CHAPTER 8

MULTILAYER FEEDFORWARD NEURAL NETWORK WITH MULTI-VALUED NEURONS FOR BRAIN–COMPUTER INTERFACING

NIKOLAY V. MANYAKOV,1 IGOR AIZENBERG,2 NIKOLAY CHUMERIN,1 AND MARC M. VAN HULLE1

1Laboratory for Neuro- & Psychophysiology, KU Leuven, Leuven, Belgium
2Computational Intelligence Laboratory, Texas A&M University-Texarkana, Texarkana, Texas, USA

In this chapter, we describe a multilayer feedforward neural network equipped with multi-valued neurons and its application to the domain of brain–computer interfacing (BCI). A new methodology for electroencephalogram (EEG)-based BCI is developed with which subjects can issue commands by looking at the corresponding targets that are flickering at the same frequency but with different initial phase. Two filter-based feature selection procedures are discussed for extracting relevant information from the phases estimated from the recorded EEGs. The proposed multichannel methodology is compared with existing single channel approaches and the results show that the former performs better in terms of accuracy and length of EEG interval considered for phase estimation.

8.1 BRAIN–COMPUTER INTERFACE (BCI)

Ever since it was acknowledged that the brain controls our intentions, motivations, decisions, muscles, and so on, theories were developed and experiments set up to better understand its neural origin. These efforts also led to the challenge to mimic
the brain by constructing artificial ones or to incorporate some of its traits into intelligent systems. One example is the artificial neural networks. They show the kind of adaptive behavior, learning capacity, and pattern recognition capabilities which we generally attribute to the brain. They also were the first to generate these capabilities by modifying synaptic connections between the artificial neurons, which is considered to be one of the ways our brain stores memories and displays adaptive behavior. Besides understanding the brain by developing models and theories, one also started to record and monitor brain activity directly, as well as to decode it in terms of the underlying sensory, cognitive and motor processes.

Soon after Hans Berger performed his first electroencephalography (EEG) recordings back in 1928, scientists started to analyze change in the electrical activity recorded over the subject’s scalp in response to external stimuli or changes in the subject’s mental or cognitive state. They found correlates providing, albeit indirectly, important insights into how the evoked EEG changes could be turned into command(s) for controlling a computer. This is in fact the principle behind the Brain–Computer Interfaces (BCIs). It records brain activity, decodes it, and issues commands, all with the aim to enable the subject, from whom the recording are made, to interact with the external world, by controlling a computer program (e.g., a game), a robot actuator, and so on, bypassing the need for muscular activity. While this was first demonstrated in the early 1970’s [51], BCIs received widespread attention only recently with the advent of EEG devices and computer technology that made it possible to perform on-line monitoring and decoding. BCI research is now widely considered as one of the most successful applications of the neuroscience as it can provide an outlook for immediately improving the quality of life of patients suffering from severe communication and motor disabilities such as in the case of (terminal stage) amyotrophic lateral sclerosis, stroke, traumatic brain or spinal cord injury, cerebral palsy, muscular dystrophy etc. For example, with the use of a BCI, a locked-in syndrome patient (re)gains the ability to communicate his/her intentions, desired motor actions, emotions, etc. [32, 37].

Any BCI system (see Fig. 8.1) consists of the following components: a recording device, a preprocessor, a decoder, and an external device, such as a robot actuator or a computer display where the result of the issued commands are shown to the subject. Brain activity is either recorded inside (invasively) or outside the brain (noninvasively), e.g., on the scalp. Apart from EEG, noninvasive BCIs [10, 15, 11] have been described that employ magnetoencephalography (MEG) and functional magnetic resonance imaging (fMRI). Invasive BCIs are based on electrode arrays implanted in brain tissue [50, 38, 39] or just above it as in the case of the electrocorticograms (ECoG) [28]. Preprocessing is done with the spatial and temporal filtering (for example, the notch filter to remove power line interference) of the recorded signals, along with the selection and construction of signal features that detect information relevant for the considered decoding task. The decoder performs the classification or regression of the preprocessed signal into the control signal of the external device.

3Sometimes the term Brain–Machine Interfaces (BMIs) is used, generalizing the type of controllable device.
device. The feedback provided to the subject, of the outcome of the decoder, is an important aspect of the BCI system as it enables one to detect mistakes and motivates in this way the subject to better modulate his/her brain activity, when adhering to a so-called neurofeedback principle, or to improve the preprocessor and the decoder, or both. Thus, when the feedback is used to improve the outcome, the BCI can be viewed as a closed-loop system.

### 8.1.1 Invasive BCI

The origin of invasive BCI’s can be traced back to 1999 when, for the first time, it was shown that ensembles of cortical neurons could directly control a robotic manipulator [13]. Since then a steady increase in the number of publications can be observed. Invasive BCIs can be divided according to the type of signal used. They can rely on action potentials (spikes), which is a short-lasting event with a rapid, stereotypical change in the membrane potential. Spike trains are extracted by high-pass filtering the extracellular recordings (above 500 Hz). For BCI purposes, recordings are made from either a single cortical area (for example, the primary motor cortical area, M1) or from multiple ones, thereby taking advantage of the distributed processing of information in the brain. For decoding purposes, either spike trains from a few neurons, with prominent tuning properties [46, 47], or from a large ensemble of neurons (hundreds of cells) [38, 12, 54] are considered. If the low-frequency component (below 300 Hz) of the extracellular recording is used, then we are dealing with local field potentials (LFPs). LFPs represent the composite extracellular potentials from hundreds or thousands of neurons around the electrode tip. They are more stable than spikes as they can be recorded for longer period of time, which makes them attractive for BCI applications [39, 40, 44]. All invasive BCIs require brain surgery; and, as a consequence, much of the research is done on animal such as monkeys [46, 47, 26], rather than directly on humans [25, 23]. Spike- or LFP-based BCIs are primary developed for motor control, for example, of an arm actuator [26, 46, 47, 54], where
for decoding a linear regression of the spike firing rate into the position and velocity of the limb is usually considered. For a review of invasive BCI, we refer to Ref. 26.

