ELSEVIER

Computers and Chemistry 25 (2001) 541–550

Computers & Chemistry

www.elsevier.com/locate/compchem

Prior-knowledge-based feedforward network simulation of true boiling point curve of crude oil

Chong-wei Chen, De-zhao Chen *

Department of Chemical Engineering, Zhejiang University, Hangzhou 310027, People's Republic of China

Received 28 August 2000; received in revised form 10 October 2000; accepted 12 December 2000

Abstract

Theoretical results and practical experience indicate that feedforward networks can approximate a wide class of functional relationships very well. This property is exploited in modeling chemical processes. Given finite and noisy training data, it is important to encode the prior knowledge in neural networks to improve the fit precision and the prediction ability of the model. In this paper, as to the three-layer feedforward networks and the monotonic constraint, the unconstrained method, Joerding's penalty function method, the interpolation method, and the constrained optimization method are analyzed first. Then two novel methods, the exponential weight method and the adaptive method, are proposed. These methods are applied in simulating the true boiling point curve of a crude oil with the condition of increasing monotonicity. The simulation experimental results show that the network models trained by the novel methods are good at approximating the actual process. Finally, all these methods are discussed and compared with each other.

Keywords: Prior knowledge; Feedforward network; Exponential weight method; Adaptive method; True boiling point curve of crude oil

1. Introduction

Three-layer feedforward networks have been proved to be universal function approximators (Hornik et al., 1989; Hornik et al. 1990). The ability to approximate arbitrarily complex functions has been exploited in modeling chemical processes. It approximates the corresponding model by using the sample data to train the network, adjusting its interconnect weights to maximize some goodness-of-fit criterion. It does not need to know the detailed knowledge of the underlying process in modeling.

However, there may be some special properties in the actual chemical process, such as monotonicity and concavity, called the prior knowledge of the model. Although Gallant and White (1982) have shown that the arbitrarily large data can make the three-layer feedforward networks approximate a given piecewise differentiable function with an arbitrarily small error, the actual training data are finite. Moreover, in many cases the data set is small and noisy due to limitations of the technology. The networks trained by the finite, sparse and noisy data may be overfitting, thus hampering the accuracy of the model and even violating the prior knowledge, because it relies completely on the data. So the trained model will not predict well, and its practicability is affected.

To solve this problem, some researchers, such as Joerding and Meador (1991) and Thompson and Kramer (1994), are investigating methods to include prior knowledge into neural networks. These methods exploit the corresponding prior knowledge while training the networks with sample data. So the trained networks will not violate the prior knowledge, and the

* Corresponding author.
E-mail address: dzchen@mail.hz.zj.cn (D.-z. Chen).

0097-8485/01/$ - see front matter
PII: S0097-8485(00)00116-9

prediction ability of the model is improved. In addition, because of the decrease of the training space, the efficiency of training may increase.

The difficulty of the methods is how to include the prior knowledge and the sample data into the neural networks simultaneously and effectively. Joerding and Meador (1991) suggested two general methods, the AC method and the WC method. Chen et al. (2000) have introduced another two effective methods. In this paper, focusing on the three-layer feedforward networks and the prior knowledge of monotonic constraint, two novel methods, the exponential weight (EW) method and the adaptive (AP) method, are proposed. Then, the two novel methods, as well as the existing ones, are applied to the simulation of the true boiling point curve of crude oil and compared with each other. The results indicate that the novel methods are feasible and effective.

2. Neural network model of the true boiling point curve of crude oil

Crude oil is a very complicated mixture, mainly containing different kinds of hydrocarbons, organic sulfur, nitrogen and oxygen compounds and trace inorganic compounds. Each component has a different boiling point. Usually, we plot the true boiling point curve of the crude oil by taking the mass percentage p of the distilled component as the x-axis, and the distilled temperature t as the y-axis. The curve can reflect the composition of the distilled crude oil. Therefore, to build a model that takes p as the independent variable and t as the dependent variable is an important problem in the petrochemical industry (Hu and Tang, 1987).

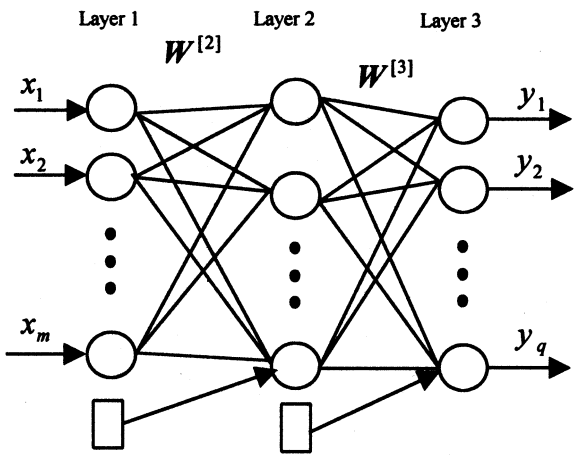


Fig. 1. Three-layer feedforward network.

