q-Softplus Function: Extensions of Activation Function and Loss Function by Using q-Space

Motoshi Abe(✉) and Takio Kurita

Hiroshima University, Higashi-Hiroshima, Japan
i13abemotoshi@gmail.com, tkurita@hiroshima-u.ac.jp

Abstract. In recent years, the performance of machine learning algorithms has been rapidly improved because of the progress of deep learning. To approximate any non-linear function, almost all models of deep learning use the non-linear activation function. Rectified linear units (ReLU) function is most commonly used. The continuous version of the ReLU function is the softplus function and it is derived by the integration of the sigmoid function. Since a sigmoid function is based on Gaussian distribution, the softplus activation function is also based on Gaussian distribution. In machine learning and statistics, most techniques assume the Gaussian distribution because Gaussian distribution is easy to handle in mathematical theory. For example, the exponential family is often assumed in information geometry which connects various branches of mathematical science in dealing with uncertainty and information based on unifying geometric concepts. The q-space is defined to extend this limitation of information geometry. On the q-space, q-multiplication, q-division, q-exponential, and q-logarithm are defined with hyperparameter q as a natural extension of multiplication and division, etc. in general space. In this paper, we propose to extend the activation function and the loss function by using q-space. By this extension, we can introduce hyperparameter q to control the shape of the function and the standard softplus function can be recovered by setting the hyperparameter $q = 1$. The effectiveness of the proposed q-softplus function, we have performed experiments in which the q-softplus function is used for the activation function of a convolutional neural network instead of the ReLU function and the loss function of metric learning Siamese and Triplet instead of max function.

Keywords: q-Space · Activation function · q-Softplus

1 Introduction

In recent years, the performance of the machine learning algorithms has been rapidly improved. Many techniques of machine learning are proposed such as support vector machine [24], neural network [4], convolutional neural network [6], and so on. Since these models can approximate any non-linear function, they are effective for classification [11,13,20,21], person recognition [7,10], object detection [25], and so on.

C. Wallraven et al. (Eds.): ACPR 2021, LNCS 13189, pp. 31–44, 2022.
https://doi.org/10.1007/978-3-031-02444-3_3

To approximate any non-linear function, almost all models of deep learning use the non-linear activation function. Rectified linear units (ReLU) function is most commonly used as non-linear activation function of the hidden layers in the deep learning models. Sigmoid function or softmax function is often used as a non-linear activation function in the output layer of the deep learning models.

The continuous version of the ReLU function is softplus function and it is derived by the integration of the sigmoid function. The sigmoid function and softmax function are defined by using exponential function and have a close relation with Gaussian distribution. It means that the input of the sigmoid or softmax function is assumed to be a Gaussian distribution. Exponential linear units (ELU) [19], Sigmoid-weighted linear unit (SiLU) [9], swish [18], and mish [16] have been proposed as extension of ReLU function. The such activation functions are derived from ReLU function or sigmoid function.

In machine learning and statistics, most techniques assume the Gaussian distribution for prior distribution or conditional distribution because Gaussian distribution is easy to handle in mathematical theory. For example, the exponential family is often assumed in information geometry which connects various branches of mathematical science in dealing with uncertainty and information based on unifying geometric concepts. In information geometry, it is famous that the exponential family is flat under the e-connection. The Gaussian distribution is a kind of the exponential family.

However, some famous probability distributions, such t-distribution, is not exponential family. As an extension of information geometry, q-space is defined [22]. On the q-space, q-multiplication, q-division, q-exponential, and q-logarithm are defined with hyperparameter q as a natural extension of multiplication and division, etc. in general space. In the q-space, the q-Gaussian distribution is derived by the maximization of the Tsallis entropy under appropriate constraints. The q-Gaussian distribution includes Gaussian distribution and t-distribution that can be represented by setting the hyperparameter q to $q = 1$ for Gaussian distribution and $q = 2.0$ for t-distribution. Since the q-Gaussian distribution can be written by scalar parameter, we can handle some probability distributions as flat in q-space.

The authors proposed to used q-Gaussian distribution for dimensionality reduction technique. The t-Distributed Stochastic Neighbor Embedding (t-SNE) [15] and the parametric t-SNE [14] are extended by using the q-Gaussian distribution instead of t-distribution as the probability distribution on low-dimensional space. They are named q-SNE [1] and the parametric q-SNE [17].

