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

Hopfield Neural Network for Seismic Velocity Picking

Kou-Yuan Huang
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
kyhuang@cs.nctu.edu.tw

Jia-Rong Yang
nikeasyanzi@gmail.com

***Abstract*—The Hopfield neural network (HNN) is adopted for velocity picking in the time-velocity semblance image of seismic data. A Lyapunov function in the HNN is set up from the velocity picking problem. We use the gradient descent method to decrease the Lyapunov function and derive the equation of motion. According to the equation of motion, each neuron is updated until no change. The converged network state represents the best polyline in velocity picking. We have experiments on simulated and real seismic data. The picking results are good and close to the human picking results.**

Keywords—Hopfield neural network; seismic velocity picking; semblance image; Lyapunov function; equation of motion

I. INTRODUCTION

Velocity analysis is very important in reflection seismic data processing. Velocity picking was to pick a series of peaks in the seismic time-velocity semblance image (stacking energy). Conventional it was done by geophysical experts and took much time. Some automatic methods had been done on seismic velocity picking [1]-[6]. In 1974, Beitzel and Davis [1] used the minimum spanning tree method to do the velocity picking. This method had to manually choose a skeleton as the final solution from results. In 1992, Schmidt and Hadsell [2] applied the multilayer perceptron (MLP) in velocity picking. They trained two MLP models. Then, they must use another MLP model that was trained by the human picked polyline to validate the polylines. In 1994, Fish and Kasuma [3] also used multilayer perceptron in velocity picking. They used the human picked peaks and the eight neighbors in a semblance image to form training patterns. The MLP was trained until it could approximate the expert picking result. The drawback of the MLP was that the human picking result was needed as the reference to validate the candidate picks. The methods in [1]-[3] needed the human involvement. In 2002, Beveridge et al. [4] took velocity picking as a problem of choosing the best polyline from semblance peaks. They defined an energy function that included inverse energy of picked points and a constraint on average turning angle. They used the steepest descent method to find the polyline. However, the constraint of the interval velocity was not taken into consideration, and that might result in an ineligible solution. In 2012 and 2013, Huang et al. [5], [6] used the simulated annealing and genetic algorithm for seismic velocity picking, but the computations were in random process and the iteration steps were complex and not efficient.

In 1982, Hopfield proposed a recurrent model [7]-[9]. It could solve the combinatorial optimization problems like traveling salesman problem [9] and n-queens problem [10]. On seismic application, it was ever applied on seismic horizon picking [11].

Here we transfer the seismic velocity picking on a semblance image to a combinatorial optimization problem. The Hopfield neural network (HNN) can get the best velocity picking result. The picking result is applied to do the normal move-out (NMO) correction and trace stacking.

II. SEISMIC DATA ACQUISITION AND VELOCITY PICKING

A. Seismic Data Acquisition and Semblance Image

We apply the HNN method on real seismic data at Nankai. We describe its seismic data acquisition [12]-[14]. Nankai is near the coast of Japan over the Nankai trough where the Philippine plate is a subduction beneath Eurasia. The data were collected by the University of Texas, the University of Tulsa, and the University of Tokyo [15]. They used end-on spread to acquire the marine seismic data. Fig. 1 shows the shot and end-on spread to get a one-shot seismogram. In each shot, the spacing of each receiver is also 33.33 m. Then, we move the shot point and receivers at the same time and get the other seismogram. The spacing of each shot, as well as the spacing of each receiver, is 33.33 m. Fig. 2 shows the seismogram of shot 1750 of Nankai data. It has 69 traces and 2,750 samples per trace with sampling interval 0.004 seconds and total 11 seconds.

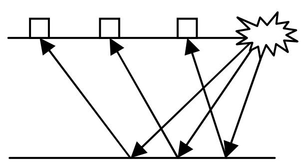


Fig. 1. Shot and receivers at one-shot seismogram.

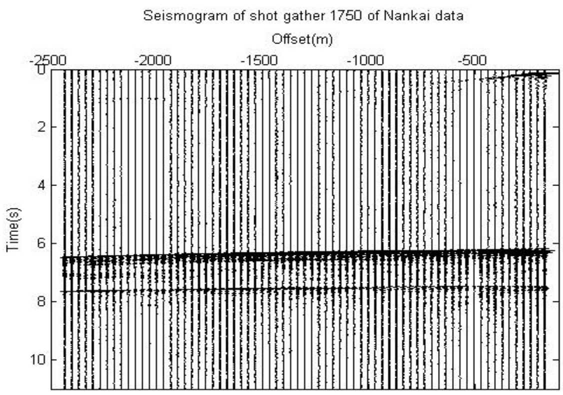


Fig. 2. Seismogram of shot 1750 of Nankai data.

