Information Sciences 585 (2022) 543–558

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Second-order information bottleneck based spiking neural networks for sEMG recognition ☆

Anguo Zhang [a,b], Yuzhen Niu [c], Yueming Gao [a,b,*], Junyi Wu [d], Zhipeng Gao [d]

[a] College of Physics and information Engineering, Fuzhou University, Fuzhou 350108, China
[b] Key Laboratory of Medical Instrumentation and Pharmaceutical Technology of Fujian Province, Fuzhou 350116, China
[c] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China
[d] AI Research Center, Xiamen Meiya Pico Information Co., Ltd, Xiamen 361000, China

ARTICLE INFO

Article history:
Received 11 September 2020
Received in revised form 14 November 2021
Accepted 18 November 2021
Available online 1 December 2021

Keywords:
Surface electromyography (sEMG) recognition
Spiking neural network
Second-order information bottleneck

ABSTRACT

The pattern recognition of surface electromyography (sEMG) signal is an important application in the realization of human-machine interface. However, due to the disturbance of human body, sensors and environment, sEMG signal usually contains lots of noise, which brings great challenges to the high-accuracy sEMG pattern recognition. In addition, embedded human wearable devices are becoming more and more popular nowadays. How to realize the sEMG recognition method with low power consumption and high noise immunity has also become a difficult and very meaningful research topic. In this paper, a spiking neural network (SNN) classification method based on second-order information bottleneck training is proposed. Firstly, the training loss function for classification neural networks is constructed based on the proposed second-order information bottleneck. The method is used to train the conventional continuous-valued neural network and convert it into an SNN model with equivalent structure and connection weights. Then, the converted SNN is used to classify the sEMG signal patterns. Through a series of theoretical analysis and experimental results, it is proved that this method has significant advantages in terms of generalization of network determination and computational efficiency. The experimental code can be accessed from https://github.com/anvien/2OIB-for-sEMG-Recognition.

1. Introduction

Surface electromyography (sEMG) is a type of one-dimensional time series electrical signal produced by the weak action potential which is generated by muscle fibers on the skin surface when skeletal muscle contracts. sEMG is detected and collected using electrodes. Its amplitude is usually between 15 and 100 μV, and the signal spectrum width is 10–500 Hz. Changes in sEMG can reflect neuromuscular activity. Therefore, the measurement of sEMG is important in research and application to the diagnosis of neuromuscular diseases and muscle ergonomic analysis. In rehabilitation medicine, sEMG

☆ This work is supported by the National Natural Science Foundation of China under Grant No. U1505251 and No. 61672158, Chinese Ministry of Science and Technology under Grant No. 2016YFE0122700, National Science Foundation of Fujian Province under Grant No. 2019J02006, Fujian Provincial Department of Science and Technology under Grant No. 2021I0005, and Major Science and Technology Project of State Development and Investment Group Co., Ltd under Grant No. SDIC2021-07.

* Corresponding author at: College of Physics and information Engineering, Fuzhou University, Fuzhou 350108, China.
E-mail address: gym@fzu.edu.cn (Y. Gao).

https://doi.org/10.1016/j.ins.2021.11.065
0020-0255/

is used for the assessment of muscle function and prosthetic limb control. sEMG also has considerable potential for assessing the degree of fatigue in sports training, as well as the non-injury prediction of muscle fiber types and the determination of anaerobic thresholds.

The classification of sEMG patterns is fundamental to understanding the characteristics of neuromuscular signals. However, due to the influence of factors like body surface, environment, and sensor noise, the sEMG signals collected sEMG signals often include complex interference signals. Thus, it is still challenging to classify sEMG signals with high precision and robustness.

A large number of sEMG-oriented recognition methods have been proposed, including wavelet transforms [15,8], fuzzy rule systems such as neuro-fuzzy classifier [23] and fuzzy Gaussian mixture model [20], support vector machine (SVM, [33]), and hidden Markov [11] and Bayesian [17] models. However, these approaches use carefully designed data features and model parameters, and the recognition accuracy is susceptible to interference from sensor noise and environmental factors. Recently, methods based on deep learning have received increasing attention [45,39,47,27,9]. Deep learning provides an end-to-end inference paradigm, which can further extract the essential characteristics of sEMG signals, while achieving higher classification accuracy. However, deep learning-based methods still have some challenges. Since sEMG signals are often polluted by noise [44], it is necessary to develop more generalized deep learning models to overcome sEMG signal noise and interference, and achieve higher robustness. Deep learning requires multiple floating-point operations to generate results, and the large power consumption involved makes it difficult to directly apply these methods to embedded or wearable devices.

Spiking neural networks (SNNs) are considered to be an effective solution to the problem of the high power consumption of conventional deep learning methods. SNN is referred to as a third-generation neural network algorithm [29], and is an event-driven, sparse computing architecture. SNNs can produce high accuracy, while achieving extremely low power consumption, and have been successfully applied on IBM's TrueNorth [30] and Intel's Loihi [12] chips. Therefore, signal processing methods based on SNNs have also attracted extensive attention. SNNs with different structures and hardware processors have been widely applied in sEMG signal recognition.

[34] analyzed the feasibility of an SNN architecture running on NeuCube [21,22], an integrated development environment, for EMG pattern recognition. They found that the NeuCube-based SNNs obtain higher classification accuracy than common machine learning techniques. In [32], the authors demonstrated that an SNN classifier could be considered as a viable option for sEMG recognition in future human-computer interface (HCI) systems for real-time prosthesis applications. Liu et al. used an SNN trained using the SpikeProp algorithm [6] to recognize the sEMG signal of a hand gesture. They claimed that the SNN could achieve a satisfactory classification result even on a small scale network, and was superior to other common machine learning methods [26].

In [28], the authors proposed the use of a spiking recurrent neural network, liquid state machine (LSM) [43], to classify spatio-temporal EMG signals. Signals from various hand gestures were collected using a Myo armband and encoded into a spike sequence. Donati et al. [14] proposed a mixed-signal neuromorphic computing framework for EMG classification. The SNN employed in their framework was a feed-forward network in which the synaptic weights of the input layer were fixed and randomly generated. Only the weights of the output layer needed to be trained. In [10], Cheng et al. proposed a spike firing time search algorithm for a "deep structure" SNN, which aimed to detect the time interval during which a new spike was fired by the neuron. Their experimental results demonstrated that the proposed method could significantly improve the training speed of the SNN. Moreover, the classification accuracy of the sEMG patterns was very close to those of other popularly used machine learning models. Rekabdar et al. proposed a real-time unsupervised SNN method for spatio-temporal human gesture recognition [35]. They built an SNN classifier based on Izhikevich [19] and Integrate-and-Fire-or-Burst [7] neuron models, and implemented them in CUDA. Their experiments showed that the SNN approaches obtained superior accuracy than other prominent methods, and were suitable for the early classification of different types of human actions in time-sensitive mobile applications such as robotics. In [32], Mukhopadhyay et al. studied the performance of a deep ANN/SNN model for the pattern classification of sEMG signals generated by hand movement. The authors used the ANN2SNN conversion method to obtain a well-trained SNN model, and demonstrated that compared with the ANN classifier, the SNN classifier had lower computational intensity because there are fewer spiking feedforward propagation paths, and activation function operations (such as ReLU) are not necessary for spiking neurons. Although considerable research has been conducted, there are still many problems in SNN-based sEMG pattern recognition, which should be solved as a matter of urgency. The existing learning algorithms are more suitable for a "shallow" neural network structure, such as an LSM-based approach [28], and the single-hidden-layer feed-forward network approach [14], which are difficult to use to adequately train a deep network. The SNN training method proposed in [10] was primarily aimed at improving the training speed; however, compared with other approaches, the method did not improve the classification accuracy of sEMG patterns.

Information bottleneck (IB), a data compression method that was first proposed in [41], can extract an efficient and compact representation of relevant information about the source of interest. IB avoids the need of a prior specification of a distortion measure by considering the mutual information between the quantizer output and the original data source. Owing to this advantage, the IB method has been widely used in optimizing noise signals representing source data [18,42,1].

Taking these issues into consideration, in this paper, we proposed an SNN classifier based on the second-order information bottleneck (2O-IB) for sEMG recognition. We first developed a new IB method called 2O-IB, and used it to train a traditional deep learning model. Then, we converted the trained model to the corresponding SNN counterpart, and used it to perform sEMG pattern recognition. The main contributions of this paper are summarized as follows:

1) A novel IB method, 2O-IB, is proposed, by which we can train a deep learning model with stronger generalization ability to overcome the noise problem of sEMG signals.
2) Taking advantage of the inherent event-driven and sparse-computing characteristics of SNNs, the trained deep learning model is converted into an SNN counterpart. The computational energy consumption can be significantly reduced, which provides theoretical feasibility for the application in energy-constrained devices.

The rest of this paper is organized as follows. Section 2 provides some background information on IB and SNNs. Section 3 introduces the proposed 2O-IB mehtod and the sEMG recognition framework. In Section 4, we describe toy experiments to demonstrate that our 2O-IB method can effectively improve the generalization of network models. Section 5 compares the performance of our sEMG recognition framework with others in term of classification accuracy. The experimental results confirmed the superiority of our framework. Finally, Section 6 summarizes this paper.