In this paper, we propose to define the activation function and the loss function by using the q-exponential and q-logarithm of q-space. Especially we define q-softplus function as an extension of the softplus function. By this extension, we can introduce hyperparameter q to control the shape of the function. For example, we can recover the standard softplus function or the shifted ReLU function by changing the hyperparameter q of the q-softplus function. To make the origin of the proposed q-softplus function the same as the one of the ReLU function, we also defined the shifted q-softplus function.

To show the effectiveness of the proposed shifted q-softplus function, we have performed experiments in which the shifted q-softplus function is used as the activation function in a convolutional neural network instead of the standard ReLU function. Also, we have performed experiments in which the q-softplus function is used for loss function of metric learning Siamese [5,8,10] and Triplet [12,23] instead of the max function. Through the experiments, the proposed q-softplus function shows better results on CIFAR10, CIFAR100, STL10, and TinyImageNet datasets.

2 Related Work

2.1 q-Space

Information geometry is an interdisciplinary field that applies the techniques of differential geometry to study probability theory and statistics [3]. It studies statistical manifolds, which are Riemannian manifolds whose points correspond to probability distributions. Tanaka [22] extended the information geometry developed on the exponential family to q-Gaussian distribution.

To do so, it is necessary to extend the standard multiplication, division, exponential, and logarithm to q-multiplication, q-division, q-exponential, and q-logarithm in [22]. Then we can consider a space in which these q-arithmetic operations are defined. In this paper, we call this space q-space.

In q-space, the q-multiplication and q-division of two functions f and g are respectively defined as

$$f \otimes_q g = \left(f^{1-q} + g^{1-q} - 1\right)^{\frac{1}{1-q}}, \tag{1}$$

and

$$f \oslash_q g = \left(f^{1-q} - g^{1-q} + 1\right)^{\frac{1}{1-q}}, \tag{2}$$

where q is a hyperparameter.

Similarly the q-exponential and q-logarithm are defined as

$$exp_q(x) = \left(1 + (1-q)\,x\right)^{\frac{1}{1-q}}, \tag{3}$$

and

$$log_q(x) = \frac{1}{1-q}\left(x^{1-q} - 1\right). \tag{4}$$

These q-arithmetic operations converge to the standard multiplication and division when $q \to 1$. In the q-space, the q-Gaussian distribution is derived by the maximization of the Tsallis entropy under appropriate constraints. The q-Gaussian distribution includes Gaussian distribution and t-distribution. Since the q-Gaussian distribution can be written with a scalar parameter q, we can handle a set of probability distributions as flat in q-space.

2.2 Activation Function

In a neural network, we use an activation function to approximate non-linear function. The ReLU function is famous and is mostly used in deep neural networks. The ReLU function is defined as

$$ReLU(x) = max(0, x). \tag{5}$$

The main reason why the ReLU function is used in deep neural network is that the ReLU function can prevent the vanishing gradient problem. The ReLU function is very simple and works well in deep neural networks. This function is also called the plus function.

The softplus function is a continuous version of the ReLU function and is defined as

$$Softplus(x) = \log(1 + \exp x). \tag{6}$$

The first derivative of this function is continuous around at 0.0 while one of the ReLU function is not. The softplus function is also derivation as integral of a sigmoid function.

Recently many activation functions have been proposed for deep neural networks [9,16,18,19]. Almost all of such activation functions are defined based on the ReLU function or sigmoid function or a combination of the ReLU function and sigmoid function.

These functions are also used to define loss function. For example, the max (ReLU) function or softplus function is used as contrastive loss or triplet loss uses in metric learning.

2.3 Metric Learning

The Siamese network and Triplet network have been proposed and often used for metric learning.

The Siamese network consists of two networks which have the shared weights and can learn metrics between two outputs. In the training, the two samples are fed to each network and the shared weights of the network are modified so that the two outputs of the network are closer together when the two samples belong to the same class, and so that the two outputs are farther apart when they belong to different classes.

