Regularized Flexible Activation Function Combinations for Deep Neural Networks

Renlong Jie, Junbin Gao, Andrey Vasnev, and Minh-Ngoc Tran

University of Sydney Business School

December 10, 2020

Methodology

The activation function in combined form:

$$
\begin{aligned}
o_i(z, \boldsymbol{\alpha}^i, \boldsymbol{\beta}^i) &= \sum_{k=1}^{K} \alpha_{ik} f_k(z, \boldsymbol{\beta}_{ik}), \\
\sum_{k=1}^{K} \alpha_{i,k} = 1, \quad 0 \leq \alpha_{i,k} &\leq 1 \quad \forall k, i
\end{aligned}
\tag{1}
$$

Updating rules:

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

Methodology

We introduce the following three principles for selecting the components:

- **Principle 1**: Each component should have the same domain as the baseline activation function.
- **Principle 2**: Each component should have an equal range as the baseline activation function.
- **Principle 3**: Each component activation functions should be expressively independent of other component functions with the following definition.

Methodology

Definition 1: If a component activation function f_k is **expressively independent** of a set of other component functions: $f_1, ..., f_n$, there does not exist a set of combination coefficients α_1,...,α_n, inner activation parameters β_1,...β_n, parameters of the previous linear layers $\boldsymbol{W'}$, $\boldsymbol{b'}$ such that for any input X, activation parameters $\boldsymbol{\beta_k}$, and parameters of the previous linear layer $\boldsymbol{W_k}$, $\boldsymbol{b_k}$, the following equation holds:

$$\begin{aligned} f_k(z_k, \boldsymbol{\beta_k}) &= f_k(\boldsymbol{W_k} X + \boldsymbol{b}_k, \boldsymbol{\beta_k}) \\ &= \sum_{i=1}^{n} \alpha_i f_i(\boldsymbol{W'} X + \boldsymbol{b'}, \boldsymbol{\beta_i}) = \sum_{i=1}^{n} \alpha_i f_i(z', \boldsymbol{\beta_i}) \end{aligned} \tag{3}$$

Proposition 1: For a single-layer network with m neurons, if a component activation function f_k, which is not expressively independent of other components, is excluded, we need at most $2m$ neurons to express the same mapping exactly.

Methodology

P-Sig-ramp: Sigmoid/Tanh Function substitute with bounded domain:

$$o(z;\alpha,\beta) = \alpha \cdot \sigma(z) + (1-\alpha) \cdot f(z;\boldsymbol{\beta}) \tag{4}$$

where $0 \leq \alpha \leq 1$ and

$$f(z;\boldsymbol{\beta}) = \begin{cases} 0 & \text{if } z < -\frac{1}{2\beta} \\ \beta z + \frac{1}{2} & \text{if } -\frac{1}{2\beta} \leq z \leq \frac{1}{2\beta} \\ 1 & \text{if } z > \frac{1}{2\beta} \end{cases} \tag{5}$$

P-E-Relu/Id: Relu Function substitute with unbounded domain:

$$\begin{aligned} o(z;\alpha,\beta) &= \alpha \mathrm{Relu}(z) + \beta \mathrm{Elu}(z) + (1-\alpha-\beta)(-\mathrm{Elu}(-z)) \\ o(z;\alpha,\beta) &= \alpha \mathrm{Relu}(z) + (1-\alpha)(-\mathrm{Elu}(-z,\beta) + \mathrm{Elu}(z,\beta)) \\ o(z;\alpha,\beta) &= \alpha z + (1-\alpha)(-\mathrm{Elu}(-z,\beta) + \mathrm{Elu}(z,\beta)) \end{aligned} \tag{6}$$