2. Preliminaries

2.1. Information theory

Let X and Y be two random variables that take the values x and y from two finite sets with distributions $p(x)$ and $p(y)$, respectively. The *Shannon entropy* $H(X)$ measures the average uncertainty of X, which is defined as follows:

$$H(X) = -\sum_{x} p(x)\log p(x) \tag{1}$$

If not specified, all the logarithms are base 2 and entropy is expressed in bits. The conditional entropy $H(Y|X)$ measures the average uncertainty associated with Y under a known outcome of X, which is defined as

$$H(Y|X) = -\sum_{x,y} p(x)p(y|x)\log p(y|x) \tag{2}$$

In general, $H(Y|X) \neq H(X|Y)$ holds, and $H(X) \geqslant H(X|Y) \geqslant 0$. Further, the mutual information between X and Y measures the shared information between X and Y, which is defined as:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= \sum_{x,y} p(x)p(y|x)\log\tfrac{p(y|x)}{p(y)} \\ &\geqslant 0 \end{aligned} \tag{3}$$

For a dataset $\mathscr{D} = \{(x_1, y_1), \ldots, \ (x_N, y_N) | x_i \in X, \ y_i \in Y, i = 1, \ldots, \ N\}$, N is the number of i.i.d. training sample pairs. We use p to denote the distributions that are fixed and q to denote the distributions that are updated throughout the optimization process. Variable X and output Y follow the marginal distributions $p(X)$ and $p(Y)$, and their joint distribution is denoted by $p(x, y)$.

Suppose T is an intermediately compressed representation of X in the way of a Markov chain $X \leftrightarrow Y \leftrightarrow T$. For a given encoding map from X to the variable T, using the conditional probability $q(t|x)$ and $q(y|t)$, we can compute the following joint distribution:

$$q(x,t) = q(t|x)p(x), \tag{4}$$

$$q(y,t) = \sum_{x} q(t|x)p(x,y), \tag{5}$$

$$q(t|x,y) = q(t|x), \tag{6}$$

Thus,

$$q(x,y,t) = p(x,y)q(t|x,y) = p(x,y)q(t|x) \tag{7}$$

$$\begin{aligned} q(t|y) &= \tfrac{1}{p(y)}\sum_{x} q(x,y,t) = \tfrac{1}{p(y)}\sum_{x} p(x,y)q(t|x) \\ &= \sum_{x} p(x|y)q(t|x) \end{aligned} \tag{8}$$

$$q(y|t) = \tfrac{1}{q(t)}\sum_{x} q(x,y,t) = \tfrac{1}{q(t)}\sum_{x} p(x,y)q(t|x) = \sum_{x} p(y|x)q(x|t) \tag{9}$$

$$q(t) = \sum_x p(x)q(t|x) \tag{10}$$

2.2. Information bottleneck

The information bottleneck (IB) principle aims to extract a compression T of X that is relevant for determining Y, where T is found to maximize the relevance with Y, formulated in terms of the mutual information $I(Y;T)$, while under the constraint of input compression, which is formulated in terms of $I(X;T)$. This constrained optimization problem can be written as:

$$\arg_{T\in\Delta} \max I(T;Y) \text{ s.t. } \mathrm{I}(X;T) \leqslant \gamma \tag{11}$$

where Δ is the variable space of T, and γ is a constraint number.

Rather than solving the constrained optimization problem in (11), the general IB method minimizes the Lagrangian cost function:

$$\min_{q(t|x)} \mathscr{L}_{\mathrm{IB}}(T) = I(Y;T) - \beta I(T;X) \tag{12}$$

where β is the Lagrange multiplier, a non-negative free parameter that controls the trade-off between the input compression and output determination. Considering that the mutual information between two probability functions is non-negative, minimizing $\mathscr{L}_{\mathrm{IB}}(T)$ is to minimize $I(X;T)$ while maximizing $I(Y;T)$.

However, in the original IB principle, it is generally challenging to compute the mutual information $I(X;T)$ for continuous and non-Gaussian random samples. To address this limitation, [2] proposed a variational information bottleneck (VIB) method. The VIB quantifies the upper bound of $I(X;T)$ by

$$\begin{aligned} I(X;T) &= D_{\mathrm{KL}}(q(T|X)\|r(T)) - D_{\mathrm{KL}}(q(T)\|r(T)) \\ &\leqslant D_{\mathrm{KL}}(q(T|X)\|r(T)) \end{aligned} \tag{13}$$

where r is the surrogate marginal distribution over the bottleneck activation T. The lower bound of $I(T,Y)$ given by [2] is as follows:

$$\begin{aligned} I(T,Y) &\geqslant \sum_{y,t} q(y,t) \log \tfrac{q(y|t)}{p(y)} \\ &= \sum_{y,t} q(y,t) \log q(y|t) - \sum_{y,t} p(y) \log p(y) \\ &= \sum_{y,t} q(y,t) \log q(y|t) + H(Y) \end{aligned} \tag{14}$$

where the term $H(Y)$ is a constant because the label Y is independent of the optimization method. Therefore, it can be ignored in the loss function of IB. Thus, the lower bound of the VIB function can be rewritten as

$$\begin{aligned} \mathscr{L}_{\mathrm{VIB}}(T) &\geqslant \mathbb{E}_{q(T,Y)} \log q(Y|T) \\ &\quad -\beta D_{\mathrm{KL}}(q(T|X)\|r(T)) + \mathrm{const} \end{aligned} \tag{15}$$

Different from the proposed approach in [2] that calculates the upper bound of $I(X;T)$, [38] substitutes the term $I(X;T)$ directly with the entropy $H(T)$, and gives the deterministic information bottleneck (DIB):

$$\mathscr{L}_{\mathrm{DIB}}(T) = I(T;Y) - \beta H(T) \tag{16}$$

Note that $I(X;T) = H(T) - H(T|X)$. The conditional entropy $H(T|X)$ is sometimes called the noise entropy, and measures the stochasticity in the mapping from X to T. [38] claimed that, by removing the stochasticity, the DIB method produces a deterministic encoding distribution of T about X.

2.3. Spiking neural network

In this paper, we used the integrate-and-fire (IF) neuron model as the computing unit. The IF model is simple and efficient spiking neuron and has been widely used in ANN2SNN conversions. The mathematical form of the IF model is as follows:

$$\tau_m \frac{\mathrm{dV}(t)}{\mathrm{dt}} = V_{\mathrm{reset}} + R_m \Omega(t) \tag{17}$$

where $V(t)$ is the membrane potential at time t, V_{reset} is the recovery potential, τ_m and R_m denote the neuronal time constant and the membrane resistance, respectively. $\Omega(t)$ is the sum of the neuronal synaptic input current. The larger the value of R_m is, the greater the influence of the input current Ω on the membrane potential V will be. The larger the time constant τ_m is, the slower the response speed of the potential V to the input Ω will be. For an IF neuron j in layer ϕ, its synaptic input current is

$$\Omega_j^{\phi}(t) = \sum_{i=1}^{N} s_i^{\phi-1}(t) \cdot w_{ji}^{\phi-1}, \tag{18}$$

where $w_{ji}^{\phi-1}$ represents the connection weight between neuron j and neuron i in the previous layer $\phi-1$, $s_i^{\phi-1}(t)$ represents the output of presynaptic neuron i at time t, and $s_i^{\phi-1}(t)=1$ means that neuron i produces a spike output, otherwise $s_i^{\phi-1}(t)=0$. When the membrane potential of an IF neuron exceeds its firing threshold V_{th}, it generates a spike signal forward. Its membrane potential V is then immediately reset to V_{reset}, and the neuron enters the refractory time t_{ref}.

2.4. Conversion of spiking neural network

Unlike conventional continuous-valued ANNs, SNNs use discrete spikes to process and transmit signals. The current widely used error back-propagation algorithms cannot be directly applied to the training of SNN models. In general, the training strategies of SNNs can be divided into spike-timing-dependent plasticity (STDP) [5,24], local training rules based on reinforcement learning [16,4,31], spike-based back-propagation [40,46,25], and ANN2SNN conversion methods [13,36,48]. The unsupervised, semi-supervised and fully supervised learning rules based on STDP/ reinforcement learning are still limited to the training of shallow SNNs, and the accuracy obtained on more complex datasets, such as Cifar10, is much lower than that of conventional ANNs. The back propagation algorithm based on peaks introduces high computational complexity, operation overhead and time consumption. The number of SNN layers to be trained is still limited to between 9 and 11.

Pre-built ANNs use the traditional backpropagation-based method for training, and can then be converted into SNNs with a corresponding structure. The specific construction method is as follows:

1) Build a deep ANN model. No bias is used in each layer. In addition, the activation function of the hidden layer is ReLU, and the output layer is Softmax.
2) Use error backpropagation to train the ANN model.
3) Replace the ReLU unit with IF neurons, and map the trained connection weights directly to the SNN of the corresponding structure.