We collect those traces with the same reflection point to become a common depth point (CDP) gather, also called common mid-point (CMP) gather as shown in Fig. 3 and 4 [12], [13]. Fig. 5 shows the CMP 933 gather of Nankai data. In order to find the correct velocities for NMO correction, we have to construct the semblance image of the CMP gather and perform velocity picking. In the generation of semblance image from CMP gather, each pixel in a seismic semblance image is the energy of all corrected traces in the CMP gather at the certain time and certain velocity. The range of stacking velocity is from 1000 m/s to 7000 m/s and the velocity sampling interval is 25 m/s. Fig. 6 shows the semblance image of CMP 933. We do the velocity picking on the it. The result of velocity picking on Fig. 6 is used to find the correct time-velocity points and polyline for the NMO correction and trace stacking. Fig. 7 shows the NMO correction on traces of CMP gather. The purpose of NMO correction is to correct the offset difference of the reflection signal to vertical reflection. Stacking those corrected traces to become one trace can enhance signal to noise ratio.

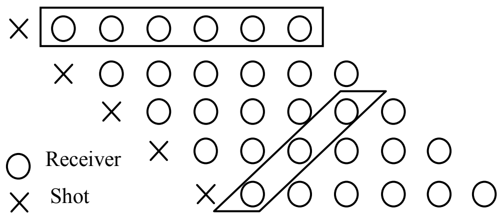


Fig. 3. Stack chart of selected traces of a CMP gather from seismograms.

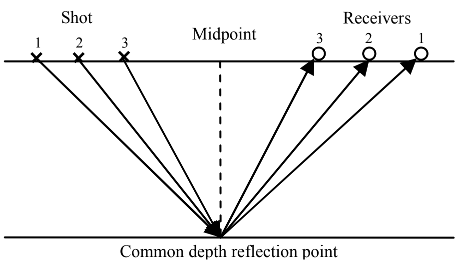


Fig. 4. Shots and receivers geometry on a CMP gather.

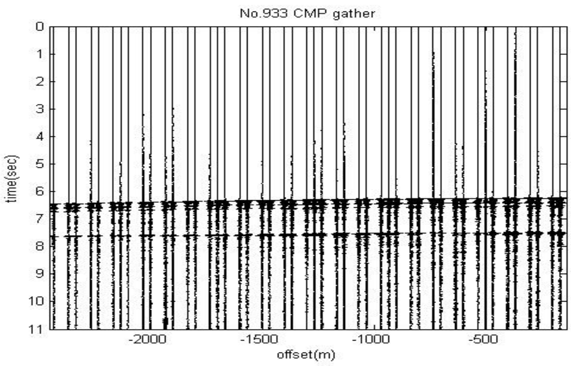


Fig. 5. CMP 933 gather of Nankai data.

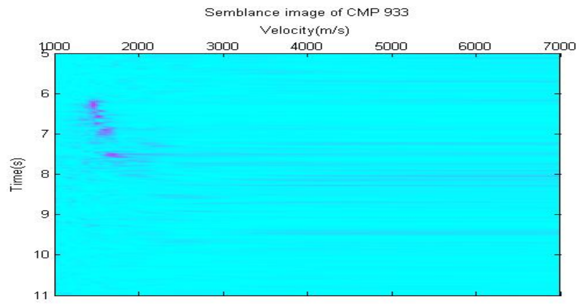


Fig. 6. Semblance image of CMP 933

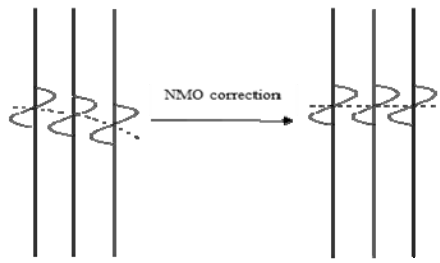
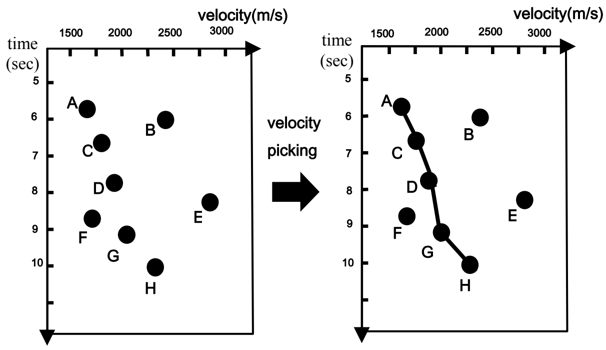


Fig. 7. NMO correction on traces of CMP gather.

B. Seismic Velocity Picking

The energy of each pixel in a seismic semblance image is a normalized coherence measure of the traces between 0 and 1 [12]-[14]. Velocity picking is to pick several peak points and get the best time-velocity polyline. The example is shown below. Fig. 8 illustrates eight peaks in the semblance image. The local peaks in time-velocity seismic semblance image are ordered in a sequence with time first, then velocity. Eight peaks are ordered in A, B, …, and H. The point with earlier time links to the point with later time, and they become a polyline. There are many possibilities in polylines. For example, in Fig. 8, A→C→D→G→H is a possible solution for velocity picking. The value of picked point is 1, otherwise 0. The state is [1 0 1 1 0 0 1 1]. We connect the picked point with value 1 as the picking result. Usually the velocity in deeper layer is increased.