Usually, we can build nonparametric models, such as splines and neural networks, on the sample data. Moreover, we also know that t is monotonically increasing in p over its domain, which is the prior knowledge of the model. Hu (1983) used the spline function to approximate the true boiling point curve. The method is comparatively accurate, but complicated in computation, and it cannot guarantee monotonicity in the domain. In this paper, we will use neural networks to simulate the true boiling point curve of crude oil and develop new methods to introduce prior knowledge of the monotonicity constraint into the network.

2.1. The structure of the feedforward network

The three-layer feedforward network *net* used in this paper is illustrated in Fig. 1. The first layer is the input layer, which accepts an m-dimension input vector $\boldsymbol{x}$. The second layer has n hidden nodes, which use the following tansig function $f(x)$ to process the weighted input:

$$f(x) = \frac{2}{1 + \mathrm{e}^{-2x}} - 1 \tag{1}$$

The third layer is the output layer, which has q nodes and uses identical function as the process function. The output vector is a q-dimension vector $\hat{\boldsymbol{y}}$, which is the prediction of the target output $\boldsymbol{y}$. $\boldsymbol{W}^{[2]}$ and $\boldsymbol{W}^{[3]}$ are the weight matrices between layers one and two, and two and three, respectively, including the biases. $w_{ij}^{[2]}$, the element of $\boldsymbol{W}^{[2]}$, denotes the interconnect weight between the jth node in layer 1 and the ith node in layer 2. $w_{ki}^{[3]}$, the element of $\boldsymbol{W}^{[3]}$, denotes the interconnect weight between the ith node in layer 2 and the kth node in layer 3. $b_i^{[2]}$ denotes the bias to the ith node in layer 2. $b_j^{[3]}$ denotes the bias to the jth node in layer 3. The function expressed by the network *net* is:

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{W}^{[3]} f(\boldsymbol{W}^{[2]} \boldsymbol{x}) \tag{2}$$

For the sake of brevity, some of the formulas in the following text will replace $\boldsymbol{W}^{[2]}$ and $\boldsymbol{W}^{[3]}$ with $\boldsymbol{W}$.

As to modeling the true boiling point curve of a crude oil, we use a full-connection network in which the number of the input nodes, the hidden nodes and the output nodes of the network are $m = 1$, $n = 7$ and $q = 1$, respectively. In addition, each hidden and output node has an input bias. Therefore, the number of connection weights between layers 1 and 2, and 2 and 3 are both seven. Including the input bias for the nodes, the total number of weights in the network is $1 \times 7 + 7 \times 1 + 7 + 1 = 22$. In the above-mentioned case, the input and output vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are all one-dimensional, so we will use scalars x and y to denote them in the following text.

Table 1
Data for the true boiling point of a crude oil

No.	p (%)	t (°C)
1	0	75
2	1.53	90.32
3	2.34	107
4	3.46	129
5	4.40	143
6	5.36	165
7	6.79	191
8	8.21	210
9	9.76	230
10	11.24	246
11	12.78	260
12	15.03	280
13	17.19	295
14	19.59	312
15	21.09	321
16	23.58	339
17	24.75	350
18	26.34	365
19	28.05	380
20	30.71	403
21	34.18	439
22	35.57	467
23	40.95	480
24	44.50	481
25	47.89	482
26	50.21	490
27	51.73	500
28	54.42	510
29	57.32	512
30	59.77	513

2.2. Sample data of a kind of crude oil

Distillation data for a typical crude oil are listed in Table 1. They will be used as the sample data in the simulation. Here, the mass percentage p corresponds to x, and the distillation temperature t corresponds to y. The size of the sample set is 30. In addition, t is normalized to

$$\frac{t-\min\{t\}}{2(\max\{t\}-\min\{t\})}$$

before further processing in order to speed up the training process.

From Table 1 we can see the sample data t is monotonically increasing versus p. So the training data have already satisfied the prior knowledge. Our goal is to prevent the network model relying on the training data from violating the prior knowledge.

2.3. Training algorithm and the network parameters

All the training methods in this paper are based on the Levenberg–Marquardt (LM) algorithm (Bazaraa and Shetty, 1979). We assume the absolute error is $e = y_k - \hat{y}(x_k)$ at the training datum (x_k, y_k). When the performance function is the summed square error $\mathrm{SSE} = \Sigma_i(y_i - \hat{y}(\boldsymbol{x}_i))^2$, in which the vector x_i should be replaced with a scalar x_i.

$$\Delta \boldsymbol{W} = -e\cdot(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} + \mu \boldsymbol{I})^{-1}\boldsymbol{J}^{\mathrm{T}} \tag{3}$$

where $\boldsymbol{J}$ is the Jacobian matrix, which contains the first derivations of the network errors with respect to the weights and biases; μ is a modulatory parameter. When μ is zero, it is just Newton's method, using the approximate Hessian matrix. When μ is very large, it becomes a gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will never be increased at each iteration step of the algorithm.