Methodology

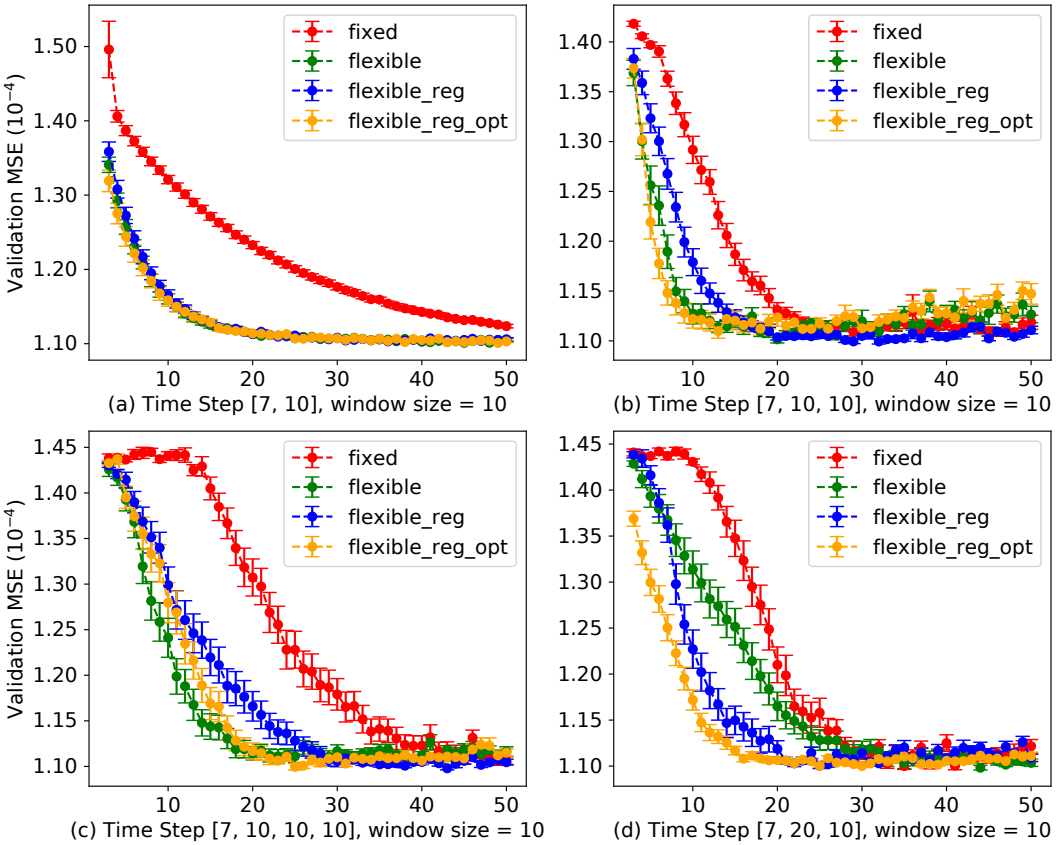
Layer-wise Regularisation for Activation Parameters:

$$
\begin{aligned}
L &= L_0 \\
&+ \delta_1 \sum_j \frac{\lambda_j}{m_j} \sum_i \sum_k ||\alpha_{ijk} - \bar{\alpha}_{jk}||^2 \quad \text{towards-mean} \\
&+ \frac{\delta_2}{n} \sum_i \sum_j \sum_k ||\alpha_{k0} - \alpha_{ijk}||^2 \quad \text{towards-default} \\
&+ \frac{\delta_3}{n} \sum_i \sum_j \sum_k (||\text{ReLu}(\alpha_{ijk^*} - (1 - \Delta))||^2 \\
&+ ||\text{Relu}(-\Delta - \alpha_{ijk^*})||^2) + \text{other terms}
\end{aligned}
\tag{7}
$$

The regularization coefficients δ_1 and δ_2 can be implemented for controlling the flexibility in activation parameter space.

Experiments

Experiment for multi-variate time series forecasting with LSTMs: (fixed = sigmoid, flexible = P-Sig-Ramp)



Experiments

Convolutional Autoencoder 1 (CAE 1):