8.1.2 Noninvasive BCI

Noninvasive BCI mostly rely on electroencephalogram (EEG) recordings, which measure cumulative simultaneous synchronous activity of thousands or millions of pyramidal neurons mainly in close proximity of the scalp. Since such recordings do not require any surgery, and therefore can be performed on human subjects directly, they have attracted more attention in the scientific community than their invasive counterparts. Noninvasive BCI can be divided according to the brain signal paradigm used. We can distinguish BCIs based on event-related potentials (ERPs) [19, 17, 16], which are stereotyped electrophysiological responses to time- and phase-locked internal or external stimuli [30]. In order to detect the ERP component in the signal, one trial is usually not enough and several trials should be averaged to reduce additive noise and other irrelevant activity in the recorded signals. The most-known BCI of this type is the one based on the cognitive P300 (‘oddball’) component of ERP in the parietal cortex, which is evoked in response to an infrequent preferred stimulus but not to a frequent, nonpreferred stimulus. The aim of the decoder is then to detect an enhanced positive-going signal component with a latency of about 300 ms when observing the rare stimulus [43]. It has been widely used to achieve a letter spelling or other type of communication system operating in visual, auditory, or tactile mode [19, 15, 49]. Besides cognitive, also early sensory components of the ERP have been used, — for example, the visually evoked potential (VEP) [51].

Whereas BCIs based on ERPs deal with information encoded in the temporal domain (the averaged ERP waveform), another BCI detects changes in the power of a particular frequency band (“rhythm”) evoked by some voluntary acts, for example, the imagination of right/left-hand movements. Since those events in EEG recordings are not phase locked to the onset of the voluntary acts, an averaging method is of no use here. The relevant information is primarily encoded in the frequency domain. By monitoring event-related desynchronization (ERD) and event-related synchronization (ERS) in the mu- and beta-bands, in response to the imagined movement, one can construct different types of BCI applications [56, 42]. The detection of other mental tasks (e.g., imagined cube rotation, number subtraction, word association [41]) also belong to this category.

8.2 BCI BASED ON STEADY-STATE VISUAL EVOKED POTENTIALS

In this chapter we focus on a particular type of VEP-based BCI, namely one based on the Steady-State Visual Evoked Potential (SSVEP). It relies on the psychophysiological properties of EEG responses recorded from the occipital pole during the periodic presentation of identical visual stimuli (i.e., flickering stimuli). When the periodic presentation is at a sufficiently high rate (above 6 Hz), the individual transient vi-
sual responses overlap, leading to a *steady-state* signal: the signal resonates at the stimulus rate and its multipliers [30]. This means that, when the subject is looking at stimuli flickering at frequency $f$, the frequency $f$ and its harmonics $2f, 3f, \ldots$ are salient in the Fourier transform of the EEG signal, as schematically illustrated in Figure 8.2. However, as the amplitude of a typical EEG signal decreases as $1/f$ in the spectral domain [8], the power of $f$ decreases with increasing frequency (as is the case with the harmonics as well).

![Figure 8.2](image-url) Schematic overview of the SSVEP decoding approach: (A) the subject looks at the Target stimulus, flickering at frequency $f$, (B) noisy EEG-signals are recorded, (C) the power spectral density plot of the EEG signal (estimated over a sufficiently large time window) shows dominant peaks at $f$, $2f$, and $3f$.

As the SSVEP is embedded in other ongoing brain activity and (recording) noise; hence, when considering a too small recording window, the flickering frequency $f$ could not be detected or erroneously detected. To overcome this problem, averaging over several time intervals [14], recording over longer time intervals [53] and/or preliminary training [18, 31, 34] are often used for increasing the *signal-to-noise ratio* (SNR) and the detectability of the responses.

An SSVEP-based BCI can be considered as a dependent one according to the classification proposed in Ref. 57. The dependent BCI does not use the brain’s normal output pathways (for example, the brain’s activation of muscles for typing a letter) to carry the message, but activity in these pathways (e.g., muscles) is needed to generate the brain activity (e.g., EEG) that does carry it. In the case of SSVEP BCI, the brain’s output channel is EEG, but the generation of the EEG signal depends on the gaze direction (subject should look at flickering stimulus) and therefore on extraocular muscles and the cranial nerves that activate them. A dependent BCI is essentially an alternative method for detecting/decoding of the messages carried by the brain’s normal output pathways. It has the advantages of a high information transfer rate (the amount of information communicated per unit time) [9] and little (or no) user training [53].

As a stimulation device for SSVEP BCI, either light-emitting diodes (LEDs) or computer screens (LCD or CRT monitors) are used [59]. While the LEDs can evoke
MULTILAYER FEEDFORWARD NEURAL NETWORK WITH MULTI-VALUED NEURONS FOR BCI

\[
f_1 = 30 \text{ Hz} 
\]

\[
f_2 = 20 \text{ Hz} 
\]

\[
f_3 = 15 \text{ Hz} 
\]

\[
f_4 = 12 \text{ Hz} 
\]

(a) ‘On’/‘off’ frequency-coded stimulation.

\[
f = 15 \text{ Hz}, \quad \Delta \phi_1 = 0 
\]

\[
f = 15 \text{ Hz}, \quad \Delta \phi_2 = \frac{\pi}{2} 
\]

\[
f = 15 \text{ Hz}, \quad \Delta \phi_3 = \pi 
\]

\[
f = 15 \text{ Hz}, \quad \Delta \phi_4 = \frac{3\pi}{2} 
\]

(b) ‘On’/‘off’ (50% duty cycle) phase-coded stimulation.