In detail, the spiking version for some traditional analog operators are:

1) The output of a spiking convolution at time t in layer ϕ is:

$$y^{\phi}(t) = \sum_{i=1}^{H}\sum_{j=1}^{W} s_{i,j}^{\phi-1}(t) \cdot w_{i,j}^{\phi-1} \tag{19}$$

where H and W are the height and width of convolution kernel, respectively. $w_{i,j}^{\phi-1}$ is the weight of the kernel element at index i,j in previous layer $\phi-1$.

2) The spiking softmax layer simply predicts the output class corresponding to the neuron that spiked most during the presentation of the stimulus.

3. Proposed method

3.1. Method framework

Figs. 1 and 2illustrates the overall process architecture of the sEMG pattern recognition based on our proposed method. The process is divided into two stages. Stage ① is training of the ANN model using the 2O-IB method. At this stage, the multi-channel sEMG signal is normalized and input into the conventional ANN using the ReLU activation function, and the ANN model is trained according to the value of the sEMG signal and the corresponding class label. Stage ② is to use the converted SNN model and sEMG spike stream for pattern recognition inference. The sEMG spike stream is generated using Poisson sampling, which can be written as

$$s(t) = \mathscr{H}\left\{ p - \frac{R_0^1(t)}{\Delta t \cdot f_{max} \cdot (\max(\mathbf{P}) - \min(\mathbf{P}))} \right\}, \tag{20}$$

where $s(t)$ is the binary value of a sampling point at time t, and $s=1$ means a spike is generated. Moreover, $\mathscr{H}(\cdot)$ is the Heaviside function, p is the original normalized value of the sEMG signal at the sampling point, Δt is the unit of time step during SNN running, and f_{max} is the maximal input firing rate. $\max(\mathbf{P})$ and $\min(\mathbf{P})$ denote the maximal and minimal normalized value over the whole sEMG dataset, respectively, and $R_0^1(t)$ is a random value in $[0,1]$ at t.

The ANN2SNN conversion used in this paper follows the method demonstrated in Section 2.4. There have been reports that the computational accuracy of the SNNs converted by ANN2SNN is lower than that of the original ANN. However, if

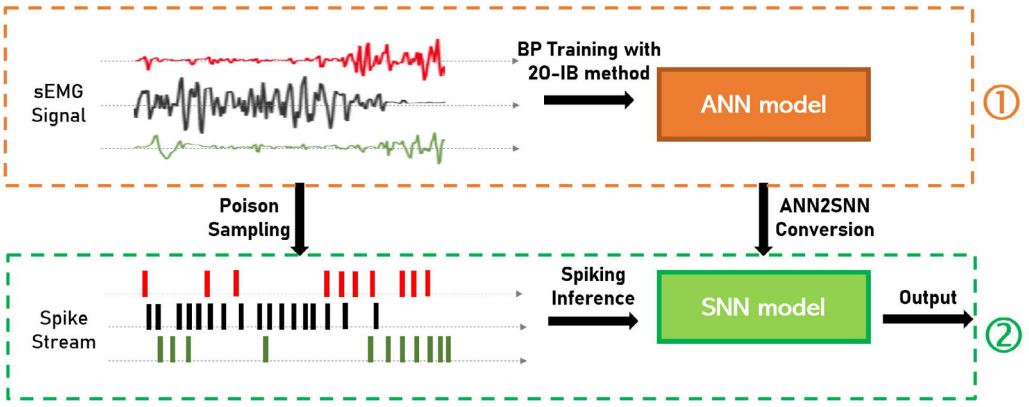


Fig. 1. Overall framework of our proposed second-order information bottleneck (2O-IB) based spiking neural network (SNN) recognition method. It can be roughly divided into two stages: ①, artificial neural network (ANN) model training and ② SNN model inference.

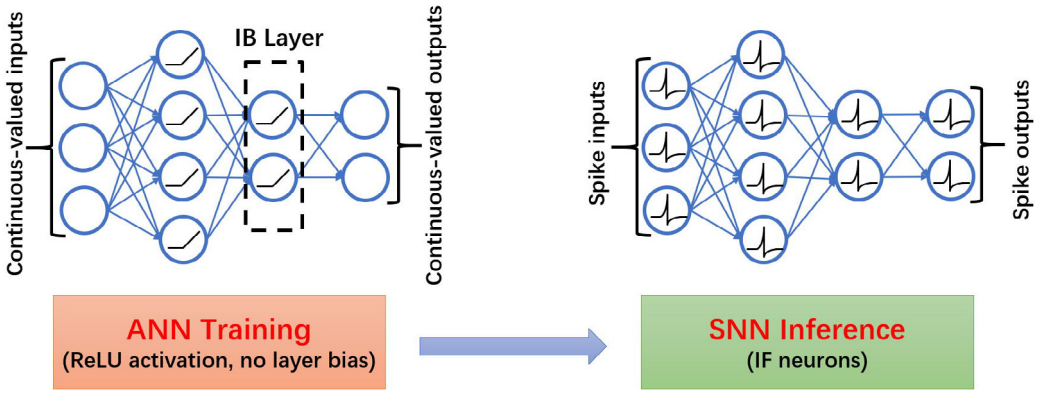


Fig. 2. Diagram of the ANN2SNN conversion.

the performance of the original ANN can be improved, the corresponding SNN counterpart can also be improved. The ANN training algorithm described in this paper uses our proposed 2O-IB loss function, which is introduced in Section 3.2 in detail.

3.2. Second-order information bottleneck

We propose a 2O-IB as follows:

$$\mathscr{L}_{2OIB}(T) = H^2(Y|T) + \beta I^2(X;T) \tag{21}$$

The L2 regularization loss is given as:

$$\mathscr{L}_{\text{reg}} = \lambda \|W\|^2 \tag{22}$$

The following total training loss is defined:

$$\begin{aligned} \mathscr{L} &= \mathscr{L}_{2OIB} + \mathscr{L}_{\text{reg}} \\ &= H^2(Y|T) + \beta I^2(X;T) + \lambda \|W\|^2, \end{aligned} \tag{23}$$

where β has the same meaning as in (12), and $\gamma\|W\|^2$ is an L2 regularization term of the network weight W with a small positive coefficient λ.

The conventional IB problem of minimizing $I(X;T)$ is a convex function of the encoding mapping $q(t|x)$ for fixed $p(x)$. Maximizing $H(Y|T) = H(Y) - I(T;Y)$ is a concave function of decoding mapping $q(y|t)$ for a fixed $p(x,y)$. The entropy $H(Y)$ is only dependent on the distribution of the variable (output) Y, and can be regarded as a constant for the whole dataset $\mathscr{D}$, or with a relatively small fluctuation for a patch of training data. Thus, the optimizing 2O-IB function $\mathscr{L}_{2OIB}$ is a concave function used to determine the global or local minimum.

We define the optimal parameter set ω of the network to achieve the best performance of the minimal value of the loss function), and define a randomly generated parameter set ϕ. Let the representation predicted (decoded) by ϕ be $\widehat{Y}$, and $\hat{y}$ is a random instance of $\widehat{Y}$. Thus, we have

$$\begin{aligned} H(q_\phi(Y|T)) &\leqslant H(q_\phi(Y|T)) + D_{KL}(q_\omega(Y|T)\|q_\phi(Y|T)) \\ &= -\mathbb{E}_{q_{\omega(Y,T)}}\left[\log q_\phi(Y|T)\right] \\ &\overset{(a)}{=} -\mathbb{E}_{q_{\omega\left(Y,\widehat{Y}\right)}}\left(\mathbb{E}_{q_{\phi\left(\widehat{Y},T\right)}}\left[\log q_\phi\left(Y|\widehat{Y}\right)\right]\right) \\ &= -\mathbb{E}_{q_{\omega\left(Y,\widehat{Y}\right)}}\left[\log q_\phi\left(Y|\widehat{Y}\right)\right] \\ &= \mathscr{C}\left(q_\phi\left(Y|\widehat{Y}\right)\right) \end{aligned} \tag{24}$$

where $\mathbb{E}\cdot$ denotes the expectation value, $\mathscr{C}(\cdot)$ is the cross-entropy, and (a) is due to

$$q_{Y|T}(y|t) = q_{Y|q(\hat{y}|t)}(y|q(\hat{y}|t)) = q_{Y|\widehat{Y}}(y|\hat{y}) \tag{25}$$

The inequality (24) becomes an equality only if $q_\phi(y|t)$ is equal to the "optimal" mapping $q_\omega(y|t)$. Further, it holds that $H(q_\phi(Y|T)) \to \mathscr{C}\left(q_\phi\left(Y|\widehat{Y}\right)\right)$ if the Kullback-Leibler (KL) divergence $D_{KL}(q_\omega(Y|T)\|q_\phi(Y|T)) \to 0$, which implies that if the distance between network parameter set ϕ and the "optimal" set ω is trained to be smaller, the term $H(q_\phi(Y|T))$ is closer to its upper bound.