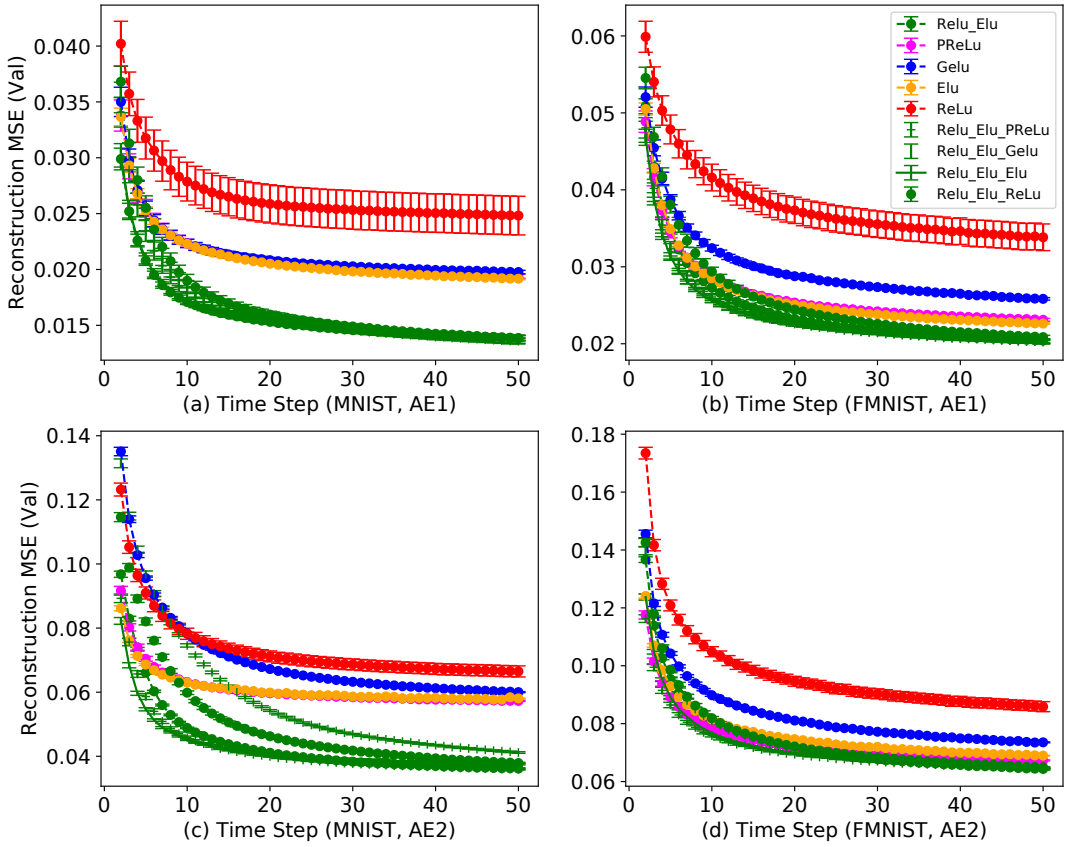
$$\text{Input}(28*28*1) \rightarrow \text{Conv2d}(16,\ 3,\ 3) \xrightarrow{\text{ReLU}} \text{MP}(2,\ 2) \rightarrow$$
$$\text{Coding} \rightarrow \text{Conv2d}(8,\ 5,\ 3) \xrightarrow{\text{ReLU}} \text{Conv2d}(1,\ 2,\ 2\ )$$
$$\xrightarrow{\text{Tanh}} \text{Output}$$

Convolutional Autoencoder 2 (CAE 2):

$$\text{Input}(28*28*1) \rightarrow \text{Conv2d}(16,\ 3,\ 3) \xrightarrow{\text{ReLU}} \text{MP}(2,\ 2) \rightarrow$$
$$\text{Conv2d}(8,\ 3,\ 2) \xrightarrow{\text{ReLU}} \text{MP}(2,\ 1) \rightarrow \text{Coding}$$
$$\rightarrow \text{Conv2d}(16,\ 3,\ 2) \xrightarrow{\text{ReLU}} \text{Conv2d}(8,\ 5,\ 3)$$
$$\xrightarrow{\text{ReLU}} \text{Conv2d}(1,\ 2,\ 2\ ) \xrightarrow{\text{Tanh}} \text{Output}$$

Experiments

Experiment for lossy image compression with CAE 1 and CAE 2 on MNIST and FMNIST: (Relu_Elu = P-E2-ReLU)



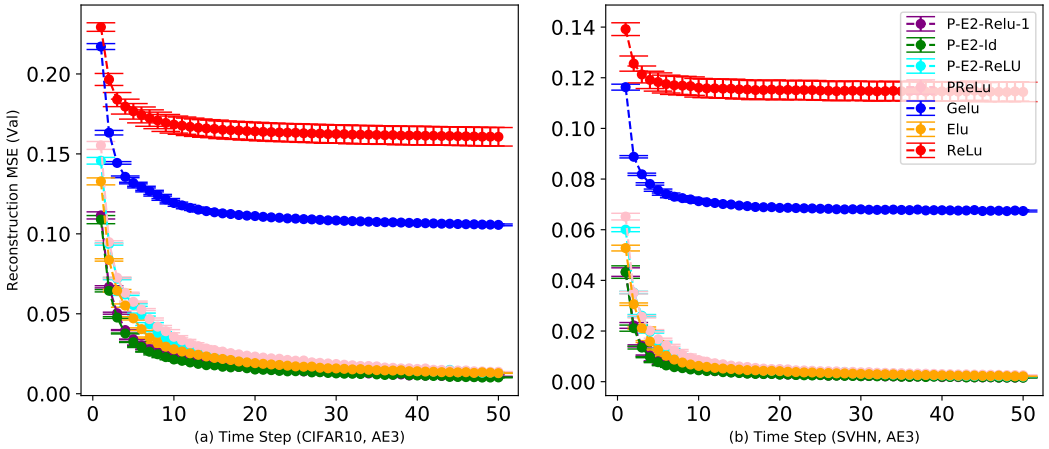
Experiments

Convolutional Autoencoder 3 (CAE 3):

