2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Finite Convergence of the Learning Algorithms for a Modified Multi-Valued Neuron

Dongpo Xu and Shuang Liang

***Abstract*—The multi-valued neuron (MVN) has a strong multiclassification ability. However, the MVN learning algorithms require the complex-valued learning rate and depends on the unknown optimal weights. To address this issue, we introduce a modified MVN that centers the neuron state in each sector. The learning algorithms of the modified MVN are able to reuse the real-valued learning rate and eliminate the dependencies on the optimal weights. We prove the convergence of the modified MVN learning algorithms with real-valued learning rate.**

Keywords*—Complex-valued neural networks, Multi-valued neuron, Derivative-free learning, Convergence

I. Introduction

In recent years, the MVN has attracted widespread attention in [3], [4], [5], [6], [7], [9], [12]. The MVN outperforms many other neurons and MVN-based neural networks have shown their high potential on multi-classification problems. The discrete MVN has a learning algorithm based on the error-correction rule. It is derivative-free, which makes it highly efficient. This property and the MVN's high functionality make this neuron attractive for the development of different applications.

The discrete MVN was introduced in [3], and its outputs are the exact k^{th} roots of unity (where k is a positive integer). The discrete MVN activation function was proposed by N.N Aizenberg et al in 1971 [10], and it is the first historically known complex-valued activation function, which has been widely used in practical problems [2], [1]. A single discrete-valued MVN is a neuron with n variables $f(x_1, x_2, \cdots, x_n)$, which is either a function $f : E_k^n \to E_k$, or a function $f : O^n \to E_k$, where $E_k = \{1, \varepsilon_k, \varepsilon_k^2, \cdots, \varepsilon_k^{k-1}\}$ is the set of the k^{th} root of unity, $\varepsilon_k = e^{i2\pi/k}$ is the primitive k^{th} root of unity, i is an imaginary unity, k is a positive integer, and O is a set of points located on the unit circle. This function is a threshold function of k-valued logic and therefore it can be represented using complex-valued weights as follows

$$f(x_1, \dots, x_n) = P(w_0 + w_1x_1 + \cdots + w_nx_n) \qquad (1)$$

where $X = (1, x_1, \dots, x_n)$ for $x_j \in E_k, j = 1, \dots, n$ is an input vector and $W = (w_0, w_1, \dots, w_n)$ is a weighted vector. The values of the function and of the variables are complex. P is the activation function of the neuron:

$$P(z) = e^{i\frac{2\pi j}{k}}, \quad \text{if } \frac{2\pi j}{k} \le \arg z < \frac{2\pi(j+1)}{k} \qquad (2)$$

where $j = 0, 1, ..., k-1$, ε_k^j are the values of k-valued logic, $z = w_0 + w_0x_1 + \ldots + w_nx_n$ is the weighted sum, and arg$z$ is the argument of the complex number. Equation (2) is illustrated in Fig. 1. The discrete MVN's outputs are always the k-th roots of unity $\varepsilon^j = e^{i\frac{2\pi j}{k}}$, $j \in \{0, 1, ..., k-1\}$.

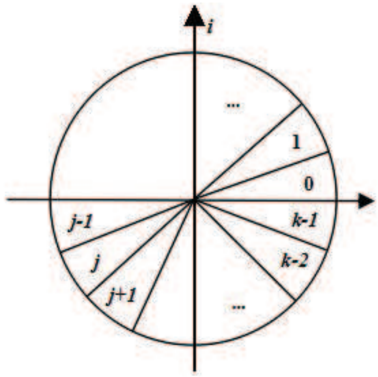

Fig. 1. The discrete MVN activation function

The activation function (2) divides a complex plane into k equal sectors and maps the whole complex plane into a set E_k of the k^{th} roots of unity. If the weighted sum is located in sector j, then the neuron's output is ε^j. MVN training is reduced to the movement along the unit circle. The MVN error-correction learning rule generalizes the classical Rosenblat's error-correction learning rule [10]. This rule and the MVN learning algorithm based on it are presented and analyzed in detail in [4], where a modified proof of the convergence of the learning algorithm has also been presented.

In the MVN error-correction learning, the direction, in which the "incorrect" actual weighted sum should move, is completely determined by the neuron's error, which is the arithmetic difference $\delta_r = \varepsilon^{q_r} - \varepsilon^{s_r}$ between the desired output ε^{q_r} and the actual output ε^{s_r} located on the unit circle. If the actual output is "incorrect", we should move the weighted sum in the direction of the desired output.

There are two MVN learning algorithms. The error-correction learning rule of discrete MVN is [3]

$$W_{r+1} = W_r + \frac{C_r}{n+1}(\varepsilon^{q_r} - \varepsilon^{s_r})\overline{X_r} \qquad (3)$$

This work was supported by the National Natural Science Foundation of China (No. 61301202), by the Research Fund for the Doctoral Program of Higher Education of China (No.20122304120028), and by the Fundamental Research Funds for the Central Universities.

