Applying Bayesian neural networks to identify pion, kaon and proton in BES Ⅱ *

XU Ye(徐晔)[1;1)] HOU Jian(侯健)[1,2] ZHU Kai-En(朱开恩)[1]

1 (Department of Physics, Nankai University, Tianjin 300071, China)
2 (Institute of High Energy Physics, CAS, Beijing 100049, China)

Abstract The Monte-Carlo samples of pion, kaon and proton generated from 0.3 GeV/c to 1.2 GeV/c by the 'tester' generator from SIMBES which are used to simulate the detector of BES Ⅱ are identified with the Bayesian neural networks (BNN). The pion identification and misidentification efficiencies are obviously better at high momentum region using BNN than the methods of χ^2 analysis of dE/dX and TOF information. The kaon identification and misidentification efficiencies are obviously better from 0.3 GeV/c to 1.2 GeV/c using BNN than the methods of χ^2 analysis. The proton identification and misidentification efficiencies using BNN are basically consistent with the ones of χ^2 analysis. The anti-proton identification and misidentification efficiencies are better below 0.6 GeV/c using BNN than the methods of χ^2 analysis.

Key words Bayesian neural networks, particle identification, pion, kaon, proton, anti-proton

PACS 07.05.Mh, 29.85.+c, 02.70.Uu

1 Introduction

Pion (π), kaon (K) and proton (p) are three of the six kinds of particles that can be directly detected by BES Ⅱ (the second generation of Beijing Spectrometer)[1], so it is important to identify π, K and p for the data analysis of BES Ⅱ experiment. π, K and p in BES Ⅱ experiment are generally identified with the χ^2 analysis of dE/dX and TOF information and the methods derived from it, and the identification capability of those methods is dependent on the momentum of the particle. π, K and p can be identified well below 0.7 GeV/c, but the π, K identification and misidentification efficiencies are obviously bad above 0.8 GeV/c. And the behavior of the anti-proton ($\bar{\text{p}}$) in the BES Ⅱ detectors is different from the one of p, so they will be identified separately. The Bayesian neural networks (BNN)[2] is an algorithm of the neural networks trained by Bayesian statistics. It is not only a non-linear function as neural networks, but also controls model complexity. So its flexibility makes it possible to discover more general relationships in data than the traditional statistical methods and its preferring simple models make it possible to solve the over-fitting problem better than the general neural networks[3]. In this paper, BNN will be applied to identify π, K, p and $\bar{\text{p}}$, respectively. And the efficiency of particle identification is compared using BNN and the χ^2 analysis of dE/dX and TOF information according to their weight from different momentum[4].

2 The classification with Bayesian neural networks[2, 5]

The idea of Bayesian neural networks is to regard the process of training a neural network as a Bayesian inference. Bayes' theorem is used to assign a posterior density to each point, $\bar{\theta}$, in the parameter space of the neural networks. Each point $\bar{\theta}$ denotes a neural network. In the method of the Bayesian neural network, one performs a weighted average over all points in the parameter space of the neural network, that is, all neural networks. The methods make use of training data $\{(x_1,t_1),\ (x_2,t_2),\ \cdots,\ (x_n,t_n)\}$, where t_i is the known label associated with data x_i. t_i=0, 1, $\cdots$, N−1, if there are N classes in the problems of classification; x_i has P components if there are P factors on which the classification is influenced. That is the set of data $x = (x_1,\ x_2,\ \cdots,\ x_n)$ which corre-

