Journal of Machine Learning Research 19 (2018) 1-39 Submitted 4/16; Revised 4/18; Published 8/18

Numerical Analysis near Singularities in RBF Networks

Weili Guo WLGUO@SEU.EDU.CN
Haikun Wei HKWEI@SEU.EDU.CN
Key Laboratory of Measurement and Control of CSE, Ministry of Education, School of Automation
Southeast University
Nanjing, Jiangsu Province 210096, P.R. China

Yew-Soon Ong ASYSONG@NTU.EDU.SG
Jaime Rubio Hervas JHERVAS@NTU.EDU.SG
School of Computer Science and Engineering
Nanyang Technological University,
50 Nanyang Avenue 639798, Singapore

Junsheng Zhao ZHAOJUNSHSHAO@163.COM
School of Mathematics Science
Liaocheng University
Liaocheng, Shandong Province 252059, P.R. China

Hai Wang HWANG@SMU.CA
Sobey School of Business
Saint Mary's University
Halifax, Nova Scotia B3H 3C3, Canada

Kanjian Zhang KJZHANG@SEU.EDU.CN
Key Laboratory of Measurement and Control of CSE, Ministry of Education, School of Automation
Southeast University
Nanjing, Jiangsu Province 210096, P.R. China

Editor: Yoshua Bengio

Abstract

The existence of singularities often affects the learning dynamics in feedforward neural networks. In this paper, based on theoretical analysis results, we numerically analyze the learning dynamics of radial basis function (RBF) networks near singularities to understand to what extent singularities influence the learning dynamics. First, we show the explicit expression of the Fisher information matrix for RBF networks. Second, we demonstrate through numerical simulations that the singularities have a significant impact on the learning dynamics of RBF networks. Our results show that overlap singularities mainly have influence on the low dimensional RBF networks and elimination singularities have a more significant impact to the learning processes than overlap singularities in both low and high dimensional RBF networks, whereas the plateau phenomena are mainly caused by the elimination singularities. The results can also be the foundation to investigate the singular learning dynamics in deep feedforward neural networks.

Keywords: RBF networks, Singularity, Learning dynamics, Numerical analysis, Deep learning

1. Introduction

The results in (Watanabe, 2007) indicate that the parameter spaces of almost all types of learning machines have singular regions where the Fisher information matrices degenerate, including layered neural networks, normal mixtures, binomial mixtures, Bayes networks, hidden Markov models, Boltzmann machines, stochastic context-free grammars, and reduced rank regressions. For the widely used feedforward neural networks, researchers have found that the learning dynamics are affected by the existing singularities. Some strange behaviors occur in the learning process, such as learning dynamics often become very slow and the learning process is trapped in plateaus.

Researchers have realized that such plateau phenomena arise from the singular structure of the parameter space and the Fisher information matrix degenerates at singularities (Fukumizu, 1996; Fukumizu and Amari, 2000; Amari and Ozeki, 2001; Amari et al., 2009). The geometrical structure of such statistical models has been studied by information geometry (Amari and Nagaoka, 2000). The standard statistical paradigm of the Cramer-Rao theorem does not hold at singularities and the model selection criteria, such as Akaike information criterion (AIC), Bayes information criterion (BIC) and minimum description length (MDL), may fail due to the existence of singularities (Amari et al., 2006). The effect of singularity in Bayesian inference was studied in (Watanabe, 2001a,b, 2010; Aoyagi, 2010), and a widely applicable Bayesian information criterion (WBIC) was proposed which remains efficient for the singular model (Watanabe, 2013). Mononen (2015) applied the WBIC to the analytically solvable Gaussian process regression case.

The error function was used instead of traditional log-sigmoid function to investigate online learning dynamics of the multilayer perceptrons (MLPs)(Biehl and Schwarze, 1995; Saad and A.Solla, 1995; Park et al., 2003). Cousseau et al. (2008) used the error function to discuss the learning dynamics of a toy model of MLPs near singularities. Guo et al. (2014, 2015) obtained the analytical expression of averaged learning equations and took the theoretical analysis of learning dynamics near overlap singularities of MLPs. For the Gaussian mixtures, Park and Ozeki (2009) analyzed the dynamics of the EM algorithm around singularities. Radial basis function (RBF) networks are typical feedforward neural networks which have been applied in many fields. Wei et al. (2008) gave a general mathematical analysis of the learning dynamics near singularities in layered networks, and obtained universal trajectories of learning near the overlap singularities. By using the methods in (Wei et al., 2008), Wei and Amari (2008) obtained the averaged learning equations of RBF networks, analyzed the learning dynamics near overlap singularities, and revealed the mechanism of plateau phenomena near the singularities. Nitta (2013, 2015) discussed the singular learning dynamics of complex-valued neural networks. Due to the existence of singularities, the standard gradient method is not Fisher efficient and the gradient descent direction is no longer the steepest descent direction. In order to overcome this problem, natural gradient method was proposed to accelerate the learning dynamics (Rattray et al., 1998; Amari, 1998; Amari et al., 2000; Park et al., 2000; Heskes, 2000; Pascanu and Bengio, 2014; Zhao et al., 2015a).

In recent years, deep learning has become a very hot topic in the machine learning community. Deep neural networks are designed based on traditional neural networks; however, it is very difficult to train deep neural networks by using the Backpropagation (BP)

algorithm. The training is computationally expensive and often presents vanishing gradient problems (Bengio et al., 1994). Till Hinton et al. (2006) proposed deep belief networks to overcome the difficulties by constructing multilayer restricted Boltzmann machines and training them layer-by-layer in a greedy fashion, many types of deep neural networks, including deep Boltzmann machine, deep convolutional neural networks, deep recurrent neural networks etc, have been applied to various fields successively, such as computer vision, pattern classification, natural language processing, nonlinear system identification, etc (Schmidhuber, 2015; Goodfellow et al., 2016).

Due to the much larger number of hidden layers and architecture size, training deep neural networks also faces many challenges (van Hasselt et al., 2016; Gulcehre et al., 2017). On the other hand, the robustness of the training effect cannot be guaranteed, even with a pre-training process (Erhan et al., 2009). Researchers are very interested in what causes the difficulties in training the deep neural networks and various analytical tools are used to study this problem. Goodfellow et al. (2014) provided some empirical evidence that the learning processes did not seem to encounter significant obstacles on a straight path from initialization to solution (obtained via gradient descent method). However, they also puzzled why the training of large models remained slow despite the scarcity of obstacles. Dauphin et al. (2014) came to the conclusion that the training difficulties were originated from the proliferation of saddle points and local minima with high error are exponentially rare in high dimensions. The saddle points caused the long plateaus in the training process. Choromanska et al. (2015) obtained the results that the gradient descent converge to the band of low critical points, and that all critical points found there are local minima of high quality. Lipton (2016) thought that large flat basins in the parameter space were the barrier to training the networks.

From the point of view of singularities in the parameter space, the above results have a certain rationality. From the theoretical results in previous literature and simulation results in this paper, we can find that the points in the elimination singularity are saddles, the points in the overlap singularity are local minima (in the batch mode learning) and the generalization error surface near the overlap singularity is very flat. It would be much clearer if the analytical form of Fisher information matrix of such deep neural networks is obtained.

Besides, Saxe et al. (2014) investigated the deep linear neural networks and found that the error did not change under a scaling transformation. This would cause the training difficulty which was called scaling symmetries in (Dauphin et al., 2014). The scaling symmetries are very similar to elimination singularities which will be discussed in Section 3. These results can be applied to a more general case. For instance, deep belief nets are based on the restricted Boltzmann machine. However, the restricted Boltzmann machine is singular, which implies the learning dynamics of deep belief nets may be seriously affected by the singularities. The learning processes of deep convolutional neural networks and deep multilayer perceptrons also face this problem. The analytical results of learning dynamics near singularities in shallow neural networks can be generalized to the deep neural networks and improve the learning efficiency. Due to overfitting issues in deep learning and the singular structure of the learning machine, it is worthy to analyze the influence of singularities in the deep neural networks in the future.

Currently, the effects of singularities to the learning dynamics of neural networks are still unknown and, therefore, it is important to examine the learning dynamics near singularities. As there are only two types of singularities (i.e. overlap and elimination singularities) in the parameter space of RBF networks and Wei and Amari (2008) has obtained the analytical form of averaged learning equations, we choose the RBF networks as the research objective in this paper. Based on the theoretical analysis results, we numerically analyze the learning dynamics near singularities through a large number of simulation experiments. From the results in (Wei and Amari, 2008; Park and Ozeki, 2009; Guo et al., 2015), it can be seen that the learning dynamics near singularities are similar in RBF networks, multilayer perceptrons and Gaussian mixtures. Thus, though the analysis is taken based on RBF networks in this paper, the statistical results can also reflect other feedforward neural networks.

For typical RBF networks with k hidden units:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{k} w_i \phi(\boldsymbol{x}, \boldsymbol{J}_i), \tag{1}$$

where $\boldsymbol{x} \in \mathcal{R}^n$ denotes the input vector, $\boldsymbol{J}_i \in \mathcal{R}^n$ is the center vector for neuron i, and w_i is the weight of neuron i in the linear output neuron. $\phi(\cdot)$ denotes the Gaussian function and $\phi(\boldsymbol{x}, \boldsymbol{J}_i) = \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{J}_i\|^2}{2\sigma^2})$, $\boldsymbol{\theta} = \{\boldsymbol{J}_1, \cdots, \boldsymbol{J}_k, w_1, \cdots, w_k\}$ represents all the parameters of the model.

Next we introduce two types of singularities. If two hidden units i and j overlap, i.e. $\boldsymbol{J}_i = \boldsymbol{J}_j$, then $w_i\phi(\boldsymbol{x}, \boldsymbol{J}_i) + w_j\phi(\boldsymbol{x}, \boldsymbol{J}_j) = (w_i + w_j)\phi(\boldsymbol{x}, \boldsymbol{J}_i)$ remains the same value when $w_i + w_j$ takes a fixed value, regardless of particular values of w_i and w_j. Therefore, we can identify their sum $w = w_i + w_j$, nevertheless, each of w_i and w_j remains unidentifiable. When one output weight $w_i = 0$, $w_i\phi(\boldsymbol{x}, \boldsymbol{J}_i) = 0$, whatever value $\boldsymbol{J}_i$ takes. These are the only two types of singularities existed in the parameter space of RBF networks (Fukumizu, 1996; Wei and Amari, 2008):
(1) Overlap singularity:

$$\mathcal{R}_1 = \{\boldsymbol{\theta} | \boldsymbol{J}_i = \boldsymbol{J}_j\},$$

(2) Elimination singularity:

$$\mathcal{R}_2 = \{\boldsymbol{\theta} | w_i = 0\}.$$

In this paper, we first derive the explicit expression of the Fisher information matrix for RBF networks. Secondly, we use the average learning equations (ALEs) to investigate the batch mode learning dynamics of RBF networks. A large number of numerical simulations are conducted. By judging whether the Fisher information matrix degenerates and tracing important variables of numerical simulations, we evaluate the learning processes which are seriously affected by the two types of singularities. We also examine the effects of the existence of singularities to RBF networks.

The rest of the paper is organized as follows. Section 2 shows the analytical expression of Fisher information matrix of RBF networks. Section 3 contains the numerical analysis near singularities for various specific cases. Finally, Section 4 presents our conclusions.

2. Analytical Expression of Fisher Information Matrix in RBF Networks

As the singularities are the regions where the Fisher information matrix of system parameters degenerates, the Fisher information matrix can be seen as an important indicator to judge whether the learning process has arrived to the singularities. We show the explicit expression of the Fisher information matrix in this section.

In the case of regression, we have a number of observed data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_t, y_t)$, which are generated by an unknown teacher function:

$$y = f_0(\boldsymbol{x}) + \varepsilon, \tag{2}$$

where ε is an additive noise, usually subject to Gaussian distribution with zero mean.

We also assume that the training input is subject to a Gaussian distribution with zero mean and a covariance matrix $\boldsymbol{\Sigma}$:

$$q(\boldsymbol{x}) = (\sqrt{2\pi})^{-n} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}\right). \tag{3}$$

As the covariance matrix plays a constant term role in the numerical analysis process and does not essentially influence the analytical results, without loss of generality, we choose the covariance to be the identity matrix, namely $\boldsymbol{\Sigma} = \mathbf{I}$. $q(\boldsymbol{x})$ can be generalized as an uniform distribution (Wei and Amari, 2008).