Dongpo Xu and Shuang Liang are with the College of Science, Harbin Engineering University, Harbin 150001, China. (E-mail: dongpoxu@gmail.com).

and with the modification suggested in [6]

$$W_{r+1} = W_r + \frac{C_r}{(n+1)|z_r|}(\varepsilon^{q_r} - \varepsilon^{s_r})\overline{X_r} \quad (4)$$

where r is the index of iteration, $\overline{X_r}$ is the complex conjugate of the input vector X_r, W_r is the weighting vector, n is the number of neuron inputs, $z_r = W_r \cdot X_r$ is the weighted sum obtained on the r-th iteration, $(\cdot)$ is a dot product of the two vectors, and C_r is the learning rate (it may always be equal to 1).

The continuous MVN has been proposed in [6], [7], and its activation function is

$$P(z) = \exp(i(\arg z)) = e^{i\mathrm{Arg}\, z} = \frac{z}{|z|} \quad (5)$$

where $\mathrm{Arg}\, z$ is the main value of the argument of the complex number z and $|z|$ is its modulus. The learning rules (3) and (4) will be modified for the continuous-valued case in the following way:

$$W_{r+1} = W_r + \frac{C_r}{n+1}\left(\varepsilon^{q_r} - \frac{z_r}{|z_r|}\right)\overline{X_r} \quad (6)$$

$$W_{r+1} = W_r + \frac{C_r}{(n+1)|z_r|}\left(\varepsilon^{q_r} - \frac{z_r}{|z_r|}\right)\overline{X_r} \quad (7)$$

The convergence of the continuous case means that the following condition must be satisfied: $|\arg(\varepsilon^q) - \arg(e^{iargz})| < \lambda$, where λ determines the precision of the learning.

Definition 1.1 ([3], [4]): The learning subsets $A_0, A_1, ..., A_{k-1}$ are called k-separable, if it is possible to find a permutation $R = (\alpha_0, \alpha_1, ..., \alpha_{k-1})$ of the elements of the set $K = \{0, 1, ..., k-1\}$, and an optimal weighting vector W_{opt} such that $W_{opt} \cdot X \neq 0$ and

$$P(W_{opt} \cdot X) = \varepsilon^{\alpha_j} \quad (8)$$

for $X \in A_j$, $j = 0, 1, ..., k-1$. Without loss of generality we may always supply (2) by $P(0) = \varepsilon^0 = 1$. This means that the function P is now determined on the entire set $\mathbb{C}$ of complex numbers.

Theorem 1.1 ([3], [4], [5]): If the learning subsets $A_0, A_1, ..., A_{k-1}$ are k-separable, then the MVN learning algorithm with either of the learning rules (3), (4), (6) and (7) converges after a finite number of steps.

II. Example of the MVN learning

Consider the following 2-separated learning subsets $A_0 = \{X_1, \cdots, X_{N-1}\}$ and $A_1 = \{X_N\}$. The initial weight is denoted by $W_1 = (a_1 + b_1 i, a_2 + b_2 i, a_3 + b_3 i)$. There exists a permutation $R = (\alpha_0, \alpha_1)$ of the elements of the set $K = (0, 1)$, and an optimal weighting vector W_{opt} such that $P(W_{opt} \cdot X_j) = \varepsilon^{\alpha_0}$ for $j = 1, 2, \cdots, N-1$, and $P(W_{opt} \cdot X_N) = \varepsilon^{\alpha_1}$.

Theorem 2.1: The MVN learning algorithms (3) and (4) do not converge for the 2-separated learning subsets $A_0 = \{X_1, \cdots, X_{N-1}\}$ and $A_1 = \{X_N\}$, if the following conditions are satisfied

$$\varepsilon^{\alpha_0}\mathrm{Im}\,(W_1 \cdot X_j) > 0 \quad (9)$$

$$\mathrm{Im}\left(\overline{X_N} \cdot X_j\right) \leq 0 \quad (10)$$

for $j = 1, ..., N$

Proof: We declare that

$$\varepsilon^{\alpha_0}\mathrm{Im}\,(W_r \cdot X_j) > 0 \quad (11)$$

holds for $1 \leq j \leq N$ and all $r \in \mathbb{N}$. We argue it by induction. For the base case $r = 1$, the equation (11) is true from (9), which starts the induction.