The training parameters for the LM algorithm are: the performance function PE, the lower limit of the performance function *goal*, the lower limit of the gradient *grad*, and the maximum number of iterations *epochs*. The training will stop if the number of iterations exceeds *epochs*, if the performance function drops below *goal*, or if the magnitude of the gradient is less than *grad*. As for the problem of the true boiling point curve of a crude oil, all the methods introduced in the following text will take the follow training parameters if there are no special explanation: $\mathrm{PE} = \mathrm{SSE} = \Sigma_l(y_l - \hat{y}(x_l))^2$, $goal = 1 \times 10^{-4}$, $grad = 1 \times 10^{-10}$, and $epochs = 100\,000$.

2.4. Performance criteria of the simulation

We will inspect the performance of the simulation network trained by different methods using the following three factors: the approximation accuracy, the prediction ability to the testing data, and the conformance of the prior knowledge.

For each training method, we use the cross-validation (Martens and Dardenne, 1998; Martens and Naes, 1989) to inspect the approximation accuracy and prediction ability of a network. That is, we take out a datum from the sample data as temporary 'testing data' in turn, and use the remaining data to model a network and predict on the testing data. Thus, by modeling time after time, we can inspect and compare the performance of the different methods equally. As for the problem of a true boiling point curve of crude oil,

because of the possibility of the non-monotonicity between the sample datum 22 and 23, we will not extract them as the testing data. Thus the size of the training set for each model in cross validation is 29 and the total number of models is 28. The mean square error (MSE) of the training data for one model in cross validation is:

$$\mathrm{MSE} = \frac{1}{l}\sum_i (y_i - \hat{y}(x_i))^2 \tag{4}$$

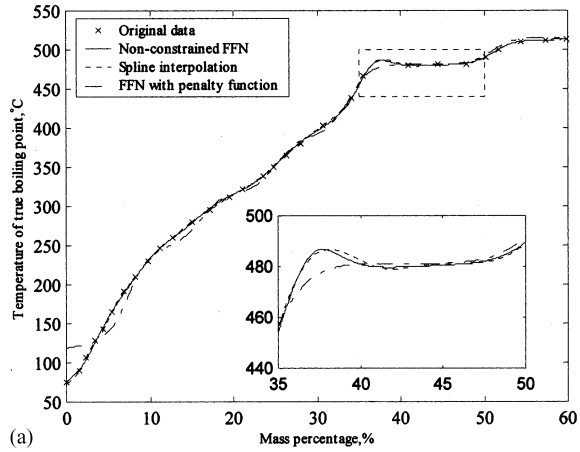
where l is the total number of the training data. The approximation accuracy for each training method is the mean of the MSEs (denoted by $\overline{\mathrm{MSE}}$) of the different models in cross validation trained by that method.

The relative prediction error on the testing data for one model in cross validation is:

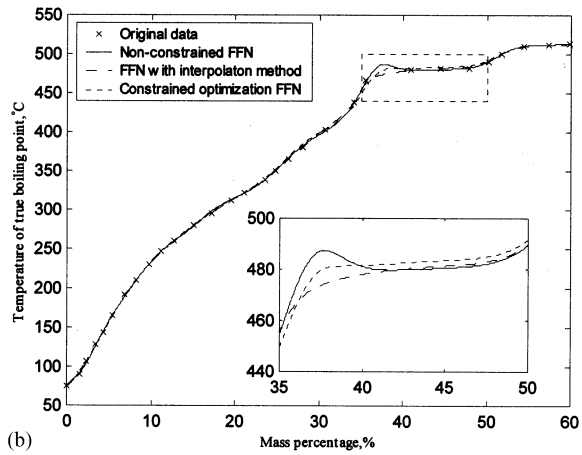
$$r_e = \frac{y - \hat{y}(x)}{y} \tag{5}$$

The prediction ability for each training method can be measured by the mean of the $|r_e|$ values (denoted by $\overline{|r_e|}$) and the standard deviation of the r_e values (denoted by δ_{r_e}) of the different models in cross validation trained by that method.

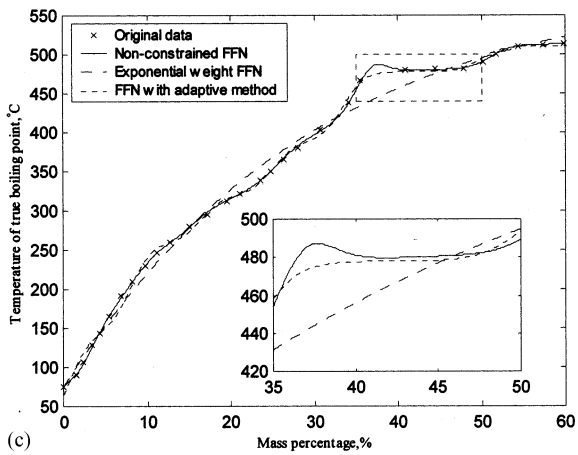
As for the conformance of the prior knowledge, in this paper, we need to analyze the non-monotonic interval of the network function $\hat{y}$ versus x. To do this, we calculate the points where $\hat{y}'(x) = 0$ by a numerical method, and eliminate the inflexions to determine the non-monotonic interval. Because the non-monotonic interval cannot be averaged among different models, we total the times when the model has a non-monotonic interval in cross validation for each training method. Moreover, we plot the curve and calculate the non-monotonic interval for one model selected at random (we just select the first model) in the cross validation as the delegate.