For the RBF networks (1), the Fisher information matrix is defined as follows (Amari and Nagaoka, 2000):

$$F(\boldsymbol{\theta}) = \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right\rangle, \tag{4}$$

where $\langle \cdot \rangle$ denotes the expectation with respect to the teacher distribution. The teacher distribution is given by:

$$p_0(y, \boldsymbol{x}) = q(\boldsymbol{x}) \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{(y - f_0(\boldsymbol{x}))^2}{2\sigma_0^2}\right). \tag{5}$$

Then by using the results obtained in (Wei and Amari, 2008) and taking further calculations, we can obtain the following theorem:

Theorem 1 *The explicit expression of Fisher information matrix for RBF networks is:*

$$\begin{aligned} F(\boldsymbol{\theta}) &= \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right\rangle \\ &= \begin{bmatrix} F_{11} & \cdots & F_{1k} & F_{1(k+1)} & \cdots & F_{1(2k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ F_{k1} & \cdots & F_{kk} & F_{k(k+1)} & \cdots & F_{k(2k)} \\ F_{(k+1)1} & \cdots & F_{(k+1)k} & F_{(k+1)(k+1)} & \cdots & F_{(k+1)(2k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ F_{(2k)1} & \cdots & F_{(2k)k} & F_{(2k)(k+1)} & \cdots & F_{(2k)(2k)} \end{bmatrix}, \end{aligned} \tag{6}$$

where:

$$C(\boldsymbol{J}_i, \boldsymbol{J}_j) = \left(\frac{\sigma^2}{\sigma^2 + 2}\right)^{\frac{N}{2}} \exp\left(-\frac{\sigma^2(\|\boldsymbol{J}_i\|^2 + \|\boldsymbol{J}_j\|^2) + \|\boldsymbol{J}_i - \boldsymbol{J}_j\|^2}{2\sigma^2(\sigma^2 + 2)}\right), \tag{7}$$

$$B(\boldsymbol{J}_i, \boldsymbol{J}_j) = -\frac{\sigma^2 \boldsymbol{J}_j - (\boldsymbol{J}_i - \boldsymbol{J}_j)}{\sigma^2(\sigma^2+2)}, \tag{8}$$

$$\left\langle \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_i)}{\partial \boldsymbol{J}_i} \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_j)}{\partial \boldsymbol{J}_j^T} \right\rangle = \frac{C(\boldsymbol{J}_i, \boldsymbol{J}_j)}{\sigma^2(\sigma^2+2)} \left(\mathbf{I}_n + (\boldsymbol{J}_j - (\sigma^2+1)\boldsymbol{J}_i) B^T(\boldsymbol{J}_i, \boldsymbol{J}_j) \right), \tag{9}$$

$$\mathbf{I}_n \textit{ is the compatable identity matrix.} \tag{10}$$

For $1 \le i \le k, \quad 1 \le j \le k,$

$$\begin{aligned} F_{ij} &= \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{J}_i} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{J}_j^T} \right\rangle = w_i w_j \left\langle \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_i)}{\partial \boldsymbol{J}_i} \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_j)}{\partial \boldsymbol{J}_j^T} \right\rangle \\ &= w_i w_j \frac{C(\boldsymbol{J}_i, \boldsymbol{J}_j)}{\sigma^2(\sigma^2+2)} \left(I_n + (\boldsymbol{J}_j - (\sigma^2+1)\boldsymbol{J}_i) B^T(\boldsymbol{J}_i, \boldsymbol{J}_j) \right). \end{aligned} \tag{11}$$

For $1 \le i \le k, \quad k+1 \le j \le 2k,$

$$\begin{aligned} F_{ij} &= \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{J}_i} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial w_{j-k}} \right\rangle = w_i \left\langle \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_i)}{\partial \boldsymbol{J}_i} \phi(\boldsymbol{x}, \boldsymbol{J}_{j-k}) \right\rangle \\ &= w_i C(\boldsymbol{J}_i, \boldsymbol{J}_{j-k}) B(\boldsymbol{J}_i, \boldsymbol{J}_{j-k}). \end{aligned} \tag{12}$$

For $k+1 \le i \le 2k, \quad 1 \le j \le k,$

$$F_{ij} = \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial w_{i-k}} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{J}_j^T} \right\rangle = F_{ji}^T. \tag{13}$$

For $k+1 \le i \le 2k, \quad k+1 \le j \le 2k,$

$$\begin{aligned} F_{ij} &= \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial w_{i-k}} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial w_{j-k}} \right\rangle = \langle \phi(\boldsymbol{x}, \boldsymbol{J}_{i-k}) \phi(\boldsymbol{x}, \boldsymbol{J}_{j-k}) \rangle \\ &= C(\boldsymbol{J}_{i-k}, \boldsymbol{J}_{j-k}). \end{aligned} \tag{14}$$

■

Remark 1: When the Fisher information matrix is near singular, the condition value of the matrix becomes very large, namely, the inverse of the condition value is near to 0. Thus the inverse of the condition value can be used to measure how close the system parameters are to the singularities. In the following numerical analysis, we record the inverse of condition value of the Fisher information matrix to show the influence of singularities on the learning process more clearly.

Remark 2: By adding the inverse of Fisher information matrix as an coefficient to the weights update in the standard gradient descent algorithm, researchers proposed the natural gradient descent method to overcome or decrease the serious influence of the singularities. Thus the Fisher information matrix plays a key role in natural gradient descent method. This means that besides being the fundamental in the following numerical analysis, obtaining the analytical form of Fisher information matrix can greatly help us in designing the modified natural gradient descent algorithms with better performance in the future.

3. Numerical Analysis near Singularities

After having obtained the analytical form of Fisher information matrix in Theorem 1, we numerically analyzed the learning dynamics of RBF networks by taking four experiments in this section, where the specific learning dynamics influenced by different types of singularities are shown and the experiment results are statistically analyzed. In Section 3.1 and Section 3.2, we conduct artificial experiments for low and medium dimensional cases, where the input distribution is known. For these cases, the Fisher information matrix can be obtained by using Theorem 1, and the relation between the stage where the singular learning dynamics occur and the stage where the Fisher information matrix degenerates can be clearly observed. In Section 3.3, the experiment for high dimensional case is carried out by a real data set to investigate the effects of the singularities.

3.1 Two-hidden-unit RBF Networks

The results obtained in (Wei and Amari, 2008) indicate that the batch mode learning dynamics are very similar to the averaged learning dynamics and we can use the averaged learning equations (ALEs) to investigate the dynamics in batch mode learning. Moreover, the ALEs do not depend on any specific sample data set which can overcome the disturbance caused by randomness of the model noise. Besides, as the ALEs are ordinary differential equations (ODEs), and for the given teacher parameters and initial values of student parameters, the learning processes of the student parameters can be obtained by solving ODEs. Thus, in this section, we use the ALEs to perform the experiments, where the analytical form of ALEs in RBF networks has been obtained in (Wei and Amari, 2008).

The student RBF network is defined in Eq.(1). We also assume that the teacher function is described by a RBF network with s hidden units:

$$f_0(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{\theta_0}) + \varepsilon = \sum_{i=1}^{s} v_i \phi(\boldsymbol{x}, \boldsymbol{t}_i) + \varepsilon, \tag{15}$$

where ε denotes zero mean Gaussian additive noise that is uncorrelated with training input $\boldsymbol{x}$. When the true teacher function $f_0(\boldsymbol{x})$ cannot be represented by a RBF network, $f(\boldsymbol{x}, \boldsymbol{\theta}_0)$ is assumed to be its best approximation by the RBF network.

The analytical form of ALEs is as follows (Wei and Amari, 2008):

$$\dot{\boldsymbol{J}}_i = \eta w_i \left(\sum_{j=1}^{s} v_j C(\boldsymbol{t}_j, \boldsymbol{J}_i) B(\boldsymbol{t}_j, \boldsymbol{J}_i) - \sum_{j=1}^{k} w_j C(\boldsymbol{J}_j, \boldsymbol{J}_i) B(\boldsymbol{J}_j, \boldsymbol{J}_i) \right), \tag{16}$$

$$\dot{w}_i = \eta \left(\sum_{j=1}^{s} v_j C(\boldsymbol{t}_j, \boldsymbol{J}_i) - \sum_{j=1}^{k} w_j C(\boldsymbol{J}_j, \boldsymbol{J}_i) \right), \tag{17}$$

where $i = 1, 2, \cdots, k$. $C(\boldsymbol{t}, \boldsymbol{J})$ and $B(\boldsymbol{t}, \boldsymbol{J})$ have the same meanings with Eq.(7) and Eq.(8), respectively.

The generalization error is:

$$L(\boldsymbol{\theta}) = \left\langle \frac{1}{2}\left(f(\boldsymbol{x},\boldsymbol{\theta}_0) - f(\boldsymbol{x},\boldsymbol{\theta})\right)^2 \right\rangle$$
$$= \frac{1}{2}\sum_{i,j} v_i v_j C(\boldsymbol{t}_i,\boldsymbol{t}_j) - \sum_{i,j} v_i w_j C(\boldsymbol{t}_i,\boldsymbol{J}_j) + \frac{1}{2}\sum_{i,j} w_i w_j C(\boldsymbol{J}_i,\boldsymbol{J}_j). \tag{18}$$

Results in (Wei and Amari, 2008) indicate that investigating the model with two hidden units is enough to capture the essence of the learning dynamics near singularities. Therefore, we first perform the numerical analysis of the RBF networks with two hidden units, and we then analyze the RBF networks in a more general case in the following sections. The learning dynamics of RBF networks are all obtained by solving ALEs for the given teacher parameters and initial student parameters.

In this subsection we analyze the case where the teacher and student models both have two hidden units.

The student model has the following form:

$$f(\boldsymbol{x},\boldsymbol{\theta}) = w_1\phi(\boldsymbol{x},\boldsymbol{J}_1) + w_2\phi(\boldsymbol{x},\boldsymbol{J}_2). \tag{19}$$

The teacher model is also described by a RBF network with two hidden units:

$$f(\boldsymbol{x},\boldsymbol{\theta_0}) = v_1\phi(\boldsymbol{x},\boldsymbol{t}_1) + v_2\phi(\boldsymbol{x},\boldsymbol{t}_2) + \varepsilon. \tag{20}$$

We choose the spread constant $\sigma = 0.5$.

In order to investigate the influence of the singularities in the learning process of RBF networks more accurately, we mainly focus on input $\boldsymbol{x}$ with dimension 1. For this type of RBF networks, the global minimum is the point where the generalization error is 0, which makes easier to analyze the simulation results.

3.1.1 Toy Model of RBF Networks

In order to visually represent the learning trajectories of parameters in the loss error surface and given that a 3D figure can only show three parameters, we initially focus on a special case of RBF networks, where part of the student parameters will remain invariable in the training process.

In the case of overlap singularity, we choose the teacher model parameters v_1 and v_2 to be the initial state of w_1 and w_2, and only J_1 and J_2 will be modified in the learning process. In all the other cases, the weights J_2 and w_2 are fixed to be the same as the teacher parameters t_2 and v_2, and therefore, only J_1 and w_1 will be modified in the learning process. Thus, for all cases, there are only two variable parameters: J_1-J_2 or J_1-w_1. When the learning process has been completed, we can plot the learning trajectories of parameters through the generalization error surface in a 3D figure. Although the student model is a toy model, the simulation results can show the influence of singularities during the learning process in a direct and visual manner.

In what follows, the teacher model is chosen as: $t_1 = -1.95$, $t_2 = -0.90$, $v_1 = 1.35$, $v_2 = 1.72$, the width spread $\sigma = 0.5$. The main reason behind only choosing one teacher function is to illustrate that the learning process of a RBF network can be affected by all

the types of singularities under different initial states. For a given initial state of student parameters, the learning trajectories of J_i and w_i can be obtained by solving Eqs.(16) and (17). The generalization error trajectory and error surface can also be obtained from Eq.(18) after J_i and w_i have been calculated.

By analyzing the simulation results, we list all the cases of learning processes as follows. In the following figures, '$\circ$' and '$\times$' represent the initial state and final state, respectively.

Case 1 (**Fast convergence**) : The learning process converges to the global minimum fast.

In this case, the singularities do not affect the learning process and the learning dynamics quickly converge to the global minimum after the beginning of learning process. An example is provided in Figure 1, which represents the trajectories in a log scale of the inverse of the condition number, generalization error and learning trajectory in the generalization error surface, respectively. In the training process, J_2 and w_2 remain invariable. As shown in Figure 1, the parameters J_1 and w_1 directly converge to the global minimum (Figure 1(c)) and the Fisher information matrix remains regular (Figure 1(a)).

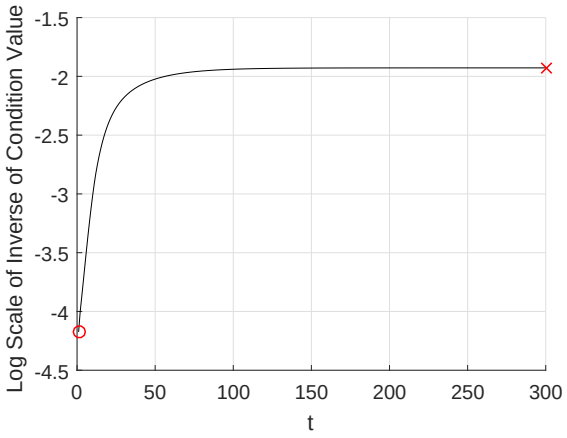
Case 2 (**Overlap singularity**) :The learning process is significantly affected by overlap singularity.

In this case, the learning trajectories of parameters J_1 and J_2 arrive at the overlap singularity, namely $J_1 = J_2$. An example is given in Figure 2, which shows the trajectories of log scale of inverse of condition number, generalization error, weights J_i and learning trajectory in the generalization error surface, respectively. In the training process, w_1 and w_2 remain invariable.

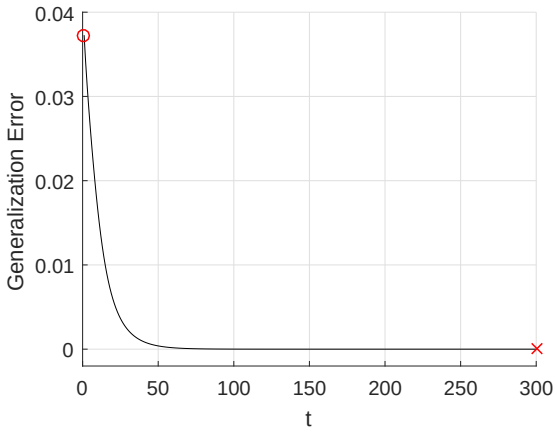
From Figure 2(a), we can see that the inverse of the condition number of Fisher information matrix gets closer and closer to 0 as the training process runs, and finally smaller than $10e-15$ which means that the Fisher information matrix nearly degenerates. Meanwhile, J_1 and J_2 nearly equal to each other (Figure 2(c)), namely the parameters arrive at the overlap singularity. The generalization error descents fast at the beginning of the learning process, and after J_1 and J_2 nearly overlap, the generalization error changes slightly. As shown in Figure 2(d), the generalization error surface is very flat near the final state of J_1 and J_2, which indicates that the parameters present difficulties escaping from the overlap singularity. In order to explore what causes the difficulties in training large-scale networks, (Lipton, 2016) revealed the high degree of nonlinearity in the learning path by analyzing the learning trajectories using the 2D PCA and thought that the large flat regions of the weight space hinder the learning process. From Figure 2(d), we can see that the error surface near the overlap singularity is very flat. Due to the so flat error surface around the overlap singularity, the learning may become very slow even if the two hidden units do not equal to each other exactly.

Remark 3: It can be seen that the log scale of the inverse of the condition number obviously fluctuates at the end of the learning process (Figure 2(a)). We think this is mainly because the value is too small (smaller than $10e-15$), and even a slight change of the parameters would cause the obvious fluctuation of the condition number of the Fisher information matrix due to the limit to the degree of accuracy of computer.

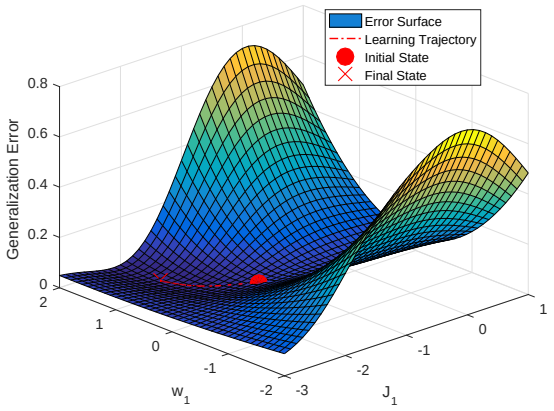
Case 3 (**Cross elimination singularity**): The learning process crosses the elimination and reaches the global optimum after training.