Received 28 May 2007
* Supported by National Natural Science Foundation of China (10605014)
1) Corresponding author, E-mail: xuye76@nankai.edu.cn

sponds to the set of target $t=(t_1,\ t_2,\ \cdots,\ t_n)$. The posterior density assigned to the point $\bar{\theta}$, that is, to a neural network, is given by Bayes' theorem

$$p(\bar{\theta}\,|\,x,t)\ =\ \frac{p(x,t\,|\,\bar{\theta})p(\bar{\theta})}{p(x,t)}=\frac{p(t\,|\,x,\bar{\theta})p(x\,|\,\bar{\theta})p(\bar{\theta})}{p(t\,|\,x)p(x)}=\frac{p(t\,|\,x,\bar{\theta})p(\bar{\theta})}{p(t\,|\,x)}\,, \tag{1}$$

where data x do not depend on $\bar{\theta}$, so $p(x\,|\,\theta)=p(x)$. We need the likelihood $p(t\,|\,x,\bar{\theta})$ and the prior density $p(\bar{\theta})$, in order to assign the posterior density $p(\bar{\theta}\,|\,x,t)$ to a neural network defined by the point $\bar{\theta}$. $p(t\,|\,x)$ is called evidence and plays the role of a normalizing constant, so we ignore the evidence. That is,

$$\text{Posterior}\propto\text{Likelihood}\times\text{Prior}\,. \tag{2}$$

We consider a class of neural networks defined by the function

$$y_m(x,\bar{\theta})\ =\ \frac{\exp[s_m(x,\bar{\theta})]}{\sum_{k=0}^{N-1}\exp[s_k(x,\bar{\theta})]},\qquad m\ =\ 0,\ 1,\ \cdots,\ N-1\,, \tag{3}$$

where

$$s_k(x,\bar{\theta})\ =\ b_k+\sum_{j=1}^{H}v_{jk}\tanh\left(a_j+\sum_{i=1}^{P}u_{ij}x_i\right),\qquad k\ =\ 0,\ 1,\ \cdots,\ N-1\,. \tag{4}$$

The neural networks have P inputs, a single hidden layer of H hidden nodes and m outputs. In the particular Bayesian neural networks described here, each neural network has the same structure. The parameter u_{ij} and v_{jk} are called the weights and a_j and b_k are called the biases. Both sets of parameters are generally referred to collectively as the weights of the Bayesian neural networks, $\bar{\theta}$. $y_m(x,\bar{\theta})$ is the probability that the event, (x,t), belongs to the mth's class. So the likelihood of n training events is

$$p(t\,|\,x,\bar{\theta})=y_{t_1}y_{t_2}\cdots y_{t_n}=\prod_{i=1}^{n}y_{t_i}\,, \tag{5}$$

where it has been assumed that the events are independent with each other.

We get the likelihood, meanwhile we need the prior to compute the posterior density. But the choice of prior is not obvious. However, experience suggests a reasonable class is the priors of Gaussian class centered at zero, which prefers smaller rather than larger weights, because smaller weights yield smoother fits to data. In the paper, a Gaussian prior is specified for each weight using the Bayesian neural networks package of Radford Neal[1)]. However, the variance for weights belonging to a given group (either input-to-hidden weights (u_{ij}), hidden-biases (a_j), hidden-to-output weights (v_{jk}) or output-biases (b_k)) is chosen to be the same: $\sigma_u^2,\ \sigma_a^2,\ \sigma_v^2,\ \sigma_b^2$, respectively. However, since we don't know, a priori, what these variances should be, their values are allowed to vary over a large range, while favoring small variances. This is done by assigning each variance a gamma prior

$$p(z)=\left(\frac{\alpha}{\mu}\right)^{\alpha}\ \frac{z^{\alpha-1}\mathrm{e}^{-z\frac{\alpha}{\mu}}}{\Gamma(\alpha)}\,, \tag{6}$$

where $z=\sigma^{-2}$, and with the mean μ and shape parameter α set to some fixed plausible values. The gamma prior is referred to as a hyperprior and the parameter of the hyperprior is called a hyperparameter.