A	B	C	D	E	F	G	H
1	0	1	1	0	0	1	1

Polyline: A->C->D->G->H

Fig. 8. Example of seismic velocity picking.

III. Seismic Velocity Picking By The Hopfield Model

The discrete HNN is shown in Fig. 9. Every neuron receives the inputs from other neurons, then after activation function it sends the output to other neurons. Once the initial state of the network is set, each neuron is updated according to the equation of motion. The procedure is repeated until the network has no change and converges to a steady state. In this application each peak point is represented by one neuron. The neuron state is 1 or 0. 1 represents that the peak point is selected in a polyline.

Fig. 10 shows the proposed steps of velocity picking by the HNN. At first, we get the local peak points on seismic time-velocity semblance image by peak detection. Then, we choose several points with higher semblance values as candidate peak points. Next, we set up a Lyapunov function from the velocity picking problem. In order to decrease the Lyapunov function, the equation of motion is derived that the function can reach the minimum. According to the equation of motion, each neuron is updated until no change. The converged network state forms the best polyline that represents the best velocity picking result.

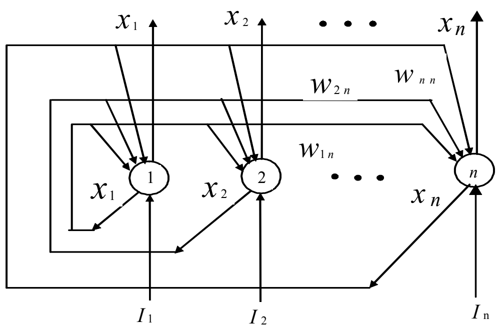


Fig. 9. Discrete Hopfield neural network.

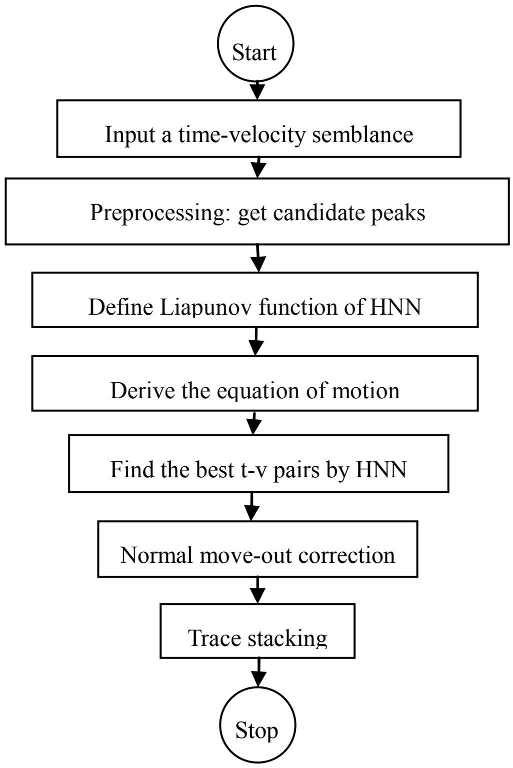


Fig. 10. Steps of velocity picking by Hopfield neural network.

A. Preprocessing

For a semblance image, we use a 5-by-5 window to get the peak points. The semblance value of a point is compared with its neighbors. The point is a peak point if it has the largest semblance value. We move the window from left to right and top to bottom to find the peak points. Then, we choose the top Q points with higher semblance values as the candidate points. The candidate points are arranged with time first, then velocity. So there are Q neurons in the HNN. And K points are selected from the Q candidate points as the result of velocity picking. We use a vector $\mathbf{x} = [x_1, x_2, \ldots, x_Q]^T$ to represent a state of HNN where x_i is 1 if the i^{th} point is picked and 0 otherwise. The points with value 1 are linked as a best polyline.

B. Lyapunov Function and Equation of Motion

The Lyapunov function of $\mathbf{x}$ is set up as follows.

$$E'(\mathbf{x}) = -\alpha_p E_p(\mathbf{x}) + \alpha_{npts} E_{npts}(\mathbf{x}) + \alpha_{vi} E_{vi}(\mathbf{x}) + \alpha_{vs} E_{vs}(\mathbf{x}) \quad (1)$$

where α_p, α_{npts} , α_{vi} , and α_{vs} are the positive parameters.

The first term in the Lyapunov function is the total semblance value of the picked points.

$$E_p(\mathbf{x}) = \sum_{i=1}^{Q} x_i p(x_i)$$

where $p(x_i)$ is the semblance value of the i^{th} peak point.

The second term is the constraint on total number of K picked points.

$$E_{npts}(\mathbf{x}) = (sum(\mathbf{x}) - K)^2$$

where K is the predefined number of picked points and $K \leq Q$.