(a) Trajectory of log scale of inverse of condition value

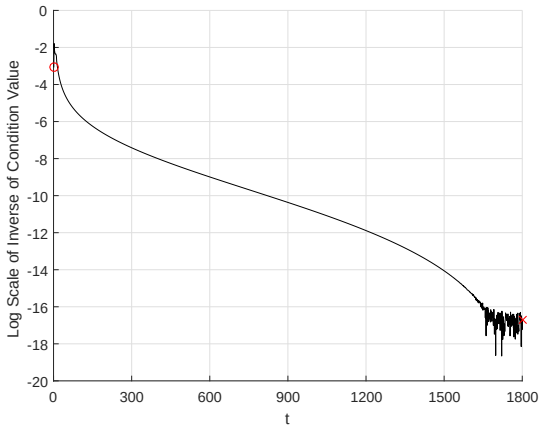


(b) Time evolution of generalization error

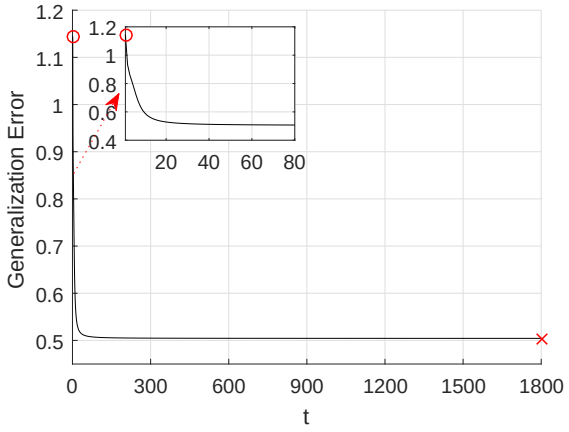


(c) Learning trajectory in generalization error surface

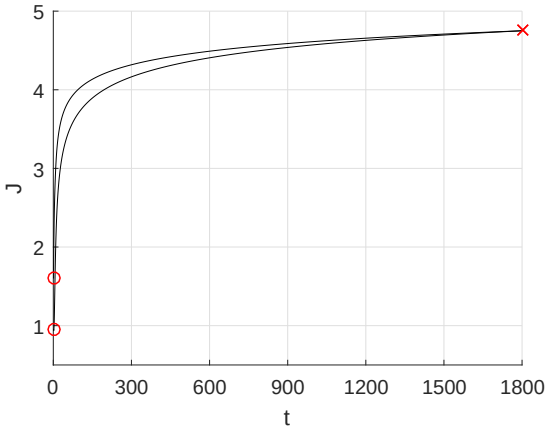
Figure 1: Case 1 (Fast convergence) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 0.30$, $w_1^{(0)} = 0.57$, $J_2^{(0)} = t_2$, $w_2^{(0)} = v_2$. In the training process J_2 and w_2 remain invariable. The final student parameters are $J_1 = -1.95$ and $w_1 = 1.35$.



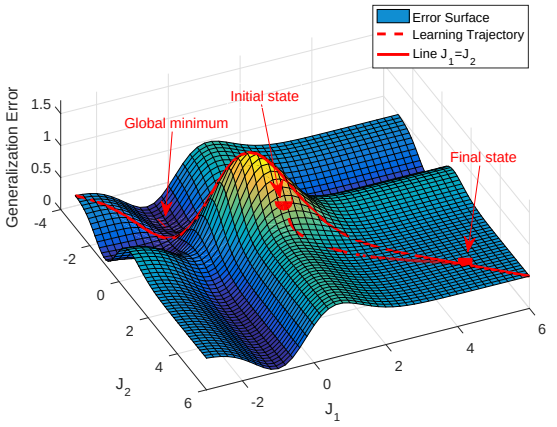
(a) Trajectory of log scale of inverse of condition value



(b) Time evolution of generalization error



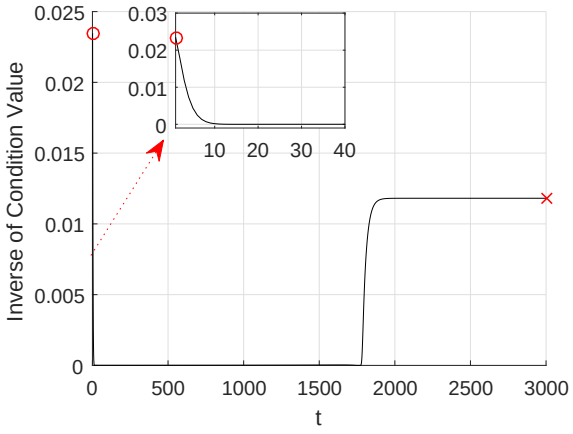
(c) Time evolution of J_1 and J_2



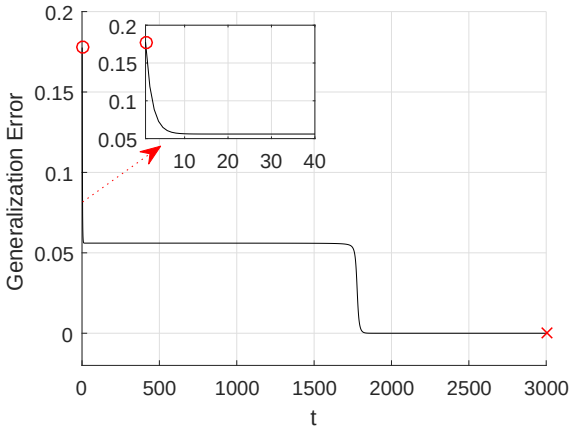
(d) Learning trajectory in generalization error surface

Figure 2: Case 2 (Overlap singularity) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 1.60$, $J_2^{(0)} = 0.95$, $w_1^{(0)} = v_1$, $w_2^{(0)} = v_2$. In the training process w_1 and w_2 remain invariable. The final student parameters are $J_1 = 4.7504$ and $J_2 = 4.7504$.

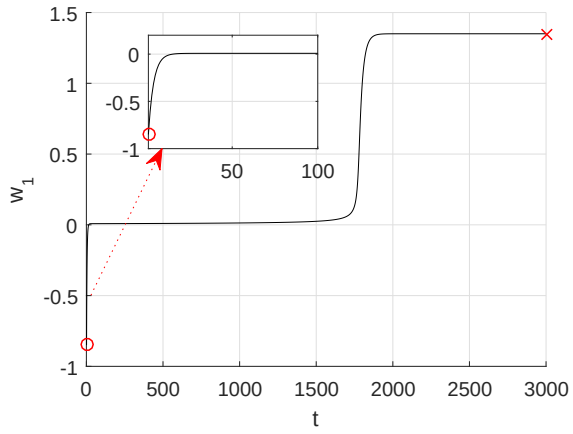
When the learning process arrives at the elimination singularity, e.g. $w_1 = 0$, the term $w_1\phi(x, J_1)$ vanishes. Hence J_1 does not affect the behavior of $f(x, \boldsymbol{\theta})$ and is not identifiable on the subspace $w_1 = 0$. An example is shown in Figure 3, where the learning process arrived at the elimination singularity and finally reached the global optimum after crossing it. Figure 3 shows the trajectories of the inverse of the condition number, generalization error, weight w_1 and learning trajectory in the generalization error surface, respectively.



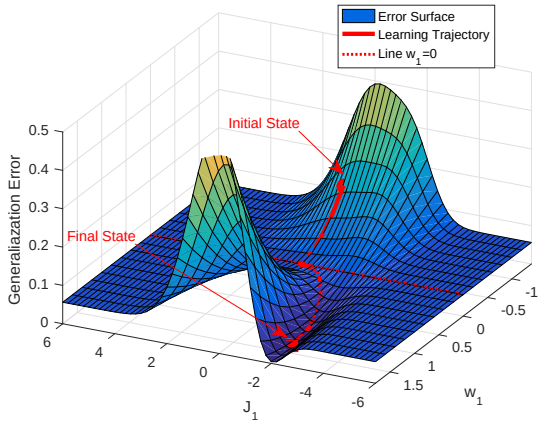
(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w_1



(d) Learning trajectory in generalization error surface

Figure 3: Case 3 (Cross elimination singularity) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 0.18$, $w_1^{(0)} = -0.85$, $J_2^{(0)} = t_2$, $w_2^{(0)} = v_2$. In the training process J_2 and w_2 remain invariable. The final student parameters are $J_1 = -1.95$ and $w_1 = 1.35$.

From the trajectory of the inverse of the condition value of Fisher information matrix in Figure 3(a), the Fisher information became nearly singular at the early stage of training and remain so for some time. w_1 nearly equaled 0 at this stage (Figure 3(c)) which means

that the learning process has arrived at the elimination singularity. It can be clearly seen from Figure 3(d) that the points on the line $w_1 = 0$ are all saddle points. Then the student parameters randomly walked around $w_1 = 0$ and finally the learning process skipped the elimination singularity and the student model exactly learned the teacher model. An obvious plateau phenomenon can be observed during the learning process as shown in Figure 3(b).

Case 4 (**Near elimination singularity**): When the student parameters are near the elimination singularity in the training, the learning process is significantly affected by the elimination singularity.

In our simulation experiments, we observed another case in which, when w_1 is close to 0 but not equal to 0, the learning process is also significantly affected by elimination singularity. Then the parameters do not skip the elimination singularity and reach the global optimal points. Figure 4 shows the trajectories of the inverse of the condition number, generalization error, weight w_1 and learning trajectory in the generalization error surface, respectively.

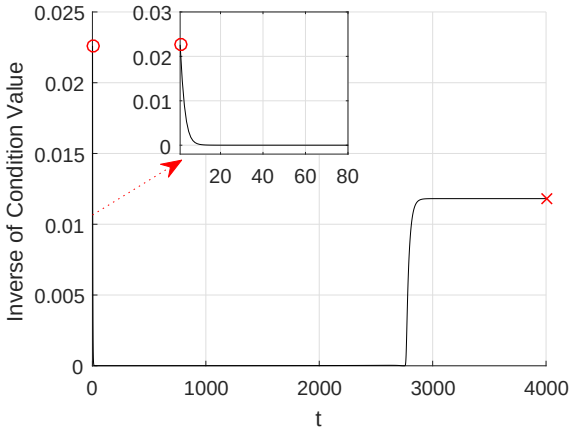
The two learning processes are similar to each other by comparing Figure 4(a) with Figure 3(a) and Figure 4(b) with Figure 3(b), respectively. However, the trajectory of w_1 in Figure 4(c) shows that w_1 is close to 0 during the training process but does not equal 0. During the stage where w_1 approaches 0 and departs from it, the learning process is significantly affected and a plateau phenomenon can clearly be observed. This means that the elimination singularity will significantly affect the learning process even if the parameters are only near to it.

By investigating the deep linear neural networks, (Saxe et al., 2014) obtained a case similar to the elimination singularity that slows down the learning process. The equation of error E is derived as $E(a, b) = \frac{1}{2\tau}(s - ab)^2$, where τ represents the inverse of the learning rate, s represents the input-output correlation information, a represents the weight from the input node to the hidden layer and b represents the weight from the hidden layer to the output node. Obviously $b = 0$ represents the elimination singularity. It can be seen that the error did not change under the scaling transformations $a \to \lambda a$, $b \to \frac{b}{\lambda}$. $a = 0$, $b = 0$ is also a fixed point. As shown in Figure 2 in (Saxe et al., 2014), we can see that certain directions of the learning point to $a = 0,\ b = 0$ which implies the parameters will converge to the point at first under an appropriate initial state. As the point $a = 0,\ b = 0$ is not stable, the parameters will escape from it and finally converge to the global minimum. During this process, long plateau can be observed. This is basically the same as with the learning trajectories in Figure 3(d) and Figure 4(d). The results illustrate the importance of investigating the singularities in deep neural networks.

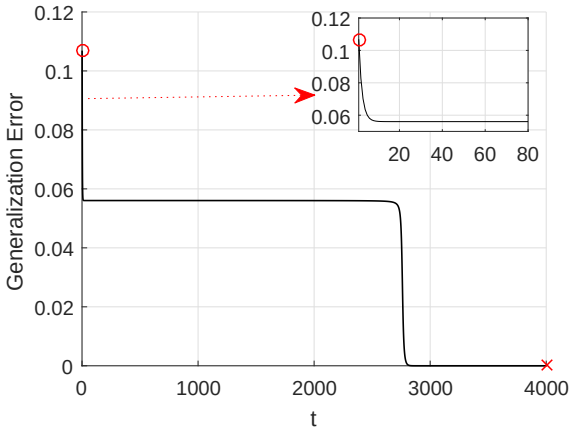
Case 5 (**Output weight 0**) : After training, output weight w_1 becomes nearly equal to 0.

In the simulation experiments, we also observe that sometimes the output weight w_1 becomes nearly equal to 0 after training. Even if the training process lasts longer, the weight also remains nearly 0. We give an example of this case in Figure 5. Figure 5 shows the trajectories of log scale of the inverse of the condition number, generalization error, weight w_1 and learning trajectory in the generalization error surface, respectively.