Figure 8.3 Examples of frame-based (“on/off”) stimulation patterns for frequency-coded (a) and phase-coded (b) SSVEP BCIs. The white-shaded squares indicate the “on” frames, while the dark-shaded squares are the “off” frames. Only 30 frames corresponding to 500 ms of stimulation (on a 60-Hz screen) are shown.

more prominent SSVEP responses [59], they still require additional equipment (considering that the feedback is presented on a computer screen). Therefore, SSVEP-based BCI systems mostly rely on computer screens for visual stimulation in order to combine stimulation and feedback presentation devices. However, the latter also have their disadvantages as the stimulation frequencies need to be related to the refresh rate of the computer screen [52] (see Section 8.2.1), whereas LEDs can be stimulated at any desirable frequency. In any case, the flickering frequency needs to be restricted to specific (subject-dependent) frequency bands to obtain good responses [34].

8.2.1 Frequency-Coded SSVEP BCI

The stimulation sequence is constructed in the following way: an intense (“on”) stimulus is shown for \(q\) frames, and a less intense (“off”) stimulus for the next \(w\) frames. Hence, the flickering period of the stimulus is \(q + w\) frames and the corresponding stimulus frequency is \(f_{scr}/(q + w)\), where \(f_{scr}\) is the refresh rate of the computer screen used for stimulation. An example of an “on/off” stimulations pattern (based on \(q = 1\) “on” frames, indicated by the white-shaded squares) for 20, 15, 12, and 10 Hz constructed for an \(f_{scr} = 60\) Hz is shown in Fig. 8.3(a). The whole subset of SSVEP stimulation frequencies for a 60-Hz screen consists of the following frequencies: 30, 20, 15, 12, 10, 8.57, 7.5, 6.66, and 6 Hz.

Frequency-coded SSVEP BCIs rely on the detection of the SSVEP induced by one of several \((f_1, \ldots, f_N)\) stimulation frequencies, one for each selectable target. Thus, the information is encoded into the frequency domain. The detection is usually based on monitoring of an increase (with respect to the normal resting condi-
tion) in the power spectral density (psd) of one of the stimulation frequencies (i.e., $f_1, \ldots, f_N$ and their subharmonics) during the time the subject is gazing at the corresponding target [21]. After detecting the increase, e.g., at frequency $f$, the BCI system produces an output (command) associated with that target.

The number of encodable targets in frequency-coded SSVEP-based BCI is factually even lower than the dividers of the screen’s refresh rate as it is not desirable to take frequencies that are dividers of one another [52]. To clarify this, consider again the previously mentioned nine frequencies for a 60-Hz screen. Assume we have 10 Hz and 20 Hz flickering frequencies. In this case, when the subject is gazing at the 10-Hz target, we have an increase in the psd at 10 Hz and consequent subharmonics, which are 20 Hz, 30 Hz, and so on. While powers in those subharmonics are decreasing with frequency, there is still a significant peak at 20 Hz, i.e., the 10-Hz stimulation produces also an increase in psd amplitude at 20 Hz. Thus, it is likely to make a mistake by erroneously interpreting the 20-Hz frequency based on EEG recordings. Due to this fact, we have to further limit the number of possible frequencies simultaneously used for stimulation.

### 8.2.2 Phase-Coded SSVEP BCI

In order to increase the number of encodable targets in SSVEP-based BCI, the phase has been proposed in addition (or as an alternative) to the frequency [24, 27, 33]: even a single frequency could be used with different phase lags for encoding different targets. For example, assume we have an $f = 15$-Hz stimulation, on a 60-Hz monitor, which is constructed by repeating one stimulation period (with length $T_{fr} = 4$ frames) of $q = 1$ intense frames followed by $w = 3$ “off” frames. This leads to the stimulation profile $\alpha(k)$ (where $k$ is the video frame index) as visualized in Fig. 8.3(b). Let us consider a phase lag $\Delta \phi$. One period of stimulation takes $T_{fr}$ frames and corresponds to $2\pi$, thus, the phase lag $\Delta \phi$ produces a corresponding stimulation delay of $\Delta k = T_{fr} \Delta \phi / 2\pi = 2 \Delta \phi / \pi$ frames. Note, that $\Delta k$ must be an integer, which is the case when $2 \Delta \phi / \pi \in \mathbb{Z}$. For example, with a phase lag $\Delta \phi = \frac{\pi}{2}$, the corresponding frame lag is $\Delta k = 1$ and the corresponding stimulation profiles for targets 2, 3, and 4 are constructed by shifting in time the profile $\alpha(k)$, by 1, 2, and 3 frames (into the future), respectively, as depicted in Fig. 8.3(b). By design, the resulting set of stimuli is circular (i.e., target 3 could be received from target 1 by shifting on $\pi$ in either direction). The phases estimated from the EEG responses on the phase-coded stimulation also demonstrate a similar property (of being circular). When the subject is presented with such phase-coded stimulations, by extracting phase information from the Fourier transform of the EEG signal at $f = 15$ Hz and by comparing it to the phase of some reference signal (for example, the phase of the EEG response for a stimulus with zero phase lag [27]), one can detect the target the subject is looking at. The comparison to the reference phase is one of the simplest solutions to the more general problem of constructing a mapping of circular data (estimated phases from EEG data may be represented on the unit circle in the complex plane) onto circular target-classes [35, 36]. By combining the frequency- and phase-based approaches, one can increase the number of encoded targets [24].
8.3 EEG SIGNAL PREPROCESSING

