Regularized Flexible Activation Function Combination for Deep Neural Networks

Renlong Jie, Junbin Gao, Andrey Vasnev, Minh-ngoc Tran

The University of Sydney Business School

{renlong.jie, junbin.gao, andrey.vasnev, minh-ngoc.tran}@sydney.edu.au

ICPR 2020
25th INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION
Milan, Italy 10 | 15 January 2021

Introduction

There are several limitations of the existing studies on flexible/trainable activation functions. First, most of existing work focus on some specific forms of parameterized activation functions rather than a more general form, or consider each component of the combination as a fixed activation function. Second, there is a lack of attention to flexible activations with bounded domain such as sigmoid and tanh. Third, existing works rarely discuss the regularization of activations parameters, which have different nature from normal model parameters. In this study, we consider the activation function as a combination of a set of component functions following several principles. Based on these principles, we develop two flexible activation functions that can be implemented for bounded or unbounded domains.

Main Objectives

1. Design a general form and updating rules for combined flexible activation functions.
2. Develop flexible activation functions with both bounded and unbounded domains.
3. Design regularization terms for the flexible activation parameters to control the flexibility.
4. Use experiments to show the advantage of proposed activation function and regularization techniques.

Methods

We implement a general form of parameterized activation functions linearly combining different activation functions as components:

$$o_i(z, \boldsymbol{\alpha}^i, \boldsymbol{\beta}^i) = \sum_{k=1}^{K} \alpha_{ik} f_k(z, \boldsymbol{\beta}_{ik}), \quad \sum_{k=1}^{K} \alpha_{i,k} = 1, \quad 0 \le \alpha_{i,k} \le 1 \quad \forall k, i. \tag{1}$$

where i indexes the neuron, and $z = z_l = \boldsymbol{W_l} X_{l-1} + \boldsymbol{b_l}$ is the input of the activation layer indexed by l. k is the index of each component and $\alpha_{i,k}$ is the corresponding combination weight. $\boldsymbol{\beta}_{ik}$ is the activation parameter vector for the k-th component activation f_k in ith neuron. The back propagation of activation parameters by stochastic gradient descent can be done by:

$$\begin{aligned} \alpha_{ik} &\to \alpha_{ik} - \gamma \frac{\partial L}{\partial \alpha_{ik}} = \alpha_{ik} - \gamma \frac{\partial L}{\partial o_i} \cdot f_{ik}(z, \boldsymbol{\beta}_{ik}), \\ \boldsymbol{\beta}_{ik} &\to \boldsymbol{\beta}_{ik} - \gamma \frac{\partial L}{\partial \boldsymbol{\beta}_{ik}} = \boldsymbol{\beta}_{ik} - \gamma \frac{\partial L}{\partial o_i} \cdot \alpha_{ik} \frac{\partial f_{ik}(z, \boldsymbol{\beta}_{ik})}{\partial \boldsymbol{\beta}_{ik}}. \end{aligned} \tag{2}$$

We propose three principles for selecting the components for combined flexible activation functions.

- Each component should have the same domain as the baseline activation function.
- Each component should have an equal range as the baseline activation function.
- Each component activation functions should be expressively independent of other component functions with the following definition, which means each component function should not be expressed by a linear combination of other components with effective training for any output from the previous layer.

Based on the general combination form and the three principles mentioned above, we design two types of activation function as follows:

P-Sig-Ramp: This can be applied to replace fixed sigmoidal functions.

$$o(z; \alpha, \beta) = \alpha \cdot \sigma(z) + (1 - \alpha) \cdot f(z; \boldsymbol{\beta}),$$

where $0 \le \alpha \le 1$ and

$$f(z; \boldsymbol{\beta}) = \begin{cases} 0 & \text{if } z < -\frac{1}{2\beta}, \\ \beta z + \frac{1}{2} & \text{if } -\frac{1}{2\beta} \le z \le \frac{1}{2\beta}, \\ 1 & \text{if } z > \frac{1}{2\beta}. \end{cases}$$

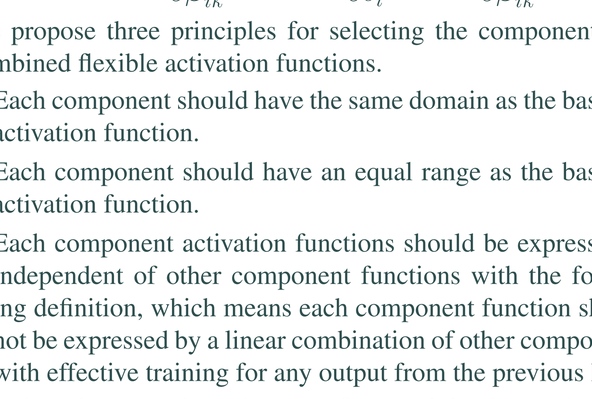

Figure 1: Examples of P-Sig-Ramp with different activation parameters.