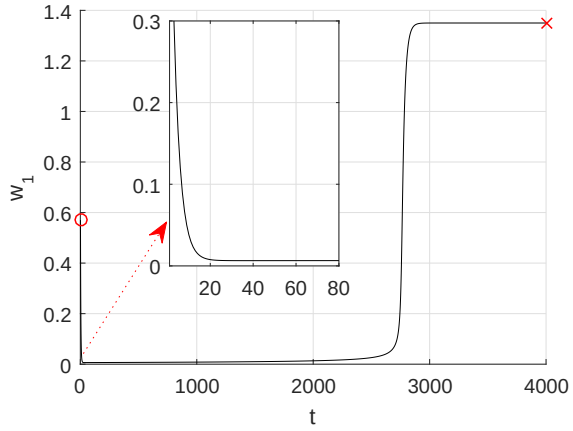
From Figure 5(d), it can be seen that w_1 quickly drops to 0 at the beginning of the training, and does not escape from it till the end. Even if we continue the training process for a longer time, the student parameters remain almost unchanged. This is mainly because



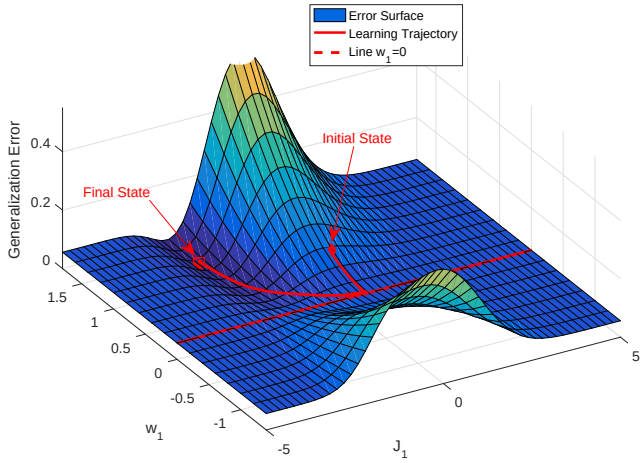
(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error

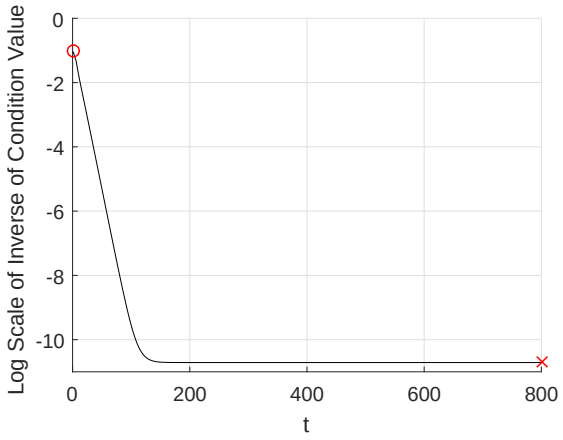


(c) Time evolution of w_1

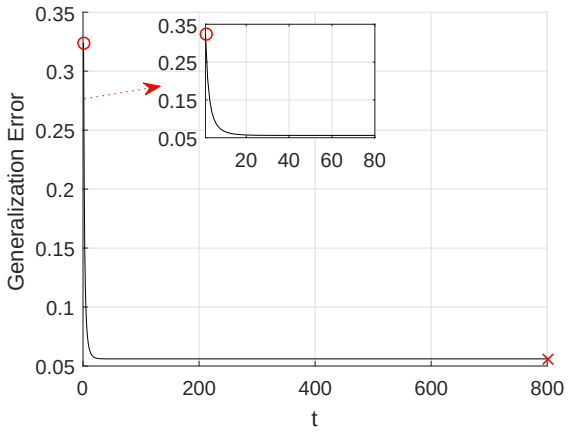


(d) Learning trajectory in generalization error surface

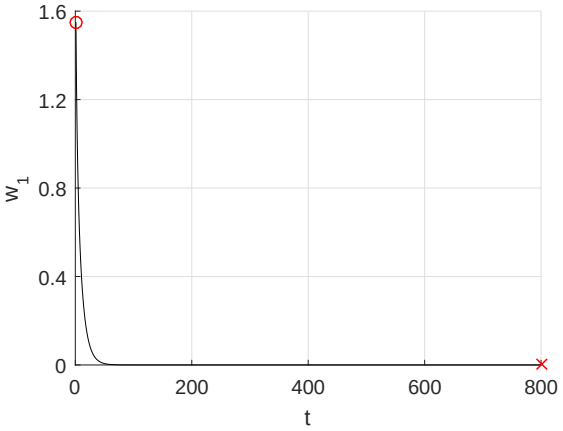
Figure 4: Case 4 (Near elimination singularity) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 0.30$, $w_1^{(0)} = 0.57$, $J_2^{(0)} = t_2$, $w_2^{(0)} = v_2$. In the training process J_2 and w_2 remain invariable. The final student parameters are $J_1 = -1.95$ and $w_1 = 1.35$.



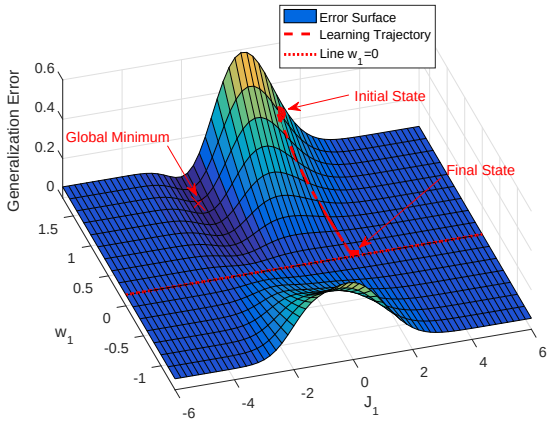
(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w_1



(d) Learning trajectory in generalization error surface

Figure 5: Case 5 (Output weight 0) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 0.95$, $w_1^{(0)} = 1.55$, $J_2^{(0)} = t_2$, $w_2^{(0)} = v_2$. In the training process J_2 and w_2 remain invariable. The final student parameters are $J_1 = 1.6192$ and $w_1 = 0$.

the radial basis function has little effect on a region that far from the center. When the centers of the teacher and the student are very far from each other, the student cannot exactly approximate the teacher and the output weight w_1 will become zero in order to avoid a bigger error. In this example, the initial student center J_1 is far away from the teacher center t_1, and w_1 is close to 0 after training. In this case, the student model is trapped in local optimum after the training process.

Hitherto, four cases of interesting learning processes in RBF networks have been visually introduced. In addition to the overlap singularity case, the other cases are actually one-hidden-unit RBF network to approximate one-hidden-unit RBF network. Even under only one hidden unit situation, the learning dynamics of RBF networks are still seriously affected by singularities.

In summary, 1) in the overlap singularity case, as the generalization error surface is very flat around the overlap singularity, the parameters cannot escape from it once they have been affected by overlap singularity. 2) For the elimination singularity case, it can be seen that the points in the elimination singularity are saddles, where part of the region is in a local minimum direction and another part of the region is in a local maximum direction. When the learning process arrives near the elimination singularity by local minimum direction, the parameters walk randomly on the singularity till they arrive at the local maximum direction, then the parameters converge to the global minimum. During the random walk stage, a plateau phenomenon can be obviously observed. If the parameters can not walk to the local maximum direction (mainly because the student center is far from the teacher center), the output weight finally nearly equals 0.

In the following subsection, we investigate the case of two-hidden-unit RBF networks approximated by normal two-hidden-unit RBF networks.

3.1.2 RBF Networks with Two Hidden Units

In this subsection, we consider three cases of v_1 and v_2: (1) v_1 and v_2 are both positive; (2) v_1 and v_2 are both negative; and (3) v_1 and v_2 have opposite sign, respectively. For each case of v_1 and v_2, we consider three cases of w_1 and w_2: (1) w_1 and w_2 are both positive; (2) w_1 and w_2 are both negative and (3) w_1 and w_2 have opposite sign. Therefore, there are 9 cases of the teacher parameters.

The procedure followed for the numerical analysis is given as:
Step 1: The teacher parameters are generated uniformly in the interval $[-2,\ 2]$. There are 9 cases. For each case, we generate 500 groups of teacher parameters.
Step 2: After each group of teacher parameters is generated, 20 groups of initial student parameters are generated uniformly in the interval $[-2,\ 2]$.
Step 3: For each group of teacher parameters and initial student parameters, we use the ALEs to accomplish the learning process. Some important variables, such as the generalization error, or the student parameters J_i and w_i, are traced and recorded.
Step 4: For each learning process, as the student parameters have been traced, the Fisher information matrix can be obtained. Then we record the inverse of the condition number of the Fisher information matrix.
Step 5: After the inverse of the condition value of the Fisher information value is recorded, a primary screening can be taken to judge whether the inverse of the condition number of

the Fisher information matrix of the learning processes has been close to 0.
Step 6: After this primary screening, we make a further analysis. If weight w_i was nearly equal to 0 in the process, then the process was affected by elimination singularity. If the two weights J_1 and J_2 nearly overlapped after training, the learning process was affected by overlap singularity. We count the numbers of the learning processes which were affected by elimination singularities and overlap singularities, respectively.

In this experiment, we totally accomplish the learning processes 90000 ($3\times3\times500\times20$) times. Given that the cases which are affected by the singularities in this subsection are exactly the same with those in Section 3.1.1, in order to keep the paper more concise, we do not show the learning trajectories belong to these cases in this subsection. Next, we count the number of learning processes which contain one of the cases above and focus on the ratio of learning processes influenced by different singularities. As the learning processes are both affected by elimination singularities in case 3 and case 4, we view case 3 and case 4 as one case in the counting process. The statistical results are shown in Table 1.

Number of total experiments	90000
Number of case 1 (Fast convergence)	61299
Number of case 2 (Overlap singularity)	6786
Number of case 3 and case 4 (Elimination singularity)	11288
Number of case 5 (Output weight 0)	10627

Table 1: Statistical results of two-hidden-unit RBF networks

From the 4 cases of observed behaviors and the statistical results shown in Table 1, we can obtain some conclusions as follows:

1) Many researchers have noticed the plateau phenomenon in the learning dynamics of feedforward neural networks (Amari et al., 2006; Saad and A.Solla, 1995; Biehl and Schwarze, 1995; Fukumizu and Amari, 2000). However, the reason why the plateau phenomenon occurs remains controversial. From the experimental results in Figure 3 and Figure 4, we found that the existence of singularities in the student parameter space results in the plateaus.

2) As shown in Table 1, nearly 68 percent of all the experiments did not get affected by the singularities and the learning dynamics converged to the global minimum fast. Almost 7.5 percent of experiments have been affected by overlap singularities and 12.5 percent of experiments have been affected by the elimination singularities. The data indicates that the singularities have a great impact on the learning processes of RBF networks. In light of the wide application of the RBF networks in practice, the influence of singularities ought to attract more attention of researchers. For the two-hidden-unit RBF networks, the initial center of the student model may be often relatively too far from the center of the teacher model, which causes the output weight of the student model to be nearly 0 after training. This case has been observed and mentioned in (Wei et al., 2007). From the results in Table 1, nearly 12 percent of experiments belong to this case.

3) From the statistical results shown in Table 1, it can be seen that the elimination singularities have much more influence in the learning dynamics than the overlap singularities. However, by now, few results in analyzing the elimination singularities have been obtained, which forms a sharp contrast to the overlap singularities. Due to the serious influence of

elimination singularities on the learning dynamics, it is worthy to take a theoretical analysis of elimination singularities.

3.2 RBF Networks in a General Case

In the previous subsection, we showed the results for the RBF networks with two hidden units. In this subsection, we generalize these results for the general RBF networks.

Without loss of generality, we introduce the student as a ten-hidden-unit model, namely:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{k} w_i \phi(\boldsymbol{x}, \boldsymbol{J}_i), \tag{21}$$

where $k = 10$.

We also assume that the teacher model is represented by a RBF network with 10 units, namely:

$$f(\boldsymbol{x}, \boldsymbol{\theta_0}) = \sum_{i=1}^{s} v_i \phi(\boldsymbol{x}, \boldsymbol{t}_i) + \varepsilon, \tag{22}$$

where $s = 10$.

We choose the spread constant $\sigma = 0.5$ and the input dimension $n = 2$.

When the number of hidden units in the student model is larger than that of the teacher, the redundant case exists. This implies that the teacher parameter might be on the singularity and the learning processes are basically affected by the singularity. In order to overcome this problem and avoid the overlap of the teacher units, we choose the minimal distance between the teacher units $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ to be bigger than $2\sigma^2$. The main reason behind this choice are based on the results obtained in (Wei and Amari, 2008). (Wei and Amari, 2008) obtained that the two teacher units are well separated when the distance between two hidden units is bigger than $2\sigma^2$.

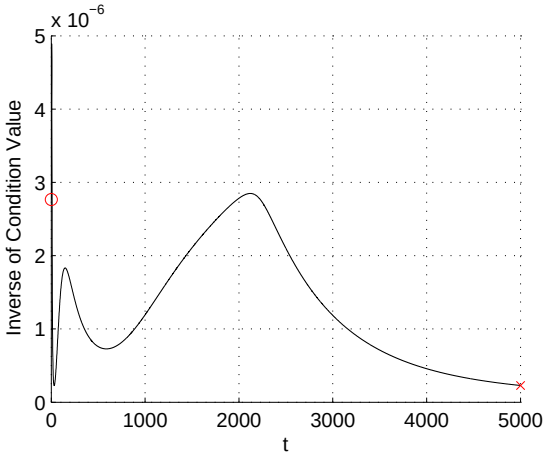
In our experiments, the teacher parameters $\boldsymbol{t}_i$, v_i, are uniformly generated in the interval $[-4,\ 4]$ and we generate 50 groups of teacher parameters. After each group of teacher parameters is generated, 20 groups of initial student parameters $\boldsymbol{J}_i^{(0)}$, $w_i^{(0)}$ are generated uniformly in the interval $[-4,\ 4]$. We use the ALEs to accomplish the learning processes. The experiment procedure is similar to that in Section 3.1.2.

By analyzing the simulation results, the cases where the learning processes present the undesirable behaviors are similar to those of RBF networks with two hidden units. To make the paper concise, the teacher parameters, the initial student parameters and the final student parameters of the following cases are listed in Appendix A. In the following figures, '$\circ$' and '$\times$' represent the initial state and final state, respectively.

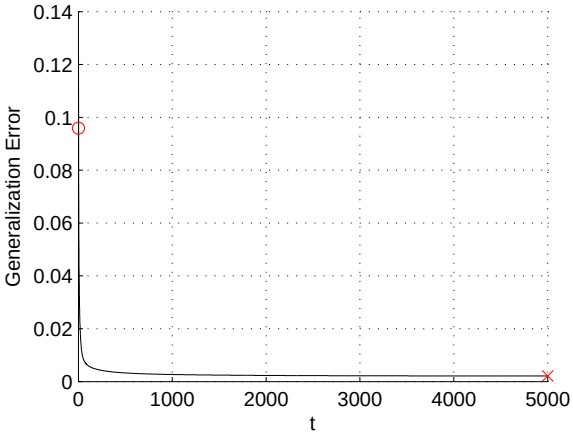
Case 1 (**Fast convergence**): The learning process quickly converges to the global minimum and the singularities do not affect the learning process.

We provide an example of this case. Figure 6 shows the trajectories of the inverse of the condition number, generalization error, and weights w_i, respectively.