Let $\{(\boldsymbol{x}_i, y_i)|i = 1 \ldots N\}$ be a set of training samples, where $\boldsymbol{x}_i$ is an image and y_i is a class label of i-th sample. The loss function of the Siamese network is defined as

$$L_{siamese} = \frac{1}{2} t_{ij} d_{ij}^2 + \frac{1}{2}(1 - t_{ij}) max(m - d_{ij}, 0)^2, \tag{7}$$

$$d_{ij} = \|f(\boldsymbol{x}_i; \theta) - f(\boldsymbol{x}_j; \theta)\|^2 \tag{8}$$

where t_{ij} is the binary indicator which shows whether the i-th and j-th samples are the same class or not, f is a function corresponding to the network, θ is

a set of shared weights of the network. This θ is learned by minimizing this loss $L_{siamese}$. The Siamese loss is called the contrastive loss. It is noticed that the max (ReLU) function is used in this loss. It is possible to use the softplus function instead of the max function.

The Triplet network consists of three networks with the shared weights and learns metrics between three outputs. In the training, the three samples are fed to each network. One sample is called an anchor. The sample that is the same class with the anchor is called a positive sample and the sample that is a different class from the anchor is called a negative sample. For the positive sample, the networks is trained such that the two outputs of anchor and positive are closer together. For the negative sample, the networks is trained such that the two outputs of anchor and negative become away from each other.

Let x_a, x_p, and x_n be the anchor, the positive, and the negative sample respectively. The loss function of the Triplet network is defined as

$$L_{triplet} = max(d_{ap} - d_{an} + m, 0), \tag{9}$$

where m is a margin, d_{ij} is a distance same as the contrastive loss. It is noticed that the max (ReLU) function is also used in this loss. We can use the softplus function instead of the max function. Since the max or softplus function is linear when $x >> 0$, they are effect to move the sample farther away. This is very important for metric learning.

3 q-Softplus Function and Shifted q-Softplus Function

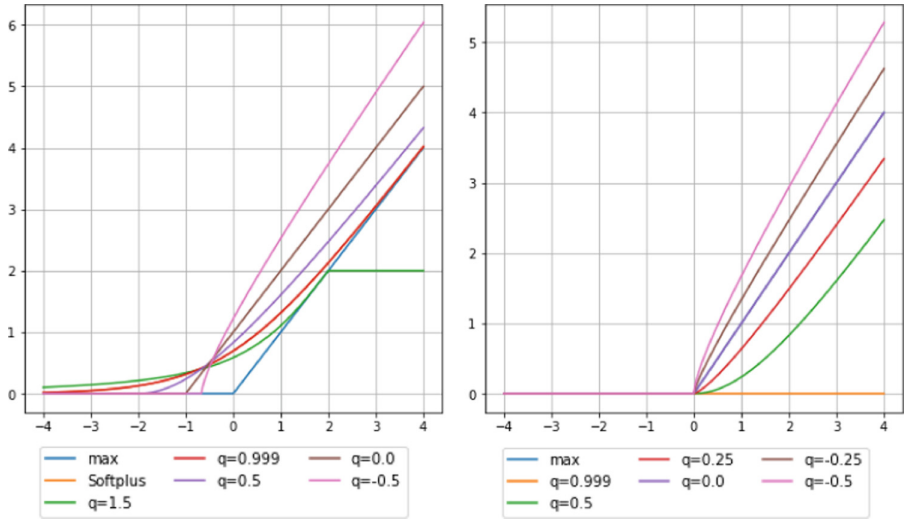
The q-Space is defined to extend information geometry developed for exponential family. By using q-space, we can consider the natural extended world. In this paper, we proposed an extension of the standard activation functions or the loss functions by using q-space. Since q-exponential and q-logarithm express the various shape of a graph by setting a hyperparameter q, we can control the shape of the activation function or the loss function by selecting the better parameter q in the q-space. In particular, in this paper, we proposed the q-softplus function as an extension of the softplus function.

3.1 q-Softplus Function

The q-softplus function is defined as

$$\begin{aligned} qsoftplus(x) &= log_q(1 + exp_q x) \\ &= \frac{1}{1-q}\left(\left(1 + max\left(1 + (1-q)\,x, 0\right)^{\frac{1}{1-q}}\right)^{1-q} - 1\right). \end{aligned} \tag{10}$$

When $q \to 1$, q-softplus function close to the original softplus function. Figure 1 (A) shows the shape of the q-softplus function compared with the max (ReLU) function and the softplus function. When $q = 0.999$ (q close to 1), q-softplus