Now suppose that (11) is true for some positive integer r, then the weight only needs to be updated for the sample X_N

$$W_{r+1} - W_r = \frac{C_r}{n+1}(\varepsilon^{\alpha_1} - \varepsilon^{\alpha_0})\overline{X_N} \quad (12)$$

Let us compute a dot product of both parts of (12) with X_j

$$W_{r+1} \cdot X_j - W_r \cdot X_j = \frac{C_r}{n+1}(\varepsilon^{\alpha_1} - \varepsilon^{\alpha_0})\overline{X_N} \cdot X_j \quad (13)$$

Note that $\varepsilon^{\alpha_1} \neq \varepsilon^{\alpha_0} \in \{-1, 1\}$ for the 2-valued logic, thus $\varepsilon^{\alpha_1} = -\varepsilon^{\alpha_0}$ and

$$\mathrm{Im}\,(W_{r+1} \cdot X_j) - \mathrm{Im}\,(W_r \cdot X_j) = -\frac{2C_r}{n+1}\varepsilon^{\alpha_0}\mathrm{Im}(\overline{X_N} \cdot X_j) \quad (14)$$

From (10) and (14), we have

$$[\mathrm{Im}\,(W_{r+1} \cdot X_j) - \mathrm{Im}\,(W_r \cdot X_j)]\,\varepsilon^{\alpha_0} \geq 0 \quad (15)$$

Case I: If $\varepsilon^{\alpha_0} = 1$, then we get $\mathrm{Im}\,(W_r \cdot X_j) > 0$ from (9), so $\mathrm{Im}(W_{r+1} \cdot X_j) \geq \mathrm{Im}(W_r \cdot X_j) > 0$ from (15), and this means that $\varepsilon^{\alpha_0}\mathrm{Im}\,(W_{r+1} \cdot X_j) > 0$.

Case II: If $\varepsilon^{\alpha_0} = -1$, then we get $\mathrm{Im}\,(W_r \cdot X_j) < 0$ from (9), so $\mathrm{Im}(W_{r+1} \cdot X_j) \leq \mathrm{Im}(W_r \cdot X_j) < 0$ from (15), and this means that $\varepsilon^{\alpha_0}\mathrm{Im}\,(W_{r+1} \cdot X_j) > 0$.

The result of the both cases is the equation (11) with r replaced by $r+1$, and this is the inductive step. Thus by induction, the equation (11) is true for all positive integers r.

From (11), we get $P\,(W_r \cdot X_N) = \varepsilon^{\alpha_0}$ for all $r \in \mathbb{N}$, so W_r needs to be updated for every iteration step r because of $P\,(W_{opt} \cdot X_N) = \varepsilon^{\alpha_1}$. This indicates that the learning rule (3) does not converge in a finite steps. The learning rule (4) can be shown in a similar way. ■

Remark 2.1: The proof of Theorem 2.1 does not restrict that the learning rate C_r is constant and only needs $C_r \in \mathbb{R}$. The conditions (9) and (10) only depend on the learning samples $\{X_1, \cdots, X_N\}$ and the initial weight W_1. There are $N+1$ inequality equations and $N+1$ unknown complex variables $\{X_1, \cdots, X_N, W_1\}$, which ensures that there exists at least one solution to satisfy (9) and (10). A specific solution is given in Example 2.1.

Example 2.1: According to (10), we choose the learning subsets $A_0 = \{X_1, X_2, X_3\}$ and $A_1 = \{X_4\}$, where

$$\begin{cases} X_1 = \left(1,\ \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i,\ -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i\right) \\ X_2 = \left(1,\ \frac{1}{2} + \frac{\sqrt{3}}{2}i,\ -\frac{1}{2} + \frac{\sqrt{3}}{2}i\right) \\ X_3 = \left(1,\ \frac{1}{3} + \frac{2\sqrt{2}}{3}i,\ -\frac{1}{3} + \frac{2\sqrt{2}}{3}i\right) \\ X_4 = (1,\ i,\ i) \end{cases} \quad (16)$$

First we give the optimal weighting vector

$$W_{opt} = \left(\frac{\sqrt{3}}{2} - \frac{1}{2}i,\ -\frac{1}{2} + \frac{\sqrt{3}}{2}i,\ \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i \right) \tag{17}$$

that established the followings

$$\begin{cases} P\left(W_{opt} \cdot X_j\right) = \varepsilon^0 = 1, j = 1, 2, 3 \\ P\left(W_{opt} \cdot X_4\right) = \varepsilon^1 = -1 \end{cases} \tag{18}$$

Set $\varepsilon^{\alpha_0} = 1$ and take (16) into (9), we have

$$\begin{cases} b_1 + \frac{\sqrt{2}}{2}b_2 + \frac{\sqrt{2}}{2}a_2 - \frac{\sqrt{2}}{2}b_3 + \frac{\sqrt{2}}{2}a_3 > 0 \\ b_1 + \frac{1}{2}b_2 + \frac{\sqrt{3}}{2}a_2 - \frac{1}{2}b_3 + \frac{\sqrt{3}}{2}a_3 > 0 \\ b_1 + \frac{1}{3}b_2 + \frac{2\sqrt{2}}{3}a_2 - \frac{1}{3}b_3 + \frac{2\sqrt{2}}{3}a_3 > 0 \\ b_1 + a_2 + a_3 > 0 \end{cases} \tag{19}$$