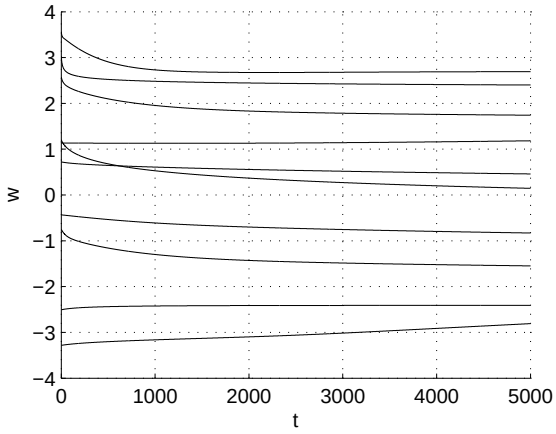
From Figure 6(a), the Fisher information matrix did not become singular during the learning process. Meanwhile, the generalization error dropped fast after the beginning of the learning process, and the singularity did not obviously affect the learning process. After the training process, the student model has converged to the global minimum.



(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w

Figure 6: Case 1 (Fast convergence) in RBF networks of general case

Case 2 (**Overlap singularity**): The learning process is affected by overlap singularity.

In this case, the learning processes are trapped in overlap singularities after training. An example belonging to this case is shown in Figure 7.

By analyzing the simulation results, it can be seen that, apart from the case where two student hidden units overlapped exactly after training, the phenomenon that two hidden units did not exactly overlap sometimes occurs. For the multidimensional parameters, we adopt the variable $h(i,j) = \frac{1}{2}\|\boldsymbol{J}_i - \boldsymbol{J}_j\|^2$ to indicate the distance between $\boldsymbol{J}_i$ and $\boldsymbol{J}_j$. When $\boldsymbol{J}_i$ and $\boldsymbol{J}_j$ nearly overlap, $h(i,j)$ nearly equals 0. Figure 7 shows the trajectories of the inverse of the condition number, generalization error, $h(4,8)$, weights w_i, respectively.

From Figure 7(a), the inverse of the condition value reduced to nearly 0 at the early stage of training process which implies that the Fisher information matrix nearly degenerated, and therefore this state remains till the end. Meanwhile, $h(4,8)$ (shown in Figure 7(c)) dropped to a very small value which implied $\boldsymbol{J}_4$ and $\boldsymbol{J}_8$ nearly overlapped, and the learning process is trapped in overlap singularities. From the final state of $\boldsymbol{J}$, it can be seen that $\boldsymbol{J}_4$ and $\boldsymbol{J}_8$ are close to each other, but do not exactly overlap. However, the gradient of the generalization error $L(\boldsymbol{\theta})$ respect to the final student parameters is nearly 0, which is too small to influence the learning process, and the student parameters will remain almost unchanged even though the learning process lasts longer. This is mainly because the error surface of RBF networks in a general case near overlap singularities is very flat. When the learning process arrives at the neighborhood of overlap singularities, although the student units have not overlapped completely, the student units will slightly change as the result of the relatively unchanged error in the remaining stage. The trajectories in Figure 7(a) and Figure 7(b) are similar to the corresponding trajectories in Figure 2(a) and Figure 2(b), respectively.

Case 3 (**Elimination singularity**): The learning process is affected by the elimination singularity and a plateau phenomenon can be observed.

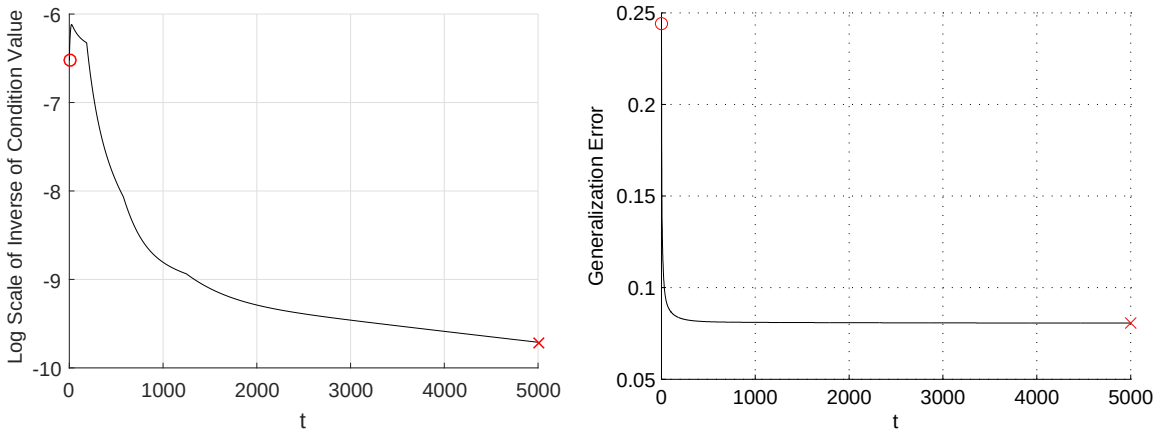
In this case, the learning process is significantly affected by the elimination singularity. This case is similar to cases 3 or 4 in Section 3.1.2. A plateau phenomenon can be observed during the learning process. We give an example of this case in Figure 8, which shows the trajectories of the inverse of the condition number, generalization error, and weights w_i.

As shown in Figure 8(a), the Fisher information matrix became nearly singular at an early stage of the learning process, and then became regular again. From the trajectories in Figure 8(b), it can be observed that w_5 (the wider line) skipped 0 when the Fisher information matrix became singular and then regular. This means that the learning process was affected by the elimination singularity. A plateau phenomenon can be observed in the trajectory of the generalization error as shown in Figure 8(b).

Case 4 (**Output weight 0**): After training, one of output weights w_i becomes nearly equal to 0.

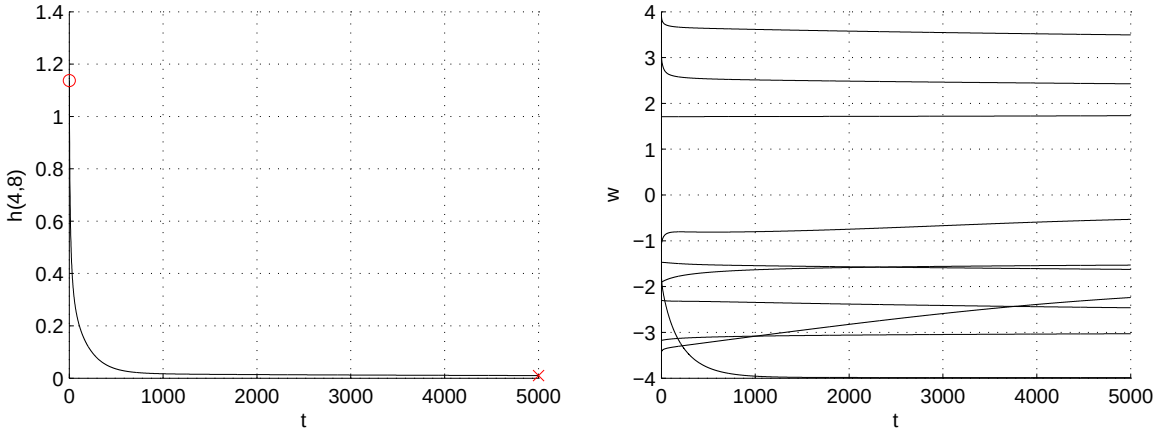
This case is similar to case 5 in Section 3.1.2. When the initial student center is too far from the center of the teacher model, the output weight w_i of the student model usually becomes nearly 0 after the training process. An example is shown in Figure 9.

Figure 9 shows the trajectories of the inverse of the condition number, generalization error, and weights w_i, respectively. From Figure 9(c), w_5 (the wider line) has become nearly 0 after training.



(a) Trajectory of inverse of condition value

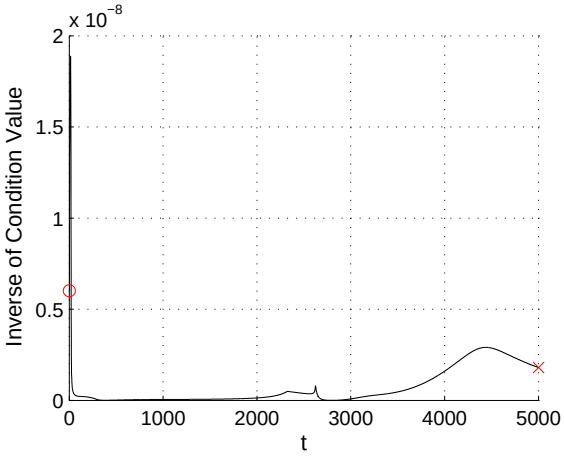
(b) Time evolution of generalization error



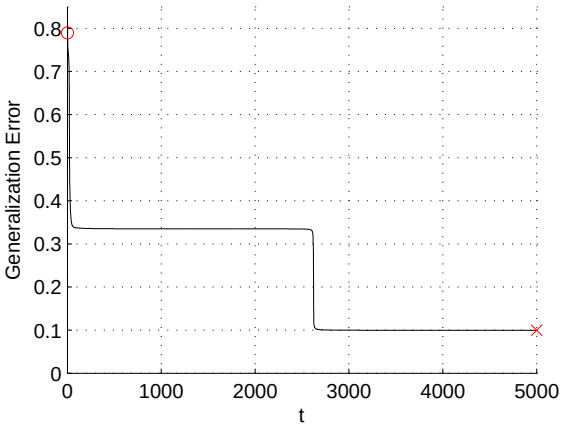
(c) Time evolution of $h(4, 8)$

(d) Time evolution of w

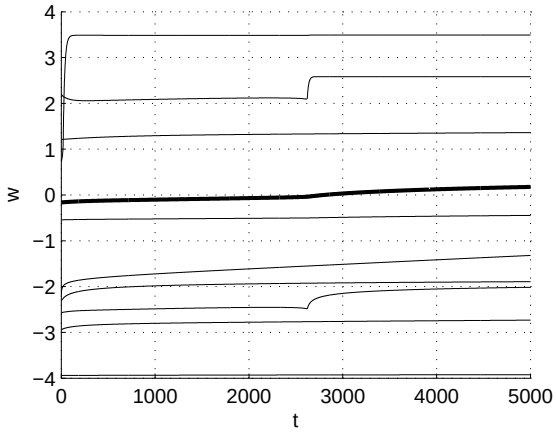
Figure 7: Case 2 (Overlap singularity) in RBF networks of general case



(a) Trajectory of inverse of condition value

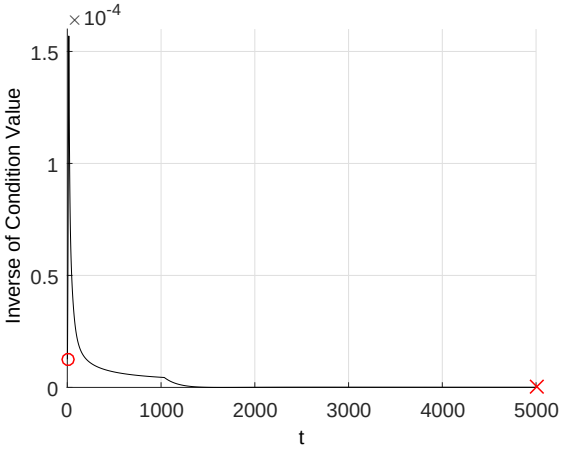


(b) Time evolution of generalization error

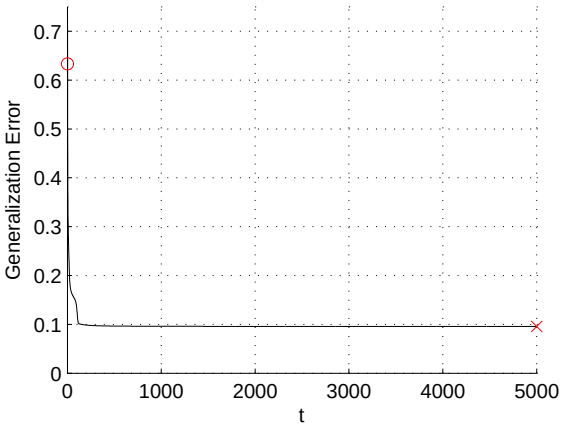


(c) Time evolution of w

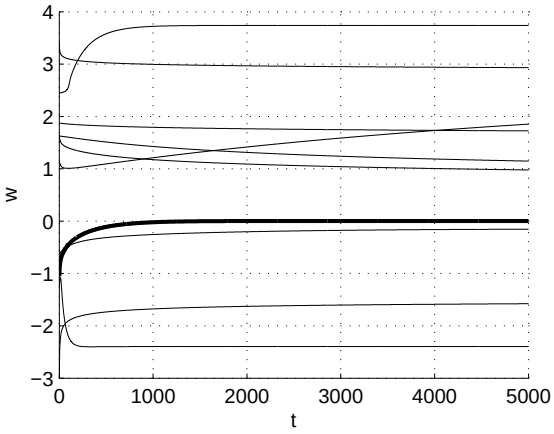
Figure 8: Case 3 (Elimination singularity) in RBF networks of general case



(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w

Figure 9: Case 4 (Output weight 0) in RBF networks of general case

Case 5 (**Overlap and elimination singularity**): The learning process is affected by not only the overlap singularity but also the elimination singularity.

Different from the case of RBF with two hidden units, we find that sometimes the learning process in a more general case is simultaneously affected by the elimination singularity and the overlap singularities. We give an example of this case in Figure 10, which shows the trajectories of log scale of the inverse of the condition number, generalization error, $h(1,9)$, and weights w_i, respectively.

From Figure 10(a), the Fisher information matrix became singular at the early stage of the learning process, and as a result the learning process arrived in singularities. As shown in Figure 10(b), a plateau phenomenon can be obviously observed. From Figure 10(d), it can be seen that w_3 (the wider line) crossed 0 when the plateau phenomenon occurred, namely the learning process crossed the elimination singularity. From Figure 10(c), $h(1,9)$ became very small along the training process. After training, $\boldsymbol{J}_1 = [2.4494, \ -2.1973]^T$ and $\boldsymbol{J}_9 = [2.4020, \ -2.2118]^T$, i.e. $\boldsymbol{J}_1$ and $\boldsymbol{J}_9$ nearly equaled to each other, so the learning process was trapped in an overlap singularity.

Case 6 (**Elimination singularity and output weight 0**): The learning process is affected by elimination singularity and one of the output weights w_i becomes nearly 0 after the learning process.

In addition to the case above, we also find a case where the learning dynamics are affected by the elimination singularities during the learning process, one of the weights w_i becomes nearly 0 after training and the student parameters are trapped in an local minimum. We give an example that belongs to this case in Figure 11.

