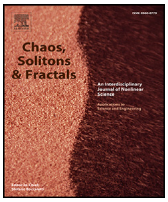
Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos



Optimized injection of noise in activation functions to improve generalization of neural networks

Fabing Duan [a,*], François Chapeau-Blondeau [b], Derek Abbott [c]

[a] *Institute of Complexity Science, Qingdao University, Qingdao 266071, PR China*
[b] *Laboratoire Angevin de Recherche en Ingénierie des Systèmes (LARIS), Université d'Angers, 62 avenue Notre Dame du Lac, 49000 Angers, France*
[c] *Centre for Biomedical Engineering (CBME) and School of Electrical and Electronic Engineering, The University of Adelaide, South Australia 5005, Australia*

ARTICLE INFO

Keywords:
Probabilistic activation function
Adaptive stochastic resonance
Injected noise
Rademacher complexity
Generalization

ABSTRACT

This paper proposes a flexible probabilistic activation function that enhances the training and operation of artificial neural networks by intentionally injecting noise to gain additional control over the response of each neuron. During the learning phase, the level of injected noise is iteratively optimized by gradient-descent, realizing a form of adaptive stochastic resonance. From simple hard-threshold non-differentiable neuronal responses, controlled injection of noise gives access to a wide range of useful activation functions, with sufficient differentiability to enable gradient-descent learning for both the neuron and the injected-noise levels. Experimental results on function approximation demonstrate injected noise generally converging to non-vanishing optimal levels associated with improved generalization abilities in the neural networks. A theoretical explanation of the generalization improvement based on the path norm bound is presented. With injected noise in the deep neural network, experimental results on classifying images also obtain non-vanishing optimal noise levels to achieve better testing accuracies. The proposed probabilistic activation functions show the potential of adaptive stochastic resonance for useful applications in machine learning.

1. Introduction

Injecting noise into activation functions is emerging as an effective way to facilitate artificial neural network training, yielding competitive results on different tasks, for example on face verification [1] and PennTreebank analysis [2]. It is argued [2] that injection of appropriate noise into the saturated regimes of activation functions facilitates the flow of gradients, whereas such noiseless activation functions may get blocked by vanishing gradients. This approach [1,2] is essentially a natural extension of injecting noise into the input, weights, expected signals or gradients for improving the generalization ability of artificial neural networks [3–11]. Specifically, using a rigorous expansion of infinitesimal injected noise variance, Bishop [12] proved that noise injection into input is equivalent to a smoothing regularization that behaves as a generalized Tikhonov regularizer in the loss function. Similarly, the superiority of injecting noise in activation functions of the hidden layer was also theoretically demonstrated for minimizing the convex loss function of the feedforward neural network, which has a smaller empirical loss than networks with injecting noise into the input [13,14].

Of particular note is that the activation function [13–15] or the estimator [16] with noise injection can be theoretically reduced to a transformed unit smoothed over the probability density function (PDF) of injected noise. This transformation allows backpropagation training of neural networks with a family of nondifferentiable activation functions [13,15], and avoids the time-consuming statistical experiments of neural network training by injecting a large number of noise samples into the input or the model parameters [12,17,18]. Moreover, such transformed units with adjustable characteristics for the injected noise can improve the neural network performance with only a small increase in the complexity of the network architecture [13,15,19–24]. Along with updating weights of networks in the training, the noise hyperparameters of the transformed unit also adaptively learn by gradient descent. Designing activation functions with learnable hyperparameters that enable fast training of accurate deep neural networks is becoming an attractive area of interest in machine learning [1,2,13–15,19–27].

However, the assumption of infinitesimal injected noise variance in the proof of the Tikhonov regularization [12,13] does not always hold, because the converged (local optimal) noise variance in activation functions is frequently far larger than unity in trials. Thus, besides experiments of various activation functions on benchmark data sets, the mathematical explanation of the performance of neural networks with general activation functions is essential for insight into the network

[*] Corresponding author.
E-mail address: fabingduan@qdu.edu.cn (F. Duan).

https://doi.org/10.1016/j.chaos.2023.114363
Received 4 October 2023; Received in revised form 27 November 2023; Accepted 3 December 2023

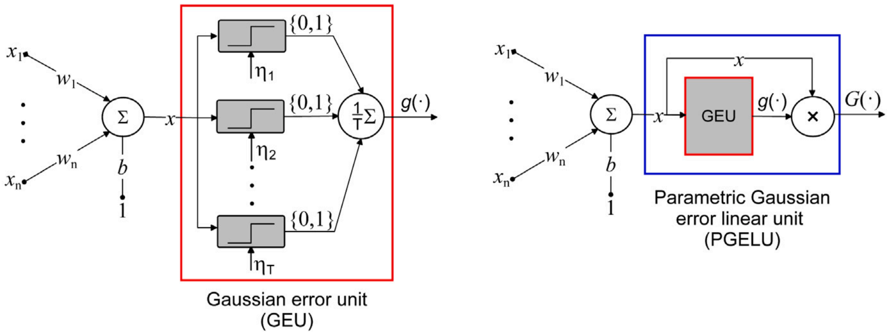


Fig. 1. Block diagram representations of general probabilistic activation functions of GEUs of Eq. (5) and PGELUs of Eq. (7). Note that other probabilistic activation functions indicated in Eq. (2) can be derived by injecting noise η with various PDFs.

generalization. In this paper, we first propose a general probabilistic activation function from the perspective of stochastic resonance [28], which exploits the constructive role of a nonzero amount of injected noise for improving the performance of certain nonlinear systems [13–16,29–39]. It is interesting to note that the proposed activation function not only bridges the gap between the fundamental McCulloch–Pitts (binary) neuron model [40,41] and some pre-specified nonlinear activation functions, e.g. sigmoid and hyperbolic tangent (tanh), but also naturally elicits some *ad hoc* activation functions. In addition, on multiplying the input by the probabilistic output of the proposed activation function, we can also obtain the unbounded nonlinearities of the rectified linear unit (ReLU) [1] and the Gaussian error linear unit (GELU) [24]. Thus, the proposed probabilistic activation function is far more inclusive.

Furthermore, we will show that the proposed activation functions outperform the traditional ones across some benchmark classification tasks, because the injected noise can adaptively modify the input–output function relationship and allow the gradient to flow more efficiently in the network training. Correspondingly, the designed network with the proposed activation functions also achieves a smaller generalization error on real data sets. Using the path-based norm measure [42,43], it is then found that the hyperparameter of injected noise in activation functions provides size-independent complexity control for a shallow feedforward neural network, which establishes a theoretical explanation of injected noise for improving generalization of the designed networks. Experimental results with deep neural networks on image classification also demonstrate several optimized noise levels larger than unity. With such optimized noise, the proposed activation functions can better preserve the features of images, and then higher testing accuracies are obtained. These theoretical and experimental results show that the proposed probabilistic activation functions established upon adaptive stochastic resonance are definitely worth exploring further for enhancing the capabilities of deep neural networks.