8.3.1 EEG Data Acquisition

We made recordings with a wireless EEG system consisting of two parts: an amplifier coupled with a wireless transmitter and a USB receiver (Fig. 8.4(a), 8.4(b)). The wireless EEG system was developed by Holst Centre\(^4\) and built around their ultra-low power eight-channel EEG amplifier chip. Each EEG channel is sampled with a resolution of 12 bit per sample at 1024 Hz. We use an EEG-cap with large filling holes and sockets for active Ag/AgCl electrodes (ActiCap, Brain Products, Fig. 8.4(c)). The recordings are made with eight electrodes located primarily on the occipital pole, namely at positions PO7, PO3, POz, PO4, PO8, O1, Oz, O2 according to the international 10–10 system (Fig. 8.4(d)). The reference and ground electrodes are placed on the right and left mastoids, respectively, mainly for comparing our results to those of [27], in which case the recordings were done with a single Oz electrode referenced to the right mastoid. For further analysis we additionally considered EEGs from the mentioned electrodes with respect to common average reference (CAR)\(^5\) and all possible bipolar combinations,\(^6\) thus leading to 44 channels \(s_d(t)\). The phases were extracted as

\[
\phi_d = \arg \left( \sum_t s_d(t) \cos(2\pi n_h f t) + i \sum_t s_d(t) \sin(2\pi n_h f t) \right)
\]  

(8.1)

where \(i = \sqrt{-1}\), \(f\) is the stimulus frequency, and \(n_h\) indicates the considered (sub-)harmonic(s). We used segments \(s_d(t)\) of length \(T\) (\(T = 1, \ldots, 5\) seconds) cropped from the stimulation stage recordings. We further restrict ourselves to only the principal harmonic, thus considering only \(n_h = 1\), leading to 44-dimensional feature space of phases \(\phi_d\). Note the difference in notations: we use \(\Delta \phi\) to denote the phase shifts in the visual stimulation, while \(\phi\) is used for the phases extracted from the recorded EEG data.

8.3.2 Experiment Description

Seven subjects (all male, aged 23–35, average 28.3 years) participated in the experiment. The subjects were sitting about 60 cm from a notebook’s LCD screen (with refresh rate \(f_{scr} = 60\) Hz) on which the stimuli of size \(6 \times 6\) cm were shown. A set of \(N = 6\) stimuli flickering at \(f = 10\) Hz with phase lags of \(\Delta \phi = \pi/3\) were simultaneously presented using the stimulation profiles shown in Fig. 8.5. The stimulus had a 50\% duty cycle (\(\frac{7}{10}\)·100\%) as this was reported to produce better detectable

\(^4\)http://www.holstcentre.com/

\(^5\)The CAR signal is a difference between the EEG from the current electrode and the average ones from all electrodes used in the recordings.

\(^6\)The resulting signal is the difference between the EEG from the current electrode and other preselected one.
SSVEP responses for mostly all frequencies and for $f = 10$ Hz in particular [48]. The stimuli were arranged in two rows and three columns, separated 7.5 cm horizontally and 7.75 cm vertically. The fixation point marker was placed for five seconds on each stimulus followed by a one second interval allowing the gaze to shift to the next stimulus (during this time no stimuli were shown). Each stimulus was attended by the subject $L = 20$ times. In total, we acquired $6 \times 20 = 120$ five-second-long EEG intervals.

### 8.3.3 Feature Selection

We wish to decode $N$ different phase shifted stimuli (in our case $N = 6$) based on $K$ (in our case $K = 44$) phase features. Since both the input features (extracted phases) and the output classes are circular, our classifier should be able to map the
\(K\)-dimensional torus into a circle. To achieve this, we firstly reduce the number of features \(K\) through a filter-based (thus, not considering the classifier output) feature selection procedure. As it was pointed out in Ref. 22, the objective of feature selection is threefold:

1. To improve the prediction performance of the classifier (by avoiding undesirable input variables that can make the classifier’s learning process to adjust to inseparable data).

2. To provide low computationally complex classifiers that are also more cost-effective (by reducing the number of input channels/variables of the classifier).

3. To yield a better understanding of the underlying process that generated the data.

To understand the necessity of selecting appropriate channels, let us consider Fig. 8.6, where phases extracted from EEG recordings are represented by shaped and colored symbols (each shape and color correspond to a particular phase shifted target in a particular round of the experiment, where the subject was asked to gaze at a particular flickering stimulus for \(T = 1\) second) for three different channels: Oz referenced to the mastoid (as it was considered in Ref. 27), bipolar Oz–POz (as it was considered in Ref. 24) and the best channel according to the standard deviation selection procedure described below. It is clearly seen that the latter yields the best separability of the classes. The feature selection procedure should be applied to training data, with further restriction to selected channels for new data coming for classification.

The feature selection procedure has to take into account the circular nature of the data. Thus, when calculating estimates, we have to rely on circular statistics [20]. In this case, the mean value for a set of phases \(\phi_{1d}, \phi_{2d}, \ldots, \phi_{Ld}\) is estimated as

\[
\text{arg} \left( \sum_{l=1}^{L} e^{i\phi_{ld}} \right)
\]

where \(i\) is the imaginary unit, \(d\) is feature index and \(l\) is the trial index \((l = 1, \ldots, L)\).