(A) q-Softplus function (B) shifted q-Softplus function

Fig. 1. This figure shows the graph of the activation functions. In (A), it shows the max (ReLU) function, softplus function and q-softplus function with difference hyperparamete q. When $q = 0.999$, q close to 1, the q-softplus function overlaps the softplus function. In (B), it shows the max (ReLU) function and shifted q-softplus function with difference hyperparamete q. When $q = 0.0$, the q-softplus function overlaps the max function.

function overlapped with the softplus function. Moreover, when $q = 0.0$, q-softplus function becomes the shifted max function. From Fig. 1 (A), it is noticed that the q-softplus function can represent the various shapes including the max (ReLU) function and the softplus function. When $1 + (1 - q)\, x > 0$ the first derivative of x is as follows,

$$\begin{aligned} \frac{dqsoftplus(x)}{dx} &= \left(1 + (1 + (1 - q)x)^{\frac{1}{1-q}}\right)^{-q} (1 + (1 - q)x)^{\frac{q}{1-q}} \\ &= (1 + exp_q x)^{-q} \, (exp_q x)^q \,, \end{aligned} \tag{11}$$

other wise is 0. When $q \to 1$, Eq. 11 closes to first derivation of softplus function.

3.2 Shifted q-Softplus Function

The q-softplus function becomes shifted max function when $q = 0.0$. To make q-softplus with $q = 0.0$ the same as the max function, we propose to shift q-softplus function by introducing sift term. We call this function the shifted q-softplus

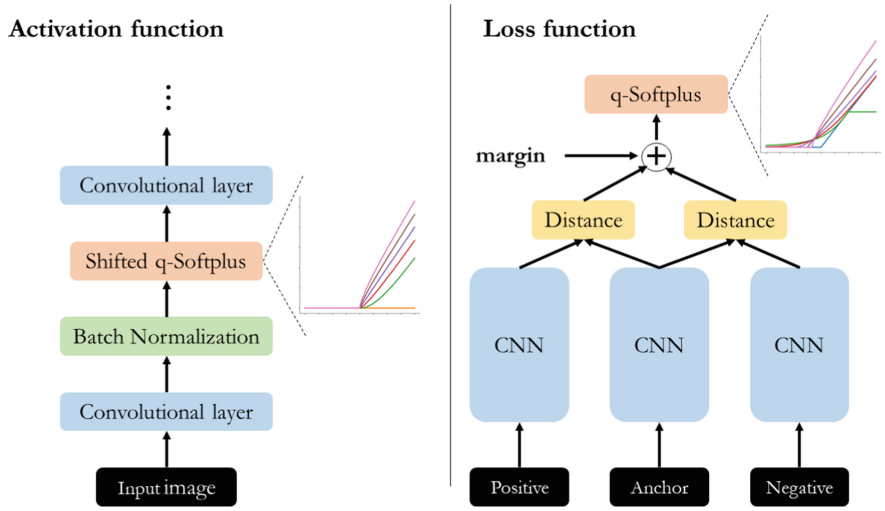


Fig. 2. This figure shows network architecture where q-softplus or shifted q-softplus function is used. As an activation function, the shifted q-softplus is replaced from ReLU function. As a loss function of triplet loss, the q-softplus function is replaced from max function.

function Then the shifted q-softplus function is defined as

$$\begin{aligned} sqsoftplus(x) &= log_q(1 + exp_q(x - \frac{1}{1-q})) \\ &= \frac{1}{1-q}\left(\left(1 + max\left(1 + (1-q)\left(x - \frac{1}{1-q}\right), 0\right)^{\frac{1}{1-q}}\right)^{1-q} - 1\right). \end{aligned} \tag{12}$$

When $q = 0.0$, the shifted q-softplus function becomes the same as the max function. Figure 1 (B) shows the shapes of the shifted q-softplus function. It is noticed that the shifted q-softplus function can represent the various shapes including the max function from this figure.

3.3 Loss Function for Metric Learning

The loss function of the Siamese network or the Triplet network, the max or softplus function is important to move the sample farther away because the max or softplus function is linear when $x >> 0$. We also propose a new loss function called q-contrastive loss and q-triplet loss by using q-softplus. The q-contrastive loss is defined as

$$L_{qsiamese} = \frac{1}{2} t_{ij} d_{ij}^2 + \frac{1}{2}(1 - t_{ij}) qsoftplus(m - d_{ij}, 0)^2. \tag{13}$$

Table 1. This table shows classification accuracy on CIFAR10, CIFAR100, STL10 and Tiny ImageNet. The hyperparameters q of all activation function on VGG11 are same. The accuracy shows percentage for train and test sample respectively.