(a)



(b)



(c)

Fig. 2. Comparison between the different methods in modeling the true boiling point curve of crude oil.

3. Some existing feedforward network modeling methods

3.1. Unconstrained feedforward network modeling

The unconstrained (NC) method only uses sample data to train the network without thinking over the prior knowledge. In the NC method, the LM algorithm (Hagan and Menhaj, 1994) is used to train the network and approximate the true boiling point of a crude oil.

The $\hat{y}(x) - x$ curve of the first model in cross validation is illustrated in Fig. 2(a). As a comparison, we also add the cubic spline interpolation curve. The detailed performance of the method is listed in Table 2. We can see that the model is accurate in approximation; however, there is a monotonically decreasing interval [37.8%, 41.7%] in the first model of cross validation, whereas other parts of the curve increase monotonically.

To inspect the monotonically decreasing interval more clearly, we list the values of $\hat{y}(x)$ versus x in Table 3 by sample on probable non-monotonic interval [37.00%, 42.00%] with sample interval 0.5% (we also

Table 2
Performance of the different methods in modeling the true boiling point of a crude oil

Method	Mean of approximation accuracy ($\overline{\text{MSE}}$)	Mean of relative prediction error $\lvert\overline{r_e}\rvert$	Standard deviation of relative prediction error δ_{r_e}	Number of non-monotonic models/Total models in cross validation	Non-monotonic interval of the first model in cross validation
NC	2.9431×10^{-6}	0.0545	0.0939	27/28	[37.80%, 41.70%]
J.PF	1.3046×10^{-4}	0.2341	0.6861	10/28	None
IP	2.2268×10^{-6}	0.0401	0.1046	1/28	None
CO	7.8737×10^{-6}	0.0318	0.0585	0/28	None
EW	1.2115×10^{-4}	0.0531	0.1006	0/28	None
AP	6.5829×10^{-5}	0.3404	1.2838	9/28	None

sample the points on which the first derivation of $\hat{y}(x)=0$) for the first model in cross validation. From Table 3, as for the NC method, we can see that the sampled $\hat{y}(x)$ values increase at first, reach a maximum at 37.80% (the point on which $\hat{y}'(x)=0$), then decrease and reach a minimum at 41.70% (the point on which $\hat{y}'(x)=0$), and then increase again. So obviously $\hat{y}(x)$ is not monotonic in interval [37.00%, 42.00%] with respect to x.

3.2. Joerding's penalty function (J.PF) method

To incorporate prior knowledge information into feedforward networks, Joerding and Meador (1991) presented an architecture constraint (AC) method and a weight constraint (WC) method. The AC method selects the architecture of the network (including the network structure, the fashion of the computation, etc.) that satisfies the prior knowledge. So every point in the weight space associated with the constrained architecture will produce a network that satisfies the desired condition. While the WC method modified the training algorithms in such a way that the training trajectory in weight space converges to a point in a weight subspace that satisfies the prior knowledge. On the basis of the latter, Joerding and Meador (1991) proposed the penalty function method (J.PF), in which a sufficient (but not necessary) condition for the monotonic increase of Eq. (2) is deduced for the feedforward network introduced in Section 2.1:

$$w_{j1}^{[2]}\cdot w_{1j}^{[3]}>0,\qquad j=1,\ 2,\ \ldots,\ n \tag{6}$$

The J.PF method uses the function:

$$\text{PE}(\boldsymbol{W})=\text{SSE}(\boldsymbol{W})+L(\boldsymbol{W}) \tag{7}$$

as the performance function of the network, where $L(\boldsymbol{W})$ is a penalty function, which is zero when $w_{j1}^{[2]}\cdot w_{1j}^{[3]}\geq 0$ and positive when $w_{j1}^{[2]}\cdot w_{1j}^{[3]}<0$. $L(\boldsymbol{W})$ has a modulatory parameter θ that can control the approximation accuracy and conformance of the prior knowledge.

As for the problem of simulating the true boiling point curve of crude oil, θ is set to a value of -1, the lower limit of the performance function is $goal=1\times10^{-4}$ (note: here the performance function is $\text{SSE}(\boldsymbol{W})+L(\boldsymbol{W})$, not $\text{SSE}(\boldsymbol{W})$), and all other parameters are same as those in Section 2.3. The training result is illustrated in Fig. 2(a). From Table 2 we can see that even though the J.PF method eliminates the non-monotonic interval, its approximation accuracy and prediction ability are not very good.