P-E2-ReLU/Id: This can be applied to replace fixed ReLU, ELU and PFeLU functions.

$$\begin{aligned} o(z; \alpha, \beta) &= \alpha \text{Relu}(z) + \beta \text{Elu}(z) + (1 - \alpha - \beta)(-\text{Elu}(-z)), \\ o(z; \alpha, \beta) &= \alpha \text{Relu}(z) + (1 - \alpha)(-\text{Elu}(-z, \beta) + \text{Elu}(z, \beta)), \\ o(z; \alpha, \beta) &= \alpha z + (1 - \alpha)(-\text{Elu}(-z, \beta) + \text{Elu}(z, \beta)). \end{aligned}$$

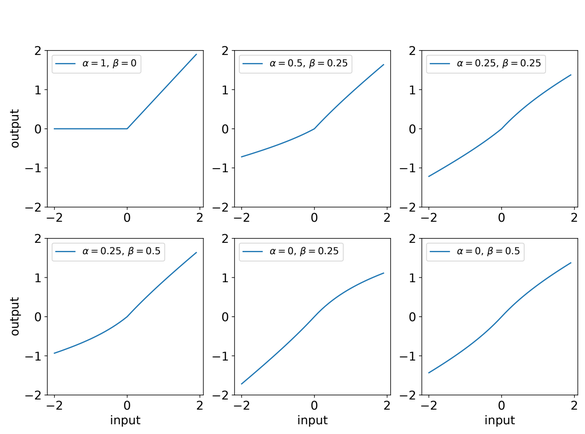

Figure 2: Examples of the 1st P-E2-ReLU with different activation parameters.

In addition, to control the flexibility and avoid over-parameterization, we introduce two regularization terms. The whole loss function can be written as follows.

$$\begin{aligned} L &= L_0 \\ &+ \delta_1 \sum_j \frac{\lambda_j}{m_j} \sum_i \sum_k ||\alpha_{ijk} - \bar{\alpha}_{jk}||^2 \\ &+ \frac{\delta_2}{n} \sum_i \sum_j \sum_k ||\alpha_{k0} - \alpha_{ijk}||^2 \\ &+ \frac{\delta_3}{n} \sum_i \sum_j \sum_k (||\text{ReLU}(\alpha_{ijk^*} - (1 - \Delta))||^2 \\ &+ ||\text{ReLU}(-\Delta - \alpha_{ijk^*})||^2) + \text{other terms}. \end{aligned} \tag{3}$$

The first regularization term controls the deviation of flexible activation functions from the average function of a layer. The second regularization term controls the flexible activation functions from the fixed baselines such as ReLU and sigmoid.

Results

First, we do multi-variate time series forecasting on G7 indices with stacked LSTMs, while four different hidden layer configurations are applied. The three candidate activation functions include fixed sigmoid function, P-Sig-Ramp with no regularization on activation parameters, P-Sig-Ramp with towards-mean regularization with a regularization coefficient of 0.025, and P-Sig-Ramp with optimized regularization coefficients. We do hyper-parameter search to find the optimal learning rate for each model and each activation functions. The learning curves of 50 runs on validation set are given in Figure 3.

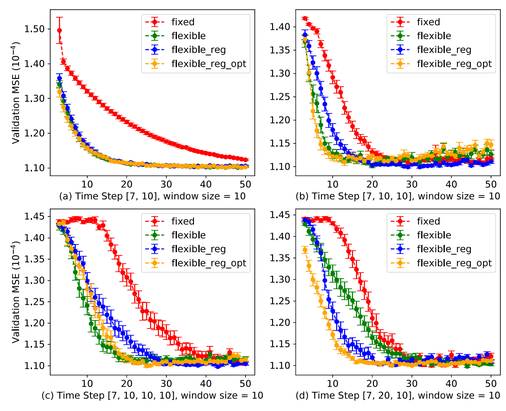

Figure 3: Comparison between the average learning curves (with error bars) of LSTM models with different activation functions.

