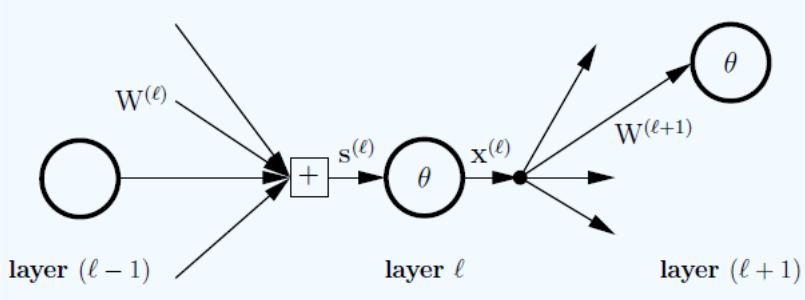
Neural Networks

Rao Sihang 15220162202309

A Brief Introduction of Various Activation Functions



$$x^l = \theta(s^l)$$

Activation Function is a extremely part of neural network, which can transfer a linear classifer to a nonlinear one.There are various activation functions available, and different functions have different performances in practice and application.

In gradient-based learning methods and backpropagation, we update weights w and find the optimal one based on the partial derivative of loss function, which is a function of derivative of activative function. Backpropagation computes gradients by the chain rule, if the absolute value of derivative of activative functuin is smaller than one, which has the effect of multiplying n of these small numbers to compute gradients of the "front" layers in an n-layer network, meaning that the gradient (error signal) decreases exponentially with n while the front layers train very slowly and weights cannot be updated, this phenomenon is called **vanishing gradient problem.** On the other hand, if absolute value of derivative of activation function is larger than one, **exploding gradient problem** would happen.

Sigmoid

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

Sigmoid, called Logistic Activation Function, historically dominated in this field, but finally, it was proved to be unsuitable for neural network because of samll derivative giving rise to vanishing gradients.

When $\sigma(x) \to \pm\infty, \sigma'(x) \to 0$,which leads to the failure of update in weight for specific neurons. By the way, the output of this function ranges from zero to one, so mean is non-zero.

Tanh

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

Similary, due to zero derivative, this activation function also suffers from vanishing gradients. Differing from the last one, Output mean of Tanh function is zero, so this one is relatively more popular than Sigmoid function in practice.

ReLU (Rectified Linear Unit)

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

$f'(x) = 1, x > 0$,so this function can guarantee the update of weights at least half of neurons,which can alleviate vanishing gradients.But for negative x, it still cannot update the weight.

Similar with Sigmoid , output mean of ReLu is not zero.

Leaky ReLU

$$f(x) = max(0.1x, x)$$

This function makes a nonzero derivative for negative zero to avoid dead ReLU, but the results are inconsistent.

Parametric ReLU

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

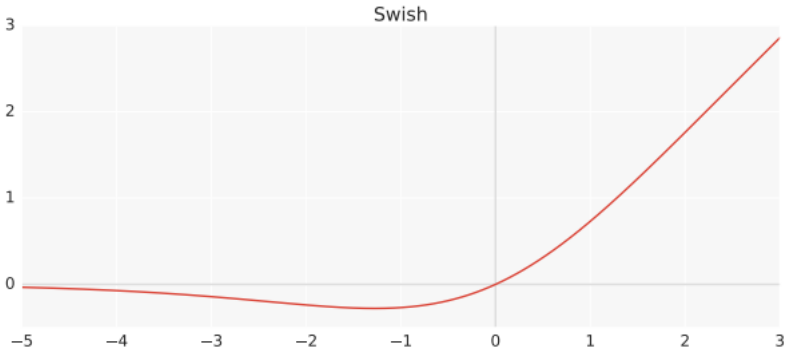
The optimal parameter α can be gained through learning.

Besides,in order to alleviate dead ReLU, there are still lots of ReLU functions, for example

- PReLU (Parameteric Rectified Linear Unit): value of α is random with uniform distribution.
- ELU (Exponential Linear Unit): $f(x) = max\{\alpha(e^x - 1), x\}$
- SELU(Scaled Exponential Linear Unit):$f(x) = max\{\lambda(\alpha(e^x - 1), \lambda x\}, \lambda = 1.0507, \alpha = 1.67326$

SWISH

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



This activation function is proposed by Ramachandran et al. in 2017, they employed automatic search to find high-performing novel activation functions and found SWICH is the best one with non-monotonicity and the gradientp reserving property,which gave rise to lots of arguments in terms of increase in learning efficiency from this function.

The property of activation function *:From Steffen et al. (2018)*

Property	Description	Problems	Examples
derivative	f'	> 1 exploding gradient (e) < 1 vanishing (v)	sigmoid (v), tanh (v), cube (e)
zero-centered	range centered around zero?	if not, slower learning	tanh (+), relu (−)
saturating	finite limits	vanishing gradient in the limit	tanh, penalized tanh, sigmoid
monotonicity	$x > y \implies f(x) \geq f(y)$	unclear	exceptions: sin, swish, minsin

A Comparison of Deep Learning Activation Functions and Penalized Tanh Function

Steffen et al. (2018) firstly perfomed the largescale comparison of 21 activation functions across eight different NLP tasks and they found "a largely unknown activation function performs most stably across all tasks, the so-called penalized tanh function". Penalized Tanh function was firstly proposed in Xu et al.(2016).

$$f(x) = \begin{cases} tanh(x) & x > 0 \\ 0.25tanh(x) & x \leq 0 \end{cases}$$

- They compared 21activation functions
- They tries in three popular NLP task types (sentence classification, document classification, sequence tagging) comprising 8 individual tasks.
- They used three different popular NLP architectures, namely, MLPs, CNNs, and RNNs.

According to general perfomance, Penalized tanh and SWISH performed best, which was followed by rectifier functions..