Neural decoding based on probabilistic neural network[*]

Yi YU[1,2,3], Shao-min ZHANG[1,2,3], Huai-jian ZHANG[1,2,3], Xiao-chun LIU[1,2,4],

Qiao-sheng ZHANG[1,2,3], Xiao-xiang ZHENG[1,2,3], Jian-hua DAI[†‡1,2,4]

(*[1]Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, China*)

(*[2]College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou 310027, China*)

(*[3]Key Laboratory of Biomedical Engineering of Ministry of Education, Zhejiang University, Hangzhou 310027, China*)

(*[4]College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: qaas@zju.edu.cn

Received Sept. 9, 2009; Revision accepted Jan. 23, 2010; Crosschecked Mar. 15, 2010

Abstract: Brain-machine interface (BMI) has been developed due to its possibility to cure severe body paralysis. This technology has been used to realize the direct control of prosthetic devices, such as robot arms, computer cursors, and paralyzed muscles. A variety of neural decoding algorithms have been designed to explore relationships between neural activities and movements of the limbs. In this paper, two novel neural decoding methods based on probabilistic neural network (PNN) in rats were introduced, the PNN decoder and the modified PNN (MPNN) decoder. In the experiment, rats were trained to obtain water by pressing a lever over a pressure threshold. Microelectrode array was implanted in the motor cortex to record neural activity, and pressure was recorded by a pressure sensor synchronously. After training, the pressure values were estimated from the neural signals by PNN and MPNN decoders. Their performances were evaluated by a correlation coefficient (CC) and a mean square error (MSE). The results show that the MPNN decoder, with a CC of 0.8657 and an MSE of 0.2563, outperformed the traditionally-used Wiener filter (WF) and Kalman filter (KF) decoders. It was also observed that the discretization level did not affect the MPNN performance, indicating that the MPNN decoder can handle different tasks in BMI system, including the detection of movement states and estimation of continuous kinematic parameters.

Key words: Brain-machine interfaces (BMI), Neural decoding, Probabilistic neural network (PNN), Microelectrode array
doi:10.1631/jzus.B0900284   **Document code:** A   **CLC number:** TP2; R741

1 Introduction

The technology of invasive brain-machine interfaces (BMI) has been developed in decades, due to its possibility to cure neurological disabilities (Lebedev and Nicolelis, 2006; Feng *et al.*, 2007; Ye *et al.*, 2008). The main task of BMI system is decoding the cortical neural activity into commands of direct brain-controlled prosthetic devices. According to a BMI theory, it is proved that the neural ensemble activity of the motor cortex is distributed but not centralized,

and at present the development of multielectrode array techniques for population-level neural recording has enabled the research of distributed coding in the motor cortex (Nicolelis, 2003).

Decoding movement information of human and animal forelimbs from neural spiking activity has been the most focused task of BMI domain. As early as 1970s, Schmidt *et al.* (1978) had proven that monkeys could learn to control the spike firing rate of a single neuron freely in the primary motor cortex in a close-loop system. In the 1980s, Georgopoulos *et al.* (1988; 1989) found the relation between the movement direction of upper limb and single neuronal firing pattern in the primary motor cortex in rhesus monkeys. These studies opened the door of BMI research. In 2000, an open-loop BMI system used for

direct neural control of a robot arm in the aotus was developed by Nicolelis' group (Wessberg *et al*., 2000). Subsequently, his group realized the prediction of velocity and force of rhesus monkey's upper limb by neural ensemble activities from its motor cortex, and the prediction signal was used to control a robot arm as well (Lebedev *et al*., 2005). In 2006, the BMI technology was firstly used in humans by Hochberg *et al*. (2006). In the system, a 96-microelectrode array was implanted in the primary motor cortex of a paralytic patient to record neural spiking signal, and decoding algorithms were developed to control a curser for direct neural control. Velliste *et al*. (2008) decoded neural signal in rhesus monkey's motor cortex to control a prosthetic device with four degrees of freedom. With the development of motor-related BMI technology, it will be possibly used to recover the motor function of paralytic patients and greatly improve their life quality.