	CIFAR10				CIFAR100			
	VGG without BN		VGG with BN		VGG without BN		VGG with BN	
	Train	Test	Train	Test	Train	Test	Train	Test
ReLU	100	87.74	100	89.00	99.98	62.65	99.98	64.91
q = −0.5	10	10	100	88.39	1	1	99.97	61.52
q = −0.25	100	86.56	100	88.85	99.98	57.98	99.98	63.99
q = −0.1	100	87.56	100	88.80	99.98	61.33	99.98	65.31
q = 0.0	100	87.74	100	89.00	99.98	62.65	99.98	64.91
q = 0.1	100	**88.20**	100	88.98	99.98	62.93	99.98	**65.48**
q = 0.2	100	87.62	100	**89.09**	99.98	**63.18**	99.98	65.07
q = 0.25	100	87.75	100	88.95	99.98	62.45	99.98	64.63
q = 0.5	10	10	100	84.98	1	1	99.98	53.62

	STL10				TinyImageNet			
	VGG without BN		VGG with BN		VGG without BN		VGG with BN	
	Train	Test	Train	Test	Train	Test	Train	Test
ReLU	100	91.76	100	92.68	85.18	52.50	92.03	56.10
q = −0.5	10	10	100	87.55	0.5	0.5	72.46	50.98
q = −0.25	100	89.75	100	91.11	62.04	47.69	86.29	54.36
q = −0.1	100	91.45	100	92.08	78.62	51.50	90.40	56.09
q = 0.0	100	91.76	100	92.68	85.18	52.50	92.03	56.10
q = 0.1	100	91.34	100	92.59	88.86	**53.24**	92.39	56.60
q = 0.2	100	90.60	100	92.34	89.97	52.94	92.19	56.33
q = 0.25	100	89.85	100	91.90	90.20	52.73	92.14	**56.87**
q = 0.5	10	10	25.32	18.23	0.5	0.5	62.35	49.13

Similarly, the q-triplet loss is defined as

$$L_{triplet} = qsoftplus(d_{ap} - d_{an} + m, 0). \tag{14}$$

By using the q-softplus function, we can control the effect of moving the sample farther away. Since the first derivative of the q-softplus function is continuous at 0, it can move the sample more farther away than the given margin. We can also use the shifted q-softplus function in loss function. Since the shifted q-softplus function has distorted linear shapes, we can control the effect of loss.

Figure 2 shows the example of the network architecture where the q-softplus function or the shifted q-softplus function is used. In this figure, the example of the triplet loss is shown.

Table 2. This table shows test classification accuracy on CIFAR10, CIFAR100, STL10 and Tiny ImageNet by using optuna. The hyperparameters q of shifted q-softplus function found by optuna are shown in Table 3. The accuracy shows percentage.

	CIFAR10		CIFAR100	
	VGG without BN	VGG with BN	VGG without BN	VGG with BN
ReLU	87.74	89.00	62.65	64.91
q-softplus	**88.34**	**89.23**	**63.58**	**65.57**

	STL10		TinyImageNet	
	VGG without BN	VGG with BN	VGG without BN	VGG with BN
ReLU	91.76	92.68	52.50	56.10
q-softplus	**91.81**	**92.80**	**53.33**	**56.97**

Table 3. This table shows the found hyperparameter q of each shifted q-softplus function on VGG11 by using optuna. VGG11 has 10 q-softplus activation functions. The qk denotes the k-th shifted q-softplus function from first layer.

	CIFAR10		CIFAR100	
	VGG without BN	VGG with BN	VGG without BN	VGG with BN
q0	0.015	0.118	0.105	0.012
q1	0.198	0.039	0.107	0.162
q2	0.075	0.086	0.007	0.249
q3	0.184	0.246	0.048	0.158
q4	0.241	0.067	0.106	0.023
q5	0.227	0.067	0.143	0.042
q6	0.201	0.145	0.204	0.113
q7	0.155	0.162	0.073	0.166
q8	0.034	0.057	0.058	0.249
q9	0.151	0.035	0.003	0.064

	STL10		TinyImageNet	
	VGG without BN	VGG with BN	VGG without BN	VGG with BN
q0	0.073	0.086	0.096	0.030
q1	0.057	0.040	0.004	0.003
q2	0.147	0.074	0.088	0.093
q3	0.062	0.078	0.000	0.008
q4	0.052	0.159	0.005	0.109
q5	0.001	0.130	0.159	0.001
q6	0.057	0.111	0.004	0.001
q7	0.004	0.144	0.010	0.205
q8	0.071	0.174	0.094	0.005
q9	0.035	0.034	0.001	0.001

4 Experiments

4.1 Experimental Dataset

To confirm the effectiveness of the proposed q-softplus based activation function and loss function, we have performed experiments using MNIST, FashionMNIST, CIFAR10, CIFAR100, STL10, and Tiny ImageNet datasets.