$$\text{Input}(32*32*3) \rightarrow \text{Conv2d}(12,\ 4,\ 2) \xrightarrow{\text{ReLU}} \text{Conv}(24,\ 4,\ 2)$$
$$\xrightarrow{\text{ReLU}} \text{Conv}(48,\ 4,\ 2) \xrightarrow{\text{ReLU}} \text{Coding} \rightarrow \text{Conv2d}(24,\ 4,\ 2)$$
$$\xrightarrow{\text{ReLU}} \text{Conv2d}(12,\ 4,\ 2) \xrightarrow{\text{ReLU}} \text{Conv2d}(3,\ 4,\ 2) \xrightarrow{\text{ReLU}} \text{Output}$$

Experiments

Experiment for lossy image compression with CAE 3 on CIFAR10:



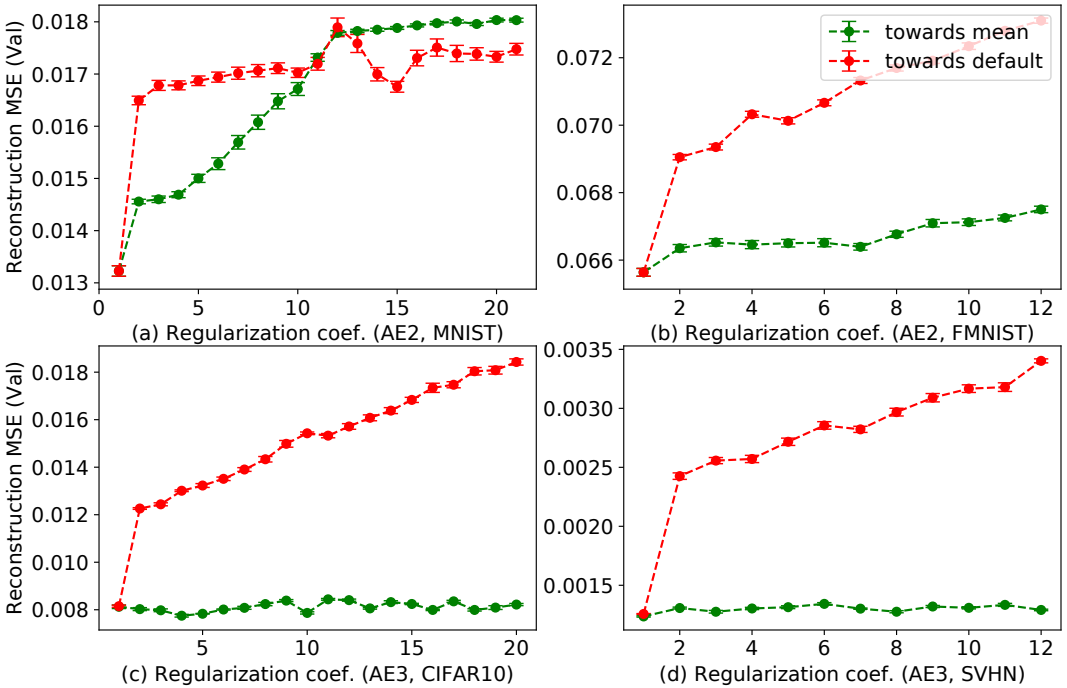
Experiments

Table: Statistical Tests for Performances on Test Set

	Activation	Dataset CIFAR10	Dataset SVHN
Model 1	P-E2-Id-1	**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: $m_1 \geq m_2$	3.64E-09	3.96E-14
Test 2	H0: $m_1 \geq m_3$	1.73E-13	3.97E-11
Test 3	H0: $m_4 \geq m_2$	2.05E-06	4.86E-7
Test 4	H0: $m_4 \geq m_3$	4.77E-10	2.03E-7

Experiments

Experiment for lossy image compression: Regularization



Conclusion

- Two types of combined flexible activation functions are proposed: P-Sig-Ramp and P-E2-ReLU.
- Two regularization terms: (1) Control the deviance from the average activation function in each layer; (2) Control the deviance from the baseline activation function.
- Experiments of LSTM on the tasks of time series forecasting and CAE for lossy image compression.

The End

Thanks for your attention.