The next experiment is to apply two convolutional auto-encoder models for lossy image compression on MNIST and FMNIST. The baseline activation functions applied include ReLU, PReLU, GeLU, ELU, also we use the proposed P-E2-ReLU. For each of the baseline activation functions, we use the optimized learning rate, while for P-E2-ReLU, we apply both its optimized learning rate and the optimized learning rate of other baseline activation functions. The learning curves on validation set demonstrate that P-E2-ReLU consistently outperform other activation functions even with the optimized learning rate of other methods. The result is given by Figure 4.

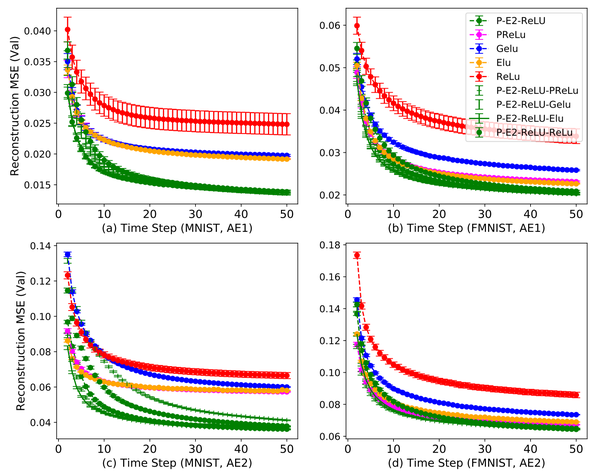

Figure 4: Comparison between the average learning curves (with errorbars) of convolutional auto-encoder models with different activationfunctions on image compression task.

Another experiment that applies another covolutional auto-encoder on CIFAR10 and SVHN with different activation functions gives the validation curves shown in Figure 5, where the proposed P-E2-Id outperforms other activation functions with statistical significance as is shown in Table 1. Also, as is indicated in Figure 6, a layer-wise activation function with shared activation parameter could be sufficiently good to make a large improvement for the optimal model performance.

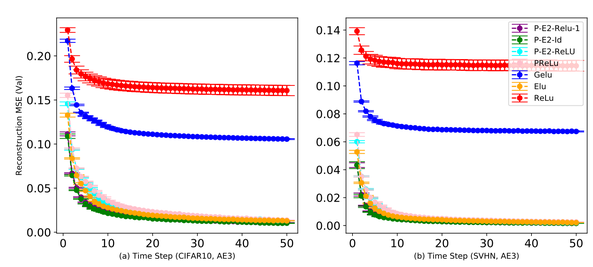

Figure 5: Comparison between the average learning curves (with errorbars) of convolutional auto-encoder models with different activationfunctions on image compression task.

	Activation	Dataset CIFAR10	Dataset SVHN
Model 1	P-E2-Id	**1.01E-2 (2.4E-4)**	**1.25E-3 (3.9E-5)**
Model 2	ELU	1.27E-2 (1.6E-4)	1.84E-3 (2.7E-5)
Model 3	PReLU	1.35E-2 (1.5E-4)	2.24E-3 (9.3E-5)
Model 4	P-E2-ReLU	1.05E-2 (2.8E-4)	1.38E-3 (6.3E-5)
	Null Hypothesis		p-value
Test 1	H0: $m_4 \ge m_2$		2.05E-06	4.86E-7
Test 2	H0: $m_4 \ge m_3$		4.77E-10	2.03E-7

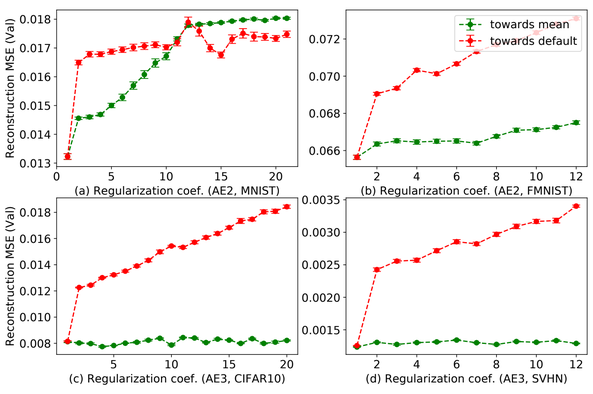

Figure 6: Comparison between the average learning curves (with error bars) of Autoencoder models with flexible activation P-E2-ReLU under different activation regularization settings.

Conclusions

- We proposed two types of combined flexible activation functions: P-Sig-Ramp and P-E2-ReLU.
- We introduced two regularization terms to: (1) Control the deviation of flexible activation functions from the average activation function in each layer; (2) Control the deviation of flexible activation functions from the baseline activation function.
- Experiments of learning tasks with LSTM and CAE show that the proposed activation function and regularization terms are effective in improving model convergence.