The term $I(X;T)$ represents the information compressed from input signal X to the intermediate activation T:

$$\begin{aligned} I(X;T) &= \sum_{x,t} q(x,t)\log\left(\frac{q(x,t)}{p(x)q(t)}\right) \\ &= \sum_{x,t} q(x,t)\log\left(\frac{q(t|x)}{q(t)}\right) \\ &= \sum_{x,t} q(x,t)\log q(t|x) - \sum_t q(t)\log q(t). \end{aligned} \tag{26}$$

However, computing the marginal distribution of T, $q(t) = \sum_x q(t|x)p(x)$ might be difficult. Inspired by the VIB [2], we also used the variational distribution $r(t)$ to approximate $q(t)$. Because the KL divergence is defined to be nonnegative, we have the following:

$$\begin{aligned} D_{KL}(q(T)\|r(T)) &= \sum_t q(t)\log q(t) - \sum_t q(t)\log r(t) \\ &\geqslant 0 \end{aligned} \tag{27}$$

According to (26) and (27), we have

$$\begin{aligned} I(X;T) &\leqslant \textstyle\sum_{x,t} q(x,t)\log q(t|x) - \sum_t q(t)\log r(t) \\ &= \textstyle\sum_{x,t} p(x)q(t|x)\log q(t|x) - \sum_{x,t} p(x)q(t|x)\log r(t) \\ &= \frac{1}{N}\sum_{n=1}^{N} q(t|x_n)\log\frac{q(t|x_n)}{r(t)} \\ &= \frac{1}{N}\sum_{n=1}^{N} D_{KL}[q(T|x_n)\|r(T)] \end{aligned} \tag{28}$$

where N is the number of data samples that has been defined before.

Combining (24) and (28) with the fact that $H(Y|T) \geqslant 0, I(X;T) \geqslant 0$, we quantified the upper bound of our proposed 2O-IB as

$$\begin{aligned} \mathscr{L}_{2OIB} &\leqslant \left[\mathscr{C}\left(q_\phi\left(Y|\widehat{Y}\right)\right)\right]^2 \\ &+\beta\left[\frac{1}{N}\sum_{n=1}^{N} D_{KL}[q(T|x_n)\|r(T)]\right]^2 \end{aligned} \tag{29}$$

Finally, the upper bound $\bar{\mathscr{L}}$ of our loss function $\mathscr{L}$ can be given by

$$\begin{aligned} \mathscr{L} \leqslant \bar{\mathscr{L}} &= \lambda\|W\|^2 + \left[\mathscr{C}\left(q_\phi\left(Y|\widehat{Y}\right)\right)\right]^2 \\ &+\beta\left[\frac{1}{N}\sum_{n=1}^{N} D_{KL}[q(T|x_n)\|r(T)]\right]^2 \end{aligned} \tag{30}$$

We can convert the problem of minimizing the loss function $\mathscr{L}$ into that of minimizing the upper bound $\bar{\mathscr{L}}$; that is, reducing $\bar{\mathscr{L}}$ to achieve the purpose of reducing $\mathscr{L}$.

4. Toy experiment

To understand the effect of our proposed 2O-IB more intuitively and easily, we trained 2O-IB-based ANN models on the simple image classification datasets MNIST and FashionMNIST, and then compared them with cross-entropy and other IB methods-based models: VIB and nonlinear-IB. *Information plane* (info-plane) [37] was used to visualize the compression item $I(X;T)$ and the decompression item $H(Y|T)$ in our proposed 2O-IB and $I(Y;T)$ in VIB and nonlinear-IB.

We measured the activity of the IB layer neurons when performing ANN model training using different loss functions - cross-entropy, regular IB, nonlinear-IB, and our 2O-IB, on the test dataset, for inference tasks. Principal component analysis (PCA) was used to project the activity onto two dimensions. Fig. 3 presents the projection results of these loss functions. For the different IB methods, we selected the optimal β value through multiple sets of experiments ($\beta = 0.05$ for regular IB, and $\beta = 0.015$ for nonlinear-IB and our 2O-IB). The figure shows that these types of loss functions can separate different classes of samples, for example, different image classes could be classified into corresponding clusters. However, the ANN models trained using nonlinear-IB and our 2O-IB could separate sample clusters more effectively, making the intra-cluster distance smaller and the inter-cluster distance larger, which means that the bottleneck states carry almost no information about input vectors beyond class identity.

In Fig. 3, the clustering of 2D projections of the neuronal activity is illustrated in the IB layer at different β values on the test set of FashionMNIST. When β is 0, the information compression term $I(X;T)$ is eliminated, and the 2O-IB degenerates into the squared function of the conventional cross-entropy loss function with L2 regularization. The training goal is only to achieve the lowest classification error rate. The larger the value of β is, the smaller the number of clusters will be.

This situation arises because a larger value of β means that the network must focus on the information compression item $I(X;T)$. The network tries to reduce the uncertainty of the neuronal activity T of the IB layer caused by different input X. Specifically, the difference between the T generated by the input X of the same class decreases, and the class interval becomes smaller. The projection of the neuronal activity T also becomes more compact. In addition, the difference of T generated by different classes may also be reduced, so that the projections of T generated by different classes also overlap), as manifested by a reduced number of projection clusters. A suitable β can keep the clusters consistent with the number of classes in the dataset, and make the clusters as compact as possible. Although an excessively large value of β further improves the compactness of the clusters, it reduces the number of clusters, which is not conducive to the accuracy of classification tasks. The best performance occurs when the number of clusters is equal to the cardinality of Y, and performance decreases as the number of clusters decreases. For example, in Fig. 4, when β was 0.02, compared with the traditional cross-entropy, the scenario in which β is 0, the IB could effectively distinguish different clusters while increasing cluster spacing. When β is 0.2, we only obtain four category clusters through the 2D projection of neuronal activity using PCA.

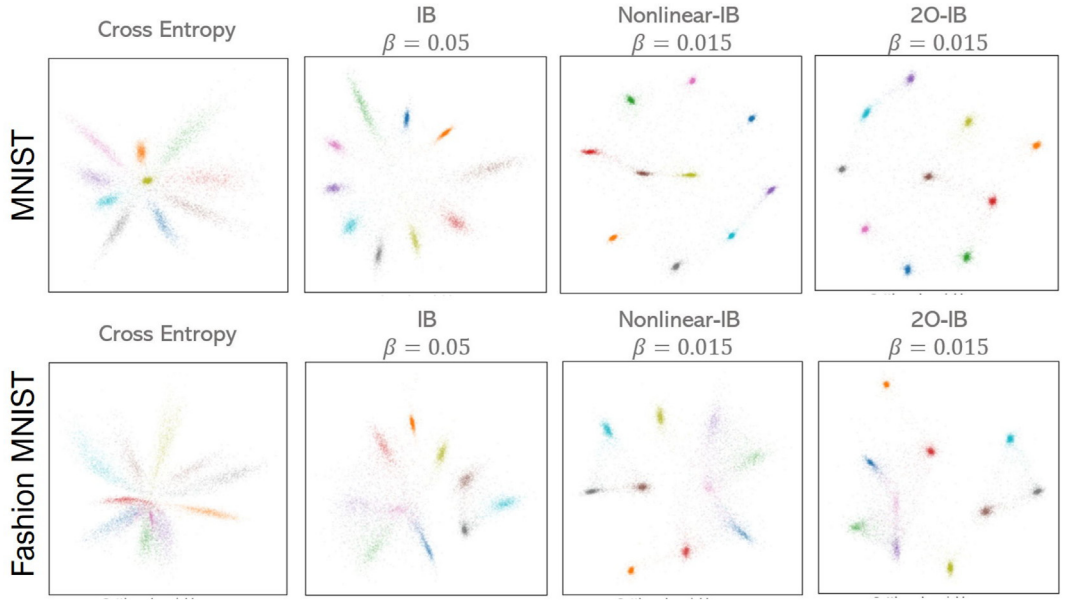


Fig. 3. Clusters of the projected neuronal activity of the information bottleneck (IB) layer for different loss function-based models.

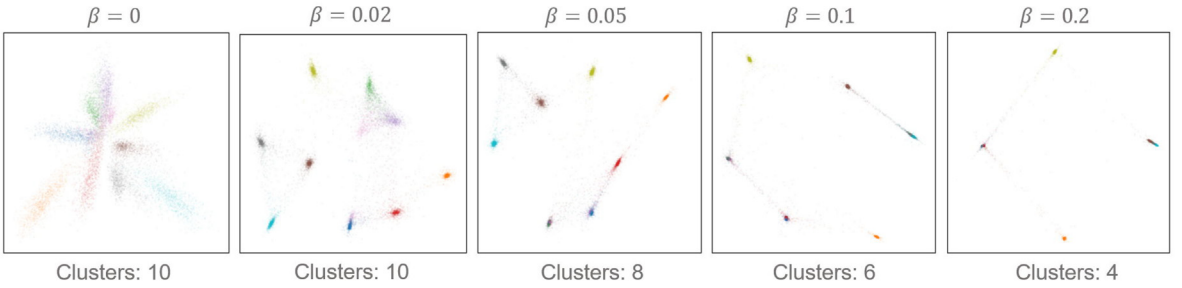


Fig. 4. Clusterization behavior of the bottleneck variable β for the second-order information bottleneck (2O-IB) Lagrangian on the FashionMNIST dataset.