The third and fourth terms are:

$$E_{vi}(\mathbf{x}) = \alpha_{vi} \sum_i \sum_{j,j>i} x_i x_j ppb(x_i, x_j) csvi(x_i, x_j)$$

$$E_{vs}(\mathbf{x}) = \alpha_{vs} \sum_i \sum_{j,j>i} x_i x_j ppb(x_i, x_j) csvs(x_i, x_j)$$

We consider the constraints on interval velocity ($vi_{n-1,n}$) and velocity slope ($vs_{n-1,n}$) to remove the ineligible polyline as in [1]. $E_{vi}(\mathbf{x})$ is to calculate the total violation times of $\mathbf{x}$ on the interval velocity constraint, and $E_{vs}(\mathbf{x})$ is to calculate the total violation times of $\mathbf{x}$ on the velocity slope constraint. $ppb(x_i, x_j)$ is defined to check whether there is no any picked point between x_i and x_j. $ppb(x_i, x_j)=1$ when there is no any picked point between x_i and x_j. Otherwise, $ppb(x_i, x_j)=0$.

The interval velocity, which is related to Dix's equation, is used to restrict the calculated interval velocity [13], [14]. Also we define the slope of stacking velocity. They are defined as

$$vi_{n-1,n} = \sqrt{\left(t_n v_n^2 - t_{n-1} v_{n-1}^2\right) / \left(t_n - t_{n-1}\right)} \quad (2)$$

$$vs_{n-1,n} = (v_n - v_{n-1}) / (t_n - t_{n-1}) \quad (3)$$

where t_{n-1} and t_n are the two-way vertical travel time of layer $n-1$ and n respectively, and v_{n-1} and v_n are the stacking velocity of layer $n-1$ and n respectively. Penalties of constrains for interval velocity ($csvi$) and velocity slope ($csvs$) are

$$csvi(vi_{n-1,n}) = \begin{cases} 0, VI_{\min} \leq vi_{n-1,n} \leq VI_{\max} \\ 1, \quad \text{otherwise} \end{cases} \quad (4)$$

$$csvs(vs_{n-1,n}) = \begin{cases} 0, VS_{\min} \leq vs_{n-1,n} \leq VS_{\max} \\ 1, \quad \text{otherwise} \end{cases} \quad (5)$$

where VI_{min}, VI_{max}, VS_{min}, and VS_{max} are the predefined values for constraints.

We can set the values of α_p, α_{npts}, α_{vi}, and α_{vs} to be 1 that each term of energy has equal weight. Therefore, if two picked points possess the largest semblance value, their total semblance value is 1+1=2. However, if they also violate the two constraints, interval velocity and velocity slope, the total penalty of violation times is also 1+1=2. Therefore, the sum of semblance values, $E_p(\mathbf{x})$, counteracts the penalty of total violation times of the two constraints, $E_{vi}(\mathbf{x})$ and $E_{vs}(\mathbf{x})$. i.e., $E'(\mathbf{x}) = 0$. From the equal weights, we can explain that the first term and sum of the third and fourth violation terms are equally important.

We remove K^2 from (1).

$$E(\mathbf{x}) = E'(\mathbf{x}) - K^2$$

$$= 2\alpha_{npts} \sum_i \sum_{j,j>i} x_i x_j$$
$$+ \sum_i \sum_{j,j>i} x_i x_j ppb(x_i, x_j)(\alpha_{vi} csvi(x_i, x_j) + \alpha_{vs} csvs(x_i, x_j))$$
$$+ \sum_i x_i(-\alpha_p p(x_i) - \alpha_{npts}(2K-1))$$

We calculate the energy difference ΔE caused by disturbing the neuron x_n from $x_n(t)$ to $x_n(t+1)$, where $x_i = x_n$ and $x_j = x_n$, and i and j are from 1 to Q. The value of each neuron is 1 or 0. We can derive

$$\Delta E = E(\mathbf{x}^{new}) - E(\mathbf{x}) = -\Delta x_n g(x_n(t))$$

where

$$g(x_n(t)) = -2\alpha_{npts} \sum_{i<n} x_i - 2\alpha_{npts} \sum_{j>n} x_j$$
$$- \sum_{i<n} x_i ppb(x_i, x_n(t))(\alpha_{vi} csvi(x_i, x_n(t)) + \alpha_{vs} csvs(x_i, x_n(t)))$$
$$- \sum_{j>n} x_j ppb(x_n(t), x_j)(\alpha_{vi} csvi(x_n(t), x_j) + \alpha_{vs} csvs(x_n(t), x_j))$$
$$+ \sum_{i<n} \sum_{j>n} x_i x_j ppb(x_i, x_n(t)) ppb(x_n(t), x_j)(\alpha_{vi} csvi(x_i, x_j) + \alpha_{vs} csvs(x_i, x_j))$$
$$+ \alpha_p p(x_n(t)) + \alpha_{npts}(2K-1)) \quad (6)$$

In order to decrease E, $\Delta E \leq 0$, we can get

$$x_n(t+1) = f_h(g(x_n(t))) \quad (7)$$

where f_h () is a hard-limiter function, i.e.,

$$x_n(t+1) = f_h = \begin{cases} 1, \text{ if } g(x_n(t)) > 0 \\ 0, \text{ if } g(x_n(t)) < 0 \end{cases}$$

Equation (7) becomes the equation of motion to change neuron value. The disturbance of a neuron is the gradient descent method. Because of $\Delta E \leq 0$, the E can reach the minimum.

Using the equation of motion, the algorithm of the HNN for seismic velocity picking is as follows.