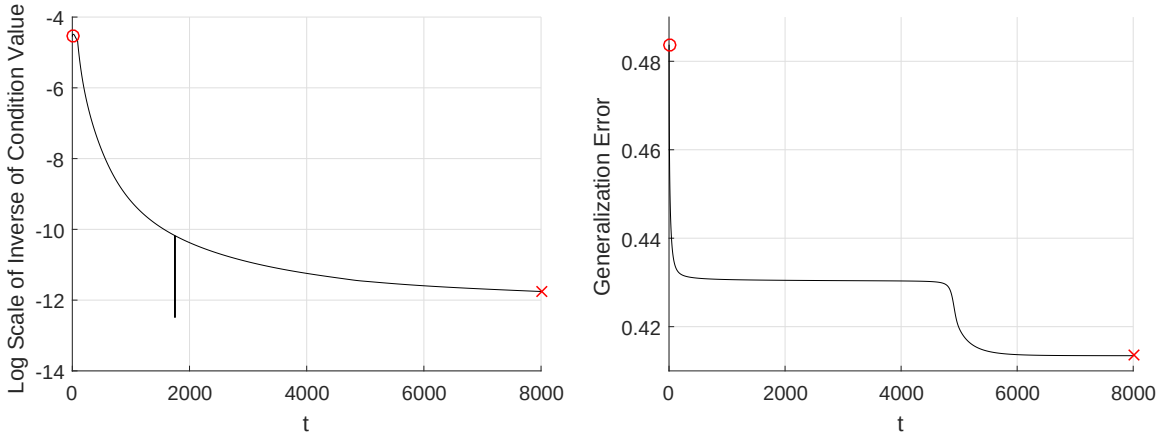
Figure 11 shows the trajectories of the inverse of the condition number, generalization error, and weights w_i, respectively. From Figure 11(b) and Figure 11(c), at the stage where w_3 crossed 0, a plateau phenomenon occurred and the learning process was affected by the elimination singularity. After training, $w_5 = -0.0004$, which is nearly equal to 0.

In comparison with the analysis results in Section 3.1.2, the RBF networks in a more general case have similar singular behaviors as those in RBF networks with two hidden units. The statistical results are summarized in Table 2.

Number of total experiments	1000
Number of case 1 (Fast convergence)	675
Number of case 2 (Overlap singularity)	109
Number of case 3 (Elimination singularity)	56
Number of case 4 (Output weight 0)	123
Number of case 5 (Overlap and elimination singularity)	16
Number of case 6 (Elimination singularity and output weight 0)	21

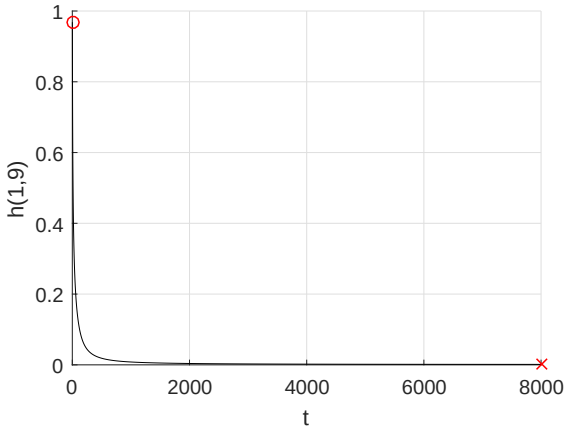
Table 2: Statistical results of RBF networks in a general case

From the results shown in Table 2, 67.5 percent of experiments did not get affected by the singularities and the learning dynamics converged to the global minimum fast. On the other hand, 20.2 percent of the learning processes are affected by the singularities. This ratio is close to the one for RBF networks with two hidden units, which implies that the existence of singularities indeed significantly affects the learning process of RBF networks. 12.3 percent of the experiments belong to case 4, which implies that the case should attract

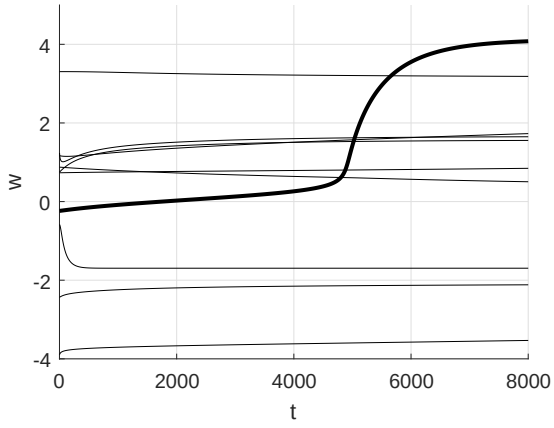


(a) Trajectory of inverse of condition value

(b) Time evolution of generalization error

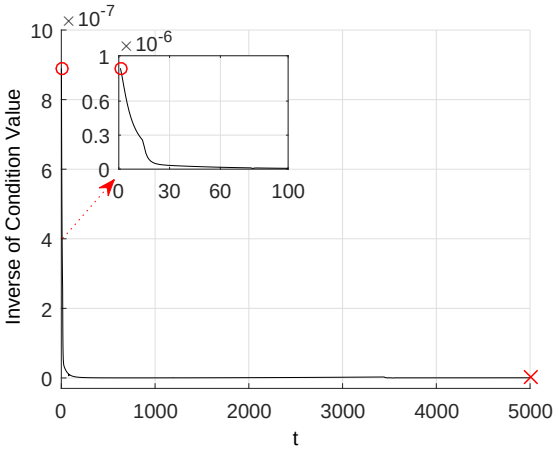


(c) Time evolution of $h(1, 9)$

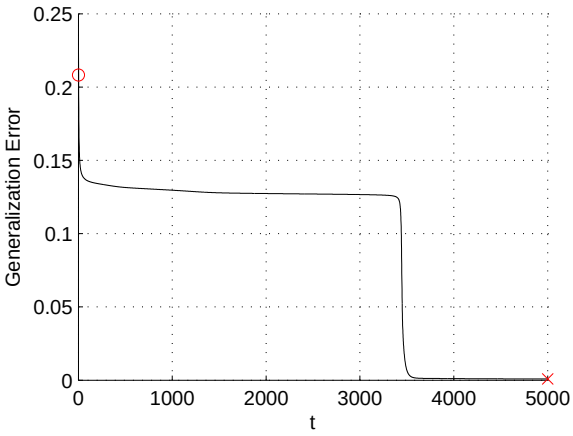


(d) Time evolution of w

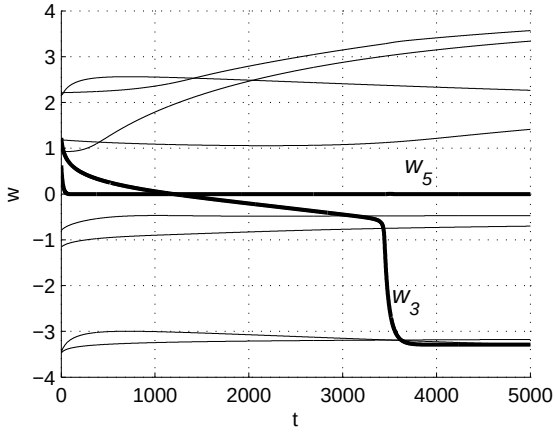
Figure 10: Case 5 (Overlap and elimination singularity) in RBF networks of general case



(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w

Figure 11: Case 6 (Elimination singularity and output weight 0) in RBF networks of general case

more attention. In case 5 and case 6, plateau phenomenons can be obviously observed where the learning dynamics are affected by the elimination singularities.

3.3 Extended Complex Scene Saliency Data set (ECSSD)

In the above experiments, we use artificial examples. We now perform an experiment by using a factual data set. Salient object detection plays a key role in many image analysis tasks that identifies important locations and structure in the visual field (Borji and Itti, 2013; Zhang et al., 2017). In recent years researchers utilize deep learning to improve the performance of saliency detection (Zhao et al., 2015b; Lee et al., 2016). As a benchmark data set in saliency detection community, extended complex scene saliency data set (ECSSD) has been widely used since its release in 2013 (Yan et al., 2013). In this experiment, we use the method proposed in (Zhang et al., 2014) to extract the features of the images in ECSSD data set as the input of the RBF networks. We get three conspicuity maps in both the rarity and the distinctiveness factors, and one conspicuity map in central bias factor. Thus the number of the nodes in the input layer is 7. The output of the training samples is '1' or '0', where '1' represents this part of the image is salient and '0' represents this part of the image is not salient.

As the distribution of input data is unknown in this experiment, we cannot obtain the analytical form of both ALEs of the training process and the Fisher information matrix. Thus we use batch mode learning to accomplish the experiment. By using a trial-and-error method, we choose the number of hidden unit in the student model to be $k = 90$ and the spread constant $\sigma = 0.5$, such that the student RBF network for the input $\boldsymbol{x}$ is given by:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{90} w_i \phi(\boldsymbol{x}, \boldsymbol{J}_i). \tag{23}$$

We use 200 samples to train the RBF network. For the learning rate $\eta = 0.002$, the model is trained by the gradient algorithm for 15000 times and the sum squared training error $E = \frac{1}{2}\sum_{i=1}^{200}(y_i - \sum_{j=1}^{90} w_j \phi(\boldsymbol{x}_i, \boldsymbol{J}_j))^2$ is used to replace the generalization error. Then we clone it 200 times. Each clone is trained with different random initial weights. The initial student parameters $\boldsymbol{J}_i^{(0)}$ and $w_i^{(0)}$ are uniformly generated in the interval $[-2,\ 2]$.

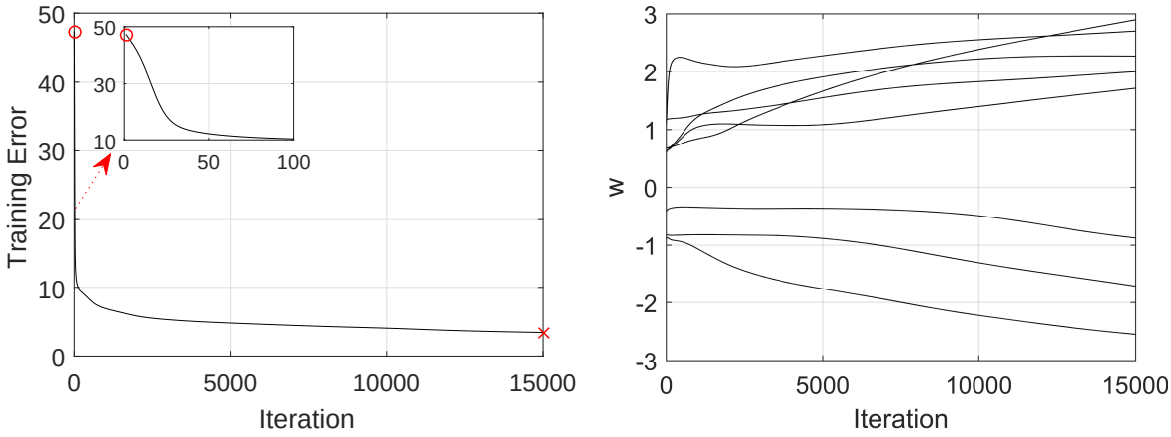
By analyzing the simulation results, the different types of learning processes are listed as follows. In the following figures, '$\circ$' and '$\times$' represent the initial state and final state, respectively.

Case 1 (**Fast convergence**): The learning process is not affected by singularities.

We give an example of this case in Figure 12, which shows the trajectories of the training error and part of output weights $\boldsymbol{w}$. From Figure 12(a), we can see that the training error comes down to a small number after the training starts and remains small till the learning stops. We do not observe that the singularities have affected the training process.

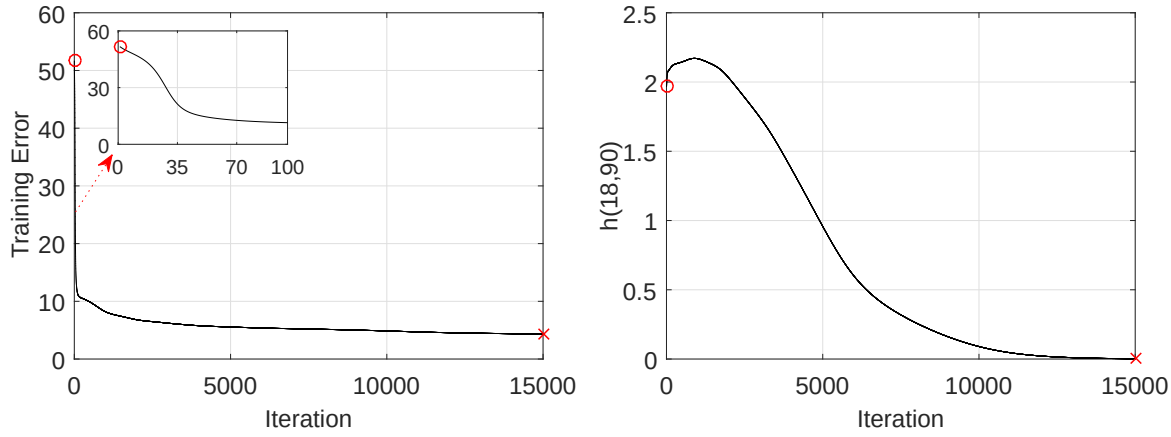
Case 2 (**Overlap singularity**): The learning process is affected by overlap singularity.

We give an example of this case in Figure 13, which shows the trajectories of the training error and $h(18, 90)$. From Figure 13(b), the Euclid distance between $\boldsymbol{J}_{18}$ and $\boldsymbol{J}_{90}$ became nearly 0, which means $\boldsymbol{J}_{18}$ and $\boldsymbol{J}_{90}$ nearly overlapped after learning.