2. Formulation of probabilistic activation functions

2.1. Model

As indicated in Fig. 1, we motivate a general probabilistic activation function by averaging T stochastic threshold classifiers [7,33,34,44,45] or McCulloch–Pitts neurons [40] denoted by

$$h(x+\eta_t) = \begin{cases} 1, & x+\eta_t \geq 0, \\ 0, & x+\eta_t < 0, \end{cases} \tag{1}$$

which are activated by the common input $x = \sum_{i=1}^{n} w_i x_i + b$ plus mutually independent white noise components η_t for $t = 1, 2, \ldots, T$. Here, w_i are weight coefficients and b is the bias. Assume that the injected noise components η_t have the common PDF $f_\eta(\eta)$, then each neuron in Eq. (1) yields a response of unity with probability $p(x) = \int_{-x}^{\infty} f_\eta(\eta)d\eta = 1 - F_\eta(-x)$, where $F_\eta(u) = \int_{-\infty}^{u} f_\eta(\eta)d\eta$ denotes the cumulative density function (CDF) of the injected noise η. By averaging outputs of T neurons, as shown in Fig. 1, we have the output $\bar{h}(x) = \frac{1}{T}\sum_{t=1}^{T} h(x+\eta_t)$ that takes the value t/T according to the binomial distribution $\binom{T}{t} p^t(1-p)^{T-t}$. It is noted this neuronal ensemble is closely associated with the occurrence of the suprathreshold stochastic resonance effect [33,34], where the optimal noise level elicits the maximum mutual information between the neuronal ensemble and the suprathreshold inputs. Here, we regard this model in Fig. 1 as a flexible probabilistic activation function to be explored in the artificial neural network as follows.

For a sufficiently large number T of neurons, the neuron ensemble output tends to the mean $p(x)$ of the binomial distribution, i.e. $g(x) = \lim_{T\to\infty} \bar{h}(x) \approx p(x)$. Therefore, a general activation function boosted by injected noise is defined as

$$g(x) = 1 - F_\eta(-x). \tag{2}$$

Since the CDF $F_\eta(x)$ satisfies $0 \leq F_\eta(x) \leq 1$, thus $0 \leq g(x) \leq 1$ and $g(x)$ is a saturating activation function with bounds 0 or 1 as $x \to \pm\infty$. Interestingly, due to the derivative $dg(x)/dx = g'(x) = f_\eta(-x)$, the general activation function $g(x)$ in Eq. (2) is Lipschitz continuous when the PDF f_η exists and is bounded almost everywhere on the domain of definition $x \in \mathbb{R}$. For instance, it is noted that the sigmoid activation function

$$g(x) = \frac{1}{1+e^{-x}} \tag{3}$$

can be deduced from the logistic noise η with its PDF $f_\eta(x) = e^{-x}/(1+e^{-x})^2 = \mathrm{sech}^2(x/2)/4$, and the saturating linear activation function

$$g(x) = \begin{cases} 1, & x \geq 1, \\ x, & 0 < x < 1, \\ 0, & x \leq 0 \end{cases} \tag{4}$$

can be derived by the uniform noise distributed in the interval of $[-1, 0]$. The tanh activation function $\tanh(x)$ can be expressed as a linear transformation of $2g(x) - 1 = 1 - 2F_\eta(-x)$, where the logistic noise PDF $f_\eta(x) = \mathrm{sech}^2(x)/2$ with the scale parameter $1/2$ yields the CDF $F_\eta(x) = e^x/(e^x + e^{-x})$. Therefore, a number of common activation functions can be generated from the proposed probabilistic activation function defined in Eq. (2).

2.2. GEU and PGELU activation functions

Furthermore, some novel probabilistic activation functions can be derived form Eq. (2), because the injected noise PDF can be arbitrarily assigned. When the injected noise is assumed to have the Gaussian PDF $f_\eta(x) = \exp(-x^2/2\sigma^2)/\sqrt{2\pi\sigma^2}$ with variance σ^2, a general activation function of Eq. (2) can be specifically expressed as

$$g(x) = 1 - F_\eta(-x) = F_\eta(x) = \Phi(x/\sigma), \tag{5}$$

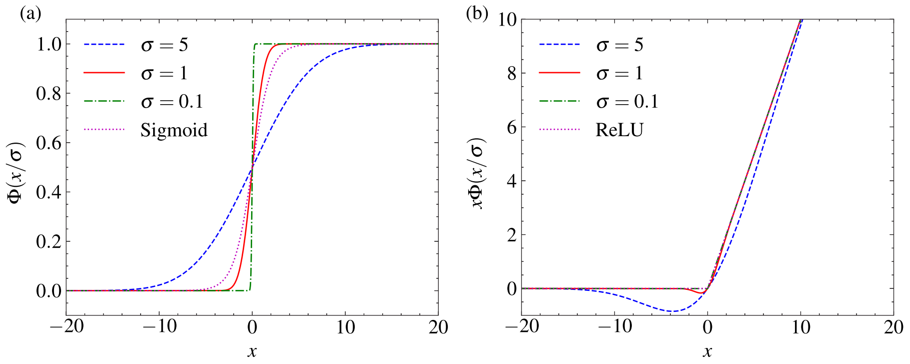


Fig. 2. (a) GEU $\Phi(x/\sigma)$ in Eq. (5) for various injected RMS noise levels. For comparison, the sigmoid activation function (dotted line) is also plotted. (b) PGELU $x\Phi(x/\sigma)$ in Eq. (7) for different injected RMS noise levels. The ReLU (dotted line) is also illustrated for comparison.

where $\Phi(x) = \int_{-\infty}^{x} \exp(-u^2/2)/\sqrt{2\pi}\ du = \frac{1}{2} + \frac{1}{2}\mathrm{erf}(x/\sqrt{2})$ denotes the CDF of a standard Gaussian distributed random variable and the Gauss error function $\mathrm{erf}(x) = 2\int_0^x e^{-t^2}dt/\sqrt{\pi}$. Here, this special activation function of Eq. (5) is called Gaussian error unit (GEU), which contains a learnable parameter σ of root-mean-square (RMS) noise. It is seen in Fig. 2(a) that, as the RMS noise decreases, the GEU becomes more similar to the threshold neuron with zero gradients. Nevertheless, for a large injected noise RMS $\sigma = 5$ (dashed line), the saturating regions of GEU in Eq. (5) appear at very large values of $|x| > 15$. For comparison, the input–output characteristic of the sigmoid activation function (dotted line) is also plotted in Fig. 2(a). In the network training procedure, the hyperparameter σ can adaptively learn by the gradient of the loss function with respect to itself, and then control the gradient flow for the variant of input.

We can multiply the probabilistic activation function $g(x)$ in Eq. (2) by the neuron input x to derive another general model

$$G(x) = xg(x) = x[1 - F_\eta(-x)], \tag{6}$$

where $g(x)$ acts a stochastic regularizer on the input x, viz. dropping the input x with a probability $g(x)$, as shown in Fig. 1(b). It is noted that some common unbounded activation functions can be also derived from Eq. (6). For instance, when the RMS noise $\sigma = 0$ (i.e. without the injected noise), the ReLU $\max(0, x) = xh(x)$ [1] is rediscovered by deterministically multiplying the input x with the McCulloch–Pitts neuron in Eq. (1). Multiplying the GEU of Eq. (5) by the input x, as shown in Fig. 2(b), the parametric Gaussian error linear unit (PGELU) is given by

$$G(x) = xg(x) = x\Phi(x/\sigma), \tag{7}$$

which reduces to GELU $x\Phi(x)$ [24] with the injected RMS noise $\sigma = 1$. It is seen in Fig. 2(b) that, in comparison with ReLU, PGELU exhibits curvature at all values of input x, and manifests a non-monotonic evaluation upon the increase of RMS noise, e.g. $\sigma = 5$ (dashed line). While, for a small RMS noise (e.g. $\sigma = 0.1$ illustrated by dashed–dotted line), PGELU in Eq. (7) almost reduces to ReLU (dotted line). In the following, we will show that the adaptively learning hyperparameter σ can improve the generalization by neural networks.