Algorithm 1: The HNN for seismic velocity picking.

Input: Q candidate peak points and their values from a seismic velocity semblance image. Set $\mathbf{x} = [x_1, x_2, ..., x_Q]^T$. Each x_i represents one neuron state and also a corresponding peak point. The value of x_i is 0 or 1. Set K as the number of picked points from Q points.

Output: Vector $\mathrm{x} = [x_1, x_2, \ldots, x_Q]^T$ corresponding to the lowest Lyapunov function value as the optimal picking result.

Step 1: Initialization.

Set up a random initial state of network $\mathbf{x} = [x_1, x_2, ..., x_Q]^T$ with network size Q. Each neuron x_n is an element of vector $\mathbf{x}$ and has a state value either 0 or 1.

Step 2: Update neurons.

1. Calculate $g(x_n(t))$ in (6), where $x_n(t)$ is the neuron state value at time t.
2. Use the equation of motion in (7), $x_n(t+1) = f_h(g(x_n(t)))$ to update the next state value of neuron $x_n(t+1)$, where n = 1, 2, …, Q.

Step 3: Repeat step and check termination

Go to Step 2 to update each neuron until the network is stable, i.e., the neurons have no change. The final state of network is the solution of velocity picking.

On semblance image we do the velocity picking by the HNN. In order to evaluate the performance, we compare the picking polyline result by the HNN with that by human. By linear interpolation, we calculate the corresponding velocities of the two polylines at each time sample. Then, the average absolute difference of velocity (V_{diff}) can be calculated as

$$V_{diff} = \frac{1}{N}\sum_{t}^{N} | V_{HNN}(t) - V_{human}(t) | \tag{8}$$

where N is the number of calculated time samples, $V_{HNN}(t)$ and $V_{human}(t)$ are the velocity picked by the HNN and human.

IV. Experimental Results

A. Experimental Results on Simulated Data

1) Simulation Data and Preprocessing:

We use Seismic Un*x system [14] to generate a geological model with twenty layers. The model is shown in Fig. 11. It has a bright spot structure [12]. At the eleventh, twelfth, thirteenth, and fourteenth layers, they are shale, gas-sand, oil-sand, and water-sand layers. Each layer has the interval velocity. We simulate the seismic data acquisition. There are 40 shots. For each shot, there are 60 receivers for two-side split spread. The spacing of each shot is 50 m. The spacing of each receiver is 50 m. The shot location ranges from location 2.5 km to 4.45 km. The receiver location ranges from location 1.025 km to 5.925 km. There are 40 one-shot seismograms. The sampling interval is 0.004 seconds and total is 6 seconds.

We rearrange traces of one-shot seismograms to CMP gathers. The full fold number is 30. There are 22 full-folded CMP gathers. We choose the CMP 70 gather for experiment because it is full-folded and located in the middle of the geologic model in Fig. 11. Fig. 12 shows the CMP gather 70. Then we generate the semblance images from the 22 CMP gathers. Fig. 13 shows the semblance image of CMP gather 70.

Before doing velocity picking, local peak detection is used to find the peak points in the semblance image. When doing local peak detection, we use a window size 5 by 5 to decide whether a point is a peak point or not. After peak detection we select top 50 points with higher semblance values as the candidate points in the experiment. In Fig. 13, the 50 selected candidate points are on it.

Because there are 20 geological layers, we manually pick 20 points to form a polyline on the semblance image of CMP gather 70. Therefore, the vector length Q is 50 and K is 20. The human picking result is shown in Fig. 14. But here we use the HNN to pick the points for comparison.

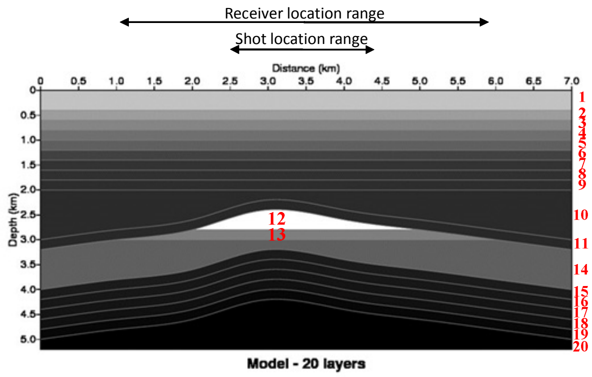


Fig. 11. Simulated 20 layer geological model.

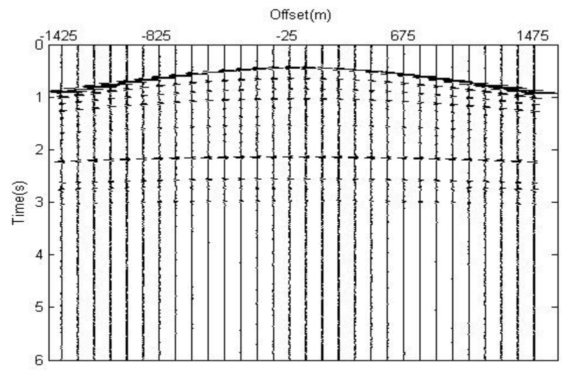


Fig. 12. CMP gather 70.