(a) Trajectory of training error (b) Trajectory of w

Figure 12: Case 1 (Fast convergence) in approximating ECSSD data set

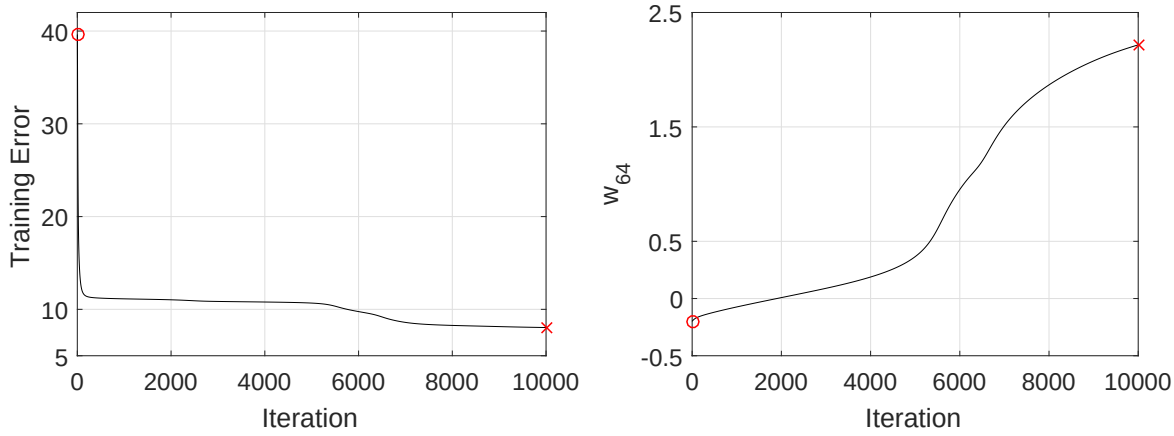


(a) Trajectory of training error (b) Trajectory of $h(18, 90)$

Figure 13: Case 2 (Overlap singularity) in approximating ECSSD data set
The initial state is:
$\boldsymbol{J}_{18}^{(0)} = [-0.4159,\ -1.0079,\ -0.3436,\ 0.1162,\ 0.6212,\ 0.3521,\ 0.9892]^T$,
$\boldsymbol{J}_{90}^{(0)} = [-0.1619,\ 0.0519,\ 0.1225,\ 0.6200,\ -0.8639,\ -0.0874,\ 0.8953]^T$.
The final state is:
$\boldsymbol{J}_{18} = [-0.2480,\ 0.0515,\ -0.1948,\ 0.2474,\ 0.0130,\ 0.0531,\ 1.2120]^T$,
$\boldsymbol{J}_{90} = [-0.2353,\ 0.0871,\ -0.1995,\ 0.2585,\ -0.0267,\ 0.0538,\ 1.1964]^T$.

Case 3 (**Elimination singularity**): The learning process is affected by elimination singularity.

We give an example of this case. Figure 14 shows the trajectories of the training error and output weight w_{64}. We can see that during the stage where w_{64} crosses 0 (Figure 14(b)), a plateau phenomenon can be observed (Figure 14(a)). The learning process is, thus, significantly affected by the elimination singularity.



(a) Trajectory of training error

(b) Trajectory of w_{64}

Figure 14: Case 3 (Elimination singularity) in approximating ECSSD data set

Case 4 (**Output weight 0**): After training, one of the output weights w_i nearly equals to 0.

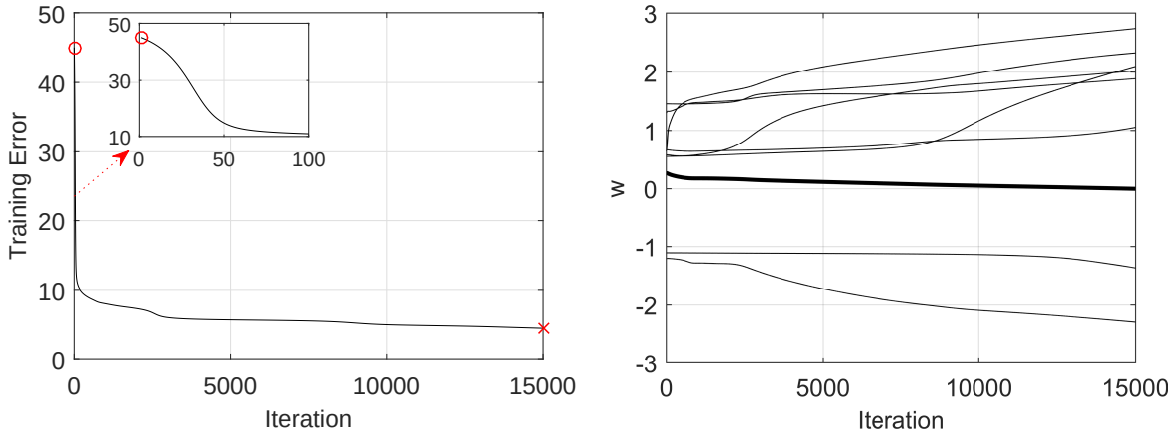
We give an example of this case. Figure 15 shows the trajectories of the training error and part of output weights $\boldsymbol{w}$. From the trajectory in Figure 15(b), it can be seen that w_{80} (the wider line) became nearly 0 after the training process.

Next, we count the learning processes which belong to each of the three cases and show them in Table 3.

Number of total experiments	200
Number of case 1 (Fast convergence)	153
Number of case 2 (Overlap singularity)	3
Number of case 3 (Elimination singularity)	42
Number of case 4 (Output weight 0)	2

Table 3: Statistical results of RBF networks in approximating ECSSD data set

It can be seen from the statistical results in Table 3 that as many as 22.5 percent of the experiments were seriously affected by the different types of singularity. There are only three experiments affected by the overlap singularity. On the other hand, we can see that 21 percent of the experiments were affected by elimination singularities. The results indicate that, in a high dimensional data scenario, the learning process is more likely affected by the elimination singularity. Therefore, it is worthy to pay more attention to investigating



(a) Trajectory of training error (b) Trajectory of w

Figure 15: Case 4 (Output weight 0) in approximating ECSSD data set

the elimination singularity. Moreover, different from our other earlier simulation results, only 1 percent of the experiments belong to case 4. The ratio is much less than those of the former experiments. The main reason is that a factual function can be represented by different suboptimal RBF networks which are equivalent to each other and the case where the initial center of the student model is far away from the center of teacher model becomes infrequent.

The statistical results confirm the previous results investigating the training difficulties in large networks from another view of point. (Dauphin et al., 2014) concluded that the local minima with high error were rare in high dimensions and the training difficulties were mainly caused by saddle points. From Table 3, we can see that the experiments affected by the overlap singularities (local minimum case) are much less than those in low dimensional networks. However, nearly all of the singular learning dynamics are affected by the elimination singularities (saddle point case). The results are in accordance with those obtained in (Dauphin et al., 2014).

4. Conclusion

Many previous works have demonstrated that the learning dynamics of feedforward neural networks are affected by the existence of singularities, but which type of singularity has more influence on the learning dynamics remains unclear. RBF networks are typical feedforward neural networks, and the learning dynamics near overlap singularities have been theoretically analyzed. Based on the obtained results, we have focused on the relationship between the existence of singularities and the learning dynamics of RBF networks in this paper. We have presented the analytical expression of the Fisher information matrix for RBF networks, as the singularities are the subspaces of the parameter space where the Fisher information matrix degenerates.

From the learning trajectories of the parameters in the generalization error surface, it can be clearly seen that the learning dynamics of RBF networks are affected by the singularities. Through a large number of numerical simulation experiments for RBF networks with two hidden units, we have identified 4 cases presenting strange learning behaviors. Nearly 7.5 percent and 12.5 percent of our experiments have shown significant effects of the overlap singularities and the elimination singularities, respectively. The points in the overlap singularity are local minima and the points in the elimination singularity are saddle points. The elimination singularities have a more significant impact to the learning processes than the overlap singularities. Our experimental results have also indicated that the plateau phenomena are mainly caused by the elimination singularities. Moreover, about 12 percent of our experiments have shown that one of the output weights of RBF networks could be close to zero after training and the student parameters are trapped into local minimum.

Through numerical simulation experiments for large scale RBF networks using a practice data set, we have found that the results are some different. Nearly all singular cases belong to the elimination singularity case and the overlap singularity case rarely occurred. This means that the large scale networks are more likely affected by the saddle points. The cases that converge to a local minimum with high error rarely appeared and the networks mainly converge to the global minimum or local minimum with good performance. The results are in accordance with the previous findings in large scale neural networks (Dauphin et al., 2014; Saxe et al., 2014; Choromanska et al., 2015).

In summary, we conclude that:

1) Overlap singularities lead to genuine local minima and elimination singularities lead to saddle points. The plateau phenomena are mainly caused by the elimination singularities.

2) The elimination singularities have a more significant impact to the learning processes than the overlap singularities. The overlap singularities mainly influence the learning dynamics of neural networks with low dimension. The large scale networks predominantly suffer from elimination singularities (saddle point case) and local minima with high error have rare influence.

Future research should pay more attention to the elimination singularities, and special treatments should be designed both for the traditional feedforward neural networks and deep neural networks to deal with the existence of singularities.

Acknowledgments

The authors would like to thank Professor S. Amari for his very constructive comments and suggestions. This work is partially supported by the Data Science and Artificial Intelligence Center (DSAIR) at the Nanyang Technological University. We would like to acknowledge support for this project from the National Science Foundation of China (NSFC) under Grant 61374006, 61773118 and 61703100, Natural Science Foundation of Jiangsu under Grant BK20170692, Jiangsu Planned Projects for Postdoctoral Research Funds under Grant 1601001A and Fundamental Research Funds for the Central Universities.

Appendix A. Proof of Theorem 1

By substituting $f(\boldsymbol{x},\boldsymbol{\theta}) = \sum_{i=1}^{k} w_i \phi(\boldsymbol{x},\boldsymbol{J}_i)$ into Eq.(4), we have

$$F(\boldsymbol{\theta}) = (F_{ij})_{(2k)\times(2k)}. \tag{A-1}$$

From Eq.(5), we have:

$$y - f_0(\boldsymbol{x}) = \varepsilon \sim \mathcal{N}(0,\sigma_0^2), \tag{A-2}$$

then

$$\frac{1}{\sqrt{2\pi}\sigma_0}\int_{-\infty}^{+\infty}\exp\left(-\frac{(y-f_0(\boldsymbol{x}))^2}{2\sigma_0^2}\right)\mathrm{d}y = \frac{1}{\sqrt{2\pi\sigma_0}}\int_{-\infty}^{+\infty}\exp(-\frac{\varepsilon^2}{2\sigma_0^2})\mathrm{d}\varepsilon = 1. \tag{A-3}$$

Thus,

$$\begin{aligned}\langle\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\rangle &= \left(\sqrt{2\pi}\right)^{-n}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}\right)\\ &\quad\times\frac{1}{\sqrt{2\pi}\sigma_0}\exp\left(-\frac{(y-f_0(\boldsymbol{x}))^2}{2\sigma_0^2}\right)\mathrm{d}y\mathrm{d}\boldsymbol{x}\\ &= \left(\sqrt{2\pi}\right)^{-n}\int_{-\infty}^{+\infty}\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}\right)\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{A-4}$$

From results in Eq.(B.6)(Wei and Amari, 2008), we have:

$$\langle\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\rangle = C(\boldsymbol{J}_i,\boldsymbol{J}_j). \tag{A-5}$$

Then we calculate $\left\langle\phi(\boldsymbol{x},\boldsymbol{J}_j)\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\right\rangle$ and $\left\langle\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_j)}{\partial\boldsymbol{J}_j^T}\right\rangle$:

$$\left\langle\phi(\boldsymbol{x},\boldsymbol{J}_j)\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\right\rangle = \frac{\partial}{\partial\boldsymbol{J}_i}\langle\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\rangle = C(\boldsymbol{J}_i,\boldsymbol{J}_j)B(\boldsymbol{J}_i,\boldsymbol{J}_j). \tag{A-6}$$

$$\begin{aligned}\left\langle\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_j)}{\partial\boldsymbol{J}_j^T}\right\rangle &= \frac{\partial}{\partial\boldsymbol{J}_j^T}\left\langle\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\phi(\boldsymbol{x},\boldsymbol{J}_j)\right\rangle\\ &= \frac{C(\boldsymbol{J}_i,\boldsymbol{J}_j)}{\sigma^2(\sigma^2+2)}\left(\mathbf{I}_n + (\boldsymbol{J}_j - (\sigma^2+1)\boldsymbol{J}_i)B^T(\boldsymbol{J}_i,\boldsymbol{J}_j)\right).\end{aligned} \tag{A-7}$$

From Eq.(A-1) and Eqs.(A-5)-(A-7), we can obtain the results in Theorem 1. ■

Appendix B. Teacher and Student Parameters of RBF Networks of General Case

Case 1 (**Fast convergence**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} 2.4872 & -1.8617 & -3.8418 & 3.2283 & -3.4445 & 2.4688 & -1.7230 & -3.8000 & 2.5460 & 3.6434\\ -0.8149 & 3.4688 & 1.3114 & 3.8362 & -1.5388 & -2.2370 & -3.9184 & 0.1137 & 0.6776 & 0.0255\end{bmatrix},$$

$\boldsymbol{v} = [2.5148, -2.6292, -2.9227, 0.3614, -0.1168, -1.1332, -2.4091, 1.4182, 0.6103, 1.9928]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} 0.9014 & 2.1329 & -1.6206 & -2.9885 & 3.7966 & -2.5606 & 2.6824 & -0.3329 & 2.2608 & -3.8733 \\ 2.0291 & -3.2296 & -3.3590 & 1.2210 & -2.8917 & 0.7866 & 2.7598 & 2.7933 & 0.0625 & 1.7027 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [3.0312, -3.2829, -2.5060, -0.7531, 1.1339, 2.5729, 0.7212, 1.2064, 3.5641, -0.4348]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} 2.0234 & 2.8324 & -1.7438 & -3.6897 & 3.5900 & -3.7365 & 2.9589 & -0.2505 & 2.5353 & -3.7203 \\ 4.3690 & -2.6481 & -3.9085 & 0.9678 & -2.8629 & 0.2693 & 2.8289 & 3.8682 & -0.7019 & 1.4435 \end{bmatrix},$$

$\boldsymbol{w} = [2.4010, -2.8088, -2.4106, -1.5482, 1.1832, 1.7415, 0.4593, 0.1452, 2.6930, -0.8301]$.