Then, the posterior density, $p(\bar{\theta}\ |\ x,t)$, is gotten according to Eqs. (2), (5) and the prior of Gaussian distribution. Given an event with data x', an estimate of the probability that it belongs to the class is given by the weighted average

$$\bar{y}_m(x'|x,t)=\int y_m(x',\bar{\theta})p(\bar{\theta}\,|\,x,t)\mathrm{d}\bar{\theta}\,. \tag{7}$$

Currently, the only way to perform the high dimensional integral in Eq. (7) is to sample the density $p(\bar{\theta}\ |\ x,t)$ with the Markov Chain Marlo Carlo (MCMC) method[2, 6—8]. In the MCMC method, one steps through the $\bar{\theta}$ parameter space in such a way that points are visited with a probability proportional to the posterior density, $p(\bar{\theta}|x,t)$. Points where $p(\bar{\theta}\,|\,x,t)$ is large will be visited more often than points where $p(\bar{\theta}\,|\,x,t)$ is small. Eq. (7) approximates the integral using the average

$$\bar{y}_m(x'\,|\,x,t)\approx\frac{1}{L}\sum_{i=1}^{L}y_m(x',\bar{\theta}_i)\,, \tag{8}$$

where L is the number of points $\bar{\theta}$ sampled from $p(\bar{\theta}|x,t)$. Each point $\bar{\theta}$ corresponds to a different neural network with the same structure. So the average is an average over neural networks, and the probability of the data x' belongs to the mth's class. The average is closer to the real value of $\bar{y}_m(x'\,|\,x,t)$, when L is sufficiently large.

3 Particle identification(PID)

The training data and test samples of π, K, p and $\bar{\text{p}}$ from 0.3 GeV/c to 1.2 GeV/c are generated by the 'tester' generator from a GEANT3-based Monte Carlo (MC) simulation program (SIMBES) with detailed consideration of the detector performance. The consistency between data and Monte Carlo has been

1) R. M. Neal, Software for Flexible Bayesian Modeling and Markov Chain Sampling, http://www.cs.utoronto.ca/~radford/fbm.software.html

checked in many high purity physics channels, and the agreement is reasonable[9]. The requirement of their polar angles is $|\cos\theta| < 0.8$. The MC samples of π, K, p, $\bar{\text{p}}$ are identified using BNN per 100 MeV/c respectively. 6000 events are uniformly generated per 100 MeV/c for π, K, p, $\bar{\text{p}}$, respectively. And 5000 events of them are for the training and 1000 events of them are for the test, respectively.

Ten variables on MDC (Main Drift Chamber), TOF (Time of Flight Counter) and BSC (Barrel Shower Counter) informations in BES II [1] are considered in the particle identification of π, K, p with BNN , and they are as follows:

1) The first variable: the most probable pulse height of dE/dX information from MDC, PHMP;

2) The second to fourth variable: the square value of the deviation between the energy loss of the particle for the ionization and its expectation if the particle is π, K, p, respectively, $(\text{XSPI})^2$, $(\text{XSK})^2$, $(\text{XSP})^2$;

3) The fifth variable: the time of flight of the particle from TOF, T;

4) The sixth to eighth variable: The weight for π, K, p from TOF, respectively, WTPi, WTK,WTP;

5) The ninth variable: the most probable pulse height of TOF, Q;

6) Then tenth variable: the ratio of the deposited energy in BSC and the momentum from MDC, E_{BSC}/P.

The ten variables are also used as inputs to BNN in the particle identification of π, K, $\bar{\text{p}}$.

3.1 PID with BNN

All ten variables are used as inputs to all neural networks, which have the same structure. In the paper, all the networks have the input layer of ten inputs, the single hidden layer of twelve nodes and the output layer of three outputs which are the probabilities of π, K, and p. And which probability is the largest one, then the particle is thought as it. The particle identification is performed per 100 MeV/c with BNN. A Markov chain of neural networks is generated using the Bayesian neural networks package of Radford Neal, with a training sample consisting of 5000 events each of π, K, p, in each process of the particle identification. One thousand iterations, of twenty MCMC steps each, are used. The neural network parameters are stored after each iteration, since the correlation between adjacent steps is very high. That is, the points in neural network parameter space are saved to lessen the correlation after twenty steps here. It is also necessary to discard the initial part of the Markov chain because the correlation between the initial point of the chain and the points of the part is very high. The initial three hundred iterations are discarded here. 1000 events each of π, K, p are used to test the identification capability of the trained BNN per 100 MeV/c from 0.3 GeV/c to 1.2 GeV/c. The particle identification of π, K, $\bar{\text{p}}$ is performed in the same way as the identification of π, K, p.

