReLU

The ReLU outperforms Tanh during learning due the fact of non-vanishing Gradient.

Definition (Rectified Linear Unit)

$$
\begin{aligned}
\mathrm{ReLU}(x) &= \max(0, x) = x \max\left(0, \frac{x}{|x|}\right) \\
\Longrightarrow \mathrm{ReLU}'(x) &= \begin{cases} 0 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} {}^{a}.
\end{aligned}
$$

[a] based on this link

Graph of ReLU and its derivative

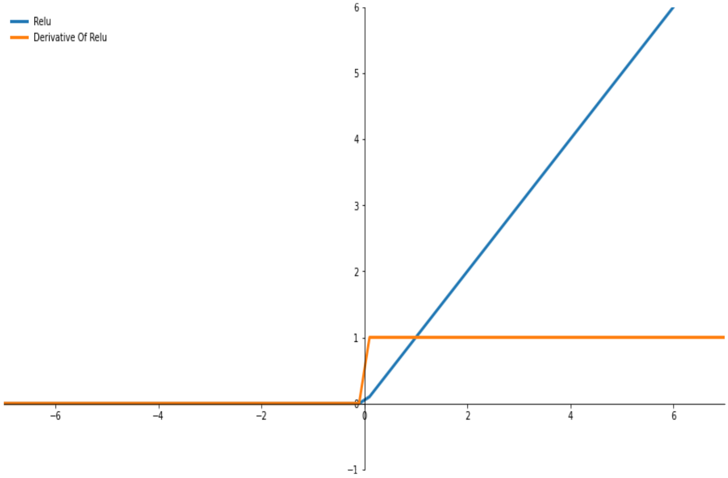

Figure: ReLU vs its derivative

Advantage and Disadvantage of ReLU

Advantages

- Cheap computation

Advantage and Disadvantage of ReLU

Advantages

- Cheap computation
- Bounded below $\implies$ Strong regularization

Advantage and Disadvantage of ReLU

Advantages

- Cheap computation
- Bounded below $\Longrightarrow$ Strong regularization
- Unbounded above $\Longrightarrow$ No vanishing gradient problem

Advantage and Disadvantage of ReLU

Advantages

- Cheap computation
- Bounded below $\Longrightarrow$ Strong regularization
- Unbounded above $\Longrightarrow$ No vanishing gradient problem

Disadvantages

- Not differentiable at 0

Advantage and Disadvantage of ReLU

Advantages

- Cheap computation
- Bounded below $\implies$ Strong regularization
- Unbounded above $\implies$ No vanishing gradient problem

Disadvantages

- Not differentiable at 0
- The gradients for negative input are zero

Swish

A lookalike ReLU parameterized activation function named *swish* was proposed in 2017 by Ramachandran et.al., which is defined as follow

Definition (swish)

Swish function and its derivative :

$$\begin{aligned} \mathrm{swish}\,(x;\beta) &= x\cdot\sigma\,(\beta x)\,,\ \forall x, \in \mathbb{R}, \beta \text{ is a constant} \\ \Longrightarrow\ \mathrm{swish}'\,(x;\beta) &= \beta\cdot\mathrm{swish}\,(x;\beta)+\sigma\,(\beta\cdot x)\,(1-\beta\cdot\mathrm{swish}\,(x;\beta)) \end{aligned}$$

Swish

A lookalike ReLU parameterized activation function named *swish* was proposed in 2017 by Ramachandran et.al., which is defined as follow

Definition (swish)

Swish function and its derivative :

$$\begin{aligned} \mathrm{swish}\,(x;\beta) &= x \cdot \sigma\,(\beta x)\,,\ \forall x, \in \mathbb{R}, \beta \text{ is a constant} \\ \Longrightarrow\ \mathrm{swish}'\,(x;\beta) &= \beta \cdot \mathrm{swish}\,(x;\beta) + \sigma\,(\beta \cdot x)\,(1 - \beta \cdot \mathrm{swish}\,(x;\beta)) \end{aligned}$$

Properties

$$\begin{aligned} \lim_{\beta \to 0} \mathrm{swish}\,(x;\beta) &= \frac{x}{2} \\ \lim_{\beta \to +\infty} \mathrm{swish}\,(x;\beta) &= \mathrm{ReLU}\,(x) \end{aligned}$$