The mean values are, for the data of each class, drawn as radial lines in Fig. 8.6. The

\(f = 10\) Hz, \(\Delta \phi_1 = 0\) \(\cdot \cdot \cdot\)
\(f = 10\) Hz, \(\Delta \phi_2 = \frac{\pi}{3}\) \(\cdot \cdot \cdot\)
\(f = 10\) Hz, \(\Delta \phi_3 = \frac{2\pi}{3}\) \(\cdot \cdot \cdot\)
\(f = 10\) Hz, \(\Delta \phi_4 = \pi\) \(\cdot \cdot \cdot\)
\(f = 10\) Hz, \(\Delta \phi_5 = \frac{4\pi}{3}\) \(\cdot \cdot \cdot\)
\(f = 10\) Hz, \(\Delta \phi_6 = \frac{5\pi}{3}\) \(\cdot \cdot \cdot\)

**Figure 8.5** Phase-coded “on/off” stimulation patterns for 10-Hz stimulation on a 60-Hz screen. As in Fig. 8.3, the white-shaded squares indicate the intensified (“on”) frames, and the dark-shaded ones indicate the “off” frames. Each row corresponds to one target stimulation profile.
Figure 8.6  Distribution of estimated phases (expressed as angular values) for subject 1 in the experiment described in Section 8.3.2 for channels Oz (left), POz–Oz (central) and POz–O2 (right). Each dot corresponds to the phase estimated from a one second interval recorded when the subject was observing a particular phase-shifted stimulus. Colors represent target-classes (with the stimulus shifted by $\Delta \phi_{mc} = m_c \pi/3$, where $m_c$ is the class index). Radial lines correspond to the circular means for each class. For the sake of visualization, each class is drawn on a circle with a different radius.

The proposed estimation of the mean value by converting the phases into a complex number $e^{i\phi_l}$ on a unit circle is a convenient way to visualize the data and the classifier output. Additionally to this, the unique correspondence between phases and unit length complex numbers can be further exploited when constructing our classifier.

If we take into account not only the argument but also the length $R_d$ of the sum vector $\sum \sum_l e^{i\phi_l}$, we can estimate the circular standard deviation according to

$$\sqrt{2(1 - R_d)}$$

(8.3)

Those standard deviations are in turn used as the basis of the following heuristic feature selection method. If we look into the distribution of our data (see Fig. 8.6), we may find that channels with a good separability between classes also possess the property that the standard deviation (scatterness of the data) within each class is minimized. Indeed, if we look, for example, into the “black” class (stars) in Fig. 8.6, we observe a decrease in the standard deviation of the right (better separable) channel compared to the channels visualized left from it. Thus, the maximal standard deviation between all encoded classes can serve as a way to arrange the channels in increasing order. By taking the first channels in this ranking, we factually perform a filter feature selection.

As another more rigorous feature selection procedure, we can rely on a statistical test for testing differences between mean values of paired classes. First of all, we assume that the data from each class $m_c$ ($m_c = 1, \ldots, N$) and each feature $d$ ($d = 1, \ldots, K$) are taken from the von Mises distribution

$$p_d^{m_c} (\varphi | \mu_d^{m_c}, \kappa_d^{m_c}) = \exp(\kappa_d^{m_c} \cos(\varphi - \mu_d^{m_c}))/\left(2\pi I_0(\kappa_d^{m_c})\right)$$

(8.4)
where $I_0$ is the modified Bessel function of order zero, and $\kappa_d^m$, and $\mu_d^m$ are the parameters responsible for the circular variance and the circular mean. By considering this assumption and the equalities of $\kappa$’s, we can perform pairwise (each class vs. each class for every channel) Watson–Williams tests, which is the circular analogue of the one-factor ANOVA. Assigning to each channel the maximal $p$-value between all pairwise tests, we can rank them in ascending order. And, as in the previously described feature selection procedure, we perform a filter feature selection procedure by taking the first channels in this ranking. Albeit we rely here on the assumption of a von Mises distribution with equal concentration parameter $\kappa$, it was shown that the Watson–Williams test is robust against deviations from this assumption [58].

8.4 DECODING BASED ON MLMVN FOR PHASE-CODED SSVEP BCI

8.4.1 Multi-Valued Neuron

The discrete multi-valued neuron (MVN) was introduced in Ref. 6 as a neural element based on the principles of multiple-valued threshold logic over the field of complex numbers. These principles have been initially formulated in Ref. 7 and then presented in Ref. 5. The concept of multiple-valued threshold logic over the field of complex numbers was recently comprehensively covered in Ref. 1.

![Geometrical interpretation of the discrete MVN activation function](image)

Figure 8.7 Geometrical interpretation of the discrete MVN activation function $P(z) = e^{i2\pi j/k}$.

The discrete MVN performs a mapping between $n$ inputs and a single output. This mapping is described by a multiple-valued ($k$-valued) function of $n$ variables $f(z_1, z_2, \ldots, z_n)$, which is a function of $k$-valued logic over the field of complex numbers [1]. For simplicity, we will omit in what follows “over the field of complex numbers” but keeping it in mind. The values of the $k$-valued logic are encoded by the $k$th roots of unity $e^{i2\pi j/k}, j \in \{0, 1, \ldots, k-1\}$ ($i$ is an imaginary unity), not by integers $0, 1, \ldots, k-1$ as in the classical $k$-valued logic. A $k$-valued function, which
can be learned by a single MVN, is called a multiple-valued \((k\text{-valued})\) threshold function or a threshold function of \(k\)-valued logic. A \(k\)-valued threshold function can be represented using \(n+1\) complex-valued weights as follows \([5, 1]\):

\[
f(x_1, x_2, \ldots, x_n) = P(w_0 + w_1x_1 + \cdots + w_nx_n) \tag{8.5}
\]

where \(x_1, x_2, \ldots, x_n\) are the variables on which this function depends (neuron inputs), and \(w_0, w_1, \ldots, w_n\) are the weights. The values of the function and of the variables are complex. They are the \(k\)th roots of unity:

\[
\epsilon_j = e^{i2\pi j/k}, \ j \in 0, 1, \ldots, k-1,
\]

\(i\) is an imaginary unity. \(P\) is the activation function of the neuron:

\[
P(z) = e^{i2\pi j/k}, \text{ if } 2\pi j/k \leq \arg z < 2\pi(j + 1)/k \tag{8.6}
\]

where \(j = 0, 1, \ldots, k-1\) are values of \(k\)-valued logic, \(z = w_0 + w_1x_1 + \cdots + w_nx_n\) is the weighted sum, and \(\arg z\) is the argument of the complex number \(z\). The activation function is illustrated in Fig. 8.7. This activation function \((8.6)\) divides the complex plane into \(k\) equal sectors with the angular size \(2\pi/k\). If the weighted sum falls into the sector number \(j \in 0, 1, \ldots, k-1\), then according to \((8.6)\) the neuron’s output is equal to \(\epsilon_j\). Thus, the neuron’s output is completely determined by the argument (phase) of the weighted sum and does not depend on its magnitude. This circularity of the MVN activation function is very important for solving those problems, which we consider in this chapter.