It is obvious that every weight that satisfies (19) can be the initial weight of the MVN. Here we can take a complex-valued initial weight

$$W_1 = \left(i,\ \frac{1}{2} + \frac{\sqrt{3}}{2}i,\ \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i \right) \tag{20}$$

or a real-valued initial weight

$$W_1 = (0.1,\ 0.2,\ -0.1) \tag{21}$$

Let t be the index of training epoch and j be the index of the training vector, so initially $t = 1$ and $j = 1$. The MVN training is performed as follows:

Epoch t:

1) *Compute the weighting sum* $z = w_0 + w_1x_1 + \cdots + w_nx_n$ for the first $(j = 1)$ pattern vector X_j

$$z_j = W_j \cdot X_j \tag{22}$$

and finally $\varepsilon^{s_j} = P(z_j)$

2) *Compute the error*. The actual output ε^{s_j} is compared to the desired output ε^{q_j}

$$\delta_j = \varepsilon^{q_j} - \varepsilon^{s_j} \tag{23}$$

If the error δ_j is not zero, then go to 3) to update the weight, otherwise, go to 1) and $j = j + 1$.

3) *Weights update*. Weight corrections can be performed using (3) that results in the following weights:

$$W_{r+1} = W_r + \frac{C_r}{n+1}(\varepsilon^{q_r} - \varepsilon^{s_r})\overline{X_r} \tag{24}$$

4) *Completion of the training epoch*. Repeat steps 1)−3) for all pattern vectors (till $j = 4$)
5) *Termination of training*. If the actual outputs of all samples are equal to the desired outputs, then the learning process stops. Otherwise, increase and perform steps 1)−4) again.

Remark 2.2: From Table I, we can see that the non-zero error is repeated in the learning process not only for the complex-valued initial weight and but also for the real-valued initial weight. It shows that the learning rule (3) of the discrete MVN falls into an invariant set, and convergence of the learning rule (3) is related to the choice of the initial weight. Note that the learning rate C_r can be taken as any positive real-valued variable, which does not affect the conclusion of this example.

Remark 2.3: Example 2.1 is only used to illustrate the inefficiencies of the MVN learning algorithms (3) and (4) for 2-separated problems. Note that the continue MVN learning algorithms (6) and (7) can solve this example by using real-valued learning rate. However, the convergence of the MVN algorithms (3), (4), (6) and (7) with real-valued learning rate can not strictly be proved so far [8].

III. Comment

An common inconsistency in the proofs of Theorem 3.17 [4] and Theorem 1 [5] is as the following:

I) **[2,(3.101)-(3.102)]**: This inequality is invalid. The original authors try to use $|\beta| \geq |\text{Re}(\beta)|$ and $|\omega_1 X_1 \cdot W_{opt} + \cdots + \omega_r X_r \cdot W_{opt}| \geq |\text{Re}(\omega_1 X_1 \cdot W_{opt}) + ... + \text{Re}(\omega_r X_r \cdot W_{opt}))| \geq ra$, where $w_j = C_j\delta_j$ and $a = \min_{1 \leq j \leq r} |\text{Re}(\omega_j X_j \cdot W_{opt})|$. Note that the last inequality is valid if and only if all $\text{Re}(\omega_j X_j \cdot W_{opt})$ $(1 \leq j \leq r)$ are same sign.

A correction to this problem has been given in [8], which changed the learning rate C_r from real-valued to be complex-valued such that

$$\text{Re}(C_j\delta_j X_j \cdot W_{opt}) \geq 0 \tag{25}$$

that is, the learning rate $C_j \in \mathbb{C}$ has to be chosen such that

$$-\frac{\pi}{2} \leq \arg C_j + \arg \delta_j + \arg(X_j \cdot W_{opt}) \leq \frac{\pi}{2} \tag{26}$$

Remark 3.1: The inequality (26) indicates that the learning rate C_j is dependent on the sample W_j and optimal weight W_{opt}. Since the optimal weight W_{opt} is unknown in the learning process of practical problems, so this correction loses flexibility to determine the learning rate C_j. This shows that the convergence of the MVN algorithms (3), (4), (6) and (7) with real-valued learning rate is still an open problem.

IV. A modified MVN

In order to overcome the difficulty of the MVN, we introduce a modified MVN in [11], its activation function is

$$Q(z) = e^{i\frac{2\pi j}{k}}, \quad \text{if } \frac{\pi(2j-1)}{k} \leq \arg z < \frac{\pi(2j+1)}{k} \tag{27}$$

where $j = 0, 1, ..., k-1$. The relation between the modified MVN and traditional MVN is described by a rotation transform

$$P(z) = Q(z\, e^{-\frac{\pi i}{k}}), \quad Q(z) = P(z\, e^{\frac{\pi i}{k}}) \tag{28}$$

Note that the output of the modified MVN is different from the MVN for the same inputs. This difference directly determines that the modified MVN has the following convergence theorems.