Fig. 5 presents the changes in the values of $I(X;T)$ and $H(Y|T)$ during model training with different β values. In the figure, $I(X;T)$ and $H(Y|T)$ had no major fluctuations after 20 training epochs because, after 20 epochs, the model has almost completed training, and subsequent training only fine-tuned the model weight slightly. As β increased, $I(X;T)$ gradually decreased after training is stabilized, while $H(Y|T)$ gradually increased because $H(Y|T)$ represents the uncertainty of Y with respect to T. A larger $H(Y|T)$ indicates a lower final classification accuracy of the network. Due to the increase in β, the model was more inclined to compress the input signal at the expense of certain discrimination ability. Moreover, $I(X;T)$ represents the relevance between the source X and the compressed state T. When β is too large, the model compresses state T to a smaller value range, reducing the ability of T to restore X.

5. Results of sEMG recognition

5.1. sEMG dataset

We tested our proposed 2O-IB based SNN method on the Non-Invasive Adaptive Prosthetics (NinaPro) dataset [3], which includes data acquired from 67 intact subjects and 11 subjects with an amputated hand, while performing several repetitive tasks such as hand movements and finger force patterns.

The number of subjects who participated in the data collection is comparatively high, especially considering the difficulty of recruiting trans-radial amputees, and the fact that intact subjects can only be used as a 'proxy' measure for amputees.

In this paper, we used the first and second sub-database of NinaPro (denoted as DB-1 and DB-2, respectively) to validate our method. NinaPro DB-1 contains 10-channel sparse sEMG signals recorded from 27 intact subjects. Each hand gesture was recorded over 10 trials at a sampling rate of 100 Hz. NinaPro DB-2 contains 12-channel sEMG signals recorded from 40 intact

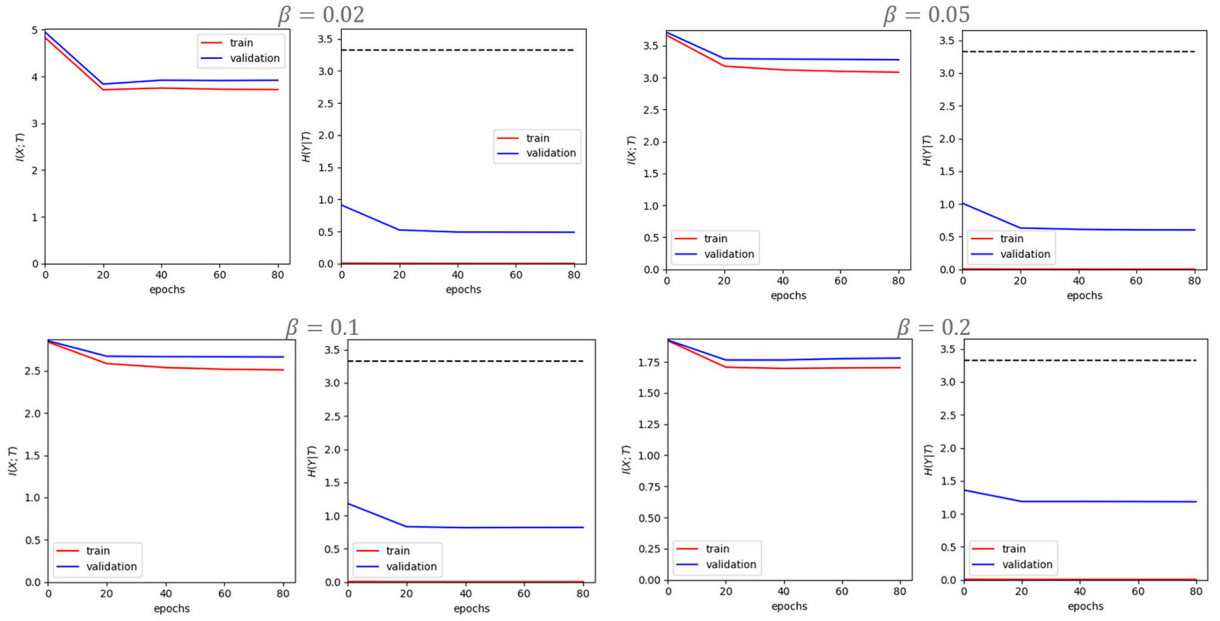


Fig. 5. Varation of $I(X;T)$ and $H(Y|T)$ during training with different β values.

subjects. Each subject was required to perform 49 types of hand movements including eight isometric and isotonic hand configurations, nine basic wrist movements, 23 grasping and functional movements and nine force patterns. Each movement was repeated six times with a 3 s rest in between. NinaPro DB-2 was recorded with a sampling rate of 2,000 Hz and was filtered with a Hampel filter to remove 50 Hz power line interference. Table 6 lists some examples of gestures in the NinaPro dataset, in which the top row represents some basic movements of the fingers, the central row presents isometric and isotonic hand configurations and basic movements of the wrist, and the bottom row represents grasping and functional movements. Fig. 6.

5.2. Network model

We used two uncomplicated neural network models to test and compare our methods (Fig. 7). Model#1 and Model#2 both contain convolutional layers, fully connected layers, softmax classification layers, and IB layers. If the traditional train-

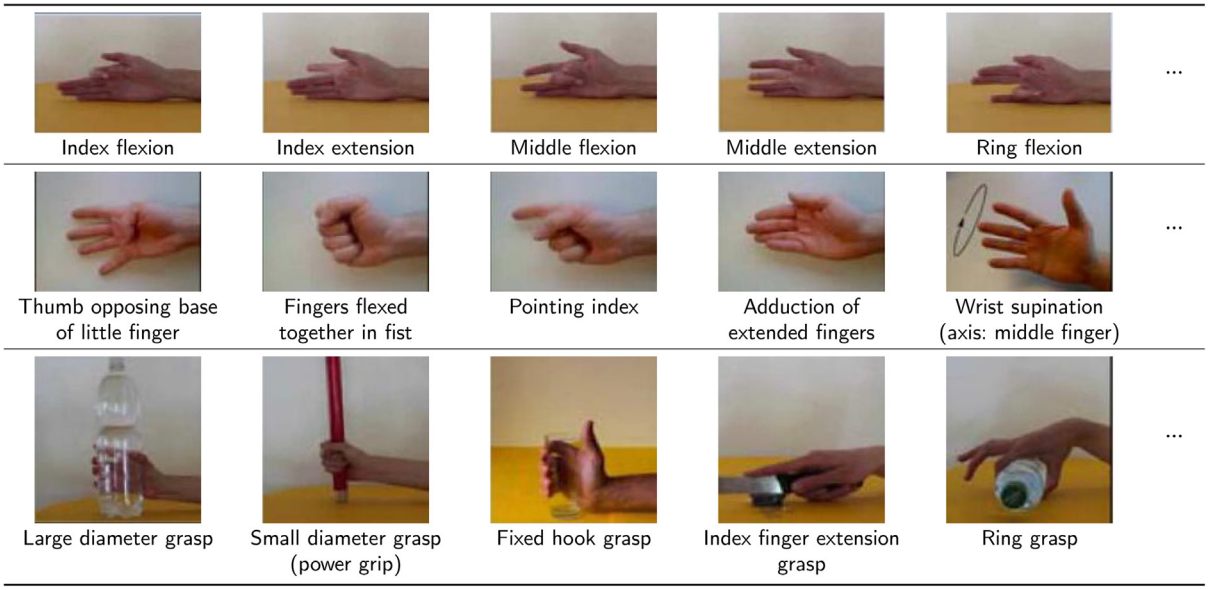


Fig. 6. Some samples of movements and force patterns of the NinaPro datasets.

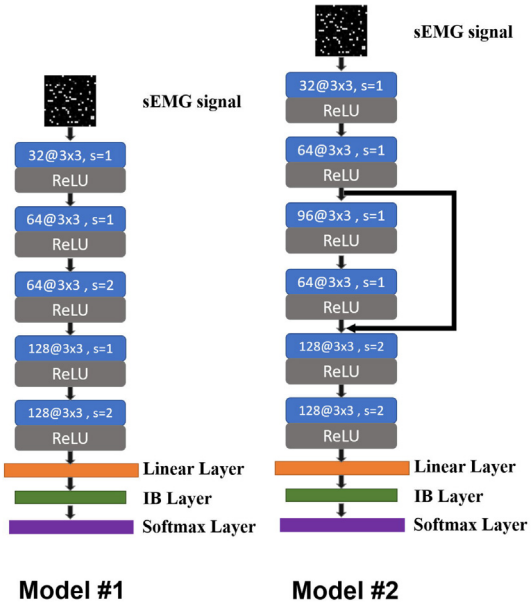


Fig. 7. Structure of two ANN models used in our experiments.

Table 1
Comparison of classification accuracy on the NinaPro-DB1 dataset of the networks trained using different loss functions (cross-entropy, regular information bottleneck (IB), and our second-order (2O)-IB).