3.3. Interpolation (IP) method

The main idea of the IP method is: add some interpolation points data that satisfy the prior knowledge to the original training data, then use the compound data to train the network, and eliminate the added points in

Table 3
Sample on the probable non-monotonic interval [37.00%, 42.00%] of the true boiling point of a crude oil (using the first model in the cross validation)

No.	p (%)	$\hat{t}$ (°C)		
		NC	EW	AP
1	37.00	484.38	441.48	473.18
2	37.50	485.91	443.98	474.77
3[a]	37.80	485.91	445.46	475.48
4	38.00	485.64	446.44	475.87
5	38.50	484.41	448.86	476.63
6	39.00	482.95	451.24	477.13
7	39.50	481.68	453.58	477.46
8	40.00	480.77	455.88	477.67
9	40.50	480.21	458.14	477.80
10	41.00	479.92	460.37	477.86
11	41.50	479.82	462.55	477.88
12[a]	41.70	479.82	463.42	477.89
13	42.00	479.84	464.70	477.89

[a] The points on which the first derivation of the function is zero.

the training process step by step. This method takes advantage of the interpolation points to embody the given prior knowledge. Usually, the interpolation points are acquired using the first-principles model and they are equally spaced with respect to the original data, so that the insertion and deletion of the interpolation points will not affect the neural network model too much.

As for the problem of simulating the true boiling point curve of crude oil, we use linear interpolation to calculate the interpolation points so that the added points are also monotonically increasing with respect to the original data. The training result is illustrated in Fig. 2(b) and Table 2. From Table 2 we can see that the approximation accuracy and the prediction ability of the IP method are good. It also eliminates the non-monotonic interval.

3.4. Constrained optimization (CO) method

The main idea of the CO method is: take the network training process as solving the following nonlinear optimization problem (Bazaraa and Shetty, 1979):

$$\min \ \mathrm{SSE}(\boldsymbol{W})$$

$$\text{s.t. } \frac{\mathrm{d}\hat{y}(x)}{\mathrm{d}x} \geq 0, \qquad \forall x \in [0,\ 1] \tag{8}$$

where the infinite inequality constraint $\mathrm{d}\hat{y}(x)/\mathrm{d}x \geq 0$ represents the monotonic prior knowledge, and [0, 1] is the domain of the input x. In the CO method, the initial weights are obtained by first training the network by the NC method. The training result is illustrated in Fig. 2(b) and Table 2. From Table 2 we can see that the approximation accuracy and the prediction ability of the CO method are comparatively good.

In the Sections 4 and 5, we will propose two new methods of including the monotonic constraint into feedforward networks and compare them with the existing methods presented in this section.

4. Exponential weight (EW) method

The necessary and sufficient condition for $\hat{y}(x)$ to be monotonically increasing is:

$$\hat{y}'(x) = \sum_i w_{1i}^{[3]} f'(w_{i1}^{[2]} x + b_i^{[2]}) w_{i1}^{[2]} > 0 \tag{9}$$

$f'(x) = (4\mathrm{e}^{-2x})/(1+\mathrm{e}^{-2x})^2$ is identically positive in its domain, so if we let:

$$w_{i1}^{[2]} > 0 \text{ and } w_{1i}^{[3]} > 0 \quad i = 1,\ 2,\ \ldots,\ m \tag{10}$$

the condition in Eq. (9) will be satisfied.

The condition in Eq. (10) is a sufficient (but not necessary) condition. The network that satisfies Eq. (10) will be monotonically increasing in its domain. On the basis of the condition in Eq. (10), the EW method replaces the original $w_{i1}^{[2]}$ and $w_{1i}^{[3]}$ in the network with $\mathrm{e}^{w_{i1}^{[2]}}$ and $\mathrm{e}^{w_{1i}^{[3]}}$ to perform all the weight adjustment and network computation (the biases are computed on its original fashion). This kind of network is called an exponential weight network. For the sake of brevity, we will use the concept of an exponential matrix in the following text, i.e. $\mathrm{e}^{\boldsymbol{W}}$ denotes a matrix the elements of which are the exponential of the elements of $\boldsymbol{W}$. Thus, the function expressed by the exponential network with the same structure as Fig. 1 is:

$$\tilde{y} = \mathrm{e}^{\boldsymbol{w}^{[3]}} \ f(\mathrm{e}^{\boldsymbol{w}^{[2]}} \ x) \tag{11}$$

where $\mathrm{e}^{\boldsymbol{w}^{[3]}}$ and $\mathrm{e}^{\boldsymbol{w}^{[2]}}$ are both exponential matrices.

Exponential weight networks have the following characteristics:

- exponential function $g(x) = \mathrm{e}^x > 0$ for $\forall x \in R$, so network will always satisfy the condition in Eq. (10), and the monotonic increasing constraint is satisfied;
- the range of the function $g(x) = \mathrm{e}^x$ is $(0, +\infty)$, so it will not introduce any other constraints or prior knowledge other than the condition in Eq. (10).