Visualizations of swish with different values of β

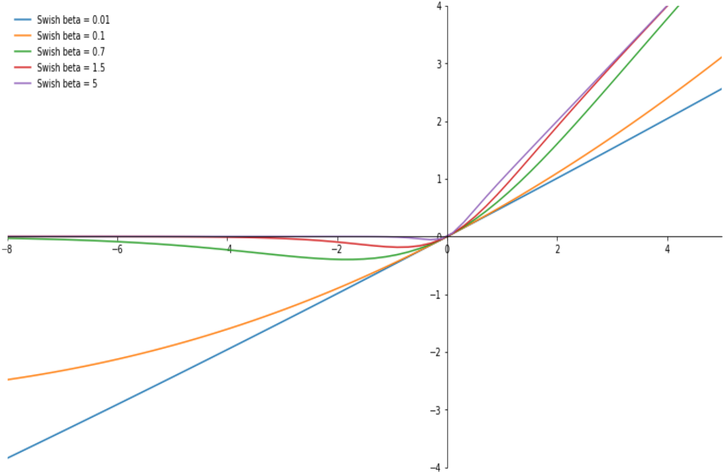

Figure: Swish and its derivatives with different values of β

Visualizations of swish and its first derivative

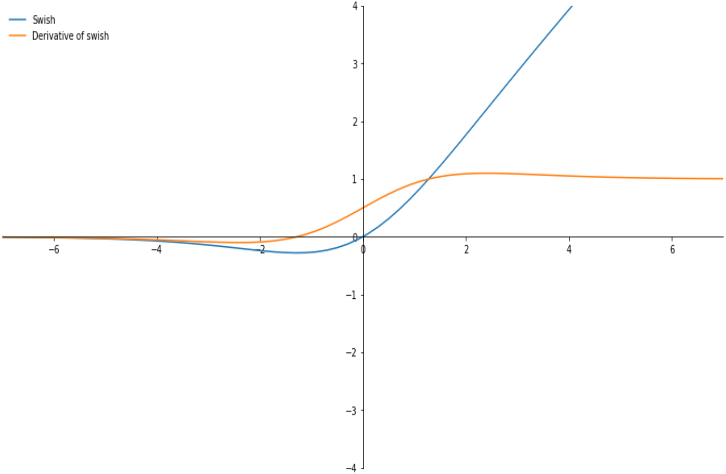

Figure: Swish and its derivatives for $\beta = 1$

Advantages - Disadvantage of swish

Advantages

- Unbounded above $\implies$ no-vanishing gradient.

Advantages - Disadvantage of swish

Advantages

- Unbounded above $\implies$ no-vanishing gradient.
- Bounded below $\implies$ regularization of the model

Advantages - Disadvantage of swish

Advantages

- Unbounded above $\implies$ no-vanishing gradient.
- Bounded below $\implies$ regularization of the model
- Smooth gradient $\implies$ fast convergence

Advantages - Disadvantage of swish

Advantages

- Unbounded above $\Longrightarrow$ no-vanishing gradient.
- Bounded below $\Longrightarrow$ regularization of the model
- Smooth gradient $\Longrightarrow$ fast convergence
- Non-monotonic $\Longrightarrow$ reduction of the influence of the weight on the output.

Advantages - Disadvantage of swish

Advantages

- Unbounded above $\implies$ no-vanishing gradient.
- Bounded below $\implies$ regularization of the model
- Smooth gradient $\implies$ fast convergence
- Non-monotonic $\implies$ reduction of the influence of the weight on the output.
- Self-gated

Advantages - Disadvantage of swish

Advantages

- Unbounded above $\implies$ no-vanishing gradient.
- Bounded below $\implies$ regularization of the model
- Smooth gradient $\implies$ fast convergence
- Non-monotonic $\implies$ reduction of the influence of the weight on the output.
- Self-gated

Disadvantage

- Computationally expensive

Mish

Definition (Mish activation function)

$$\begin{aligned} \operatorname{softplus}(x) &= \ln(1 + e^x) \\ \operatorname{mish}(x) &= x \cdot \tanh(\operatorname{softplus}(x)) \end{aligned}$$

The first derivative of Mish is given by

Derivative

$$\begin{aligned} \operatorname{mish}'(x) &= \frac{\operatorname{mish}(x)}{x} + x \cdot \sigma(x)\left(1 - \tanh^2(\operatorname{softplus}(x))\right) \\ &= \frac{\operatorname{mish}(x)}{x} + \operatorname{swish}(x) \cdot \operatorname{sech}^2(\operatorname{softplus}(x)) \end{aligned}$$

Visualizations of mish and its first derivative

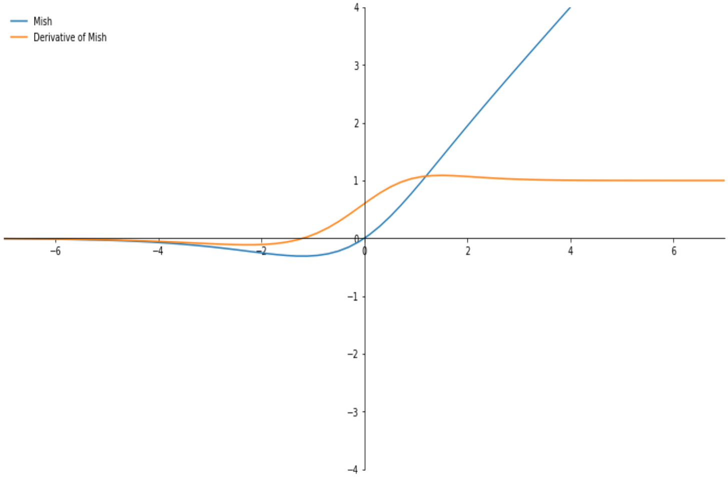

Figure: Mish vs its derivative

Difference of mish versus swish

Properties

- $\forall x \in \mathbb{R}, \tanh(\text{softplus}(x)) \geq \sigma(x)$
- $\forall x \geq 0, \text{mish}(x) \geq \text{swish}(x) \geq 0$
- $\forall x < 0, \text{mish}(x) < \text{swish}(x) < 0$