Theorem 4.1: If the learning subsets $A_0, A_1, ..., A_{k-1}$ are k-separable $(k \geq 2)$, then the modified MVN learning rules (3) and (4) with the real-valued learning rate converges after a finite number of steps.

TABLE I. THE MVN TRAINING OF EXAMPLE 2.1

Epoch t	Pattern Vector j	Weights	Weights	Actual Output	Error δ (23)
1	1	$W_1 = \left(i,\ \frac{1}{2} + \frac{\sqrt{3}}{2}i,\ \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i\right)$	$W_1 = (0.1, 0.2, -0.1)$	1	0
	2	W_1	W_1	1	0
	3	W_1	W_1	1	0
	4	W_1	W_1	1	-2
2	1	$W_2 = (-0.6667 + i, 0.5 + 1.5327i, 0.7071 - 0.0404i)$	$W_2 = (-0.5667, 0.2 + 0.6667i, -0.1 + 0.6667i)$	1	0
	2	W_2	W_2	1	0
	3	W_2	W_2	1	0
	4	W_2	W_2	1	-2
3	1	$W_3 = (-1.3333 + i, 0.5 + 2.1994i, 0.7071 + 0.6262i)$	$W_3 = (-1.2333, 0.2 + 1.3333i, -0.1 + 1.3333i)$	1	0
	2	W_3	W_3	1	0
	3	W_3	W_3	1	0
	4	W_3	W_3	1	-2
4	1	$W_4 = (-2 + i, 0.5 + 2.8660i, 0.7071 + 1.2929i)$	$W_4 = (-1.9, 0.2 + 2i - 0.1 + 2i)$	1	0
	2	W_4	W_4	1	0
	3	W_4	W_4	1	0
	4	W_4	W_4	1	-2
5	1	$W_5 = (-2.6667 + i, 0.5 + 3.5327i, 0.7071 + 1.9596i)$	$W_5 = (-2.5667, 0.2 + 2.6667i, -0.1 + 2.6667i)$	1	0
	2	W_5	W_5	1	0
	3	W_5	W_5	1	0
	4	W_5	W_5	1	-2
6	1	$W_6 = (-3.3333 + i, 0.5 + 4.1994i, 0.7071 + 2.6262i)$	$W_6 = (-3.2333, 0.2 + 3.3333i, -0.1 + 3.3333i)$	1	0
	2	W_6	W_6	1	0
	3	W_6	W_6	1	0
	4	W_6	W_6	1	-2
7	1	$W_7 = (-4 + i, 0.5 + 4.8660i, 0.7071 + 3.2929i)$	$W_7 = (-3.9, 0.2 + 4i - 0.1 + 4i)$	1	0
	2	W_7	W_7	1	0
	3	W_7	W_7	1	0
	4	W_7	W_7	1	-2

Proof: Since the learning subsets are k-separable, this means that there exists an optimal weighting vector $W^\star$ such that

$$P(W_{opt} \cdot X_r) = \varepsilon^{q_r} \tag{29}$$

Let $X_r^{'} = \varepsilon^{-q_r} X_r$, then the last equation is equivalent to

$$P(W_{opt} \cdot X_r^{'}) = \varepsilon^0 = 1 \tag{30}$$

Thus, by (2) we can get

$$0 \leq \arg[W_{opt} \cdot X_r^{'}] < \frac{2\pi}{k} \tag{31}$$

Let $\theta = \min_{1 \leq j \leq N}\{\frac{2\pi}{k} - \arg[W_{opt} \cdot X_j^{'}]\}$, N is the number of the learning samples, so $\theta > 0$ and

$$0 \leq \arg[W_{opt} \cdot X_r^{'}] \leq \frac{2\pi}{k} - \theta \tag{32}$$

Let $V_{opt} = e^{-i(\frac{\pi}{k} - \frac{\theta}{2})} W_{opt}$, then

$$-\frac{\pi}{k} + \frac{\theta}{2} \leq \arg[V_{opt} \cdot X_r^{'}] \leq \frac{\pi}{k} - \frac{\theta}{2} \tag{33}$$

The parameter θ is a measure of how close the solution decision boundary is to the input patterns.

In the learning process, the weights need to be updated if and only if $\varepsilon^{s_r} \neq \varepsilon^{q_r}$ (that is $s_r \neq q_r$) holds, so $W_{r+1} \neq W_r$ is right for $r \in \mathbb{N}$. The Theorem will be proven if we show that r (the index of iteration) has a upper bound.