Exponential weight networks can also be trained by the LM algorithm, but usually the training speed is very slow. In this paper, we will use the LM algorithm to do the work, and the computation of the Jacobian matrix can be performed using the general BP algorithm, i.e.:

$$\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{W}^{[2]}} = (\mathrm{e}^{\boldsymbol{w}^{[2]}} \ \boldsymbol{x})^{\mathrm{T}} (f'(\mathrm{e}^{\boldsymbol{w}^{[2]}} \ \boldsymbol{x}) \circ (\mathrm{e}^{\boldsymbol{w}^{[3]}})^{\mathrm{T}}) \tag{12}$$

$$\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{W}^{[3]}} = \mathrm{e}^{\boldsymbol{w}^{[3]}} \ f(\mathrm{e}^{\boldsymbol{w}^{[2]}} \ \boldsymbol{x}) \tag{13}$$

$$\boldsymbol{J} = \left[\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{W}^{[2]}} \frac{\partial \boldsymbol{E}}{\partial \boldsymbol{W}^{[3]}} \right] \tag{14}$$

where $\circ$ is the Hadamard product. By substituting Eq. (14) into Eq. (3), we can calculate $\Delta \boldsymbol{W}$.

The training results in simulating the true boiling curve by the EW method are shown in Fig. 2(c), Tables 2 and 3. There are no non-monotonic intervals in its domain, and the approximation accuracy and the prediction ability of it are better than the NC one.

5. The AP method

The AP method is a class of methods that has been widely used in the field of science and engineering. From the viewpoint of system theory (Li et al., 1991), the system using the AP method is stable in its initial status. When the system enters an unstable status because of a change in the environment or its self-evolvement, the AP method will adjust the system parameters to induce it to the original stable status or enter into a new stable status. The adjustment ability of the AP

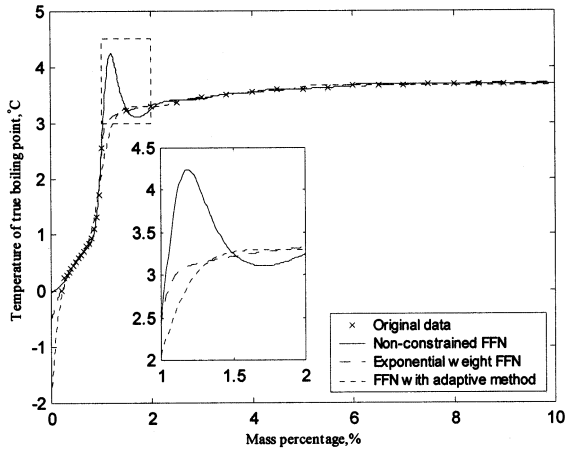


Fig. 3. Comparison between the new introduced methods in modeling the effect pressure on entropy of crude oil.

method is also called negative feedback. The adjustment process is illustrated in Fig. 3.

The AP method can be applied to the prior-knowledge-based neural network modeling. Here the neural network is considered as a self-organizing system. When the network satisfies the prior knowledge, we say it is in a stable status; contrarily, we say it is unstable. The training process of the network can be taken as the evolvement of the system (i.e. the evolvement in the positive direction), while the training target is the evolutional target. Thus, the detailed procedure to train the network with a given prior knowledge is performed as follows:

1. set the initial weight of the network so that it will satisfy the given prior knowledge, i.e. the initial status of the system is stable;
2. train the network by adjusting the weights of the network using some algorithms, i.e. the evolvement of the system;
3. when the network does not satisfy the prior knowledge, i.e. the system is unstable, use some methods to adjust the network or to modify the training strategy so that the system can eliminate the unstable factor and return to the original stable status or enter into a new stable status;
4. if the training goal has been obtained or the training time is longer than a predefined limit, the training process will be stopped; otherwise, go to step 2 and continue training.

Using the above evolvement process, the network will converge to the goal and satisfy the prior knowledge at the same time.

In order to approximate the true boiling point curve, we use the AP method based on the LM algorithm in this paper. The procedure performed is as follows:

1. initialize the weights of the network, so that it will satisfy the sufficient condition in Eq. (6) for the monotonicity of the network:
   1.1. initialize the weights of the network with non-zero random numbers;
   1.2. if $w_{j1}^{[2]} \cdot w_{1j}^{[3]} < 0$, $j = 1,2,\ldots,n$, $-w_{1j}^{[3]} \to w_{1j}^{[3]}$; otherwise, keep the weight;
2. calculate the $\Delta \boldsymbol{W}$ by Eq. (3) and do a tentative adjustment on the weights;
3. solve the equation $\hat{y}'(x) = 0$ using a numerical method; if there exists a solution x_0 that satisfies $\hat{y}''(x_0) \neq 0$, or the performance function is increased, the weights will be reset to the original value and the training strategy is adjusted by increasing μ in Eq. (3); contrarily, the adjustment on the weights is kept and μ is decreased;
4. if the performance function is smaller than the given goal, or the training time is longer than a predefined value, the iteration is stopped; otherwise, go to step 2.

The training result of the network using the AP method is illustrated in Fig. 2(c), and Tables 2 and 3. Although it eliminates the non-monotonic interval, the approximation accuracy and the prediction ability of it are not very good. The reason is analyzed in Section 7.