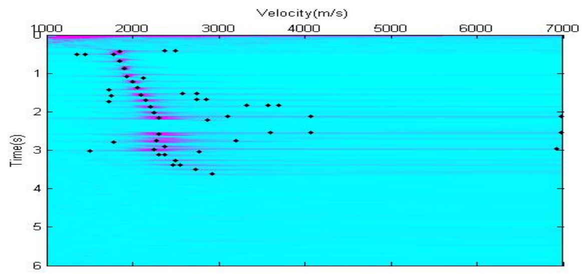


Fig. 13. 50 candidate points (Q = 50) on semblance image of CMP gather 70.

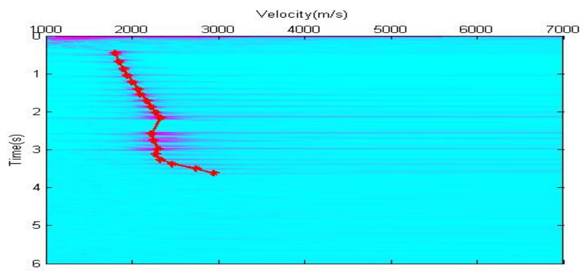


Fig. 14. Human picking result on semblance image of CMP gather 70.

2) Results of Simulation Experiments:

We use the equation of motion in (7) to do velocity picking by the HNN. We set the total peaks Q = 50, the number of picked points K = 20, α_p =1, α_{npts} =1, α_{vi} =1, and α_{vs} =1, the ranges of interval velocity and velocity slope constraints: VI_{min}=1000, VI_{max}=7000, VS_{min}= -100, VS_{max}=1000 [12]-[14]. Within the ranges, the interval velocity and velocity slope have meaning.

We do velocity picking on CMP gather 70 with 2000 experiments. For each experiment, we calculate the V_{diff} by (8). After 2000 experiments, we calculate the mean of V_{diff}. Finally we use the best velocity picking result to do the NMO correction and stacking.

We show the best result on CMP 70 in 2000 experiments. Fig. 15 shows the energy versus iteration of the best result. The network converges to a stable state after 453 iterations. Fig. 16 shows the best velocity picking result by the HNN. The black dots and line is the picking result by the HNN, and the red-cross symbols and line is picking result by human. Fig. 17 (a) and (b) show the results of NMO correction, and the stacked trace on CMP gather 70 using the best velocity picking result. In Fig. 17(b), the stacked signal is enhanced.

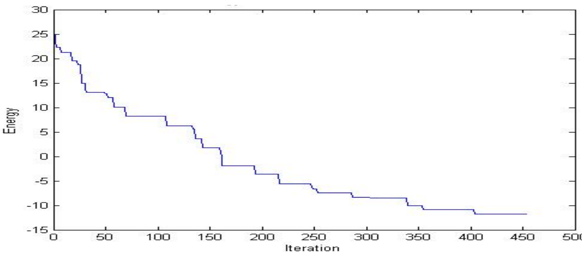


Fig. 15. Energy versus iteration of best result by HNN on CMP gather 70.

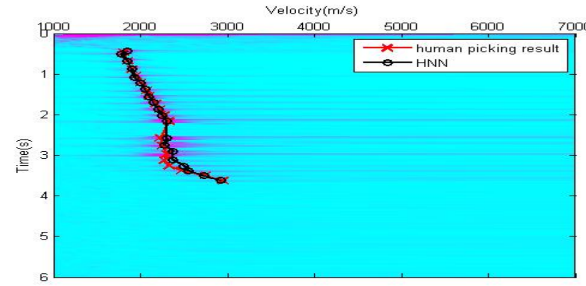
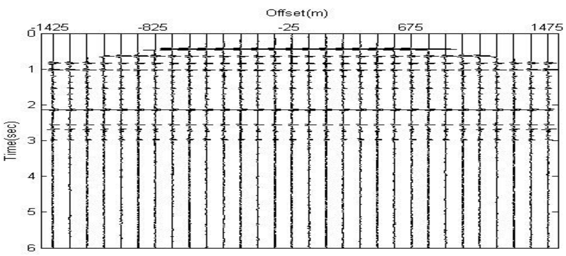
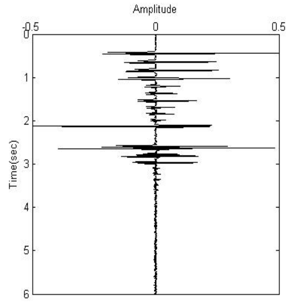


Fig. 16. Best velocity picking by HNN with black line on CMP gather 70.



(a)



(b)

Fig. 17. NMO correction and stacking result of CMP gather 70 by HNN using best picking result. (a) NMO correction, (b) stacked trace.

B. Experimental Results on Nankai Real Data

The data acquisition and semblance image on Nankai real data are described in the previous section. We use the HNN method on the semblance images of 15 CMP gathers: CMP 933, 958, 983, 1008, 1033, 1058, 1083, 1108, 1133, 1158, 1183, 1208, 1233, 1258, and 1283. They have the high folding trace number at CMP gather that can enhance signal to noise ratio. Fig. 18 shows the CMP gather 1233. Fig. 19 shows its semblance image.