Recently, a great number of neural decoding algorithms have been developed, including the population vector algorithm, Wiener filter (WF), and Bayesian filter (BF). The population vector (PV) algorithm has proven to be a successful neural decoding method, in which each neuron's activity is characterized by its preferred direction and firing rate. PV is an optimal solution when the tuning functions are linear and the preferred directions are uniformly distributed (Georgopoulos *et al*., 1988; 1989; Zhang *et al*., 1998; Taylor *et al*., 2002). WF is used in BMI systems when the preferred directions are not uniformly distributed, which uses the least square rule (Wessberg *et al*., 2000; Serruya *et al*., 2002; Carmena *et al*., 2003; Hatsopoulos *et al*., 2004; Hochberg *et al*., 2006; Gage *et al*., 2005). It is noted that WF is a more general method, and PV is a special case of WF. Both of them assume that the firing rates are linearly related to the underlying movement parameters. BF usually assumes that the kinematic decoding procedure accords with the first order Markov model, whose states are the kinematic parameters, and the measurements are the neural spiking activity. The Kalman filter (KF), one of these approaches, uses the linear assumption and least square rule, providing an efficient recursive algorithm to Bayesian rule when the likelihood and prior models are linear and Gaussian (Wu *et al*., 2002; 2003; 2004; 2006). As a more general method than KF, particle filter (PF) combines Monte Carlo sampling methods with Bayesian inference for kinematic state estimation, which has no constrain on linear and Gaussian assumptions (Brockwell *et al*., 2004; Shoham *et al*., 2005). Although the PF is more powerful than other methods, the heavy computational cost makes it difficult to implement. Spike training analysis was also described as a point process based on linear and nonlinear models (Brown *et al*., 1998; Smith and Brown, 2003; Eden *et al*., 2004; Wang *et al*., 2009). In this paper, probabilistic neural network (PNN) was firstly used to decode neural spiking activity. PNN has no linear assumption constrain on the relation between neural firing and kinematic parameter, and it costs much less computational consumption than PF (Specht, 1990). In addition, as a classification method, PNN decoder could change the number of decision attributes. A smaller number of decision attributes are enough when we just want to find the happening of events, and a larger number of decision attributes should be used when precise decoding is required. By modulating the number of decision attributes, this method is more computational managing and controllable than PF. And as a nonlinear method, it has better performance than KF and WF in our experiment.

In the experiment, a 2×8-microelectrode array was implanted in the motor cortex of rats to record neural spiking activity, and water reward was given when rat pressed a lever by its forelimb. Spikes were detected and sorted, and the pressure signal of the lever was also recorded by a set of pressure sensors. We then used PNN as a decoding algorithm to decode rat's neural firing into pressure values, which has two steps. In the training process, the structure of PNN was trained by training data, which contain both binned neural spiking data and synchronously recorded pressure signal. Then in the testing process, the pressure value was estimated, given PNN structure and current neural activity.

In the present study, two types of PNN decoders were defined. One uses current neural activity as the input of the PNN, which is similar to the WF. The other has the input vector containing both current neural activity and previous estimated pressure value. The latter is a modified PNN (MPNN), with an auto-regressive algorithm in the spirit of a hidden Markov model, but not equivalent to it, since there is no distribution kept as state model. Furthermore, two

commonly-used decoding methods, WF and KF, were also used here, and their performances were compared with PNN decoders. Correlation coefficient (CC) and mean square error (MSE) were used to evaluate the performance of the decoders. The results show that the MPNN decoder performed the best, which had higher CC and lower MSE than others.

2 Materials and methods

2.1 Probabilistic neural network

PNN is a typical nonlinear classifier, which uses minimum Bayesian risk criterion (Specht, 1990). In this paper, we discretized pressure value into several levels as the decision attributes of PNN. The number of the discretization levels could be changed for different demands. Only 2–5 levels are needed for judging whether the rat has pressed the lever. For more precise decoding, the pressure value should be discretized into much more levels, i.e., 10–100 levels in the current study. Compared with traditional Back-Propogation neural network, PNN takes less time in training and does not need to train again when more training data are added or reduced. Moreover, it can always obtain the optimal solution in Bayesian criterion when the training data are sufficient, regardless of the complexity of the classification problem.

In our case, multidimensional data $X \in R^q$ are to be classified into N ($N \geq 2$) classes, $C_1, C_2, \ldots, C_i, \ldots, C_N$ corresponding to levels of pressure values. The Bayes' decision C_i is made by the rule: vector X belongs to C_i when $H_iL_if_i(X) > H_jL_jf_j(X)$ for all $j=1, 2, \ldots, N$ and $j \neq i$, where H_i denotes the prior probability that X belongs to C_i, L_i is the loss factor when a mistake occurs for each C_i, and $f_i(X)$, as the spherical Gaussian radial basis function, is a Parzen probability density function estimator as

$$f_i(X) = \frac{1}{(2\pi)^{q/2}\sigma^q} \frac{1}{M_i} \sum_{k=1}^{M_i} \exp\left(\frac{-(X-X_{ik})^{\mathrm{T}}(X-X_{ik})}{2\sigma^2}\right),$$