Table 4. This table shows classification accuracy of test sample by Siamese network on MNIST, FashionMNIST and CIFAR10. The accuracy shows percentage for train and test sample respectively by k-nn.

	MNIST						FashionMNIST					
	q-softplus		q-softplus with m-1		sq-softplus		q-softplus		q-softplus with m-1		sq-softplus	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
max	99.99	99.37	99.99	98.37	99.99	99.37	99.83	91.29	99.37	91.26	99.83	91.29
q = −0.5	99.99	**99.41**	99.99	**99.42**	99.99	**99.43**	99.84	91.08	99.82	91.33	99.83	**91.53**
q = −0.25	99.99	99.39	99.99	99.41	99.99	99.41	99.86	90.95	99.83	91.27	99.81	91.29
q = 0.0	99.99	99.41	99.99	99.37	99.99	99.37	99.86	90.91	99.83	91.29	99.83	91.29
q = 0.25	99.99	99.36	99.99	99.35	99.99	99.33	99.86	**91.29**	99.79	**91.36**	99.25	91.28
q = 0.5	99.99	99.38	99.99	99.36	99.99	99.29	99.86	91.26	99.72	91.25	95.83	90.30
q = 0.75	99.99	99.35	99.99	99.30	99.82	98.99	99.85	91.22	99.56	91.30	75.37	65.81
q = 1.5	74.76	64.74	74.76	64.74	–	–	76.96	67.54	77.03	67.64	–	–

	CIFAR10					
	q-softplus		q-softplus with m-1		sq-softplus	
	Train	Test	Train	Test	Train	Test
max	91.32	78.91	83.35	72.85	99.83	91.29
q = −0.5	92.07	**79.27**	91.59	**79.84**	92.06	**79.81**
q = −0.25	91.67	78.87	91.67	79.29	91.78	79.35
q = 0.0	91.32	78.91	91.32	78.91	91.32	78.91
q = 0.25	91.16	78.33	90.45	78.30	89.56	77.80
q = 0.5	90.72	77.93	89.08	77.43	78.81	66.58
q = 0.75	90.30	78.25	87.19	75.90	10	10
q = 1.5	89.43	77.69	80.76	68.89	–	–

The MNIST has grey images of 10 class hand-written digits. The size of each image is 28×28 pixels. The number of training samples is 60,000 and the number of test samples is 10,000. The FashionMNIST has grey images of 10 classes of fashion items. The size of each image is 28×28 pixels. The number of training samples is 60,000 and the number of test samples is 10,000. The CIFAR10 has colored images of 10 class objects. The size of each image is 32×32 pixels. The number of training samples is 50,000 and the number of test samples is 10,000. The CIFAR100 has colored images of 100 class objects. The size of each image is 32×32 pixels. The number of training samples is 50,000 and the number of test samples is 10,000. The STL10 has colored images of 10 class objects. The size of each image is 96×96 pixels. The number of training samples is 500 and the number of test samples is 800. The TinyImageNet has colored images of 200 objects. The size of each image is 64×64 pixels. The number of training samples is 100,000 and the number of test samples is 10,000.

Table 5. This table shows classification accuracy of test sample by Triplet network on MNIST, FashionMNIST and CIFAR10. The accuracy shows percentage for train and test sample respectively by k-nn.