3.2 PID with the χ^2 analysis of dE/dX and TOF information

In this paper, the results of the particle identification of π, K, p using the χ^2 analysis are from the work done by Qin Hu, et al.[4]. The particle identification of Ref. [4] is performed in the way that the χ^2 analysis of dE/dX and TOF are added with different, not equal, weight according to different momentum region. The results of the method are better than the ones of equal weight, especially the identification of π and K at high momentum region.

4 Results and discussion

The results of the particle identification of π, K, p and $\bar{\text{p}}$ with BNN and the works done by Qin Hu, et al.[4] are shown in Fig. 1, Fig. 2, Fig. 3, Fig. 4, Fig. 5, Fig. 6, respectively. Fig. 1 shows that above 0.8 GeV/c the π identification efficiency is obviously higher while the π misidentification efficiency is obviously lower using BNN than the method in Ref. [4]. But the π identification and misidentification efficiencies are basically invariant below 0.8 GeV/c using BNN and the method in Ref. [4]. Fig. 2 shows the K identification and misidentification efficiencies are obviously better from 0.3 GeV/c to 1.2 GeV/c using BNN than the method in Ref. [4]. Fig. 3 and Fig. 4 show the p identification and misidentification efficiencies are basically invariant using BNN and the method in Ref. [4]. Fig. 5 and Fig. 6 show the $\bar{\text{p}}$ identification and misidentification efficiencies are better below 0.6 GeV/c using BNN than the method in Ref. [4]. In a word, BNN can be well applied to identify π, K, p and $\bar{\text{p}}$ in the BES II experiment, especially to distinguish π from K and identify $\bar{\text{p}}$, BNN is more advantageous than the method of χ^2 analysis and the algorithms derived from it.

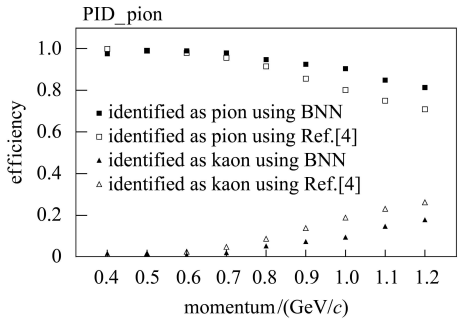

Fig. 1. The π identification and misidentification efficiencies using BNN and the method of Ref. [4], respectively.

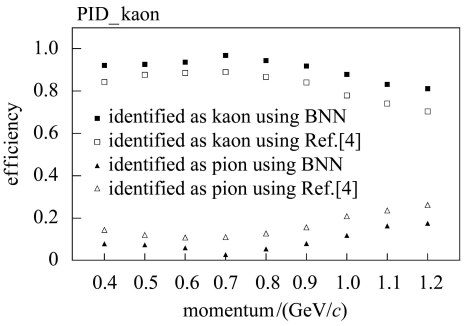

Fig. 2. The K identification and misidentification efficiencies using BNN and the method of Ref. [4], respectively.

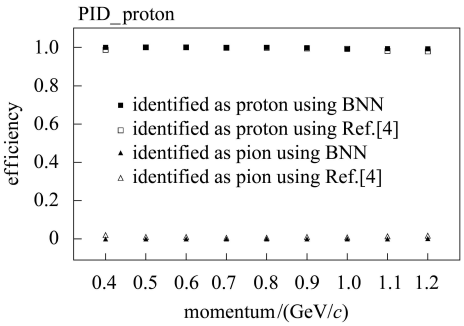

Fig. 3. The p identification and misidentification efficiencies using BNN and the method of Ref. [4], respectively.