3. Injected RMS noise on the generalization by neural networks

We here focus on the interesting property of the learnable hyperparameter of RMS noise σ for improving generalization or preventing over-fitting by the designed neural networks with proposed activation functions in Eqs. (2) and (6).

3.1. Generalization by the designed neural network on function approximation

As a motivating example,[1] we first consider a neural network for fitting a sinusoidal function $\sin(\pi x)$ on observations $y = \sin(\pi x) + \xi$ corrupted by Gaussian background noise ξ. Here, ξ is with zero-mean and variance $\sigma_\xi^2 = 0.2^2$. The samples of the training set $\{x_i, y_i\}_{i=1}^{n=21}$ (●) in the interval $[-1, 1]$ is shown in Fig. 3(a). A fully connected $N \times K \times M$ neural network is employed and has an input layer with $N = 1$ linear neuron, one hidden layer containing 10 neurons of Eq. (2) and $M = 1$ linear neuron as the output layer. After 2×10^4 training epochs using the Adam optimizer [46], it is seen in Fig. 3(a) that the output (solid line) of the sigmoid network fits the noisy data well with a small mean square error (MSE) of 8.46×10^{-5}, but has poor generalization to new observation data. For instance, the trained sigmoid network presents higher MSEs in the order of 5×10^{-2}–8×10^{-2} for 5 samples of the testing set $\{x_j, y_j\}_{j=1}^{n=21}$.

Conversely, as shown in Fig. 3(b), the 1 × 10 × 1 fully connected neural network with $K = 10$ GEUs of Eq. (5) achieves a MSE of 3.01×10^{-2}, and provides a better fit to the true function of $\sin(\pi x)$ (dashed line) than the sigmoid network does (see Fig. 3(a)). For 5 samples of testing set $\{x_j, y_j\}_{j=1}^{n=21}$, the GEU network still attains the MSE in the range of 2×10^{-2}–4×10^{-2} as the same order as the MSE of 3.01×10^{-2} in training. Therefore, the generalization by the designed GEU network to new observation data is very effective. The reason for the superiority of the GEU network over the sigmoid network lies in the learnable hyperparameter σ that controls the input–output nonlinearity of the GEU in the hidden layer. It is shown in Fig. 3(c) that, during network training, the injected RMS noise σ starts from the initial value 16 and converges on a non-zero local optimum value 6.5655 (solid line). The trained GEU neuron $\Phi(x/6.5655)$ is not so sensitive to the variety of noisy data in a wide region of input, as indicated in Fig. 2(a), and the designed network yields a smooth output that approaches the target sinusoidal function. This is the reason for the generalization improvement in the GEU network for function approximation.

From Eqs. (2) and (3), it is natural to ask whether introducing the injected noise scale parameter σ of the logistic noise η into the sigmoid activation function of Eq. (3) can improve the generalization by the sigmoid network? The answer is positive. Since the logistic noise η has its PDF $f_\eta(x) = e^{-x/\sigma}/[\sigma(1 + e^{-x/\sigma})^2]$ with variance $\sigma^2\pi^2/3$, then a variant sigmoid activation function

$$g(x) = 1 - F_\eta(-x) = \left(1 + e^{-\frac{x}{\sigma}}\right)^{-1} \tag{8}$$

can be deduced from Eq. (2). We also train the 1 × 10 × 1 neural network with $K = 10$ activation functions of Eq. (8) by noisy observations. It is illustrated in Fig. 3(a) that the output (dashed–dotted

[1] Source code: https://github.com/YuuhoRen/GEUactivation.

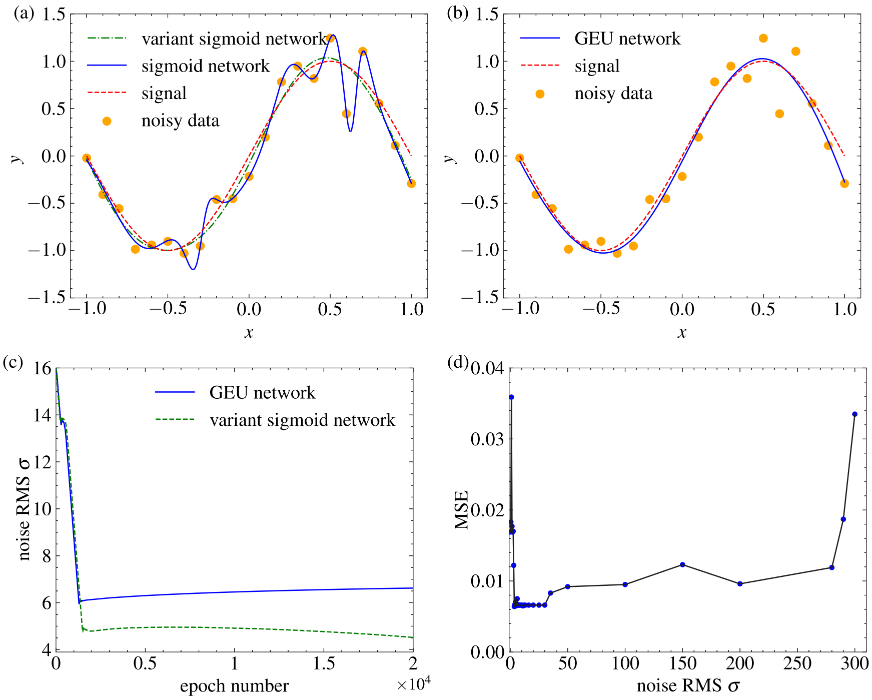


Fig. 3. Outputs of (a) the sigmoid network (solid line), the variant sigmoid network (dashed–dotted line) indicated in Eq. (8) and (b) the GEU network (solid line) indicated in Eq. (5). For comparison, the target function $\sin(\pi x)$ (dashed line) and the observations (•) are also plotted. (c) Learning curve of the injected RMS noise σ of the GEU network in the training. (d) Stochastic resonance effect of the testing MSE versus the injected RMS noise σ for the GEU network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

line) also demonstrates the role of the hyperparameter σ in enhancing the generalization of the designed network. The corresponding MSE becomes 2.99×10^{-2} in the training and the injected RMS noise σ converges to 4.5190 (dashed line), as shown in Fig. 3(c).