Pattern	Model#1-CE	Model#1-IB	Model#1-2OIB	Model#2-CE	Model#2-IB	Model#2-2OIB	Pattern	Model#1-CE	Model#1-IB	Model#1-2OIB	Model#2-CE	Model#2-IB	Model#2-2OIB
p1	63.4%	**84.5%**	78.9%	60.6%	73.2%	**78.9%**	p27	77.5%	80.0%	**85.0%**	82.5%	82.5%	82.5%
p2	86.4%	**88.6%**	86.4%	81.8%	86.4%	**88.6%**	p28	58.6%	75.9%	**79.3%**	75.9%	**77.6%**	74.1%
p3	73.9%	81.1%	**82.6%**	76.8%	75.4%	**87.0%**	p29	44.0%	**57.3%**	**57.3%**	53.3%	53.3%	**66.7%**
p4	69.6%	82.6%	**84.9%**	**89.1%**	78.3%	87.0%	p30	77.8%	**88.9%**	81.5%	81.5%	83.3%	**87.0%**
p5	81.4%	82.9%	**87.1%**	85.7%	**91.4%**	85.7%	p31	61.3%	**66.1%**	62.9%	59.7%	62.9%	**67.7%**
p6	53.2%	**63.8%**	**63.8%**	61.7%	**65.9%**	63.8%	p32	**79.3%**	77.6%	74.1%	75.9%	**82.8%**	74.1%
p7	72.0%	**80.0%**	**80.0%**	74.0%	66.0%	**78.0%**	p33	52.8%	60.4%	**75.5%**	62.3%	66.0%	**71.7%**
p8	65.1%	**79.1%**	**79.1%**	76.7%	**81.4%**	74.4%	p34	50.8%	**67.7%**	66.2%	63.1%	66.2%	**67.7%**
p9	68.8%	**72.9%**	**72.9%**	77.1%	70.8%	**79.2%**	p35	57.6%	**68.2%**	66.7%	53.0%	**66.7%**	60.6%
p10	**88.2%**	86.3%	84.3%	86.3%	88.2%	**90.2%**	p36	50.0%	61.9%	**71.4%**	64.3%	64.3%	**66.7%**
p11	48.9%	**62.2%**	**62.2%**	60.0%	62.2%	**64.4%**	p37	54.7%	66.0%	**73.6%**	56.6%	**69.8%**	64.2%
p12	72.0%	**80.0%**	**80.0%**	70.0%	70.0%	**78.9%**	p38	47.5%	**70.5%**	57.4%	59.0%	**68.9%**	**68.9%**
p13	60.0%	**77.1%**	**77.1%**	68.6%	**71.4%**	71.2%	p39	40.0%	43.8%	**55.0%**	43.8%	43.8%	**55.0%**
p14	68.4%	**78.9%**	**78.9%**	80.7%	**82.5%**	64.6%	p40	67.6%	66.2%	**70.6%**	**73.5%**	69.1%	67.6%
p15	65.4%	**78.8%**	73.1%	67.3%	67.3%	**71.7%**	p41	73.3%	**85.0%**	80.0%	76.7%	**83.3%**	81.7%
p16	45.8%	70.8%	**81.3%**	58.3%	62.5%	**64.6%**	p42	60.0%	**74.5%**	69.1%	58.2%	60.0%	**61.8%**
p17	65.2%	73.9%	**76.1%**	**73.9%**	65.2%	71.7%	p43	80.0%	83.3%	**86.7%**	78.3%	76.7%	**85.0%**
p18	68.8%	70.8%	**77.1%**	**79.2%**	66.7%	72.9%	p44	60.7%	73.8%	**77.0%**	67.2%	**77.0%**	72.1%
p19	74.6%	**88.1%**	84.7%	78.0%	81.4%	**83.1%**	p45	69.6%	**78.3%**	75.4%	66.7%	**75.4%**	73.9%
p20	**89.8%**	83.7%	87.8%	81.6%	81.6%	**89.8%**	p46	72.3%	**84.6%**	83.1%	80.0%	**84.6%**	83.1%
p21	58.7%	68.3%	**69.8%**	58.7%	69.8%	**71.4%**	p47	70.0%	78.3%	**81.7%**	73.3%	68.3%	**81.7%**
p22	**82.7%**	**82.7%**	76.9%	78.8%	75.0%	**82.7%**	p48	66.7%	**75.9%**	**75.9%**	**88.9%**	77.8%	75.9%
p23	48.8%	72.1%	**79.1%**	62.8%	67.4%	**74.4%**	p49	73.8%	80.0%	**82.5%**	73.8%	**75.0%**	**75.0%**
p24	46.5%	**56.3%**	**56.3%**	46.5%	53.5%	**62.0%**	p50	63.3%	70.9%	**73.4%**	60.8%	**74.7%**	68.4%
p25	59.6%	63.8%	**74.5%**	63.8%	**74.5%**	72.3%	p51	74.7%	**82.3%**	**82.3%**	73.4%	81.0%	**86.1%**
p26	60.6%	**84.8%**	75.8%	69.7%	69.7%	**72.8%**	p52	73.4%	82.5%	**87.5%**	78.8%	**83.8%**	78.8%
							Average	65.1%	72.5%	**75.5%**	69.6%	74.6%	**76.5%**

Table 2
Comparison of classification accuracy on the NinaPro-DB2 dataset of the networks trained using different loss functions (cross-entropy, regular information bottleneck (IB), our second-order (2O)-IB).

Pattern	Model#1-CE	Model#1-IB	Model#1-2OIB	Model#2-CE	Model#2-IB	Model#2-2OIB	Pattern	Model#1-CE	Model#1-IB	Model#1-2OIB	Model#2-CE	Model#2-IB	Model#2-2OIB
p1	63.6%	**83.6%**	81.8%	56.4%	81.8%	**83.6%**	p26	**82.1%**	**82.1%**	64.3%	64.3%	75.0%	**78.6%**
p2	85.7%	85.7%	**91.4%**	**91.4%**	55.6%	88.6%	p27	65.6%	**78.1%**	**78.1%**	**81.3%**	78.1%	69.8%
p3	**76.5%**	74.5%	72.5%	68.3%	**80.4%**	76.5%	p28	62.3%	75.5%	**79.2%**	67.9%	66.0%	**72.7%**
p4	81.1%	**89.2%**	83.8%	81.1%	89.2%	**91.9%**	p29	56.8%	61.4%	**72.7%**	47.7%	63.6%	**82.2%**
p5	89.1%	**91.3%**	89.1%	91.3%	**95.7%**	**95.7%**	p30	73.3%	80.0%	**86.7%**	71.1%	**80.0%**	67.3%
p6	42.1%	**63.2%**	57.9%	55.3%	**63.2%**	57.9%	p31	51.0%	**63.3%**	61.2%	49.0%	67.3%	**76.4%**
p7	77.1%	77.1%	**85.7%**	80.0%	80.0%	**82.9%**	p32	58.2%	**74.5%**	72.7%	52.7%	**74.5%**	59.6%
p8	62.5%	**81.3%**	78.1%	65.6%	**84.4%**	81.3%	p33	59.6%	**66.0%**	53.2%	48.9%	63.8%	**69.8%**
p9	52.8%	47.2%	**63.9%**	55.6%	55.6%	**58.3%**	p34	55.8%	60.5%	**67.4%**	51.2%	**74.4%**	74.0%
p10	71.7%	**93.5%**	89.1%	73.9%	**91.3%**	**91.3%**	p35	60.0%	68.0%	**76.0%**	60.0%	**70.0%**	67.4%
p11	**60.0%**	56.0%	**60.0%**	52.0%	**72.0%**	**72.0%**	p36	48.8%	**72.1%**	69.8%	51.2%	65.1%	**65.2%**
p12	61.9%	80.9%	**83.3%**	59.5%	78.6%	**81.0%**	p37	54.3%	56.5%	**58.7%**	43.5%	56.5%	**60.0%**
p13	61.0%	**70.7%**	63.4%	65.9%	**70.7%**	68.3%	p38	48.6%	**60.0%**	54.3%	40.0%	**51.4%**	47.9%
p14	81.4%	**83.7%**	81.4%	79.1%	86.0%	**88.4%**	p39	37.5%	**54.2%**	41.7%	35.4%	41.7%	**73.1%**
p15	59.4%	**71.9%**	62.5%	59.4%	**68.8%**	**68.8%**	p40	61.5%	65.4%	**71.2%**	61.5%	71.2%	**86.7%**
p16	57.1%	61.9%	**66.7%**	52.4%	61.9%	**64.3%**	p41	64.4%	**86.7%**	77.8%	64.4%	**88.9%**	82.1%
p17	64.9%	75.7%	**78.4%**	62.2%	**86.5%**	75.7%	p42	66.7%	71.8%	**87.2%**	59.0%	76.9%	**71.4%**
p18	74.1%	**79.6%**	**79.6%**	79.6%	83.3%	**87.0%**	p43	61.2%	**69.4%**	59.2%	59.2%	**69.4%**	69.0%
p19	69.4%	75.0%	**80.6%**	72.2%	80.6%	**86.1%**	p44	**69.0%**	**69.0%**	66.7%	59.5%	78.7%	**83.0%**
p20	78.9%	**89.5%**	86.8%	76.3%	**86.8%**	**86.8%**	p45	70.2%	**78.7%**	76.6%	72.3%	77.2%	**72.7%**
p21	**64.4%**	60.0%	57.8%	46.7%	**71.1%**	57.8%	p46	63.6%	72.7%	**88.6%**	61.4%	79.2%	**81.3%**
p22	88.6%	**94.3%**	85.7%	68.6%	94.3%	**97.1%**	p47	72.9%	**77.1%**	**77.1%**	66.7%	**86.0%**	83.7%
p23	54.3%	**71.4%**	62.9%	42.9%	**71.4%**	68.6%	p48	76.7%	83.7%	**86.0%**	72.1%	**81.8%**	**81.8%**
p24	48.6%	51.4%	**62.2%**	49.7%	**56.8%**	**56.8%**	p49	63.6%	76.4%	**80.0%**	**70.1%**	66.1%	64.5%
p25	59.1%	**68.2%**	65.9%	59.1%	72.3%	**77.3%**	Average	64.7%	73.3%	**74.2%**	62.3%	74.5%	**75.4%**