On the semblance image of each CMP gather we have the number of human picked points [12], [15]. We use it as the number of picked points K. For example, for CMP gather 1233, K=3. The number of selected peaks is Q=50.

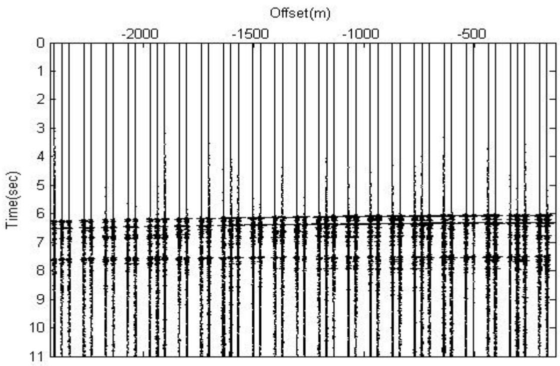


Fig. 18. CMP gather 1233.

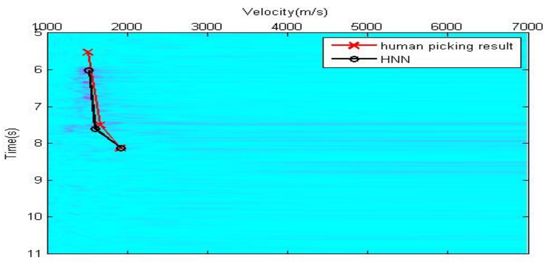


Fig.19. Semblance image of CMP gather 1233 and picking results by human and the HNN.

Because it is the marine seismic data it has constant velocity in water layer between 0 sec and 5.5 sec, there is no seismic event. We just perform velocity picking between 5.5 sec and 11 sec in the semblance image. We get the mean of V_{diff} calculated from the velocity picking results on each CMP gather by the HNN and human. The smallest mean of V_{diff} among the semblance image of 15 CMP gathers is CMP gather 1233. Q=50, K=3. We show the best experiment result in 2000 experiments on CMP gather 1233. Fig. 20 shows the energy versus iteration of the best result. The network converges to a stable state after 94 iterations. The best velocity picking result by the HNN is shown in Fig. 19 with the black dots and line. The velocity picking result by human is also shown in Fig. 19 with the red-cross symbol and line. We can compare the picking results with human picking and the HNN by (8). The gather after NMO correction and the stacking result using the best velocity picking result on CMP gather 1233 are shown in Fig. 21 (a) and (b) respectively. In Fig. 21(b), the stacked signal is enhanced.

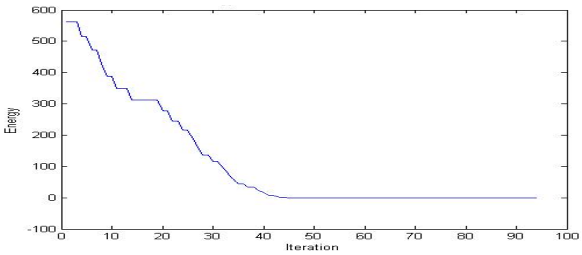
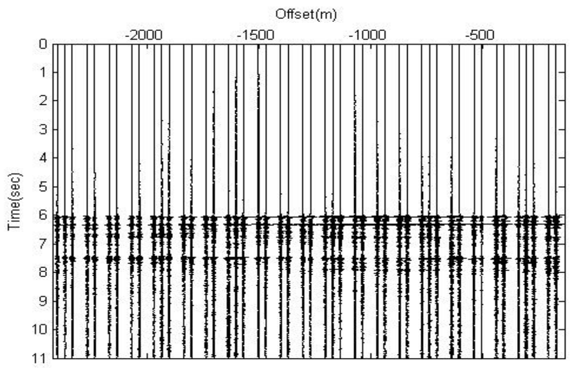
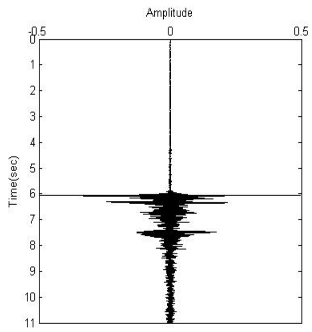


Fig. 20. Energy versus iteration of best result by HNN on CMP gather 1233.



(a)



(b)

Fig. 21. NMO correction and stacking result of CMP gather 1233 by HNN using best picking result. (a) NMO correction, (b) stacked trace.

A. Comparison of Mean of CPU time

We also use the simulated annealing (SA) and genetic algorithm (GA) on the semblance images of simulation data and real data with 2000 experiments respectively. The computer we used is ACER VERITON M670 with 6 GB RAM, and the programing language is Matlab 7.12.0 (R2011a). Table I shows the mean of CPU time of finding the best polyline by three methods on CMP gather 70. Table II shows the mean of CPU time using three methods on CMP gather 1233 of real data. The HNN has the shortest CPU time.