The presence of non-zero converged RMS noise σ in Fig. 3(c) provides evidence for the occurrence of adaptive stochastic resonance effect in the designed neural network for function approximation. To further elucidate this effect, we plot the testing MSE of the designed GEU network as a function of the injected RMS noise σ in Fig. 3(d). Here, each point of testing MSE is obtained by fixing noise RMS σ but training network weights with the backpropagation learning rule. As depicted in Fig. 3(d), it can be observed that the optimized noise RMS σ is consistent with the converged value of 6.5655 as shown in Fig. 3(c). Significantly, the testing MSE of the designed GEU network exhibits an increasing trend for both high and low values of the noise RMS σ. This is the typical resonance curve of the testing MSE versus the RMS noise σ, which also demonstrates the practicality of the stochastic resonance effect in the domain of machine learning.

3.2. Theoretical explanation of the learnable RMS noise on the Rademacher complexity

It is clearly seen in Fig. 3(c) that the converged RMS noise $\sigma > 1$ is far beyond the theoretical explanation of the injected noise as a generalized Tikhonov regularizer for training the designed threshold network [12], which only holds for the assumption of an infinitesimal injected noise variance. Then, the complexity measure that monotonically relates to the generalization error needs to be analyzed for these neural networks with the proposed general activation functions. Here, we utilize the path-based norm to analyze how the learnable hyperparameter σ does control the Rademacher complexity that measures the degree to which a hypothesis set can fit random noise [42–45,47,48].

Let $S = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ denote the training set sampled from the observations $y_i = s(\boldsymbol{x}_i) + \xi_i$, where s is the target function with its domain in $[0, 1]$ and the d-dimensional input vector $\boldsymbol{x} \in \mathbb{R}^d$ [43]. The background white noise ξ is with zero-mean and finite variance σ_ξ^2. Let $\psi_m(\boldsymbol{x}, \theta) = \sum_{k=1}^m a_k g(\boldsymbol{w}_k \boldsymbol{x} + b_k, \sigma)$ represent the three-layer network model with the learnable parameter set $\boldsymbol{\theta} = \{\{a_k, \boldsymbol{w}_k, b_k\}_{k=1}^m, \sigma\}$ and m neurons in the hidden layer. Here, $\boldsymbol{w}_k$ denotes the weight vector in the input layer, b_k is the bias parameter for the kth neuron in the hidden layer, and the weight a_k connects the kth neuron and the single linear neuron in the output layer. Consider the truncated square loss $\ell(\boldsymbol{x}, y, \boldsymbol{\theta}) = [\mathcal{T}(\psi) - y]^2/2$ with the truncation operator $\mathcal{T}(z) = \min\{\max\{z(\boldsymbol{x}), 0\}, 1\}$ for any function $z : \mathbb{R}^d \mapsto \mathbb{R}$ [42,43]. Here, the expected loss $\mathcal{L}(\boldsymbol{\theta})$ and the empirical loss $\hat{\mathcal{L}}_n(\boldsymbol{\theta})$ of the network model are given by

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x},y}[\ell(\boldsymbol{x}, y, \boldsymbol{\theta})],\ \hat{\mathcal{L}}_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{x}_i, y_i, \boldsymbol{\theta}). \tag{9}$$

Then, the generalization error between $\mathcal{L}(\boldsymbol{\theta})$ and $\hat{\mathcal{L}}_n(\boldsymbol{\theta})$ is related to the Rademacher complexity [44,45]

$$\hat{\mathcal{R}}_S(\mathcal{F}) = \mathbb{E}_{\zeta}\left[\sup_{\psi\in\Psi} \frac{1}{n}\sum_{i=1}^{n} \zeta_i \psi(\boldsymbol{x}_i)\right] \tag{10}$$

with respect to the data set S and for a family of functions $\psi \in \Psi$. Here, $\boldsymbol{\zeta} = \{\zeta_i\}_{i=1}^n$ are the independent Rademacher random variables uniformly taking values in $\{-1, 1\}$ [44,45]. For any $0 < \delta < 1$, with probability at least $1 - \delta$ over the draw of a sample S of size n, it is found [44,45] that

$$\hat{\mathcal{L}}_n(\boldsymbol{\theta}) \le \mathcal{L}(\boldsymbol{\theta}) + 2\mathbb{E}_S[\hat{\mathcal{R}}_S(\Psi)] + B\sqrt{\frac{2\ln(2/\delta)}{n}} \tag{11}$$

holds for all $\psi \in \Psi$ and $|\psi(x)| < B$ for the upper limit $B > 0$ [44,45].

Furthermore, it is proved in [43] that, for an arbitrary continuous and two-order differentiable activation function $g : \mathbb{R} \mapsto \mathbb{R}$, there exists a three-layer ReLU neural network $\psi^o(\boldsymbol{x}; \theta)$ with an finite width m (i.e. the neuron number of hidden layer), such that $\sup_{x\in\mathbb{R}} |g(x) - \psi^o(\boldsymbol{x}; \theta)| \le \varepsilon$, $\|\theta\|_p \le \gamma(g) + \varepsilon$ for an arbitrary constant $0 < \varepsilon \ll 1$. Here, the path norm $\|\theta\|_p$ $(p \ge 1)$ of $\psi^o(\boldsymbol{x}; \theta)$ is defined as [42] $\|\theta\|_p = \sum_{i=1}^m |a_i|(\|\boldsymbol{w}_i\|_1 + |b_i|)$ with its upper bound [43]

$$\gamma(g) = \int_{\mathbb{R}} |g''(x)|(|x| + 1)dx + \inf_{x\in\mathbb{R}} |g(x)| + (|x| + 2)|g'(x)|. \tag{12}$$

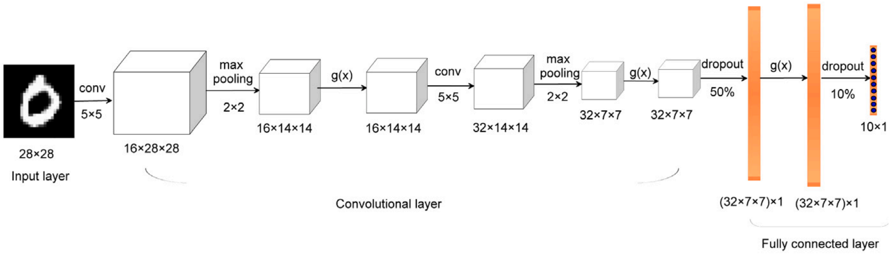


Fig. 4. Block diagram representations of the convolutional neural network architecture in Table 1 for MNIST classification.

For such three-layer network $\psi(\boldsymbol{x}, \theta)$ and $\|\theta\|_p + \sum_{k=1}^{m} |a_k| \leq Q$ with a certain $Q > 0$, the Rademacher complexity satisfies [43]

$$\hat{\mathcal{R}}_S(\Psi) \leq 2\gamma(g)Q\sqrt{\frac{2\ln(2d)}{n}}. \tag{13}$$

From Eqs. (11)–(13), we see that the generalization error of the network model with the general activation functions $g(x)$ is closely related to the upper bound $\gamma(g)$ in Eq. (12). In particular, the norm bound $\gamma(g) = 1$ for the ReLU, $\gamma(g) = 1 + \alpha$ for leaky ReLU $g(x) = \max\{\alpha x, x\}$ as $\alpha \in [0, 1)$, and $\gamma(g) = 1.5$ for the sigmoid activation function [43].