6. Modeling the curve of effect of pressure on entropy

Other than the true boiling point curve, we also verify the methods by modeling the curve of effect of pressure on entropy. The sample data (American Petroleum Institute, 1970) for the effect of pressure P_r on entropy S (reduced temperature $T_r = 1.00$) of a kind of crude oil are listed in Table 4. From the curve in Figure 7H1.5 from American Petroleum Institute (1970), we can see that the effect of pressure on entropy is monotonically increasing over its domain, which is the prior knowledge of the model. Simulation models of neural networks can replace the curve. However, the model trained by the NC method will be non-monotonic at some intervals. All other methods can overcome the problem. The performance data of the different methods are listed in Table 5. In the simulation, P_r and S are normalized to $P_r/10$ and

$$\frac{S - \min\{S\}}{2(\max\{S\} - \min\{S\})}$$

before further processing. The sample data 17, 18 and 19 are not extracted as the testing data in cross validation because of the possibility of the non-monotonicity among them. The structure of the neural network is the same as that modeling the true boiling point of the crude oil. The training parameters are: $\mathrm{PE} = \mathrm{SSE} = \Sigma_l (y_l - \hat{y}(x_l))^2$, $goal = 1 \times 10^{-3}$, $grad = 1 \times 10^{-10}$ and $epochs = 100\,000$. The simulation curves are plotted in Fig. 4.

Table 4
Data for the effect pressure on entropy for a crude oil[a]

No.	P_r	S
1	0.2	0
2	0.25	0.23
3	0.3	0.29
4	0.35	0.33
5	0.40	0.40
6	0.45	0.46
7	0.50	0.51
8	0.55	0.60
9	0.60	0.64
10	0.65	0.71
11	0.70	0.79
12	0.75	0.84
13	0.80	0.93
14	0.85	1.10
15	0.90	1.32
16	0.95	1.71
17	1.00	2.55
18	1.50	3.22
19	2.00	3.30
20	2.50	3.37
21	3.00	3.45
22	3.50	3.51
23	4.00	3.55
24	4.50	3.60
25	5.00	3.61
26	5.50	3.63
27	6.00	3.66
28	6.50	3.67
29	7.00	3.68
30	7.50	3.69
31	8.00	3.70
32	8.50	3.70
33	9.00	3.70

[a] Reduced temperature $T_r = 1.00$.

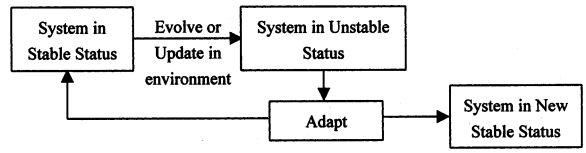


Fig. 4. Adaptation of the system in evolvement.

7. Analysis and conclusion

7.1. Classification of the methods

The AC and WC methods presented by Joerding conform to the prior knowledge only from the viewpoint of the network itself. However, we can also obtain the same goal by adding some interpolation points to the training set. The methods that encode the prior knowledge in the network by preprocessing the sample data are called data constraint (DC) methods in this paper. The AC, WC and DC methods have exhausted all possibilities for constraining feedforward networks.

According to the relationship between the algorithm and the network, the algorithms can also be divided into two classes: internal and external methods. The internal method restricts the architecture and parameters of the network so that it will satisfy the given prior knowledge, whereas the external method encodes the prior knowledge to the network using indirect ways. The applicability of the external method is wider and its operations are relatively easier; however, sometimes it cannot ensure that the network will conform to the given prior knowledge accurately. However, though the internal method usually requires some skillful design in application, and its applicability is relatively narrower, it usually can embody the prior knowledge very well. According to this classification, the AC methods belong to the internal method, and the DC methods belong to the external method.

Table 5
Performance of the different methods in modeling the effect of pressure on entropy for a crude oil

Method	Mean of approximation accuracy ($\overline{\mathrm{MSE}}$)	Mean of relative prediction error $\overline{\lvert r_e \rvert}$	Standard deviation of relative prediction error δ_{r_e}	Number of non-monotonic models/Total models in cross validation	Non-monotonic interval of the first model in cross validation
NC	2.9577×10^{-5}	1.0622	0.3042	20/30	[1.20, 1.72]
J.PF	5.4455×10^{-4}	0.1298	0.3409	14/30	None
IP	2.6481×10^{-5}	0.0667	0.1907	10/30	[1.08, 1.18]
CO	1.2378×10^{-4}	0.1047	0.3372	0/30	None
EW	3.0292×10^{-5}	0.0738	0.1861	0/30	None
AP	2.0061×10^{-4}	0.0458	0.0805	10/30	None

7.2. Discussion of the EW method

The approximation accuracy and the prediction ability (which is more important than the approximation accuracy) of the EW method presented in this paper are better than the J.PF method and the NC method. A decrease of the mean relative error means an increase of the prediction accuracy, whereas a decrease of the variance means an increase of the prediction stability. So the model approximates the actual object better by including the prior knowledge.

The EW method is a kind of AC method that encodes the prior knowledge into the network by designing the fashion of computation.