Case 2 (**Overlap singularity**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} 2.6147 & -1.9288 & -2.0314 & -1.6500 & 3.6916 & 0.0861 & -3.9215 & 1.8375 & 0.4179 & -2.8322 \\ 2.3125 & -1.8576 & 0.4307 & -3.5281 & -3.3686 & 2.5186 & 3.0892 & 0.5955 & -1.7836 & -1.3025 \end{bmatrix},$$

$\boldsymbol{v} = [3.6158, -0.3521, 1.9219, 1.9590, -3.3921, -3.9902, 2.8331, -0.5973, 1.8690, -1.9287]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} -2.5084 & 3.9684 & 0.3200 & 3.9631 & -0.8001 & -1.1329 & -0.4638 & 2.5393 & -2.0853 & -2.8973 \\ 3.3105 & 3.4805 & -3.9175 & -0.2992 & 2.0027 & -3.3396 & 2.6410 & -0.7964 & -1.1499 & -1.0947 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [-1.4711, 1.7100, -3.1748, -2.3028, 3.0340, -1.9053, -1.8244, 3.8838, -1.0591, -3.4191]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} -2.2057 & 3.7966 & 0.4940 & 3.9223 & -2.3099 & -1.3950 & 0.0860 & 4.0588 & -2.2064 & -3.0322 \\ 3.3668 & 3.3314 & -4.9285 & -0.7460 & 3.4631 & -4.7794 & 2.5181 & -0.7819 & -1.7263 & -1.4229 \end{bmatrix},$$

$\boldsymbol{w} = [-1.6246, 1.7315, -3.0298, -2.4627, 2.4273, -1.5311, -3.9856, 3.4951, -0.5322, -2.2387]$.

Case 3 (**Elimination singularity**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} -3.8655 & -1.9977 & -0.9693 & -0.6684 & 0.8680 & 1.0865 & 1.1128 & 2.2843 & 3.0604 & 3.7053 \\ -0.2720 & -3.6433 & 0.3606 & -1.3656 & -1.6714 & 3.0623 & -0.1191 & 1.5496 & 0.5356 & -0.5962 \end{bmatrix},$$

$\boldsymbol{v} = [-0.3246,\ 1.3070,\ 2.5382,\ 1.4447,\ 1.8875,\ -1.5562,\ 3.3401,\ 2.5875,\ -1.1534,\ 0.8754]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} 2.5721 & -2.0317 & -3.0520 & 2.2606 & -3.7871 & -3.4318 & -3.4161 & -3.3289 & -2.9387 & 2.9206 \\ -0.9149 & 2.3647 & 1.9332 & 0.3286 & 2.6422 & -2.5984 & 2.3670 & -3.9398 & -2.0198 & -0.9915 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [0.7242,\ 2.1926,\ -2.5681,\ -2.1687,\ -0.5425,\ 1.2113,\ -0.1622,\ -3.9397,\ -2.9404,\ -2.3096]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} -3.9125 & -3.6358 & -3.5306 & -3.4661 & -3.4104 & -3.3225 & -0.9603 & 1.0969 & 3.6156 & 4.4242 \\ 2.7119 & -2.5288 & 2.7775 & -4.0849 & 2.3727 & -2.4674 & 0.3054 & -0.1894 & 0.3816 & -1.9401 \end{bmatrix},$$

$\boldsymbol{w} = [-0.4462,\ -2.7326,\ -2.0156,\ -3.9234,\ 0.1763,\ 1.3621,\ 2.5836,\ 3.4922,\ -1.3202,\ -1.8928]$.

Case 4 (**Output weight 0**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} -2.9683 & -3.2510 & 1.8798 & 2.8565 & -0.9188 & -2.3089 & 1.2504 & -2.2858 & -2.3246 & 3.5718 \\ -1.2588 & 0.5826 & 0.5730 & 2.3925 & -2.0164 & -0.0036 & -1.2349 & -2.2290 & 3.8091 & 3.6664 \end{bmatrix},$$

$\boldsymbol{v} = [-3.8032,\ 3.7781,\ 0.2144,\ 0.4885,\ -3.8106,\ 3.0554,\ -2.3919,\ 2.8606,\ -2.1003,\ -3.9480]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} -1.6664 & 1.7394 & -1.7774 & -1.6198 & 0.5730 & 3.1482 & 0.9029 & 0.6478 & 1.7433 & -3.7033 \\ 3.3900 & 0.1411 & -2.1530 & 2.1560 & 1.9557 & -2.7220 & -1.9711 & 0.8890 & -2.1818 & 1.4784 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [1.6267,\ -1.2823,\ 1.1560,\ -0.6281,\ -1.0531,\ 1.8734,\ 3.4009,\ -2.9951,\ 1.6401,\ 2.4469]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} -1.9184 & 1.2261 & -2.6233 & -1.8287 & 0.4141 & 3.7659 & 1.4521 & 2.7995 & 2.8480 & -2.4240 \\ 4.0887 & -1.2480 & -2.6963 & 2.8370 & 2.8316 & -3.0412 & -4.6956 & 3.7300 & -3.3875 & 0.1493 \end{bmatrix},$$

$\boldsymbol{w} = [1.1474,\ -2.3942,\ 1.8534,\ -0.1541,\ 0.0024,\ 1.7245,\ 2.9327,\ -1.5772,\ 0.9781,\ 3.7390]$.

Case 5 (**Overlap and elimination singularity**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} -0.0872 & 2.5761 & -2.5050 & -2.7155 & 2.3981 & 3.6351 & -2.5770 & 1.7904 & -0.9167 & -0.1812 \\ 2.5066 & 0.0111 & -2.3643 & 0.6209 & -2.1824 & -2.0130 & 2.3249 & 1.0783 & -2.6681 & -0.9122 \end{bmatrix},$$

$\boldsymbol{v} = [-1.5554,\ -0.9298,\ -0.2798,\ 2.5378,\ 3.0691,\ 3.3465,\ -0.7750,\ -1.5487,\ 3.6799,\ -3.2292]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} 2.4527 & 3.3154 & 0.3844 & -2.5047 & 2.8795 & 3.2582 & -2.7824 & 1.0668 & 1.1586 & -3.0520 \\ -2.2713 & -0.8962 & -3.8169 & 1.9851 & -3.4619 & -1.8005 & 3.6875 & 0.7938 & -1.7594 & -2.5302 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [0.7493,\ 1.1685,\ -0.2370,\ -3.8941,\ 3.3049,\ -2.4357,\ 0.7329,\ -0.5840,\ 1.2373,\ 0.8759]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} 2.4494 & 3.5238 & -0.9825 & -3.1353 & 3.5427 & 4.7567 & -2.8540 & 1.8474 & 2.4020 & -3.2640 \\ -2.1973 & -1.9146 & -2.8404 & 3.1545 & -3.4283 & -1.8511 & 3.5258 & 0.9770 & -2.2118 & -2.9316 \end{bmatrix},$$

$\boldsymbol{w} = [1.5556,\ 1.7289,\ 4.0780,\ -3.5320,\ 3.1846,\ -2.1169,\ 0.8460,\ -1.6949,\ 1.6490,\ 0.5044]$.

Case 6 (**Elimination singularity and output weight 0**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} -1.0613 & -3.8576 & 3.4049 & 0.5341 & -2.6745 & -2.2680 & 3.4429 & -2.8965 & -3.3202 & -0.4482 \\ -1.5431 & 3.7297 & 3.7183 & -2.7340 & -3.0857 & 2.1992 & 1.9510 & -1.8745 & 0.8943 & 3.8086 \end{bmatrix},$$

$\boldsymbol{v} = [-3.2804,\ 3.1900,\ 2.8246,\ -0.3471,\ 3.8166,\ -1.0072,\ -2.4852,\ 3.4311,\ 3.9074,\ 3.0455]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} 1.8378 & -2.5054 & 0.2820 & -2.4732 & 0.2117 & -1.6407 & -3.1161 & 1.1045 & 1.9858 & -2.9216 \\ 0.7437 & 2.2043 & -2.1807 & 2.6032 & 1.0711 & 2.6934 & -0.3361 & 3.6327 & 1.7780 & -3.6724 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [2.2498,\ -3.4754,\ 1.2272,\ -1.1583,\ 0.6268,\ -0.7964,\ 0.9334,\ 1.1823,\ -3.4548,\ 2.2123]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} 2.8022 & -3.7679 & -1.0605 & -2.4646 & 0.2115 & -2.1265 & -3.2653 & -0.3505 & 2.9145 & -2.8238 \\ 1.6773 & 3.3496 & -1.5449 & 2.4451 & 1.1889 & 2.0453 & 0.8777 & 3.5703 & 1.7208 & -1.9812 \end{bmatrix},$$

$\boldsymbol{w} = [2.2651,\ -3.1769,\ -3.2902,\ -0.7017,\ -0.0004,\ -0.4704,\ 3.3391,\ 1.4123,\ -3.2949,\ 3.5688]$.

References

S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

S. Amari and H. Nagaoka. *Methods of Information Geometry*. AMS and Oxford University, New York, USA, 2000.

S. Amari and T. Ozeki. Differential and algebraic geometry of multilayer perceptrons. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer System*, E84(A(1)):31–38, 2001.

S. Amari, H. Park, and K. Fukumizu. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12(6):1399–1409, 2000.

S. Amari, H. Park, and T. Ozeki. Singularities affect dynamics of learning in neuromanifolds. *Neural Computation*, 18(5):1007–1065, 2006.

S. Amari, T. Ozeki, F. Cousseau, and H. Wei. Dynamics of learning in hierarchical models – singularity and milnor attractor. *Second International Conference on Cognitive Neurodynamics*, pages 3–9, 2009.

M. Aoyagi. Stochastic complexity and generalization error of a restricted Boltzmann machine in Bayesian estimation. *Journal of Machine Learning Research*, 11:1243–1272, 2010.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

M. Biehl and H. Schwarze. Learning by on-line gradient descent. *Journal of Physics A: Mathematical and General*, 28(3):643–656, 1995.

A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2013.

A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surface of multilayer networks. *18th International Conference on Artificial Intelligence and Statistics(AISTATS 2015)*, pages 192–204, 2015.

F. Cousseau, T. Ozeki, and S. Amari. Dynamics of learning in multilayer perceptrons near singularities. *IEEE Transactions on Neural Networks*, 19(8):1313–1328, 2008.

Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems (NIPS)*, pages 2933–2941, 2014.

D. Erhan, P. A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 153–160, 2009.

K. Fukumizu. A regularity condition of information matrix of a multilayer perceptron network. *Neural Networks*, 9(5):871–879, 1996.

K. Fukumizu and S. Amari. Local minima and plateaus in hierarchical structure of multilayer perceptrons. *Neural Networks*, 13(3):317–327, 2000.

I. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. *URL: http://arxiv. org/abs/1412.6544*, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. *MIT Press URL: http://www.deeplearningbook.org*, 2016.

C. Gulcehre, J. Sotelo, M. Moczulski, and Y. Bengio. A robust adaptive stochastic gradient method for deep learning. *International Joint Conference on Neural Networks (IJCNN2017)*, pages 125–132, 2017.

W. Guo, H. Wei, J. Zhao, and K. Zhang. Averaged learning equations of error-function-based multilayer perceptrons. *Neural Computing & Applications*, 25(3-4):825–832, 2014.

W. Guo, H. Wei, J. Zhao, and K. Zhang. Theoretical and numerical analysis of learning dynamics near singularity in multilayer perceptrons. *Neurocomputing*, 151:390–400, 2015.

T. Heskes. On "natural" learning and pruning in multilayered perceptrons. *Neural Computation*, 12(4):881–901, 2000.

G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

G. Lee, Y. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 660–668, 2016.

Z. C. Lipton. Stuck in a what? adventures in weight space. *URL: https://arxiv.org/abs/1602.07320*, 2016.

T. Mononen. A case study of the widely applicable Bayesian information criterion and its optimality. *Statistics and Computing*, 25(5):929–940, 2015.

T. Nitta. Local minima in hierarchical structures of complex-valued neural networks. *Neural Networks*, 43:1–7, 2013.

T. Nitta. Learning dynamics of a single polar variable complex-valued neuron. *Neural Computation*, 27(5):1120–1141, 2015.

H. Park and T. Ozeki. Singularity and slow convergence of the EM algorithm for Gaussian mixtures. *Neural Processing Letters*, 29(1):45–59, 2009.

H. Park, S. Amari, and K. Fukumizu. Adaptive natural gradient learning algorithms for various stochastic models. *Neural Networks*, 13(7):755–764, 2000.

H. Park, M. Inoue, and M. Okada. Online learning dynamics of multilayer perceptrons with unidentifiable parameters. *Journal of Physics A: Mathematical and General*, 36 (47):11753–11764, 2003.

R. Pascanu and Y. Bengio. Revisiting natural gradient for deep networks. *URL: http://arxiv.org/abs/1301.3584v7*, 2014.

M. Rattray, D. Saad, and S. Amari. Natural gradient descent for on-line learning. *Physical Review Letters*, 81(24):5461–5464, 1998.

D. Saad and A.Solla. Exact solution for online learning in multilayer neural networks. *Physical Review Letters*, 74(21):4337–4340, 1995.

A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *URL: http://arXiv preprint arXiv:1312.6120*, 2014.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61: 85–117, 2015.

H. van Hasselt, A. Guez, M. Hessel, and D. Silver. Learning functions across many orders of magnitudes. *URL: http://arXiv preprint arXiv:1602.07714*, 2016.

S. Watanabe. Algebraic analysis for non-identifiable learning machines. *Neural Computation*, 13(4):899–933, 2001a.

S. Watanabe. Algebraic geometrical methods for hierarchical learning machines. *Neural Works*, 14(8):1049–1060, 2001b.

S. Watanabe. Almost all learning machines are singular. *IEEE Symposium on Foundations of Computational Intelligence*, pages 383–388, 2007.

S. Watanabe. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11:3571–3594, 2010.

S. Watanabe. A widely applicable Bayesian information criterion. *Journal of Machine Learning Research*, 14:867–897, 2013.

H. Wei and S. Amari. Dynamics of learning near singularities in radial basis function networks. *Neural Networks*, 21(7):989–1005, 2008.

H. Wei, Q. Li, and W. Song. Gradient learning dynamics of radial basis function networks. *Control Theory & Applications*, 24(3):356–360, 2007.

H. Wei, J. Zhang, F. Cousseau, T. Ozeki, and S. Amari. Dynamics of learning near singularities in layered networks. *Neural Computation*, 20(34):813–843, 2008.

Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1162, 2013.

J. Zhang, J. Ding, and J. Yang. Exploiting global rarity, local contrast and central bias for salient region learning. *Neurocomputing*, 144:569–580, 2014.

J. Zhang, K. A. Ehinger, H. Wei, K. Zhang, and J. Yang. A novel graph-based optimization framework for salient object detection. *Pattern Recognition*, 64:39–50, 2017.

J. Zhao, H. Wei, C. Zhang, W. Li, W. Guo, and K. Zhang. Natural gradient learning algorithms for RBF networks. *Neural Computation*, 27(2):481–505, 2015a.

R. Zhao, W. Ouyang, H. Li, and X. Wang. Daliency detection by multi-context deep learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1274, 2015b.