In Appendix, we prove that, if the second-order derivative $g''(x) > 0$ $(x < 0)$ and $g''(x) < 0$ $(x > 0)$, then the upper bound $\gamma(g)$ in Eq. (12) of the proposed activation function $g(x)$ in Eq. (2) can be computed as

$$\gamma(g) = 1 + 2f_\eta(0). \tag{14}$$

For the GEU activation function in Eq. (5), $g''(x) = -xe^{-\frac{x^2}{2\sigma^2}}/(\sqrt{2\pi}\sigma^3)$ and the upper bound $\gamma(g) = 1 + \sqrt{2}/(\sqrt{\pi}\sigma)$. It is also interesting to note that, at the converged RMS noise $\sigma = 6.5655 \gg 1$ shown in Fig. 3(c), the corresponding norm bound $\gamma_\sigma(g) = 1.076$ of GEU is less than $\gamma(g) = 1.5$ of the sigmoid activation function in Eq. (3). Similarly, for the variant sigmoid activation function in Eq. (8), $g''(x) = (e^{\frac{x}{\sigma}} - e^{\frac{2x}{\sigma}})/[\sigma^2(1 + e^{\frac{x}{\sigma}})^3]$ and the upper bound $\gamma(g) = 1 + 1/(2\sigma)$. It is seen in Fig. 3(c) that the converged RMS noise $\sigma = 4.5190$ and the corresponding norm bound $\gamma_\sigma(g) = 1.1106$ is also less than $\gamma(g) = 1.5$ of the sigmoid activation function. Thus, the Rademacher complexity of the designed GNE network can be controlled at a lower level, and the generalization by the designed neural network, as shown in Fig. 3(b), can be improved as the learnable RMS noise $\sigma > 1$.

3.3. Learnable RMS noise on image classification

Unfortunately, the Rademacher complexity of a neural network with general activation functions based on the path norm [42,43] is not easily extended to deep neural networks, because the depth-dependent capacity control of the deep neural network is still unresolved [42–45, 48]. Therefore, here, we mainly experimentally evaluate and compare deep neural networks with the proposed and with the traditional activation functions on MNIST classification (gray images with 10 classes, 6×10^4 training and 10^4 testing examples) [49], CIFAR-10 classification (color images with 10 classes, 5×10^4 training and 10^4 testing examples) [50] and CIFAR-100 classification (color images with 100 classes, 500 training images and 100 testing images per class) [50].

A deep convolutional neural network is designed with the architecture shown in Fig. 4, wherein the activation function $g(x)$ in three layers can be selected from Sigmoid, ReLU, PReLU or GEU, respectively. For the activation function $g(x)$ of GEU in Eq. (5) or PGELU in Eq. (7), the convolutional layer includes two RMS noises σ_1 and σ_2, and the fully connected layer has one σ_3 to be learned in the training. The training epochs is 20, the batch size takes 128 and the Adam optimizer [46] is used to optimize both weights and the RMS noise levels [13]. The learning rate keeps 0.002 for the total training examples. The Xavier initialization [51] is adapted to initialize weights of the convolutional neural network such that the variance of the activations are the same across each layer. In order to validate the generalization by the neural network, the 6×10^4 training examples are normalized by subtracting the mean (0.1307) and dividing by the standard deviation (0.3081) [49, 52].

After normalization, the training images are corrupted by Gaussian noise with different variances. Here, the averaged testing accuracies are obtained for 5 trails, and the same is for the following testing results. It is listed in Table 1 that the PGELU and GEU neural networks achieve higher testing accuracies in comparison with networks consisting of other activation functions. Here, PReLU indicates the leaky ReLU with adaptively learning parameter of the rectifiers [53]. For instance, without the corrupting noise, Fig. 5(a) and (b) illustrate the train and the test accuracies, respectively. In fact, for the corrupting noise variances 0, 1 and 2, the PGELU neural networks perform better slightly. But, the ReLU convolutional neural network is invalidated with a low testing accuracy 79.63% at the large corrupting noise variance 5, as given in Table 1. In similar situations, the GEU neural network still have a high accuracy of 93.97%, which performs better by 1.60% compared with the PGELU network.

Moreover, for the GEU convolutional neural network in Fig. 4, the injected RMS noise levels are initialized to the constant 2.1 and all converges to values larger than unity, for instance, $\sigma_1 = 3.3860$, $\sigma_2 = 3.9697$ and $\sigma_3 = 3.7796$ for training images with the corrupting noise variances 5. Thus, for classifying gray images in MNIST data set, the saturating activation function of GEU, assisted by the learnable injected RMS noise σ, is demonstrated to be more robust to the noisy input. Although the capabilities are difficult to be theoretically analyzed for the deep convolutional neural network, the experimental results also demonstrate the conclusion indicated in Eq. (13) that a large RMS noise can reduce the saturating regions of the activation function and then extend the generalization by the designed neural network.

We further demonstrate the positive role of the injected RMS noise for classifying color images. The general activation functions of GEU in Eq. (2) and PGELU in Eq. (5) are evaluated by CIFAR-10 data set on the ResNet network with the deep residual learning framework [53], wherein one RMS noise is initialized to the constant 5 outside the residual blocks and 12 RMS noise levels are set to the constant 3 in the residual blocks. The detailed architecture of the ResNet network[2] is shown in Fig. 6, and all the parameters only occupy about 0.2-Mb of computer RAM (for comparison, the ResNet-18 network occupies about 11.4-Mb of memory). The batch size is 100 and the Adam optimizer is used to optimize both weights and RMS noise levels in 80 epochs. The initial learning rate takes 0.001, and decreases to a third of the last one after 20 epochs. Since CIFAR-10 is an established computer-vision data set used for object recognition [50] in real world, then these color images themselves contain background noise. So, no corrupted noise is added to the data set in trails.

[2] https://github.com/pytorch/vision/blob/main/torchvision/models/regnet.py;
https://github.com/YuuhoRen/GEUactivation

Table 1
Testing accuracy (%) versus corrupting noise variance in convolution networks.

Activation function \ Corrupting noise variance	0	1	2	3	5
Sigmoid	98.95	98.88	98.62	96.24	91.70
ReLU	99.15	99.13	98.08	94.70	79.63
GELU	99.22	99.04	98.66	95.76	91.17
PReLU	99.18	99.17	98.56	96.42	90.26
PGELU	**99.27**	**99.19**	**98.67**	96.21	91.75
GEU	99.06	99.01	98.59	**97.42**	**93.35**

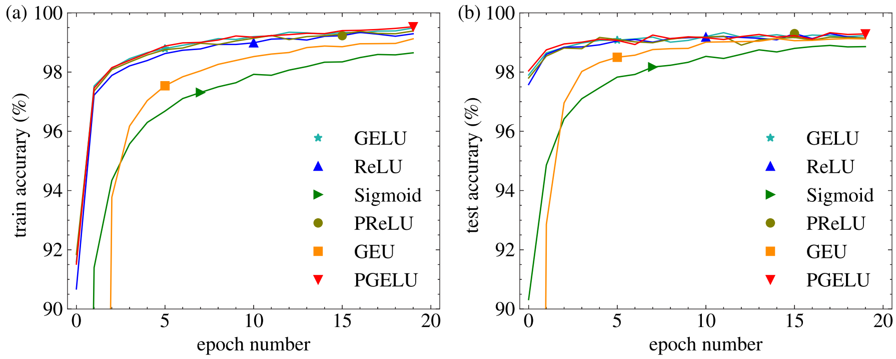


Fig. 5. (a) Train and (b) test accuracies for convolutional neural networks with different activation functions on MNIST classification.