The continuous MVN has been proposed in Ref. 3. Its inputs and output are located on the unit circle and they can be arbitrary points. The continuous MVN activation function is

\[
P(z) = e^{i\text{Arg } z} = z/|z| \tag{8.7}
\]

where \(z = w_0 + w_1x_1 + \cdots + w_nx_n\) is the weighted sum, and \(\text{Arg } z\) is the main value of the argument of the complex number \(z\). Thus, for the continuous MVN, its output is the projection of the weighted sum on the unit circle, as it is determined by the activation function \((8.7)\) (see Fig. 8.8).

MVN learning is detailed in Ref. 1. The most efficient MVN learning algorithm is based on the error-correction rule. The one for the discrete MVN was proposed in Ref. 5, which was then generalized in Ref. 2 for the continuous MVN. The convergence of the learning algorithm for the discrete MVN is proven in Refs. 5 and 1. It is shown in Ref. 4 that for the continuous MVN the learning process is reduced to that of the discrete MVN with \(k \rightarrow \infty\) in \((8.6)\) and it converges as well. The error-correction learning rule for both the discrete and the continuous MVN is as follows. Let \(D\) and \(Y\) be, respectively, the desired and actual outputs of the MVN. Then, the weights adjustment is performed according to the rule

\[
W_{r+1} = W_r + \frac{\alpha_r}{n + 1}(D - Y)\bar{X} \tag{8.8}
\]

where \(\bar{X}\) is the neuron’s input vector with complex-conjugated components, \(r\) is the number of the current weighting vector, \(n\) is the number of neuron inputs (the length of the input vector), \(W_r\) and \(W_{r+1}\) are the weighting vectors, before and after correction, respectively, \(\alpha_r\) is the learning rate (it should always be equal to 1). The
training of the neuron using this rule is performed without requiring differentiability of the activation function (actually, both activation functions (8.6) and (8.7) are not differentiable being functions of a complex variable). In Ref. 2, the following modification of the learning rule (8.8) was suggested:

$$W_{r+1} = W_r + \frac{\alpha_r}{|z_r|} (D - Y) \bar{X}$$  

(8.9)

where $|z_r|$ is the absolute value of the current weighted sum. The learning rule (8.9) works better for those input/output mappings that have many irregular jumps. The factor $1/|z_r|$ in (8.9) is de facto a self-adaptive learning rate. Geometrically, learning rules (8.8) and (8.9) are reduced to a movement along the unit circle in the shortest possible way from the “incorrect” actual output to the “correct” desired output, which is determined by the error (see Fig. 8.9).

8.4.2 Multilayer Feedforward Neural Network with Multi-Valued Neurons (MLMVN)

*Multilayer Feedforward Neural Network with Multi-Valued Neurons (MLMVN)* is a neural network with a standard feedforward topology [45]. This is a multilayer neural network for which all neurons from a given layer receive input from the neurons from the preceding layer. However, the use of MVN as a basic neuron for MLMVN has some important differences and advantages compared to a standard multilayer feedforward neural network (MLFBP) based on sigmoidal neurons (the latter is often referred to as a multilayer perceptron, i.e., MLP). The most important difference is with the way the backpropagation learning process is organized.
As shown in Refs. 2, 4, and 1, the MLMVN learning algorithm is based on the same error-correction learning rule as in the case of a single MVN. Let us consider how the MLMVN learning algorithm is organized.

Let us consider a MLMVN consisting of \( m - 1 \) hidden layers and one output layer. For the output layer (the \( m \)th layer) the weights for the \( k \)th neuron of this \( m \)th layer have to be adjusted according to the following rule:

\[
\bar{w}_{km} = w_{km} + \frac{\alpha_{km}}{N_{m-1} + 1} \delta_{km} \bar{Y}_{l,m-1}, \quad l = 1, \ldots, N_{m-1},
\]

\[
\bar{w}_{0m} = w_{0m} + \frac{\alpha_{km}}{N_{m-1} + 1} \delta_{km}
\]

where \( \bar{w} \) represents the corrected weight, \( N_{m-1} \) the number of neurons in the \( m - 1 \)st layer (the last hidden layer preceding the output layer; evidently, this is also the number of inputs of all neurons in the \( m \)th layer), \( \alpha_{km} \) the learning rate (it should always be equal to 1), \( \bar{Y}_{l,m-1} \) the actual output of the \( i \)th neuron of the \( m - 1 \)st layer, which is corrected when it has the tilde (\( \bar{\cdot} \)) supersign and conjugated when it has the bar (\( \bar{\cdot} \)) supersign, and \( \delta_{km} = \frac{1}{N_{m-1} + 1} \delta_{km}^* \) the error of the \( k \)th neuron of the \( m \)th (output) layer which is obtained from the global network error taken from the same neuron:

\[
\delta_{km}^* = D_{km} - Y_{km}
\]

where \( D_{km} \) and \( Y_{km} \) are, respectively, the desired and actual outputs of the \( k \)th neuron from the \( m \)th layer.