	MNIST						FashionMNIST					
	q-softplus		q-softplus with m-1		sq-softplus		q-softplus		q-softplus with m-1		sq-softplus	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
max	99.84	99.42	99.78	98.38	99.84	99.42	98.06	91.69	97.97	91.62	98.06	91.69
q = −0.5	99.87	**99.43**	99.84	**99.42**	99.85	**99.45**	97.57	91.48	97.52	91.55	97.35	91.51
q = −0.25	99.88	99.42	99.84	99.41	99.85	99.41	97.98	91.60	97.88	91.48	98.04	91.65
q = 0.0	99.87	99.41	99.84	99.42	99.84	99.42	98.07	91.53	98.06	91.69	98.06	91.69
q = 0.25	99.88	99.41	99.84	99.42	99.82	99.39	98.33	**91.69**	98.31	**91.75**	98.20	91.57
q = 0.5	99.87	99.39	99.84	99.36	99.76	99.35	98.39	91.63	98.37	91.71	98.09	91.57
q = 0.75	99.87	99.42	99.83	99.36	99.31	98.92	98.52	91.65	98.39	91.65	94.52	90.24
q = 1.5	77.00	68.05	45.58	23.53	–	–	76.53	67.10	76.53	67.10	–	–

	CIFAR10					
	q-softplus		q-softplus with m-1		sq-softplus	
	Train	Test	Train	Test	Train	Test
max	86.47	75.36	86.47	75.36	86.47	75.36
q = −0.5	80.34	69.44	83.35	72.75	81.17	69.99
q = −0.25	81.12	70.06	84.81	74.02	84.11	73.19
q = 0.0	81.79	70.47	86.47	75.36	86.47	75.36
q = 0.25	82.59	71.44	86.97	75.43	89.27	**77.69**
q = 0.5	83.96	72.88	86.80	**75.46**	88.24	77.05
q = 0.75	83.83	72.37	86.66	75.15	10	10
q = 1.5	35.00	15.31	82.12	71.10	–	–

4.2 Shifted q-Softplus as an Activation Function

To confirm the effectiveness to use the shifted q-softplus function as an activation function, we have performed experiments in which the shifted q-softplus function in CNN is used instead of the ReLU function. The classification accuracy is measured for the datasets CIFAR10, CIFAR100, STL10, and Tiny ImageNet. VGG11 [20] is used as the CNN model and the effect of Batch Normalization (BN) is also investigated. Stochastic gradient descent (SGD) with a momentum of 0.9 is used for optimization. The learning rate is at first set to 0.01 and is multiplied by 0.1 at 20 and 40 epochs. The parameter of the weight decay is set to 0.0001. The batch size is set to 100 training samples and the training is done for 100 epochs.

Table 1 shows the classification accuracy for different q. The score is calculated as the average of 5 trials with a different random seed. From this table, the shifted q-softplus function gives better classification accuracy than the ReLU function. From this table we can notice that the best hyperparameter q is around 0.2. When the hyperparameter q is positive, namely $q > 0.0$, the shape of the shifted q-softplus function becomes lower than the ReLU function. This means

that better classification accuracy is obtained when the outputs of each layer are smaller than the outputs of the ReLU function.

We have also performed experiments to find the best hyperparameter q of the shifted q-softplus function for each dataset by using optuna [2]. The optuna is developed for python language to find the best hyperparameter of the machine learning models. The objective function to find the best hyperparameter q is the validation loss. We used 0.1% of training dataset as the validation samples. The trials of finding phase is set to 30.

The results of test accuracy for each dataset are shown in Table 2. Again, the values in the table are the averages of 5 trials with a different random seed. The best hyperparameters q of the shifted q-softplus function for each dataset are shown in Table 3. It is noticed that the best hyperparameter q is larger than 0.0 and smaller than 0.2 for almost all cases.

4.3 q-Softplus as an Loss Function of Metric Learning

To confirm the effectiveness of the q-softplus function as loss function, we have performed experiments in which the q-softplus function is used to define the loss function of the Siamese network and the Triplet network instead of the max function. We call these loss functions q-contrastive loss and q-triplet loss. MNIST, FashionMNIST, and CIFAR10 datasets are used in the experiments. The simple CNN with 2 convolutional layers and 3 fully connected layers is used for MNIST and FashionMNIST datasets. The ReLU function is used as the activation function in the hidden layers of the network. On the other hand, VGG11 with batch normalize is used for CIFAR10 dataset. The dimension of the final output is 10 for all datasets. Stochastic gradient descent (SGD) with a momentum of 0.9 is used for optimization. The learning rate is at first set to 0.01 and is multiplied by 0.1 at 20 and 40 epochs. The parameter of the weight decay is set to 0.0001. The batch size is to 100 samples and the training is done for 100 epochs. The margin in the loss function is determined by preliminary experiments.