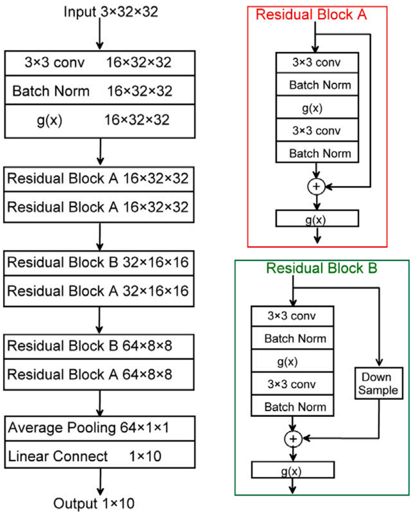


Fig. 6. Diagram representation of the architecture of the ResNet.

Table 2
Train and test accuracies (%) for various activation functions on the ResNet network.

Activation function	Sigmoid	GEU	ReLU	PReLU	GELU	PGELU
Train accuracy	70.48	70.87	89.81	95.43	95.07	**96.90**
Test accuracy	68.63	69.21	84.41	87.69	87.50	**88.58**

In Fig. 7(a) and (b), the train and the test accuracies are shown at each epoch for ResNets constructed by various activation functions, respectively. After 80 epoches of training, the test accuracies of the trained ResNets are listed in Table 2 for these considered ResNets. It is obviously seen in Table 2 that, for recognizing color images, the networks composed of unbounded nonlinearities of ReLU, PReLU, GELU and PGELU have higher testing accuracies in comparison with that of networks with the saturating activation functions of sigmoid and GEU. This indicates that, for the unbounded nonlinearities in the positive region of the input $x \in \mathbb{R}$, the non-vanishing gradient plays an important role during training. However, whatever the injected RMS noise increases, GEU still has the vanishing gradient in its saturating regions for a very large input x. This is an essential difference between the two types of activation functions. It is also indicated in Table 2 that, among four unbounded activation functions, the PGELU network achieves the best testing accuracy of 88.58%. It is also interestingly noted in Fig. 8 that all converged values of injected RMS noise levels of the trained PGELU network are larger than unity. As illustrated in Fig. 2(b), the larger the injected RMS noise is, the negative feedback for different values of the input will be more obvious. Thus, with these converged RMS noise levels larger than unity, the features of color images represented in the negative region are better preserved by the designed PGELU network, resulting in a high testing accuracy of image classification.

We applied a light-weighted ResNet to the CIFAR-100 dataset by pruning a residual block of the ResNet network, as illustrated in Fig. 6. Initially, we set 11 RMS noise levels to a constant value of 2.5. After 30 epochs of training, the PGELU ResNet achieved the highest test accuracy of 75.85%. This result outperformed ResNets with other activation functions, such as ReLU ResNet and PRELU ResNet, which achieved test accuracies of 74.25% and 74.00%, respectively. The curves of the test accuracies at each epoch are illustrated in Fig. 9(a) for ReLU, PReLU, GELU and PGELU ResNets, respectively. Moreover, it can be observed in Fig. 9(b) that, after training the PGELU ResNet, four converged RMS noise levels exceeded unity, while the remaining seven were only slightly smaller than unity. This observation highlights the beneficial impact of learnable RMS noises on enhancing the generalization performance of the designed neural network.

4. Conclusion

In this paper, we propose a flexible probabilistic activation function based on the mechanism of adaptive stochastic resonance that intelligently exploits the constructive role of injected noise. Under the guidance of this view, many traditional activation functions, such as sigmoid, tanh and ReLU, can be deduced form this general probabilistic model indicated in Fig. 1. In addition, the saturating (e.g. GEU) and the unbounded type (e.g. PGELU) of activation functions are specifically derived with the learnable Gaussian RMS noise level updated in the network training. For illustration, in the present work, the proposed flexible activation functions demonstrate how to associate the binary McCulloch–Pitts neuron model [40,41] with common activation functions in deep learning. Experimental results on the generalization by

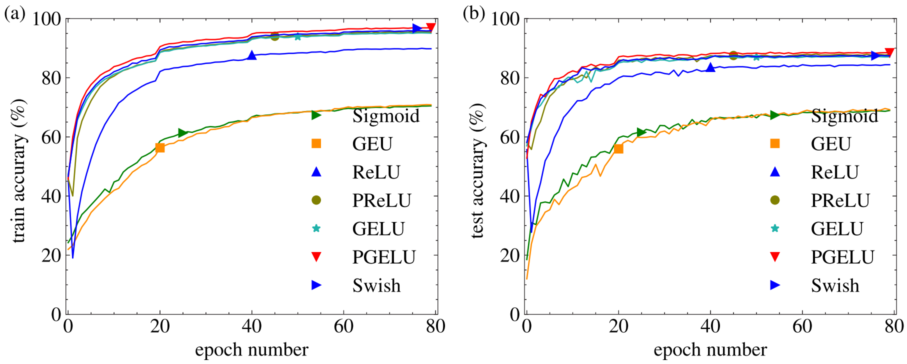


Fig. 7. (a) Train and (b) test accuracies for ResNet neural networks with different activation functions on CIFAR-10 classification.

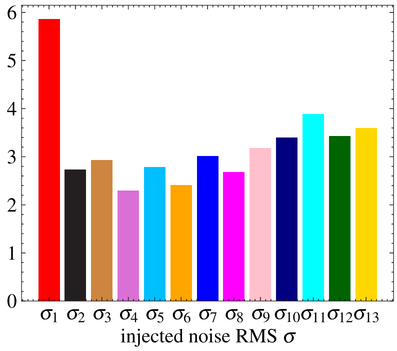


Fig. 8. Injected RMS noise levels of the trained PGELU ResNet network on CIFAR-10 dataset.

the designed neural network show that the GEU and PGELU neural networks outperform the neural networks with traditional activation functions in the testing accuracies of benchmark data sets, as the injected RMS noise levels of GEU (or PGELU) usually converge to local optimum values larger than unity. This performance does not resort to the explanation of the small injected noise variance acting as the Tikhonov regularizer [12,13]. Based on the path-norm upper bound, it is demonstrated that the large RMS noise trained by the designed GEU neural network can reduce the Rademacher complexity that measures the fitting capacity of the network. Thus, it provides a way to avoid overfitting, through improving the generalization of the designed neural network by the introduction of injected noise.