(1)

where $(\cdot)^{\mathrm{T}}$ indicates transpose of a matrix, q is the dimension of X, M_i is the number of training vector in class C_i, X_{ik} is the kth training vector in class C_i, and σ is the smooth factor or bandwidth. When both training and test vectors are normalized, we have

$$(X-X_{ik})^{\mathrm{T}}(X-X_{ik}) = -2(X^{\mathrm{T}}X_{ik}-1). \quad (2)$$

We can get a simplified classification rule from Eq. (2), when considering $H_i=M_i/M_{\mathrm{total}}$ from Bayes' decision rule ($H_iL_if_i(X) > H_jL_jf_j(X)$ for $j \neq i$), as Eq. (3):

$$L_i \sum_{k=1}^{M_i} \exp\left[\frac{X^{\mathrm{T}}X_{ik}-1}{\sigma^2}\right] > L_j \sum_{k=1}^{M_j} \exp\left[\frac{X^{\mathrm{T}}X_{jk}-1}{\sigma^2}\right]. \quad (3)$$

The PNN network is simply a parallel 4-layer structure: input, pattern, summation, and decision layers (Fig. 1). The input layer receives and normalizes input vector; each unit in pattern layer represents a training vector with response function $\exp[(X^{\mathrm{T}}X_{ik}-1)/\sigma^2]$. Summation layer computes the summation of each pattern and multiplies the loss factor. Decision layer selects the largest one in summation layer as the classification result. Along with the increasing prior knowledge, the PNN can be expanded horizontally and the classification capacity increased.
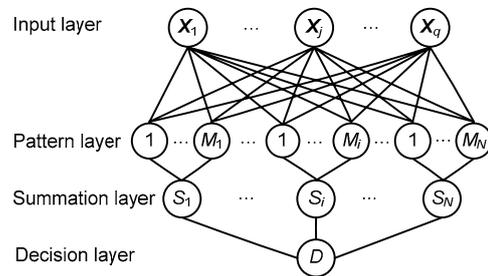
Fig. 1 Probabilistic neural network structure
X_j: input vector; M_i: number of training vector in class C_i; S_i: summation of probability density in class C_i; D: decision of the model

2.2 Experiment environment and data processing

Approval for all experimental procedures was obtained from the Animal Care Committee at Zhejiang University, China, which follows the guide for the care and use of laboratory animals (Ministry of Health of the People's Republic of China, 2000). Methods for surgery and multineuron recording have been published (Paxinos *et al.*, 1980). In this paper, Sprague-Dawley (SD) rats (approximately 280 g, provided by Zhejiang Academy of Medical Sciences, China) were trained to perform conditional operant task. In the task, when the rat press the lever by its forelimb and the pressure value was over a threshold,

the animal was rewarded by a drop of water. All rats were trained to achieve a success rate of >75% before surgery. After that, a 2×8-channel chronical microwire electrodes array (35 µm in diameter, 300 µm for a space between rows, and 200 µm for a space within rows; California Fine Wire, CA, USA) was implanted into rat's forepaw region of the forelimb primary motor cortex. The electrode tips were positioned in the layer *V*, and the approximate depth in layer *V* was in the range of 1.1–1.8 mm beneath the pia in the cerebral cortex. Layer *V* contained cell bodies that projected directly to the spinal cord to activate peripheral motor neurons. To decrease the common mode noise, one of the 16 electrodes was used as a reference electrode, and 16-channel neural analog signals were recorded in a 30-kHz sample rate, filtered, and amplified using Cerebus 128™ (Cyberkinetics Inc., USA). Then pressure signal was synchronously recorded by pressure sensor at 500 Hz during the experiment.

Neural spiking activities were extracted from 15-channel neural signals by threshold method (except the reference electrode in the array). After that, spikes from each channel were classified into 1–3 types by typically-used principal components analysis and *k*-means clustering, and each type represented one neuron physiologically. A total of 22–58 neurons were found in all 15 channels per rat. To compute neural spiking frequency for each neuron, firing number in a time bin (Δt=100 ms) was counted. Meanwhile, the pressure signal was recorded and computed into bin size (Δt=100 ms) by averaging pressure value in the bin period. Discretization was implemented to transfer the pressure value into discrete decision attributes of PNN. The discretization procedure of pressure value was realized as

$$p(n)' = \text{floor}\left(\frac{p(n) - \min\left(p(n)\right)}{\max\left(p(n)\right) - \min\left(p(n)\right)} \times N \right), \quad (4)$$

where $p(n)$ denotes pressure value in the *n*th bin.