**Figure 8.9** Geometrical interpretation of the MVN learning rule.
For the hidden layers neurons, except for the first one, the error is calculated by backpropagating it from the next layer followed by the weight adjustment:

\[
\begin{align*}
\delta_{kj} &= \frac{1}{N_{j-1}+1} \sum_{i=1}^{N_{j-1}+1} \delta_{i,j+1}(w_{k}^{i,j+1})^{-1}, \\
\tilde{w}_{k}^{lj} &= w_{k}^{lj} + \frac{\alpha_k}{(N_j-1)|z_{kj}|} \delta_{kj} \tilde{Y}_{l,j-1}, \quad l = 1, \ldots, N_j - 1, \\
\tilde{w}_{k}^{0j} &= w_{k}^{0j} + \frac{\alpha_k}{(N_j-1)|z_{kj}|} \tilde{\delta}_{kj}
\end{align*}
\]  

(8.12)

where the indexes \(k,j\) stand for the \(k\)th neuron from the \(j\)th layer, whose weights are adjusted, \(|z_{kj}|\) the absolute value of the current weighted sum of this neuron, \(N_{j-1}\) the number of neurons in the \((j-1)\)st layer (this is also the number of inputs of all neurons in the \(j\)th layer), \(\alpha_k\) the learning rate (it should always be equal to 1), \(\tilde{Y}_{l,j-1}\) the actual output of the \(l\)th neuron of the \((j-1)\)st layer, which is corrected when it has the supersign and conjugated when it has the supersign, and \(\delta_{kj}\) the error of the \(k\)th neuron of the \(j\)th layer. Finally, for the first hidden layer, the error is backpropagated from the second one after which the weights are adjusted as follows:

\[
\begin{align*}
\delta_{k1} &= \sum_{i=1}^{N_2} \delta_{2}(w_{k}^{2})^{-1}, \\
\tilde{w}_{k}^{1l} &= w_{k}^{1l} + \frac{\alpha_k}{(n+1)|z_{kj}|} \delta_{kj} \bar{x}_{l}, \quad l = 1, \ldots, n, \\
\tilde{w}_{k}^{01} &= w_{k}^{01} + \frac{\alpha_k}{(n+1)|z_{kj}|} \tilde{\delta}_{kj}
\end{align*}
\]  

(8.13)

where the indexes \(k,1\) stand for the \(k\)th neuron of the 1st layer, \(x_l\) the \(l\)th input component complex conjugated, \(n\) the number of network inputs (it is also the number of inputs of all 1st hidden layer neurons), \(\alpha_k\) the learning rate (it should always be equal to 1), and \(\delta_{k1}\) the error of the \(k\)th neuron of the 1st layer. The convergence of the learning process based on the learning rules (8.10)-(8.13) is proven in Ref. 1.

The MLMVN has shown very nice results for a number of benchmark problems [2] and real-world problems such as the identification of the type of blur and its parameters in image restoration [4], time-series prediction, among others [1]. It was shown that the MLMVN outperforms a standard MLFBP and many other kernel-based and neuro-fuzzy techniques in terms of the classification/prediction rate. It also employs fewer parameters to solve a particular problem.

### 8.4.3 MLMVN for Phase-Coded SSVEP BCI

For our phase-coded SSVEP BCI system, we want to perform mapping from the phases, estimated from the EEG recordings, onto the target-class indicators. Since both input and output are circular, we can represent them as complex numbers with unit length. For doing so, we convert all preselected phases \(\varphi_d\) (according to the feature selection algorithm described in Section 8.3.3) into complex numbers \(e^{j \varphi_d}\).

Those numbers will be used as an input into our MLMVN. During training, for the training samples of class \(m_c\) \((m_c = 1, \ldots, N)\), where \(N\) is the number of target classes) we used as the network’s desired output value \(D_{m_c} = \exp(i2\pi(m_c-\frac{1}{2})/N)\). Then, after the training, from the output \(Y_{1m}\) of the network \((k = 1\), since we have a network with a single output\), the resulting class index \(\hat{m}\) is deduced as an integer satisfying two conditions: \(2\pi(\hat{m} - 1)/N \leq \arg Y_{1m} < 2\pi\hat{m}/N\) and \(1 \leq \hat{m} \leq N\).
During training we kept track of an angular variant of the root mean square error (RMSE) \[1\]. The training was stopped when the RMSE got lower than a predefined threshold which, in our experiments, was set to 0.1 radian.

For our experiments, we used an MLMVN with a single hidden layer. This choice was motivated by the next observations. The use of a single multi-valued neuron for our problem did not allow us to achieve a proper separability between the classes, since the training did not decrease below a predefined training error value due to the more complex, nonlinear nature of the separation problem. This calls for a MLMVN. Since we did not observe any significant improvement in performance when increasing the number of hidden layers, but only an increase in training time, we stick to the minimal number of hidden layers, i.e., one.

### 8.5 SYSTEM VALIDATION

Figure 8.10 shows the result of a five-fold cross-validation performed on the methods of Refs. 27 and 24 and the proposed one using a MLMVN classifier, for different EEG interval lengths \(T\) used for phase estimation. It is clear that the results obtained with the MLMVN significantly outperform those of the other considered methods, according to a repeated-measures ANOVA \((p < 0.01)\), at least for the tested subjects. By applying the single-channel methods from Refs. 24 and 27 to the optimal channel (obtained via a wrapper-like exhaustive search through all channels \(s_d\) on the training data), we also observe the superiority of the proposed multichannel classifier (see \(p\)-values in Fig. 8.10).

We have also verified the number of neurons in the hidden layer \(N_h\) (considering range from 2 to 20) of the MLMVN and the number of best features required for obtaining a satisfactory decoding accuracy. The best features were selected according to the two proposed in Section 8.3.3 methods namely, a feature selection based on the circular standard deviation and on the Watson–Williams test. As it can be seen from Fig. 8.11, both feature selection methods perform quite equally in terms of achieved accuracy.