Table I. Mean of CPU time on CMP gather 70 by three methods.

	SA	GA	HNN
Mean of CPU time (sec)	1.76	3.46	0.07

Table II. Mean of CPU time on CMP gather 1233 by three methods.

	SA	GA	HNN
Mean of CPU time (sec)	3.04	2.97	0.02

V. Conclusions And Discussions

The Hopfield neural network (HNN) is adopted for velocity picking in the time-velocity semblance image of seismic data. Each candidate peak on the semblance image is assigned to each neuron. The most important of the HNN is to set up a Lyapunov function from the velocity picking problem that includes the total semblance values of picked points, and constraints on the number of picked points, interval velocity, and velocity slope in a seismic time-velocity semblance image. We use the gradient descent method to decrease the Lyapunov function and derive the equation of motion. The Lyapunov function can reach the minimum. According to the equation of motion, each neuron is updated until no change. The converged network state represents the best polyline. We have experiments on simulated and real seismic data. The picking results are good and close to the human picking results. The best picking results are used for the normal move-out (NMO) correction and stacking. The stacking results show that the signals are enhanced. This method can improve the seismic data processing and interpretation.

In the comparison with two optimization methods: simulated annealing and genetic algorithm, the HNN has the shortest CPU time.

To find the optimization in (1) of the Lyapunov function, the parameters can be varied and tested by the experiments as in [9]. The HNN gets stuck in local minima. We may use other methods for solving combinatorial optimization problem [16], [17] and make a comparison in the experiments.

Acknowledgment

This work was supported in part by the National Science Council, Taiwan, under Grant NSC 101-2221-E-009-147 and NSC 102-2221-E-009-165.

REFERENCES

[1] J. E. Beitzel and J. M. Davis, "A computer oriented velocity analysis interpretation technique," *Geophysics*, vol. 39, pp. 619-632, 1974.

[2] J. Schmidt and F. A. Hadsell, "Neural network stacking velocity picking," *SEG Technical Program Expanded Abstracts*, vol. 11, pp. 18-21, 1992.

[3] B. C. Fish and T. Kusuma, "A neural network approach to automate velocity picking," *SEG Technical Program Expanded Abstracts*, vol. 13. pp. 185-188, 1994.

[4] J. R. Beveridge, C. Ross, D. Whitely, and B. Fish, "Augmented geophysical data interpretation through automated velocity picking in semblance velocity images," *Machine Vision and Applications*, vol. 13, pp. 141-148, 2002.

[5] Kou-Yuan Huang, Kai-Ju Chen, Jia-Rong Yang, "Seismic velocity picking by simulated annealing," *IEEE International Geoscience and Remote Sensing Symposium* (*IGARSS)*, Munich, Germany, pp.6079-6082, Jul. 22-27, 2012.

[6] Kou-Yuan Huang, Kai-Ju Chen, and Jia-Rong Yang, "Genetic algorithm for seismic velocity picking," *International Joint Conference on Neural Networks* (*IJCNN*), Dallas, Texas, USA, pp.2722-2729, Aug. 4-9, 2013.

[7] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences,* vol. 79, pp. 2554-2558, 1982.

[8] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences,* vol. 81, pp. 3088-3092, 1984.

[9] J. J. Hopfield and D. W. Tank, ""Neural" computation of decisions in optimization problems," *Biological Cybernetics,* vol. 52, pp. 141-152, 1985.

[10] J. Mańdziuk, "Solving the n-queens problem with a binary Hopfield-type network," *Biological Cybernetics,* vol. 72, pp. 439-445, 1995.

[11] K. Y. Huang, "Neural network for seismic horizon picking," *International Joint Conference on Neural Networks Proceedings,* vol. 3, pp. 1840-1844, 1998.

[12] M. B. Dobrin and C. H. Savit, *Introduction to Geophysical Prospecting*: McGraw-Hill Book Co., 1988.

[13] Ö. Yilmaz and S. M. Doherty, *Seismic data analysis: processing, inversion, and interpretation of seismic data*: Society of Exploration Geophysicists, 2001.

[14] D. Forel, T. Benz, and W. D. Pennington, *Seismic data processing with Seismic Un*x: a 2D seismic data processing primer*, Society of Exploration Geophysicists, pp. 11-1 - pp. 11-11, 2005.

[15] G. F. Moore, T. H. Shipley, P. L. Stoffa, D. E. Karig, A. Taira, S. Kuramoto, H. Tokuyama, and K. Suyehiro, "Structure of the Nankai trough accretionary zone from multichannel seismic reflection data," *Journal of Geophysical Research,* vol. 95, pp. 8753-8765, 1990.

[16] L. N. Chen and K. Aihara, "Chaotic Simulated Annealing by a Neural Network Model with Transient Chaos," *Neural Networks*, vol. 8, no. 6, pp. 915-930, 1995.

[17] L. P. Wang, Sa Li, Fuyu Tian, and Xiuju Fu, "A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing", *IEEE Trans. System, Man, Cybern, Part B-Cybernetics*, vol.34, no.5, pp.2119-2125, October, 2004.