After that, PNN approach was used to estimate pressure value from neural ensemble spike firing. The issue of neural decoding was transferred into a classification problem. Two kinds of PNN decoders were defined here with different input vectors as Eqs. (5) and (6):

$$\boldsymbol{X}(n) = \begin{bmatrix} f_1(n) & f_2(n) & \dots & f_{q-1}(n) \end{bmatrix}, \quad (5)$$

$$\boldsymbol{X}(n) = \begin{bmatrix} f_1(n) & f_2(n) & \dots & f_{q-1}(n) & \tilde{p}(n-1) \end{bmatrix}, \quad (6)$$

where $f_i(n)$ denotes firing rate of the *j*th neuron in the *n*th bin, and $\tilde{p}(n-1)$ denotes previous estimated pressure. The decoder with input vector Eq. (5) was named PNN decoder, and the decoder with input vector Eq. (6) was named MPNN decoder. The decoding process of the PNN decoder was as follows: firstly, the structure of PNN was trained by a length of training data, whose decision attributes was $p(n)$; and in the testing step, we estimated $\tilde{p}(n)$ by classifying the class of $\boldsymbol{X}(n)$ with the PNN structure. The process of MPNN decoder was similar. In the training process, the $\tilde{p}(n-1)$ in Eq. (6) was replaced by the actually-recorded pressure value $p(n-1)$.

WF and KF were also implemented as comparisons to PNN methods. The process of WF decoder was the same as PNN decoder, which used least square rule to train the linear transform matrix, and estimated pressure value with input vector as Eq. (5). The KF decoder separated the neural decoding issue into two steps: relating the neural activity to movements and evolving movements over bins (Wu *et al.*, 2003), generative model, and system model. The detail of these methods was not shown.

The estimated pressure value by decoders and the sensor-detected pressure value were compared by CC and MSE. The CC was computed as Eq. (7), and MSE as Eq. (8):

$$\rho_{P,\tilde{P}} = \frac{E(P\tilde{P}) - E(P)E(\tilde{P})}{\sqrt{E(P^2) - E^2(P)}\sqrt{E(\tilde{P}^2) - E^2(\tilde{P})}}, \quad (7)$$

$$\text{MSE}(P,\tilde{P}) = E\left((\tilde{P} - P)^2\right), \quad (8)$$

where P denotes real pressure train and $\tilde{P}$ denotes estimated pressure train.

3 Results

During the task training, rats were kept on water restriction and water was maintained by rewarding from pressing the lever, 0.05 ml each time. Each rat pressed the lever for 200–300 times a day, about

10–15 ml water rewarded. Three rats (S9-03, S9-04, S9-12) were implanted with a 2×8-microelectrode array. After restoring from surgery for one week, all the three rats could obtain water by pressing lever repeatedly. The pressing events were repeated in a stable period, 3–5 s each time (1 s for pressing the lever and 2–4 s for drinking water). Signal processing procedures preparing for neural decoding of S9-03 are shown in Fig. 2. After a second bandwidth Butterworth filter (750–7000 Hz), the neural signal was recorded in 30 kHz (Fig. 2a), which contains two channels (5, 13). By spike detection (threshold crossing) and sorting algorithm (principal components analysis and *k*-means clustering), spikes were extracted and classified (Fig. 2b) from two channels (Fig. 2a). Fig. 2c shows the spiking raster of four neurons in the same timeline as Fig. 2a. The pressure value was synchronously recorded with neural signal by press sensor in the lever, with a sample rate of 500 Hz (Fig. 2d). Spike firing rate in bin period was computed later, as well as averaging and discretizing the pressure value in bin. Fig. 2e shows the result of S9-03, where the color of the grids indicates the firing rate of the neurons. In each rat, scores of neurons were extracted, 31 neurons for S9-03, 22 for S9-04, and 58 for S9-12. Some neurons that had only a small amount of spike firing were not included.

In the current study, two types of neural decoders based on PNN were used, the PNN decoder and the MPNN decoder, depending on two kinds of input vectors as Eqs. (5) and (6). Fig. 2 shows that the bin size was 100 ms and the discretization level was 100. The examples of decoders' outputs are shown in Figs. 3a and 3b, where the dot-line indicates the pressure signal recorded by pressure sensor and the solid-line the decoding outputs. The result of MPNN decoder generated smoother waveform than the PNN decoder (Fig. 3a). There were cleavages in yellow line peaks in PNN decoder output (Fig. 3b), which looked like high frequency noise. The CC (0.8657) of modified PNN decoder was larger than that (0.7061) of PNN decoder, and the MSE (0.2563) of MPNN decoder was smaller than that (0.5866) of the PNN decoder. However, both of them indicated that the addition of $\tilde{p}(n-1)$ as part of input vector would improve the performance of PNN decoding.