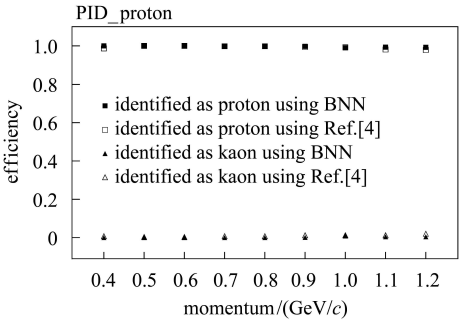

Fig. 4. The p identification and misidentification efficiencies using BNN and the method of Ref. [4], respectively.

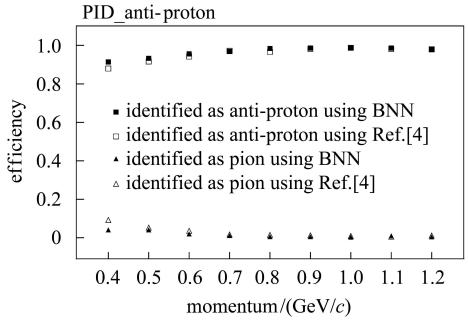

Fig. 5. The $\bar{\mathrm{p}}$ identification and misidentification efficiencies using BNN and the method of Ref. [4], respectively.

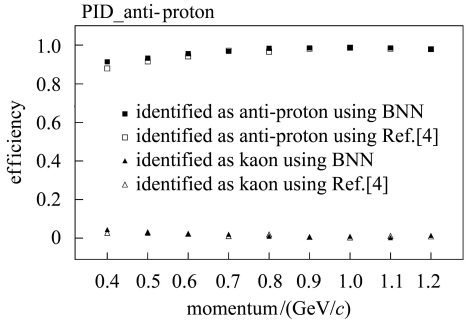

Fig. 6. The $\bar{\mathrm{p}}$ identification and misidentification efficiencies using BNN and the method of Ref. [4], respectively.

Separating π, K and identifying $\bar{\mathrm{p}}$ better than χ^2 analysis and the algorithms derived from it, BNN can be applied to the data analysis of the BES II experiment and the better results of physics can be achieved. Although the tests in this paper are only for the BES II experiment, it is expected that the algorithm of BNN can also be applied to the data analysis of the BESIII experiment (the third generation of Beijing Spectrometer) in the future and will find wide application in the experiments of high energy physics.

We wish to express our gratitude to the BES Collaboration for their excellent work on the Monte Carlo simulation.

References

1 BAI J Z et al. (BES Collaboration). Nucl. Instrum. Methods A, 2001, **458**: 627
2 Neal R M. Bayesian Learning of Neural Networks. New York: Springer-Verlag, 1996
3 Beale R, Jackson T. Neural Computing: An Introduction. New York: Adam Hilger, 1991
4 QIN Hu et al. HEP & NP, 2004, **28**(7): 738—743 (in Chinese)
5 Bhat P C, Prosper H P. Bayesian Neural Networks. In: Lyons L, Unel M K ed. Proceedings of Statistical Problems in Particle Physics, Astrophysics and Cosmology, Oxford, UK 12-15, September 2005. London: Imperial college Press. 2006. 151—154
6 Duane S, Kennedy A D, Pendleton B J et al. Physics Letters B, 1987, **195**: 216—222
7 Creutz M, Gocksch A. Physical Review Letters, 1989, **63**: 9—12
8 Mackenzie P B. Physics Letters B, 1989, **226**: 369—371
9 Ablikim M et al. (BES Collaboration). Nucl. Instrum. Methods A, 2005, **552**: 344