We transform the learning rule (3) to

$$\overline{W_{r+1}} = \overline{W_r} + \frac{C_r}{n+1}(1 - \varepsilon^{q_r - s_r}) X_r^{'} \tag{34}$$

where $s_r \neq q_r \in \{0, 1, ..., k-1\}$. We may iteratively solve this equation for $\overline{W_{r+1}}$ and obtain the result

$$\overline{W_{r+1}} - \overline{W_1} = \frac{1}{n+1}\sum_{p=1}^{r} C_p (1 - \varepsilon^{r_p}) X_p^{'} \tag{35}$$

where W_1 is the initial weighting, $r_p = (q_p - s_p) \bmod k$, and $1 \leq r_p \leq k-1$. We compute a dot product of both parts of (35) with V_{opt}

$$\begin{aligned} & V_{opt} \cdot \left(\overline{W_{r+1}} - \overline{W_1}\right) \\ & = \frac{1}{n+1}\sum_{p=1}^{r} C_p (1 - \varepsilon^{r_p})(V_{opt} \cdot X_p^{'}) \end{aligned} \tag{36}$$

From $1 \leq r_p \leq k-1$, it is evident that

$$\arg[C_p(1 - \varepsilon^{r_p})] = -\frac{\pi}{2} + \frac{\pi r_p}{k} \tag{37}$$

Taking into account that

$$\begin{aligned} & \arg[C_p(1 - \varepsilon^{r_p})(V_{opt} \cdot X_p^{'})] \\ & = \arg[C_p(1 - \varepsilon^{r_p})] + \arg[V_{opt} \cdot X_p^{'}] \end{aligned} \tag{38}$$

from the inequality (33) and equation (37), we obtain

$$\begin{aligned} \frac{\pi(r_p - 1)}{k} - \frac{\pi - \theta}{2} & \leq \arg[C_p(1 - \varepsilon^{r_p})(V_{opt} \cdot X_p^{'})] \\ & \leq \frac{\pi(r_p + 1)}{k} - \frac{\theta + \pi}{2} \end{aligned} \tag{39}$$

Let us substitute $\min_{p \geq 1}\{r_p\} = 1$ and $\max_{p \geq 1}\{r_p\} = k - 1$ in the left-hand and the right-hand sides of the last inequality, respectively. Hence we obtain the following

$$-\frac{\pi-\theta}{2} \le \arg[C_p(1-\varepsilon^{r_p})(V_{opt}\cdot X_p')] \le \frac{\pi-\theta}{2} \quad (40)$$

The last inequality shows that the complex numbers $(1-\varepsilon^{r_p})(V_{opt}\cdot X_p')$ belong to the right semi-planes, then

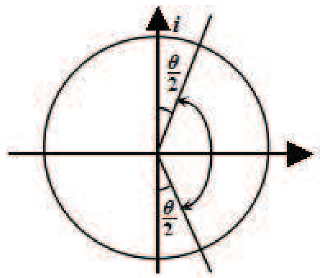

Fig. 2. The diagram of angular inequality (40)

$$\mathrm{Re}[C_p(1-\varepsilon^{r_p})(V_{opt}\cdot X_p')] \ge 0 \quad (41)$$

We estimate the minimum of $\frac{1}{n+1}\mathrm{Re}[C_p(1-\varepsilon^{r_p})(V_{opt}\cdot X_p')]$

$$\begin{aligned}\alpha &= \min_{p\ge1}\left\{\frac{1}{n+1}\mathrm{Re}\left[C_p\left(1-\varepsilon^{r_p}\right)\left(V_{opt}\cdot X_p'\right)\right]\right\}\\ &\ge \frac{1}{n+1}\min_{p\ge1}C_p\min_{p\ge1}\left|1-\varepsilon^{r_p}\right|\min_{p\ge1}\left|V_{opt}\cdot X_p'\right|\cos\frac{\pi-\theta}{2}\\ &= \frac{1}{n+1}\min_{p\ge1}\left|V_{opt}\cdot X_p'\right|C_{min}2\sin\frac{\pi}{k}\cos\frac{\pi-\theta}{2}\\ &= \frac{1}{n+1}\min_{1\le j\le N}\left|W_{opt}\cdot X_j\right|2C_{min}\sin\frac{\pi}{k}\sin\frac{\theta}{2} > 0 \quad (42)\end{aligned}$$

where $C_{min} = \min_{p\ge1}\{C_p\}$ and N is the number of the learning samples. By (36) and (42), we have

$$\begin{aligned}&\mathrm{Re}\left[V_{opt}\cdot\left(\overline{W_{r+1}}-\overline{W_1}\right)\right]\\ &= \frac{1}{n+1}\sum_{p=1}^{r}\mathrm{Re}\left[C_p\left(1-\varepsilon^{r_p}\right)\left(V_{opt}\cdot X_p'\right)\right] \ge r\alpha \quad (43)\end{aligned}$$