In addition, we compared these two decoders with other two commonly-used neural decoding al-gorithms, WF and KF. The bin size of all the four was 100 ms. The output of WF was similar to other movements decoding BMI systems. It fitted the real pressure value accurately but contained much high frequency noise, with CC=0.7778 and MSE=0.2487 (Fig. 3d). KF performed more smoothly than WF and estimated the pressing events (as the peaks in Fig. 3c) better than WF, with CC=0.7973 and MSE=0.4047. But the output of KF between the peaks was more disordered than that of WF. It is because that the pressing events in our experiment seemingly hap-pened accidently and the pressure value changed abruptly. It was somehow different from some other BMI systems using KF, which usually decode kine-matic parameters that change gradually. These linear methods did not perform as well as MPNN decoder in terms of their CC and MSE values. However, both of them had larger CC and smaller MSE than PNN de-coder, since the cleavages in pressure peak in PNN output played an import part to the computation of CC and MSE. The results of other two rats, S9-04 and S9-12, are also shown in Table 1.

Table 1 CC and MSE of neural decoders[*]

| Decoder	CC			MSE		
	S9-03	S9-04	S9-12	S9-03	S9-04	S9-12
WF	0.7778	0.8817	0.7066	0.2487	0.1200	0.4448
KF	0.7973	0.8018	0.7583	0.4047	0.3956	0.4825
PNN	0.7061	0.5782	0.6151	0.5866	0.8420	0.7682
MPNN	0.8657	0.9322	0.8840	0.2563	0.1749	0.2248

[*] Bin size=100 ms

In the data processing of S9-03, a length of 200 s data was used to train the decoders; WF spent 0.003 s, KF 0.0075 s, PNN 0.04 s, and MPNN 0.04 s on training. In the testing process, for each bin (100 ms), WF spent less than $5×10^{-7}$ s, KF less than $2.5×10^{-4}$ s, PNN less than $1.2×10^{-3}$ s, and MPNN less than $7×10^{-3}$ s. Although MPNN seemed to take the longest time among these algorithms, the training and testing consumptions were acceptable and easily met the real-time implementation. Algorithms were tested in Matlab 7.8 (Mathworks Inc., USA) in the computer with Intel Core 2 Duo CPU 2.33 GHz.

To find out the effect of discretization level on MPNN decoder, we tested the output of MPNN de-coder in a range of discretization level. In the ex-periment, the discretization level ranged from 10 to

(a)
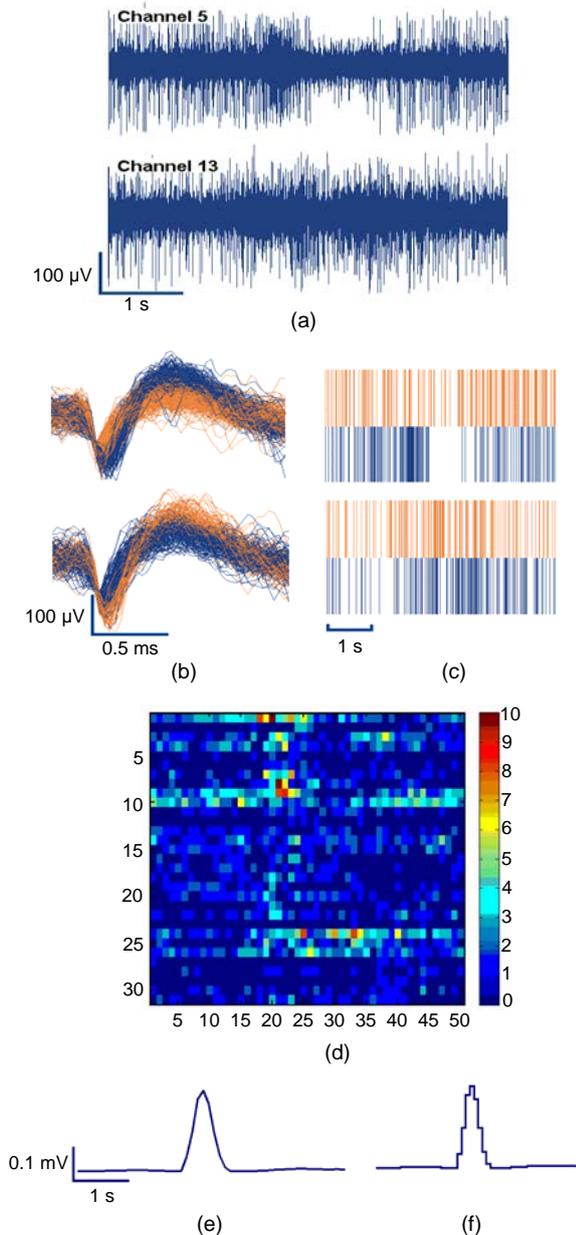
Fig. 2 Simultaneously recorded neural signal and pressure signal