ing model of a cross entropy loss function was used, the fully connected layer was used instead. The input to these two models was a single-channel 2D sEMG signal mapping, in which each element had a value of 0 or 1 (indicating whether or not the element is fired to generate a spike). The input sEMG mapping passed through multiple consecutive convolutional layers activated by the ReLU function. The specific convolution parameters are given in Fig. 7. For example, *32@3 × 3, s = 1* means that the layer had 32 convolution kernels with a size of 3 × 3, and their stride was 1. The numbers of neurons in the linear layers of Model#1 and Model#2 were 256 and 384, respectively. The number of neurons in the IB layer was fixed at 256. The number of nodes in the classification output layer was the same as the number of pattern classes in the training dataset.

The depth of Model#2 was greater than that of Model#1, and Model#2 contained a residual connection, a neural connection module that can improve network training speed and calculation accuracy. The models we built were not aimed at obtaining state-of-the-art recognition accuracy, but to compare the differences in the accuracy of the models using different training strategies.

The input firing rate of both Model#1 and Model#2 were set to 500 Hz, and the parameters β and λ in (30) were 0.015 and 0.01, respectively.

5.3. Results

Tables 1,2 show the classification accuracy of NinaPro-DB1 and NinaPro-DB2, respectively, under different training strategies. It can be seen from the two tables that compared with the conventional and commonly used cross-entropy loss function, the IB methods could effectively improve the model computational accuracy for most sEMG patterns, as well as the overall average accuracy over the whole sEMG datasets. On NinaPro-DB1, the average classification accuracy of Model#1 and Model#2 based on cross-entropy loss function training was 65.1% and 69.6%, respectively, while Model#1 and Model#2 based on the conventional information bottleneck method obtained 72.5% and 74.6% accuracy, respectively, achieving significant performance improvements. With the 2O-IB method we proposed, the classification accuracy of the sEMG model was further improved, by 3.0% and 1.9%, respectively. The results in the dataset NinaPro-DB2 were also similar. Compared with general cross-entropy training, the methods based on IB improved the classification accuracy of the model by nearly 10%, and the 2O-IB improvement effect we proposed was more obvious. The classification accuracy of the model based on our proposed 2O-IB method was the highest, refecting the effectiveness of 2O-IB.

6. Conclusion

In this paper, we introduce a method for training neural networks for sEMG pattern recognition, based on our proposed 2O-IB and converting the trained models to SNN counterparts. The advantages of this method are as follows: firstly, through the deep learning model incorporating 2O-IB training proposed by us, the generalizability of the neural network model can be effectively improved, so that the model can effectively reduce the noise while maintaining good computational accuracy and generalized inference ability. Owing to the SNN characteristics of event triggering and sparse computing, our sEMG pattern inference model could effectively reduce computing power consumption compared with traditional deep learning models, which is conducive to the deployment of sEMG pattern recognition tasks on embedded or wearable devices. Secondly, the realization method of loss function based on 2O-IB was deduced theoretically, and the validity of this method for a simple classification task was demonstrated using the MNIST and FashionMNIST datasets. Finally, on a real sEMG dataset, we compared our method with the currently widely used cross entropy loss function and the traditional IB method. The experimental results showed that the 2O-IB method could effectively improve the calculation accuracy and noise performance of sEMG pattern recognition.

CRediT authorship contribution statement

Anguo Zhang: Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Yuzhen Niu:** Visualization, Investigation. **Yueming Gao:** Supervision, Writing – review & editing. **Junyi Wu:** Software, Validation. **Zhipeng Gao:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Achille, S. Soatto, Information Dropout: Learning Optimal Representations Through Noisy Computation, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 2897–2905, https://doi.org/10.1109/TPAMI.2017.2784440, arXiv:1611.01353.
[2] A.A. Alemi, I. Fischer, J.V. Dillon, K. Murphy, Deep Variational Information Bottleneck, in: International Conference on Representation Learning, 2017, 1–19. arXiv:1612.00410..
[3] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.G.M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, H. Müller, Electromyography data for non-invasive naturally-controlled robotic hand prostheses, Scientific Data 1 (2014) 1–13, https://doi.org/10.1038/sdata.2014.53.