Some open questions still remain. The weights of the networks are usually initialized by Kaiming [19] or Xavier initialization [51]. The optimal initialization of the injected RMS noise levels in the GEU and PGELU neural networks is still an open problem. In practical trials, the initial value of the RMS noise is empirically chosen to be larger than unity, and the generalization capability of the trained neural network is found to be much improved for the noisy input. However, how large the injected RMS noise initially should be cannot be determined, especially for the case of a number of injected RMS noise levels in the deep neural network. Moreover, from viewpoints of sufficiently fast and easy implementation, the designed neural networks with the proposed GEU and PGELU activation functions, in comparison with the ReLU network, require more time for computing and the integral operation of GEU in Eq. (2) is unfavorable for hardwired implementation. It is noted in the probabilistic model of Fig. 1 that the proposed activation function can be asymptotically implemented by injecting a large number of mutually independent noise components into binary McCulloch–Pitts neurons. This paper mainly focuses on the complexity capacity of the designed network. It is noted that, in the practical testing experiments, the probabilistic activation function indicated in Fig. 1 can be mimicked by a finite number of threshold elements operated in the same noisy environment. The noise-smoothed threshold convolutional neural networks has been implemented by decoupling into a finite set of threshold functions driven by mutually independent noise components, which endows a hardware-friendly feature of the designed neural network [54]. Thus, the practical realizations of GEU and PGELU can be constructed by easily implemented binary units plus noise, and this motivates exploration of the generalization capability of deep neural networks for future study. Another natural question is whether the proposed PGELU neural network can potentially perform well to classify the 1.2 million high-resolution images in the ImageNet [55]. For this question, designing deeper neural networks with more learnable noise parameters is a challenging task.

Furthermore, when fitting observations of the sinusoidal function depicted in Fig. 3(a) using a one-hidden layer network, the optimized RMS noise σ facilitates the temporal scale of the network output to align with the frequency of the sinusoidal input. As the number of layers increases, can the designed neural network with a number of noise RMS values of σ, effectively match multi-scale high dimensional input data? This is worth to be deeply investigated in the future. In the context of image classification using a designed deep neural network, the role of injected noise can be analyzed from the perspective of the information transfer, which is closely related to the mechanism of suprathreshold stochastic resonance [33] and the information botttleneck theory in deep learning [56]. After network training, it is of interest to further demonstrate that the optimized RMS σ can maximize the mutual information between the layer output and the desired prediction, while simultaneously compressing the high-dimensional input as much as possible.

CRediT authorship contribution statement

Fabing Duan: Conceptualization, Data curation, Investigation, Methodology, Writing – original draft, Formal analysis. **François Chapeau-Blondeau:** Conceptualization, Data curation, Methodology, Writing – review & editing. **Derek Abbott:** Data curation, Formal analysis, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the Natural Science Foundation of Shandong Province, China (ZR2021MF051), and the Taishan Scholar Project of Shandong Province (TS20190930), China.

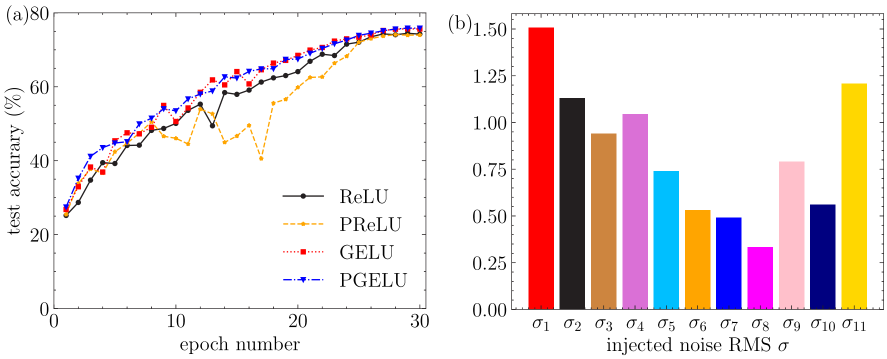


Fig. 9. (a) Test accuracies for ResNet neural networks with different activation functions, and (b) the injected RMS noise levels of the trained PGELU ResNet network on CIFAR-100 classification.

Appendix. Upper bound of path norm

From Eq. (2), we find $g'(x) = f_\eta(-x) \geq 0$ as $x \in \mathbb{R}$. If $g''(x) > 0$ $(x < 0)$ and $g''(x) < 0$ $(x > 0)$, then the first term of Eq. (12) can be calculated as

$$\begin{aligned}\int_{\mathbb{R}} |g''(x)|(|x|+1)dx &= -\int_0^{\infty} g''(x)(x+1)dx + \int_{-\infty}^{0} g''(x)(-x+1)dx \\ &= \lim_{x\to\infty}[g(x) - g'(\infty)x] - \lim_{x\to-\infty}[g(x) - g'(-\infty)x] \\ &\quad -[g'(\infty) + g'(-\infty)] + 2g'(0) \\ &= 1 + 2f_\eta(0).\end{aligned}$$

Here, noting the regularity conditions of CDF F_η and PDF f_η, i.e. $F_\eta(-\infty) = 0$, $F_\eta(+\infty) = 1$, $f_\eta(x) \geq 0$ and $\lim_{x\to\pm\infty} f_\eta(x) = 0$, we can find $g'(\pm\infty) = 0$, $g(-\infty) = 1 - F_\eta(\infty) = 0$, $g(\infty) = 1$ and $g'(0) = f_\eta(0)$. Furthermore, the second term of Eq. (12) becomes

$$\begin{aligned}\inf_{x\in\mathbb{R}} |g(x)| + (|x|+2)|g'(x)| &= \inf_{x\in\mathbb{R}} 1 - F_\eta(-x) + (|x|+2)f_\eta(-x) \\ &= \lim_{x\to-\infty} 1 - F_\eta(-x) + (|x|+2)f_\eta(-x) = 0.\end{aligned}$$

References

[1] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: 27th International conference on machine learning. San Juan, Puerto Rico, 2010, p. 807–14.

[2] Gulcehre C, Moczulski M, Denil M, Bingio Y. Noisy activation functions. 2016, arXiv, Available: https://arxiv.org/abs/1603.00391v3.

[3] Sietsma J, Dow R. Neural network pruning–why and how. In: Proceeding of IEEE international conference of neural networks, I. San Diego, CA, USA; 1988, p. 325–33.

[4] Sietsma J, Dow R. Creating artificial neural networks that generalize. Neural Netw 1991;4(1):67–79.

[5] Holmström L, Koistinen P. Using additive noise in back-propagation training. IEEE Trans Neural Netw 1992;3(1):24–38.

[6] Matsuoka K. Noise injection into inputs in back-propagation learning. IEEE Trans Syst Man Cybern 1992;22(3):436–40.

[7] Bartlett PL, Downs T. Using random weights to train multilayer networks of hard-limiting units. IEEE Trans Neural Netw 1992;3(2):202–10.

[8] Grandvalet Y, Canu S. Noise injection: Theoretical prospects. Neural Comput 1997;9(5):1093–108.

[9] Bohorquez J, Lambert MF, Alexander B, Simpson AR, Abbott D. Stochastic resonance enhancement for leak detection in pipelines using fluid transients and convolutional neural networks. J Water Resour Plan Manag 2022;148(3).

[10] Orvieto A, Kersting H, Proske F, Bach F, Lucchi A. Anticorrelated noise injection for improved generalization. In: Proceedings of the 39th international conference on machine learning, vol. 162. Baltimore, Maryland; 2022, p. 17094–116.

[11] Orvieto A, Raj A, Kersting H, Bach F. Explicit regularization in overparametrized models via noise injection. 2023, arXiv, Available: https://arxiv.org/pdf/2206.04613.

[12] Bishop CM. Training with noise is equivalent to Tikhonov regularization. Neural Comput 1995;7:108–16.