We also see that selecting more features actually decreases the classification performance. For the comparison described above we considered the four best separating features according to our heuristic method based on the standard deviation. As to the number of neurons in the hidden layer \(N_h\), we observed for the optimal number of selected channels, in the case of a good separability between classes, only a slight increase in accuracy for \(N_h\) from two to six (see Fig. 8.11). For all other cases, we observe that the accuracy increases for \(N_h\) increasing until about 10 and further decreases (see Fig. 8.11). Based on this, we decided to use \(N_h = 10\) for the comparison reported above.

### 8.6 DISCUSSION

As it can be seen from Fig. 8.6, the channel combination and the referencing method influence the separability of the classes. Thus, a proper selection of the channels
Figure 8.10  Average (among all subjects) discrimination accuracy as a function of the EEG segment length for Lee et al.’s method [27] using channel Oz referenced to the mastoid (blue), Jia et al.’s method [24] using the bipolar POz–Oz channel (light-blue), Lee et al.’s method [27] for the optimal channel (green), Jia et al.’s method [24] for the optimal channel (orange), and the proposed multichannel method based on MLMVN and feature selection (brown). The numbers above the horizontal braces (at the top of the chart) are the repeated-measures ANOVA $p$-values for the differences between the results of the proposed method and the optimal channel version methods.

Figure 8.11  Dependency of the averaged accuracy in a five-fold cross-validation test of the MLMVN classifier on the number of neurons $N_h$ in the hidden layer and the number $d$ of best features used. The best features were estimated according to a feature selection based on standard deviation (left) and based on a Watson–Williams test (right). Results are presented for subject 1 while using five seconds of EEG recording for classifying six phase-shifted targets flickering at 10 Hz on the 60-Hz LCD screen.
should always be verified, which was also indicated in Ref. 24. But as they only considered one channel and did not provide a discriminative way to select it, our results for the multichannel combination and automatic feature selection can be viewed as an improvement (and also a generalization). Additionally to this, it is clear that only some channels combinations provide a good separability. This calls for a search for an appropriate spatial filtering approach for weighting the channels in such a way that the best separability is obtained. As a step in this direction, in Ref. 29 a canonical correlation analysis is used to incorporate all recorded EEG channels into one weighted sum.

In Ref. 24 the authors considered for their decoder one optimal channel, while we used a classifier that incorporates several best channels. Here the question arises: do we actually gain anything by considering several optimal channels? As one can see from Fig. 8.12, the classification accuracy rises up to its maximum at about 4–5 best features. This tells us that the combination of several best channels outperforms the case of only one optimal channel. This confirms the use of multichannel classifiers (as, for example, the MLMVN used in this study) and underlines their benefit with respect to the single channel decoders as the one of Refs. 24 and 27.

One way to assess any new BCI system is by estimating the information transfer rate (ITR) [55] which favors more accurate, faster systems with more encoded targets. In our design we did not aim to increase the ITR, but rather to prove the introduced concept of multichannel detection based on MLMVN. In order to increase
the ITR, we can incorporate more phase-shifted targets, choose another frequency with more space for encoding such targets. Or we can even consider phase shift and frequency combinations, as it was shown in Ref. 24.

Appendix: Decoding Methods

A.1 METHOD OF JIA AND CO-WORKERS

In Ref. 24, the following method was proposed. Based on the training data, the reference phases $\phi_{m_c,f}^{\text{ref}}$ are estimated by averaging, over the whole training set, the Fourier coefficients (obtained by using FFT) at the stimulation frequency $f$ of each target class $m_c = 1,\ldots,N$ (see Eq. (8.2)). Here, only one optimal electrode (according to Ref. 24) or Oz–POz bipolar combination is considered. When decoding new data (thus, the test set), the Fourier coefficients at stimulation frequency $f$ are estimated from the same, previously selected optimal electrode. This coefficient is projected onto all reference $\phi_{m_c,f}^{\text{ref}}$ phases and the resulting target class $\tilde{m}$ is selected as the one with the maximal projected value $\rho_{\tilde{m}}$. If we consider not only the fundamental frequency $f$, but also its harmonics, then we compute the lengths of the coefficient projections not only for the fundamental frequency, but also for its harmonics, on the respective reference phases, and sum up the resulting lengths for each target class separately, and then take the $\arg \max$ of the resulting sums to infer the index of the target class.

A.2 METHOD OF LEE AND CO-WORKERS

In Ref. 27, the following method was proposed. The EEG signals recorded from Oz channel referenced to the mastoid were band pass filtered in the range $[f - 2, f + 2]$ Hz, where $f$ is the stimulation frequency. Based on the training data (1 min of recordings on a subject observing a flickering stimulus with “zero” phase lag), an SSVEP$^{\text{ref}}$ is generated by averaging all epochs (EEG recordings corresponding to one period of stimulation, from one channel). The reference value $t_{\text{ref}}$ is defined from the obtained SSVEP$^{\text{ref}}$ as the latency of the maximum amplitude peak. In the decoding stage, the phase lag between SSVEP$^{\text{ref}}$ and SSVEP$^{\text{gaze}}$ is evaluated to infer the target class. This was done by cutting the SSVEP$^{\text{gaze}}$ signal into one-period-long segments, and take the average, and determine in this average the latency of the maximum amplitude peak, called $t_{\text{peak}}$. Next, the difference $\Delta t = t_{\text{ref}} - t_{\text{peak}}$ is transformed into a phase difference $\theta = 2\pi \Delta t f$. This phase difference $\theta$ is then wrapped to the interval $[0, 2\pi)$ by adding or subtracting (if necessary) $2\pi$. The achieved phase distance $\theta$ is compared to the expected phase delays $\theta_{m_c} = 2\pi (m_c - 1)/N$ ($m_c = 1,\ldots,N$) through an estimation of the angular distance as $D_{m_c} = |\theta_{m_c} - \theta|$. The resulting target class is then derived as $\arg \min_{m_c} D_{m_c}$. 
REFERENCES


REFERENCES