(a) Two-channel raw neuronal signals in rat S9-03's motor cortex during 5 s; (b) Sorted spikes in two channels respectively; (c) Raster of neural ensemble firing. Each vertical short line in the figure denoted a spike firing; (d) Binned neuronal spike firing. Each row represented a neuron, and each column represented a bin period, where bin size was 100 ms. The color denoted the firing frequency; (e) Pressure value of the lever from pressure sensor. An abrupt change of the pressure value indicates a pressing event of rat. When the rat did not press the lever, the pressure value rested in the base line; (f) Discretized pressure when the bin size was 100 ms and the discretization level was 100
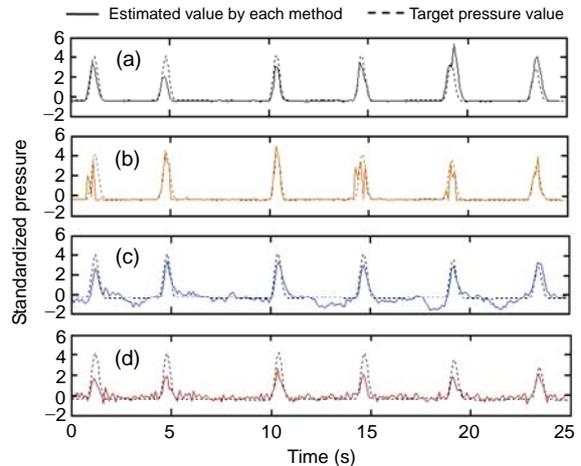
Fig. 3 Examples of pressure decoding results (rat S9-03)
(a) MPNN; (b) PNN; (c) KF; (d) WF
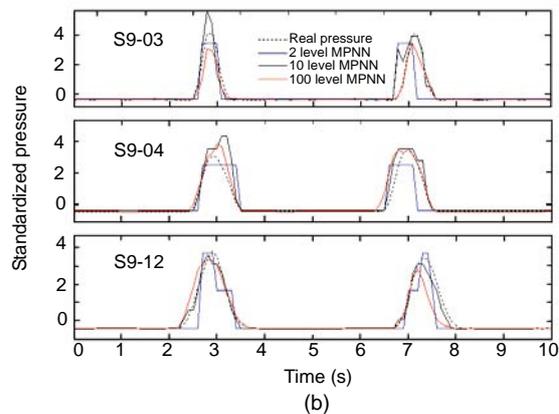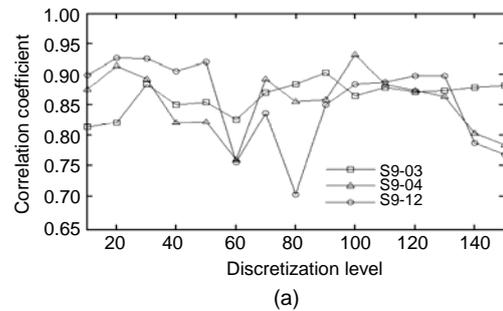
Fig. 4 Effect of discretization level on MPNN decoder
(a) Performances of MPNN decoder in a range of discretization level of three rats; (b) Waveforms of MPNN outputs in a range of discretization levels of three rats

150, and the results of all the three rats were computed. Although the CC values might oscillate between different discretization levels, there was no fixed relation between the discretization level and the decoding performance (Fig. 4a). The results of MPNN decoder in four discretization levels are listed in Fig. 4b. At level 3, there were only three values for

decoder output, and the result seemed like square wave (green line, Fig. 4b). When the discretization level increased to 10, the results gradually became smoother. At level 100, the results of MPNN look like a continuous line.

It can be concluded that smaller discretization levels are needed for only estimating the lever-pressing event, which decreases computational consumption. A larger number of discretization levels are demanded when more precise and smoother decoding output is required. The effect of bin size on MPNN was also tested. For rat S9-03, when the bin size ranged from 50 to 150 ms (50, 75, 100, 125, and 150 ms), the CC values were 0.6228, 0.7617, 0.8657, 0.7512, and 0.8114, respectively, showing an increasing trend.

4 Discussion

In the present study, we introduced PNN and MPNN decoders to decode neural ensemble activities around the pressing events. Results show that MPNN decoder had better performance than traditional algorithms, WF and KF decoders. In addition, MPNN decoder does not bring much more computational complexity.