[13] Bai S, Duan F, Chapeau-Blondeau F, Abbott D. Generalization of stochastic-resonance-based threshold networks with tikhonov regularization. Phys Rev E 2022;106(1):L012101.

[14] Duan L, Duan F, Chapeau-Blondeau F, Abbott D. Noise-boosted backpropagation learning of feedforward threshold neural networks for function approximation. IEEE Trans Instrum Meas 2021;70:3121502.

[15] Ikemoto S, Dallalibera F, Hosoda K. Noise-modulated neural networks as an application of stochastic resonance. Neurocomputing 2017;277:29–37.

[16] Uhlich S. Bayes risk reduction of estimators using artificial observation noise. IEEE Trans Signal Process 2015;63(20):5535–45.

[17] Reed R, Marks RJ. Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitter. IEEE Trans Neural Netw 1995;6(3):529–38.

[18] An G. The effects of adding noise during backpropagation training on a generalization performance. Neural Comput 1996;8(3):643–74.

[19] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: IEEE international conference on computer vision. (ICCV), Santiago, Chile; 2015, p. 1026–34.

[20] Agostinelli F, Hoffman MD, Sadowski PJ, Baldi P. Learning activation functions to improve deep neural networks. In: Bengio Yoshua, LeCun Yann, editors. 3rd International conference on learning representations, San Diego, CA, USA, Workshop track proceedings. 2015, Available: http://arxiv.org/abs/1412.6830.

[21] Balaji S, Kavya T, Sebastian N. Learnable parameter guided activation functions. 2019, http://dx.doi.org/10.48550/arXiv.1912.10752, arXiv, Available.

[22] Goyal M, Goyal R, Lall B. Learning activation functions: A new paradigm of understanding neural networks. 2020, http://dx.doi.org/10.48550/arXiv.1906.09529, arXiv, Available.

[23] Maniatopoulos A, Mitianoudis N. Learnable leaky ReLU (LeLeLU): An alternative accuracy-optimizaed activation function. Information 2021;12:513.

[24] Hendrycks D, Gimpel K. Gaussian error linear units (GELus). 2020, http://dx.doi.org/10.48550/arXiv.1606.08415, arXiv.

[25] Peng Y, Xiao L, Heidergott B, Hong L, Lam H, H. A new likelihood ratio method for training artificial neural networks. INFORMS J Comput 2022;34(1):638–55.

[26] Xiao L, Zhang Z, Jiang J, Peng Y. Noise optimization in artificial neural networks. In: IEEE 18th international conference on automation science and engineering. 2022, p. 1595–600.

[27] Chen L, An K, Huang D, Wang X, Xia M, Lu S. Noise-boosted convolutional neural network for edge based motor fault diagnosis with limited samples. IEEE Trans Ind Inf 2023;19:9491–502.

[28] Benzi R, Sutera A, Vulpiani A. The mechanism of stochastic resonance. J Phys A: Math Gen 1981;14(11):L453–7.

[29] Chapeau-Blondeau F, Godivier X. Theory of stochastic resonance in signal transmission by static nonlinear systems. Phys Rev E 1997;55:1478–95.

[30] Chapeau-Blondeau F, Rousseau D. Noise-enhanced performance for an optimal Bayesian estimator. IEEE Trans Signal Process 2004;52(5):1327–34.

[31] McDonnell MD, Abbott D. What is stochastic resonance? definitions, misconceptions, debates, and its relevance to biology. PLoS Comput Biol 2009;5(5):e1000348.

[32] Kosko B, Audhkhasi K, Osoba O. Noise can speed backpropagation learning and deep bidirectional pretraining. Neural Netw 2020;129:359–84.

[33] Stocks NG. Suprathreshold stochastic resonance in multilevel threshold systems. Phys Rev Lett 2000;84(11):2310–3.

[34] McDonnell MD, Stocks NG, Pearce CEM, Abbott D. Stochastic resonance: From suprathreshold stochastic resonance to stochastic signal quantization. New York: Cambridge University Press; 2008.

[35] Fu Y, Chen G G. Stochastic resonance based visual perception using spiking neural networks. Front Comput Neurosci 2020;14:24.

[36] Liao Z, Wang Z, Yamahara H, Tabata H. Low-power-consumption physical reservoir computing model based on overdamped bistable stochastic resonance system. Neurocomputing 2022;468:137–47.

[37] Andò B, Baglio S, Bulsara A, Marletta V. A nonlinear energy harvester operated in the stochastic resonance regime for signal detection/measurement applications. IEEE Trans Instrum Meas 2020;69(8):5930–40.

[38] Liao Z, Ma K, Sarker MS, Yamahara H, Seki M, Tabata H. Quadstable logical stochastic resonance-based reconfigurable boolean operation subjected to heavy noise floor. Results Phys 2022;42:105968.

[39] Zhao S, Shi P. A novel piecewise tri-stable stochastic resonance system driven by dichotomous noise. Sensors 2023;23(2):1022.

[40] McCulloch W, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943;5(4):115–33.

[41] Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci 1982;79(8):2554–8.

[42] Neyshabur B, Tomioka R, Srebro N. Norm-based capacity control in neural networks. 2015, http://dx.doi.org/10.48550/arXiv.1503.00036, arXiv, Available.

[43] Li Z, Ma C, Wu L. Complexity measures for neural networks with general activation functions using path-based norms. 2020, http://dx.doi.org/10.48550/arXiv.2009.06132, arXiv, Available.

[44] Shalev-Shwartz S, Ben-David S. Uderstanding machine learning: From theory to algorithm. New York, USA: Cambridge University Press; 2014.

[45] Mohri M, Rostamizadeh A, Talwalkar A. Foundations of machine laerning. Cambridge, MA: The MIT Press; 2018.

[46] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: 3rd International conference on learning representations. ICLR, San Diego, CA, USA; 2015, p. 7–9.

[47] Bartlett PL, Mendelson S. Rademacher and Gaussian complexities: Risk counds and structural eesults. J Mach Learn Res 2002;3:463–82.

[48] Jiang Y, Neyshabur B, Mobahi H, Krishnan D, Bengio S. Fantastic generalization measures and where to find them. 2020, http://dx.doi.org/10.48550/arXiv.1912.02178, arXiv, Available.

[49] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86(11):2278–324.

[50] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Technical Report, University of Toronto; 2009.

[51] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, Chia Laguna Resort, Sardinia, Italy. 2010, p. 249–56.

[52] Hoffman J, Roberts DA, Yaida S. Robust learning with jacobian regularization. 2019, http://dx.doi.org/10.48550/arxiv.1908.02729, arXiv, Available.

[53] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition. (CVPR), Las Vegas, NV, USA; 2016, p. 770–8.

[54] Duan L, Ren Y, Duan F. Adaptive stochastic resonance based convolutional neural network for image classification. Chaos Solitons Fractals 2022;162:112429.

[55] Deng J, Dong W, Socher R, Li L-J, Li K, Li FF. ImageNet: A large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition, Miami, FL, USA. 2009, p. 248–55.

[56] Tishby N, Zaslavsky N. Deep learning and the information bottleneck principle. In: IEEE information theory workshop (ITW), Jerusalem, Israel. 2015, p. 1–5.