[4] D. Baras, R. Meir, Reinforcement Learning, Spike-Time-Dependent Plasticity, and the BCM Rule, Neural Comput. 19 (2007) 2245–2279, https://doi.org/10.1162/neco.2007.19.8.2245.
[5] J. Bill, R. Legenstein, A compound memristive synapse model for statistical learning through STDP in spiking neural networks, Front. Neurosci. 8 (2014) 1–18, https://doi.org/10.3389/fnins.2014.00412.
[6] S.M. Bohte, J.N. Kok, H.L. Poutré, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48 (2002) 17–37.
[7] A.R.R. Casti, A. Omurtag, A. Sornborger, E. Kaplan, B. Knight, J. Victor, L. Sirovich, A population study of integrate-and-fire-or-burst neurons, Neural Comput. 14 (2006) 957–986.
[8] H.Y. Chae, K. Lee, J. Jang, K. Park, J.J. Kim, A Wearable sEMG Pattern-Recognition Integrated Interface Embedding Analog Pseudo-Wavelet Preprocessing, IEEE Access 7 (2019) 151320–151328, https://doi.org/10.1109/ACCESS.2019.2948090.
[9] X. Chen, Y. Li, R. Hu, X. Zhang, X. Chen, Hand Gesture Recognition based on Surface Electromyography using Convolutional Neural Network with Transfer Learning Method, IEEE J. Biomed. Health Inf. 2194 (2020), https://doi.org/10.1109/jbhi.2020.3009383, 1–1.
[10] L. Cheng, Y. Liu, Z.G. Hou, M. Tan, D. Du, M. Fei, A Rapid Spiking Neural Network Approach with an Application on Hand Gesture Recognition, IEEE Trans. Cogn. Develop. Syst. 8920 (2019) 1–12, https://doi.org/10.1109/TCDS.2019.2918228.
[11] J. Chiang, Z.J. Wang, M.J. McKeown, A hidden Markov, multivariate autoregressive (HMM-mAR) network framework for analysis of surface EMG (sEMG) data, IEEE Trans. Signal Process. 56 (2008) 4069–4081, https://doi.org/10.1109/TSP.2008.925246.
[12] M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. Weng, A. Wild, Y. Yang, H. Wang, Loihi: A neuromorphic manycore processor with on-chip learning, IEEE Micro 38 (2018) 82–99, https://doi.org/10.1109/MM.2018.112130359.
[13] P.U. Diehl, D. Neil, J. Binas, M. Cook, S.C. Liu, M. Pfeiffer, Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing, in: Proceedings of the International Joint Conference on Neural Networks 2015-Septe, 2015. doi:10.1109/IJCNN.2015.7280696, arXiv: arXiv:1011.1669v3..
[14] E. Donati, M. Payvand, N. Risi, R. Krause, G. Indiveri, Discrimination of EMG Signals Using a Neuromorphic Implementation of a Spiking Neural Network, IEEE Trans. Biomed. Circuits Syst. 13 (2019) 793–801, https://doi.org/10.1109/TBCAS.2019.2925454.
[15] F. Duan, L. Dai, Recognizing the Gradual Changes in sEMG Characteristics Based on Incremental Learning of Wavelet Neural Network Ensemble, IEEE Trans. Industr. Electron. 64 (2017) 4276–4286, https://doi.org/10.1109/TIE.2016.2593693.
[16] V. Florian, Reinforcement Learning Through Modulation of Spike-Timing-Dependent Synaptic Plasticity, Neural Comput. 1502 (2007) 1468–1502.
[17] H. Han, S. Jo, Supervised hierarchical bayesian model-based electomyographic control and analysis, IEEE J. Biomed. Health Inf. 18 (2014) 1214–1224, https://doi.org/10.1109/JBHI.2013.2284476.
[18] R.M. Hecht, E. Noor, G. Dobry, Y. Zigel, A. Bar-Hillel, N. Tishby, Effective model representation by information bottleneck principle, IEEE Trans. Audio Speech Lang. Process. 21 (2013) 1755–1759, https://doi.org/10.1109/TASL.2013.2253097.
[19] E.M. Izhikevich, Simple model of spiking neurons, IEEE Trans. Neural Networks 14 (2003) 1569–1572.
[20] Z. Ju, G. Ouyang, M. Wilamowska-Korsak, H. Liu, Surface EMG based hand manipulation identification via nonlinear feature extraction and classification, IEEE Sens. J. 13 (2013) 3302–3311, https://doi.org/10.1109/JSEN.2013.2259051.
[21] N.K. Kasabov, Neucube evospike architecture for spatio-temporal modelling and pattern recognition of brain signal, Artif. Neural Networks Pattern Recogn. (2012) 225–243.
[22] N.K. Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, Neural Networks 52 (2014) 62–76, https://doi.org/10.1016/j.neunet.2014.01.006.
[23] M. Khezri, M. Jahed, A neuro-fuzzy inference system for sEMG-based identification of hand motion commands, IEEE Trans. Industr. Electron. 58 (2011) 1952–1960, https://doi.org/10.1109/TIE.2010.2053334.
[24] C. Lee, G. Srinivasan, P. Panda, K. Roy, Deep Spiking Convolutional Neural Network Trained with Unsupervised Spike Timing Dependent Plasticity, IEEE Trans. Cogn. Develop. Syst. (2018), https://doi.org/10.1109/TCDS.2018.2833071.
[25] J.H. Lee, T. Delbruck, M. Pfeiffer, Training deep spiking neural networks using backpropagation, Front. Neurosci. 10 (2016), https://doi.org/10.3389/fnins.2016.00508, arXiv:1608.08782.
[26] Y. Liu, L. Cheng, Spiking-Neural-Network Based Fugl-Meyer Hand Gesture Recognition for Wearable Hand Rehabilitation Robot, in: Proceedings of the International Joint Conference on Neural Networks 2018-July, 1–6, 2018. doi:10.1109/IJCNN.2018.8489141..
[27] R. Luo, S. Sun, X. Zhang, Z. Tang, W. Wang, A Low-Cost End-to-End sEMG-Based Gait Sub-Phase Recognition System, IEEE Trans. Neural Syst. Rehabil. Eng. 28 (2020) 267–276, https://doi.org/10.1109/TNSRE.2019.2950096.
[28] Y. Ma, E. Donati, B. Chen, P. Ren, N. Zheng, G. Indiveri, Neuromorphic Implementation of a Recurrent Neural Network for EMG Classification, in: 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), IEEE, 2020, pp. 69–73, https://doi.org/10.1109/AICAS48895.2020.9073810.
[29] W. Maass, Networks of spiking neurons: The third generation of neural network models, Neural Networks 10 (1997) 1659–1671, https://doi.org/10.1016/S0893-6080(97)00011-7.
[30] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, et al, A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (2014) 668–673, https://doi.org/10.1126/science.1254642 668(2014.
[31] M. Mozafari, S.R. Kheradpisheh, T. Masquelier, A. Nowzari-Dalini, M. Ganjtabesh, First-spike-based visual categorization using reward-modulated STDP, IEEE Trans. Neural Networks Learn. Syst. 29 (2018) 6178–6190, https://doi.org/10.1109/TNNLS.2018.2826721, arXiv:1705.09132.
[32] A.K. Mukhopadhyay, I. Chakrabarti, M. Sharad, Classification of Hand Movements by Surface Myoelectric Signal Using Artificial-Spiking Neural Network Model, Proc. IEEE Sens. 2018-Octob, 2018–2021. doi:10.1109/ICSENS.2018.8589757..
[33] G.R. Naik, D.K. Kumar, Jayadeva, Twin SVM for gesture classification using the surface electromyogram, IEEE Trans. Inf. Technol. Biomed. 14 (2010) 301–308. doi:10.1109/TITB.2009.2037752..
[34] L. Peng, Z.G. Hou, N. Kasabov, G.B. Bian, L. Vladareanu, H. Yu, Feasibility of NeuCube spiking neural network architecture for EMG pattern recognition, in: 2015 International Conference on Advanced Mechatronic Systems (ICAMechS), IEEE, 2015, pp. 365–369, https://doi.org/10.1109/ICAMechS.2015.7287090.
[35] B. Rekabdar, L. Fraser, M. Nicolescu, M. Nicolescu, A real-time spike-timing classifier of spatio-temporal patterns, Neurocomputing 311 (2018) 183–196, https://doi.org/10.1016/j.neucom.2018.05.069.
[36] B. Rueckauer, I.A. Lungu, Y. Hu, M. Pfeiffer, S.C. Liu, Conversion of continuous-valued deep networks to efficient event-driven networks for image classification, Front. Neurosci. 11 (2017) 1–12, https://doi.org/10.3389/fnins.2017.00682.
[37] R. Shwartz-Ziv, N. Tishby, Opening the Black Box of Deep Neural Networks via Information, 1–19, 2017. arXiv:1703.00810..
[38] D. Strouse, D.J. Schwab, The Deterministic Information Bottleneck, Neural Comput. 29 (2017) 1611–1630, https://doi.org/10.1162/NECO_a_00961, arXiv:1803.01446.
[39] S. Tam, M. Boukadoum, A. Campeau-Lecours, B. Gosselin, A Fully Embedded Adaptive Real-Time Hand Gesture Classifier Leveraging HD-sEMG and Deep Learning, IEEE Trans. Biomed. Circuits Syst. 14 (2020) 232–243, https://doi.org/10.1109/TBCAS.2019.2955641.
[40] P. Tino, A.J.S. Mills, Learning beyond finite memory in recurrent networks of spiking neurons, Neural Comput. 3611 (2006) 591–613.
[41] N. Tishby, F.C. Pereira, W. Bialek, The information bottleneck method, University of Illinois 411 (2000) 368–377.
[42] M. Vera, P. Piantanida, L.R. Vega, The Role of the Information Bottleneck in Representation Learning, in: 2018 IEEE International Symposium on Information Theory (ISIT), IEEE, 2018, pp. 1580–1584, https://doi.org/10.1109/ISIT.2018.8437679.
[43] W. Waass, T. Natschlger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. 14 (2002) 2531–2560.

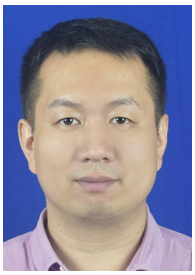
[44] K. Wang, X. Chen, L. Wu, X. Zhang, X. Chen, Z.J. Wang, High-Density Surface EMG Denoising Using Independent Vector Analysis, IEEE Trans. Neural Syst. Rehabil. Eng. 28 (2020) 1271–1281, https://doi.org/10.1109/TNSRE.2020.2987709.
[45] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, W. Geng, Surface-Electromyography-Based Gesture Recognition by Multi-View Deep Learning, IEEE Trans. Biomed. Eng. 66 (2019) 2964–2973, https://doi.org/10.1109/TBME.2019.2899222.
[46] J. Yang, W. Yang, W. Wu, A remark on the error-backpropagation learning algorithm for spiking neural networks, Appl. Math. Lett. 25 (2012) 1118–1120, https://doi.org/10.1016/j.aml.2012.02.016.
[47] M. Zanghieri, S. Benatti, A. Burrello, V. Kartsch, F. Conti, L. Benini, Robust Real-Time Embedded EMG Recognition Framework Using Temporal Convolutional Networks on a Multicore IoT Processor, IEEE Trans. Biomed. Circuits Syst. 14 (2020) 244–256, https://doi.org/10.1109/TBCAS.2019.2959160.
[48] A. Zhang, H. Zhou, X. Li, W. Zhu, Fast and robust learning in Spiking Feed-forward Neural Networks based on Intrinsic Plasticity mechanism, Neurocomputing 365 (2019) 102–112, https://doi.org/10.1016/j.neucom.2019.07.009.

Anguo Zhang was born in Hefei city, Anhui province, China in 1990. He received his bachelor's degree and master's degree in control engineering from Chongqing University in 2012 and 2016, respectively. He has been a researcher and senior engineer with the Research Institute of Ruijie, Ruijie Networks Co., Ltd. He is also pursuing the Ph.D in Communication and Information Systems at Fuzhou University. His research interest includes machine learning, artificial neural networks, control theory and applications.

Yuzhen Niu received her Ph.D. in Computer Science from Shandong University of China in 2010. She was a Post-Doctoral Researcher with the Department of Computer Science, Portland State University, Portland, OR. She is currently a Professor with the College of Mathematics and Computer Science, Fuzhou University, China. Her current research interests include image and video processing, computer vision, artificial intelligence, and multimedia.

Yueming Gao received the Ph.D. degree in electrical engineering from Fuzhou University, Fuzhou, China, in 2010. Since 2004, he has been involved in research in the areas of biomedical signal detection and processing. He is currently a professor in the college of physical and information engineering, Fuzhou University.

Junyi Wu received the B.S. degree in electronic information engineering from the Xiamen University of Technology,China,in 2017 and the M.S. degree from Fuzhou University,China,in 2020.He is currently working as an engineer in AI Research Center, Xiamen Meiya Pico Information Co.,Ltd.. His research interests include deep learning and person re-identification.



Zhipeng Gao was born in Xianyang city, Shanxi provice, China in 1984. Since 2010, he has been a researcher and senior engineer in the AI Research Center, Xaimen Meiya Pico Information Co., Ltd. His research interests include deep learning, electronic evidence collection and computer vision.