MPNN decoder, whose input vector includes previous estimated pressure value $\tilde{p}(n-1)$, has smoother and higher CC values than PNN. Although the structure of PNN decoder is also nonlinear in Bayesian criterion, it provides no estimation of uncertainty and is difficult to analyze complex temporal movement. We, therefore, found high frequency noise in PNN decoder (Fig. 3b). Moreover, we also found that MPNN decoder was better than KF, for the output of KF between pressing events was disordered. The KF is one realization of Bayesian filter under linear and Gaussian assumption, and this model has evolved to produce a smoother result (Wu *et al.*, 2003). However, the pressing events in our experiment appeared to happen accidently and the pressure value seemed changed abruptly, which cannot be described by any linear system model. So, under the least square rule, the training of evolving matrix was compromised for the abruptly changed pressure value, resulting in disorder between pressing events.

Although various decoding models had different levels of performance, the overall results were not much affected. Perhaps it is because that we evaluated the performance of regression estimate by CC but not the correct rate of classification. The results also indicate that the bin size affected the performance of MPNN. It can be that the neural ensemble firing is likely to be a non-stationary random process and a larger bin size usually diminishes the randomness of neural firing. When we enlarged the size of the bin, the performance was improved. On the other hand, it is needed to note that the duration of pressing event was only about 1 s. When the bin size was oversized, the waveform of PNN output would be distorted and the CC would be decreased.

In this study, we solved a problem of regression using classification method by discretizing the pressure signal into levels. Firstly brought forward by Specht (1990), PNN method has been successfully applied to machine learning, artificial intelligence, automatic control, and many other fields. Compared with the multi-layer feedforward network, PNN is a simpler mathematical principle and is easy to implement. The method proposed in this paper for neural decoding could be used as a general method in various BMI systems, such as regression tasks, decoding continuous movement states or trajectories by MPNN decoder, pattern recognition tasks, and judging figure or limb movements by PNN decoder. Although, general regression neural network has already been brought forward to be used for regression tasks widely, the decoder based on PNN was used in the current study in detecting the pressing events and finding precise pressure value to face different requirements.

5 Conclusion

We presented novel neural decoding methods based on probabilistic neural network, the PNN and the MPNN decoders. By comparing with other commonly-used decoding methods, WF and KF, we found that MPNN performed best among others, with higher CC and lower MSE. By changing the discretization level, MPNN decoder could handle different problems in BMI system, such as movement state detection and decoding continuous kinematic parameter estimation.

In summary, the MPNN decoder based on PNN is a viable and effective method for neural decoding in BMI systems.

6 Acknowledgment

References

Brockwell, A.E., Rojas, A.L., Kass, R.E., 2004. Recursive bayesian decoding of motor cortical signals by particle filtering. *J. Neurophysiol.*, **91**(4):1899-1907. [doi:10.1152/jn.00438.2003]

Brown, E.N., Frank, L.M, Tang, D., Quirk, M.C., Wilson, M.A., 1998. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J. Neurosci.*, **18**(18):7411-7425.

Carmena, J.M., Lebedev, M.A., Crist, R.E., O'Doherty, J.E., Santucci, D.M., 2003. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS. Biol.*, **1**(2):E42. [doi:10.1371/journal.pbio.0000042]

Eden, U.T., Frank, L.M., Barbieri, R., Solo, V., Brown, E.N., 2004. Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Comput.*, **16**(5):971-998. [doi:10.1162/089976604773135069]

Feng, Z.Y., Chen, W.D., Ye, X.S., 2007. A remote control training system for rat navigation in complicated environment. *J. Zhejiang Univ.-Sci. A*, **8**(2):323-330. [doi:10.1631/jzus.2007.A0323]

Gage, G.J., Ludwig, K.A., Otto, K.J., Ionides, E.L., Kipke, D.R., 2005. Naive coadaptive cortical control. *J. Neural Eng.*, **2**(2):52-63. [doi:10.1088/1741-2560/2/2/006]

Georgopoulos, A.P., Kettner, R.E., Schwartz, A.B., 1988. Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *J. Neurosci.*, **8**(8):2928-2937.

Georgopoulos, A.P., Lurito, J.T., Petrides, M., 1989. Mental rotation of the neuronal population vector. *Science*, **243**(4888):234-236. [doi:10.1126/science.2911737]

Hatsopoulos, N., Joshi, J., O'Leary, J.G., 2004. Decoding continuous and discrete motor behaviors using motor and premotor cortical ensembles. *J. Neurophysiol.*, **92**(2):1165-1174. [doi:10.1152/jn.01245.2003]

Hochberg, L.R., Serruya, M.D., Friehs, G.M., Mukand, J.A., Saleh, M., 2006. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, **442**(7099):164-171. [doi:10.1038/nature04970]

Lebedev, M.A., Nicolelis, M.A., 2006. Brain-machine interfaces: past, present and future. *Trends Neurosci.*, **29**(9):536-546. [doi:10.1016/j.tins.2006.07.004]

Lebedev, M.A., Carmena, J.M., O'Doherty, J.E., 2005. Cortical ensemble adaptation to represent velocity of an artificial actuator controlled by a brain-machine interface. *J. Neurosci.*, **25**(19):4681-4693. [doi:10.1523/JNEUROSCI.4088-04.2005]

Ministry of Health of the People's Republic of China, 2000. Guide for Animal Experiment Technology. Ministry of Health of the People's Republic of China, p.9-20.