By the last inequality and the Schwartz inequality, we obtain

$$\begin{aligned}r\alpha &\le \mathrm{Re}\left[V_{opt}\cdot\left(\overline{W_{r+1}}-\overline{W_1}\right)\right]\\ &\le \left|V_{opt}\cdot\left(\overline{W_{r+1}}-\overline{W_1}\right)\right| \le \|V_{opt}\|\left\|\left(\overline{W_{r+1}}-\overline{W_1}\right)\right\|\\ &= \|W_{opt}\|\left\|\left(\overline{W_{r+1}}-\overline{W_1}\right)\right\| \quad (44)\end{aligned}$$

where $\|\cdot\|$ is the Euclidean 2-norm. Then it follows from the last inequality that

$$\frac{r\alpha}{\|W_{opt}\|} \le \|W_{r+1}-W_1\| \le \|W_{r+1}\|+\|W_1\| \quad (45)$$

or, equivalently,

$$\frac{r\alpha}{\|W_{opt}\|} - \|W_1\| \le \|W_{r+1}\| \quad (46)$$

Case I: $\frac{r\alpha}{\|W_{opt}\|}-\|W_1\|<0$, that is $r<\frac{1}{\alpha}\|W_{opt}\|\|W_1\|$, then the upper bound of the number of iteration r has been got.
Case II: $\frac{r\alpha}{\|W_{opt}\|}-\|W_1\|\ge0$, (46) can be refined to

$$0 \le \frac{r\alpha}{\|W_{opt}\|} - \|W_1\| \le \|W_{r+1}\| \quad (47)$$

In Case II, we need to obtain another estimate. Let $d_p = \frac{C_p}{n+1}(\varepsilon^{r_p}-1)$, we can rewrite (3) in the form

$$\begin{aligned}W_{p+1} &= W_p + \frac{C_p}{n+1}(\varepsilon^{r_p}-1)\varepsilon^{s_p}\overline{X_p}\\ &= W_p + d_p\,\varepsilon^{s_p}\overline{X_p} \quad (48)\end{aligned}$$

By taking the squared Euclidean norm of both sides of the last equation and observe that $z_p = W_p\cdot X_p$, we obtain

$$\begin{aligned}\|W_{p+1}\|^2 &= W_{p+1}\cdot\overline{W_{p+1}}\\ &= \left(W_p + d_p\,\varepsilon^{s_p}\overline{X_p}\right)\cdot\left(\overline{W_p}+\overline{d_p}\,\varepsilon^{-s_p}X_p\right)\\ &= \|W_p\|^2+|d_p|^2\|X_p\|^2+2\,\mathrm{Re}\left[\overline{d_p}\,\varepsilon^{-s_p}z_p\right] \quad (49)\end{aligned}$$

Note that $\varepsilon^{s_p} = Q(z_p)$ and $Q(\varepsilon^{-s_p}z_p) = \varepsilon^0 = 1$. By using (27), it can be shown that

$$-\frac{\pi}{k} \le \arg[\varepsilon^{-s_p}z_p] < \frac{\pi}{k} \quad (50)$$

Observe that $1\le r_p\le k-1$ and $d_p = \frac{C_p}{n+1}(\varepsilon^{r_p}-1)$, we have

$$\arg[\,\overline{d_p}\,] = \frac{3\pi}{2} - \frac{\pi r_p}{k} \quad (51)$$

From the inequality (50) and equality (51), we can obtain

$$\frac{3\pi}{2}-\frac{\pi(r_p+1)}{k} \le \arg[\,\overline{d_p}\,\varepsilon^{-s_p}z_p] < \frac{3\pi}{2}-\frac{\pi(r_p-1)}{k} \quad (52)$$

Let us substitute $\max(r_p) = k-1$ and $\min(r_p) = 1$ in the left-hand and the right-hand sides of the last inequality, respectively. Hence we obtain the following

$$\frac{\pi}{2} \le \arg[\,\overline{d_p}\,\varepsilon^{-s_p}z_p] < \frac{3\pi}{2} \quad (53)$$

The last inequality shows that the complex numbers $\overline{d_p}\,\varepsilon^{-s_p}z_p$ belong to the left semi-planes, then

$$\mathrm{Re}[\,\overline{d_p}\,\varepsilon^{-s_p}z_p] \le 0 \quad (54)$$

We therefore deduce from (49) that

$$\|W_{p+1}\|^2 \le \|W_p\|^2+|d_p|^2\|X_p\|^2 \quad (55)$$

Adding the left-hand and right-hand sides of the last inequality for $p = 1, 2, ..., r$, we get the inequality

$$\|W_{r+1}\|^2 \le \|W_1\|^2+\sum_{p=1}^{r}|d_p|^2\|X_p\|^2 \le \|W_1\|^2+r\beta \quad (56)$$

where β ia a positive number given by

$$\begin{aligned}\beta &= \max_{p\ge1}|d_p|^2\|X_p\|^2 \le \max_{p\ge1}|d_p|^2\max_{p\ge1}\|X_p\|^2\\ &\le \frac{4C_{max}^2}{(n+1)^2}\max_{1\le j\le N}\|X_j\|^2 \quad (57)\end{aligned}$$

where $C_{max} = \max_{p\geq 1}\{C_p\}$. The equation (56) states that $\|W_{r+1}\|^2$ grows at most linearly with the iteration index r.