The goodness of the feature vectors obtained by the trained network is evaluated by measuring the classification accuracy obtained by using k nearest neighbor (k-nn) in the 10-dimensional feature space. In the following experiment, k is set to 5 for k-nn. Since the q-softplus function becomes shifted max function when $q = 0.0$, we also included experiments with margin - 1.

Table 4 shows the classification accuracy obtained by the Siamese network and Table 5 shows the classification accuracy obtained by Triplet network. The score is the average of 5 trials with a different random seed.

It is noticed that the q-softplus function gives better classification accuracy than the max function. The best hyperparameter q is around -0.5. Since the shape of the q-softplus function becomes higher than the max function when $q < 0.0$, to make the output larger is probably better to move the sample farther away.

5 Conclusion

In this paper, we proposed the q-softplus function and the shifted q-softplus function as an extension of the softplus function. Through the experiments of the classification task, we confirmed that the network in which the shifted q-softplus function is used as activation function in the hidden layers gives the better classification accuracy than the network using the ReLU function. Also, we found that the best q in the shifted q-softplus function is around 0.2. This results suggest that better classification accuracy is obtained when the outputs of each layer are smaller than the outputs of the ReLU function. Through the experiments of metric learning, we confirmed that the q-softplus function can improve the contrastive loss of the Siamese network and the triplet loss of the Triplet network. For the metric learning, the best q is around -0.5. This results suggest that better features can be obtained when the outputs are larger than the output of the max function.

Acknowledgment. This research was motivated from the insightful book by Prof. Masaru Tanaka at Fukuoka University. This work was partly supported by JSPS KAKENHI Grant Number 21K12049.

References

1. Abe, M., Miyao, J., Kurita, T.: q-SNE: visualizing data using q-Gaussian distributed stochastic neighbor embedding. arXiv preprint arXiv:2012.00999 (2020)
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
3. Amari, S.: Differential-Geometrical Methods in Statistics, vol. 28. Lecture Notes in Statistics. Springer, New York (1985). https://doi.org/10.1007/978-1-4612-5056-2
4. Anthony, M., Bartlett, P.L.: Neural Network Learning: Theoretical Foundations. Cambridge University Press, Cambridge (2009)
5. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "Siamese" time delay neural network. In: Advances in Neural Information Processing Systems, pp. 737–744 (1994)
6. Canziani, A., Paszke, A., Culurciello, E.: An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678 (2016)
7. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 403–412 (2017)
8. Chopra, S., Hadsell, R., LeCun, Y., et al.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR (1), pp. 539–546 (2005)
9. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Netw. **107**, 3–11 (2018)
10. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), vol. 2, pp. 1735–1742. IEEE (2006)

11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
12. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Feragen, A., Pelillo, M., Loog, M. (eds.) SIMBAD 2015. LNCS, vol. 9370, pp. 84–92. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24261-3_7
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Adv. Neural. Inf. Process. Syst. **25**, 1097–1105 (2012)
14. van der Maaten, L.: Learning a parametric embedding by preserving local structure. In: van Dyk, D., Welling, M. (eds.) Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 5, pp. 384–391. PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 April 2009. http://proceedings.mlr.press/v5/maaten09a.html
15. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(86), 2579–2605 (2008). http://jmlr.org/papers/v9/vandermaaten08a.html
16. Misra, D.: Mish: a self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681 (2019)
17. Motoshi Abe, J.M., Kurita, T.: Parametric q-Gaussian distributed stochastic neighbor embedding with convolutional neural network. In: Proceedings of International Joint Conference on Neural Network (IJCNN) (accepted) (2021)
18. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017)
19. Shah, A., Kadam, E., Shah, H., Shinde, S., Shingade, S.: Deep residual networks with exponential linear unit. In: Proceedings of the Third International Symposium on Computer Vision and the Internet, pp. 59–65 (2016)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
22. Tanaka, M.: Geometry of Entropy. Series on Stochastic Models in Informatics and Data Science. Corona Publishing Co., LTD. (2019). (in Japanese)
23. Wang, J., et al.: Learning fine-grained image similarity with deep ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1386–1393 (2014)
24. Wang, L.: Support Vector Machines: Theory and Applications, vol. 177. Springer, Heidelberg (2005). https://doi.org/10.1007/b95439
25. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene CNNs. arXiv preprint arXiv:1412.6856 (2014)