Mish vs Swish

- This last inequality implies that swish is more regularized than mish.
- Harder to compute the gradient compared to swish.

Visualizations of mish versus swish

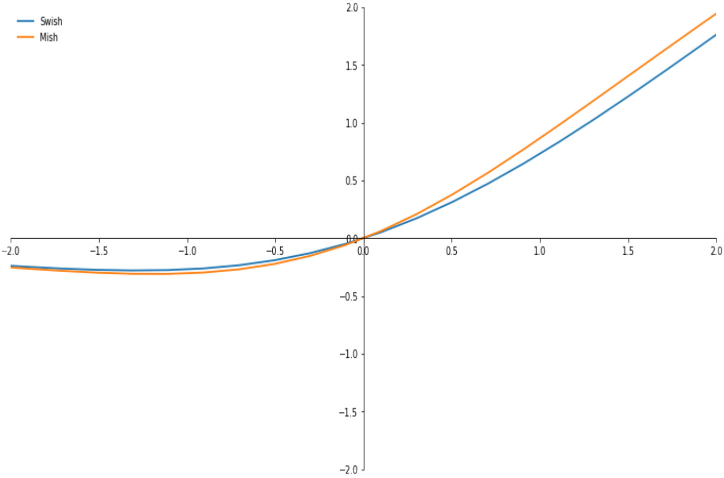

Figure: Mish vs swish

Visualizations of mish versus swish

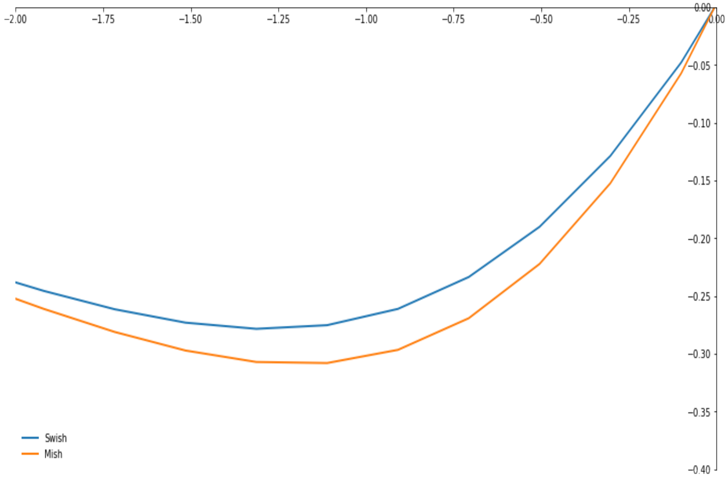

Figure: Zoom in the negatives values

Presentation of the used data and model architecture

For the experiment, we used the MNIST dataset.

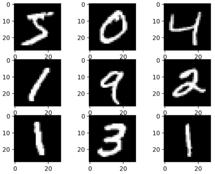

Figure: NMIST data

Presentation of the used data and model architecture

For the experiment, we used the MNIST dataset.

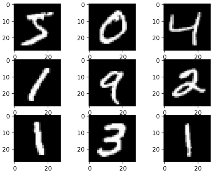

Figure: NMIST data

The **model architecture** is as followed :

$$
\begin{tikzcd}
\text{Conv1} + \text{af} \arrow[r] & \text{Conv2} + \text{af} \arrow[r] & \text{Conv3} + \text{af} \arrow[r] & \text{Conv4} + \text{af} \arrow[d] \\
 & & \text{Fully connected} & \text{Conv5} + \text{af} \arrow[l]
\end{tikzcd}
$$

Experimental result

Model parameters

batchsize : 64
optimizer : SGD
lr : 0.01
momentum : 0.9
loss function : cross entropy

Experimental result

Model parameters

batchsize : 64
optimizer : SGD
lr : 0.01
momentum : 0.9
loss function : cross entropy

Activation function | Num epochs	5	10	30
Tanh	97.69%	98.32%	98.74%
ReLU	98.21%	11.35%	98.78%
Swish	97.99%	98.50%	98.67%
Mish	98.01%	98.49%	98.90%

Table: Test Accuracy.

Observation

We observe the following in general.

- Tanh performs the least.
- Mish performs the most.

conclusion on the perforamnce of the activation functions

Mish $>$ swish $\approx$ ReLU

Thank you for listening ...