Finally, we combine the inequalities (47) and (56) to conclude that

$$\left(\frac{r\alpha}{\|W_{opt}\|} - \|W_1\|\right)^2 \leq \|W_{r+1}\|^2 \leq \|W_1\|^2 + r\beta \qquad (58)$$

or, equivalently,

$$r \leq \frac{2}{\alpha}\|W_1\|\,\|W_{opt}\| + \frac{\beta}{\alpha^2}\|W_{opt}\|^2 \qquad (59)$$

According to the conditions of Theorem 4.1, (42) and (57), we know that W_1, W_{opt}, α and β are the constants. Therefore, the number of iteration r always has an upper bound. That means that the learning rule (3) will converge in a finite number of iterations.

The finite convergence of the learning rule (4) can be proved in a similar way. The only difference is C_r turns into $\frac{C_r}{|Z_r|}$. So we give that

$$\begin{aligned} C_{min} &\triangleq \min_{p\geq 1}\frac{C_p}{|z_p|} = \min_{p\geq 1}\frac{1}{|W_p \cdot X_p|} \\ &\geq \frac{1}{\max_{p\geq 1}\|W_p\| \max_{1\leq j\leq N}\|X_p\|} \end{aligned} \qquad (60)$$

The rest part is essentially the same. ■

Remark 4.1: The proof of Theorem 4.1 eliminates the restrictions on the learning rate in (26) and makes the learning rate revert to be real-valued and be independent on the learning samples and the optimal weight. This proof does not need the finiteness condition of $\|W_r\|$ required in [4] and [5].

Remark 4.2: The maximum number of iterations (changes to the weight vector) is inversely related to α (cf.(59)). This parameter is linearly related to $\sin\frac{\pi}{k}\sin\frac{\theta}{2}$ (cf.(42)). This means that if input classes are difficult to separate (are close to the decision boundary) or k (value of k-valued logic) in (2) is too large, it will take many iterations for the algorithm to converge.

V. Conclusions

In this paper, we introduce a modified discrete MVN, which enables the algorithms (3) and (4) to reuse the real-valued learning rate for the k-separated problems ($k \geq 2$). We have given a rigorous convergence proof of the modified MVN learning algorithms (3) and (4) with the real-valued learning rate. Our proof does not require restricting the learning rate to be complex-valued and the finiteness condition of $\|W_r\|$.

Acknowledgment

We thank Prof. Igor Aizenberg, Prof. Danilo P. Mandic and the anonymous reviewers for their valuable discussions and insightful comments.

References

[1] A. Hirose, *Complex-Valued Neural Networks, Second Edition.* Berlin, Heidelberg: Springer, 2012.

[2] D. P. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models.* New York, USA: Wiley, 2009.

[3] I. Aizenberg, N. Aizenberg, and J. Vandewalle, *Multi-Valued and Universal Binary Neurons: Theory, Learning, Applications.* Boston, MA: Kluwer, 2000.

[4] I. Aizenberg, *Complex-valued neural networks with multivalued neurons.* Springer-Verlag, Berlin Heidelberg, 2011.

[5] I. Aizenberg, "A periodic activation function and a modified learning algorithm for a multi-valued neuron," *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1939-1949, Dec. 2010.

[6] I. Aizenberg and C. Moraga, "Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm," *Soft Comput.*, vol. 11, no. 2, pp. 169-183, Jan. 2007.

[7] I. Aizenberg, D. V. Paliy, J. M. Zurada and J. T. Astola, "Blur identification by multilayer neural network based on multi-valued neurons," *IEEE Trans. Neural Netw.*, vol. 19, no. 5, pp. 883-898, May 2008.

[8] I. Aizenberg, "Adjustments to the proofs of the convergence theorems," Availabe at: *http://www.eagle.tamut.edu/faculty/igor/CVNN-MVN_book_Convergence_Proofs_Adjustments.htm.*

[9] M. K. Muezzino, C. Guzelis, and J. M. Zurada, "A New Design Method for the Complex-Valued Multistate Hopfield Associative Memory," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 891–899, July 2003.

[10] N. N. Aizenberg, Yu L. Ivaskiv and D. A. Pospelov,"About one generalization of the threshold function" *Doklady Akademii Nauk SSSR (The reports of the Academy of Sciences of the USSR).*, vol. 196, no 6, pp. 1286-1290 (in Russian), 1971.

[11] S. Jankowski, A. Lozowski, and J.M. Zwrada, "Complex-valued Multistate Neural Assoliative Memory," *IEEE Trans. Neural Netw.*, vol.7, no 6, pp. 1491-1496, November 1996.

[12] T. Isokawa, H. Nishimura , and N. Matsui, "An Iterative Learning Scheme for Multistate Complex-Valued and Quaternionic Hopfield Neural Networks," *Proc. of IJCNN2009.*, pp. 1365–1371, INNS/IEEE, June 2009.