Nicolelis, M.A., 2003. Brain-machine interfaces to restore motor function and probe neural circuits. *Nature Rev. Neurosci.*, **4**(5):417-422. [doi:10.1038/nrn1105]

Paxinos, G., Watson, C.R., Emson, P.C., 1980. AChE-stained horizontal sections of the rat brain in stereotaxic coordinates. *J. Neurosci. Methods*, **3**(2):129-149. [doi:10.1016/0165-0270(80)90021-7]

Schmidt, E.M., McIntosh, J.S., Durelli, L., 1978. Fine control of operantly conditioned firing patterns of cortical neurons. *Exp. Neurol.*, **61**(2):349-369. [doi:10.1016/0014-4886(78)90252-2]

Serruya, M.D., Hatsopoulos, N.G., Paninski, L., Fellows, M.R., Donoghue, J.P., 2002. Instant neural control of a movement signal. *Nature*, **416**(6877):141-142. [doi:10.1038/416141a]

Shoham, S., Paninski, L.M., Fellows, M.R., Hatsopoulos, N.G., Donoghue, J.P., 2005. Statistical encoding model for a primary motor cortical brain-machine interface. *IEEE Trans. Biomed. Eng.*, **52**(7):1312-1322. [doi:10.1109/TBME.2005.847542]

Smith, A.C., Brown, E.N., 2003. Estimating a state-space model from point process observations. *Neural Comput.*, **15**(5):965-991. [doi:10.1162/089976603765202622]

Specht, D.F., 1990. Probabilistic neural networks. *Neural Netw.*, **3**(1):109-118. [doi:10.1016/0893-6080(90)90049-Q]

Taylor, D.M., Tillery, S.I., Schwartz, A.B., 2002. Direct cortical control of 3D neuroprosthetic devices. *Science*, **296**(5574):1829-1832. [doi:10.1126/science.1070291]

Velliste, M., Perel, S., Spalding, M.C., 2008. Cortical control of a prosthetic arm for self-feeding. *Nature*, **453**(7198):1098-1101. [doi:10.1038/nature06996]

Wang, Y., Paiva, A.R.C., Príncipe, J.C., Sanchez, J.C., 2009. Sequential monte carlo point-process estimation of kinematics from neural spiking activity for brain-machine interfaces. *Neural Comput.*, **21**(10):2894-2930. [doi:10.1162/neco.2009.01-08-699]

Wessberg, J., Stambaugh, C.R., Kralik, J.D., Beck, P.D., Laubach, M., Chapin, J.K., Kim, J., Biggs, S.J., Srinivasan, M.A., Nicolelis, M.A.L., 2000. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, **408**(6810):361-365. [doi:10.1038/35042582]

Wu, W., Black, M.J., Gao, Y., 2002. Inferring Hand Motion

from Multi-Cell Recordings in Motor Cortex Using a Kalman Filter. Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices. Edinburgh, Scotland (UK), p.66-73.

Wu, W., Black, M.J., Gao, Y., Bienenstock, E., Serruya, M., 2003. Neural Deocoding of Cursor Motion Using a Kalman Filter. Advances in Neural Information Processing Systems 15. Cambridge, MIT Press, MA, p.1-8.

Wu, W., Black, M.J., Mumford, D., Gao, Y., Bienenstock, E., 2004. Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Trans. Biomed. Eng.*, **51**(6):933-942. [doi:10.1109/TBME.2004.826666]

Wu, W., Gao, Y., Bienenstock, E., Donoghue, J.P., Black, M.J., 2006. Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Comput.*, **18**(1): 80-118. [doi:10.1162/089976606774841585]

Ye, X.S., Wang, P., Liu, J., 2008. A portable telemetry system for brain stimulation and neuronal activity recording in freely behaving small animals. *J. Neurosci. Methods*, **174**(2):186-193. [doi:10.1016/j.jneumeth.2008.07.002]

Zhang, K., Ginzburg, I., McNaughton, B.L., 1998. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *J. Neurophysiol.*, **79**(2):1017-1044.