In the EW method, exponential function e^x will increase fast with increase in x. Therefore, if the initial weight is too large, the value of e^x will increase to numerical positive infinity so that the computation cannot continue. To avoid this problem, the exponential function can be replaced with the following two functions:

$$g_1(w) = \begin{cases} e^w & w \le 0 \\ w+1 & w > 0 \end{cases}, \quad g_2(w) = \begin{cases} \dfrac{1}{1+e^{-2w}} & w \le 0 \\ \frac{1}{2}w + \frac{1}{2} & w > 0 \end{cases}$$

Functions $g_1(w)$ and $g_2(w)$ are first-order differentiable in their domains, and increase at polynomial speed when $w > 0$. Therefore, there are no special requirements (initial weight restriction) for the weight matrix when applying these functions. The disadvantage is that they use piecewise functions and the computation is complex.

The main disadvantage of the EW method is that the training speed is very slow and sometimes it is too slow to converge, since the convergence surface of the exponential weight network is more complex than the normal ones.

7.3. Discussion of the AP method

The AP method is a kind of WC method, because in the evolvement it restricts the weights of the network in the dynamic training process. Moreover, it can be considered as being a kind of external abductive method when using an external force (such as a decrement of μ in the LM algorithm) to induce the network system to the stable status, or as a kind of internal method if using the internal mechanism (such as modifying the weights and biases).

The disadvantage of the AP method is that, generally, it is difficult to find an appropriate method to adjust the parameters of the network or the training algorithm so that the system can turn from an unstable status to a stable one. A commonly used method is to decrease the training step in order to slow down the evolvement of the system; however, this may also prevent the system from evolving to the given goal. In addition, the initial status of the system is also important to the posterior evolvement. A mis-selected initial status may induce the system to a local minimum. That is the reason why the approximation accuracy and the prediction ability of the AP method are usually not very good.

7.4. Comparison between the different methods

In simulating of the true boiling point of a crude oil the size of the data set is 30, which is small relative to the number of network parameters. So, although the approximation accuracy and the prediction ability of the NC method are good, it violates the prior knowledge on interval [37.80%, 41.70%], like the cubic spline interpolation method, and it may overfit the training data. On the contrary, the J.PF, IP, CO, EW and AP methods introduce monotonicity to the model successfully and solve the problem of encoding the prior knowledge into the neural networks effectively. They prevent the trained model from violating the prior knowledge. In addition, they take advantage of the prior knowledge as a supplement to the lack of sample data to prevent overfitting. They also use the prior knowledge to decrease the free degree of the network parameters so that it does not have enough power to overfit the data. Therefore, the prior-knowledge-based training methods can also be considered as approaches to solving the overfitting problems in a sense.

The approximation accuracy and the prediction ability of the J.PF method are not satisfactory. In the network training process, although the non-monotonic interval can be decreased or eliminated by adding a penalty function to the performance function, the fit accuracy to the sample data will also be affected. Therefore, it is important to select a good penalty or an appropriate value of θ in order to find a tradeoff between the approximation accuracy and the constraint of the prior knowledge. Sometimes this may be very difficult.

The IP method uses interpolation points to constrain the connection weights in the networks. So it is a kind of DC method. The approximation accuracy and the prediction ability of the IP method in simulating the true boiling point curve of crude oil are better than all the other methods. The reason is that the interpolation points generated by linear interpolation are fairly close to the original model. So the interpolation is similar to providing more sample to the model and leads to a more accurate approximation.

The CO method uses a constrained optimization algorithm to constrain the connection weights in the network. So it is a kind of WC method. Usually, the weight space is too large and complex for the optimiza-

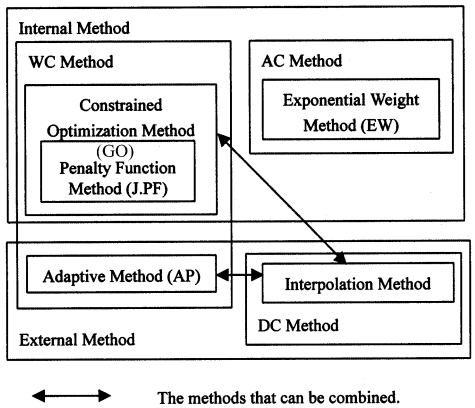


Fig. 5. The relationship between the prior-knowledge-based methods.

tion algorithm to do a thorough search. Therefore, it is important to select a good initial value. A simple and effective initializing method is first to train the network using the NC method. The subsequent optimization is based on the NC weights and only requires some small adjustments. So the approximation accuracy and the prediction ability of the CO method are usually good.

The relationship between the J.PF, IP, CO, EW and AP methods is illustrated in Fig. 5. Usually, the DC method can be combined with the WC method to obtain better performances. We will discuss such hybrid models in other papers.

Acknowledgements

This project was supported by the National Natural Science Foundation (20076041) of China.

References

American Petroleum Institute, 1970. Technical Data Book — Petroleum Refining. Port City Press, Washington, DC, pp. 7–198.

Bazaraa, M.S., Shetty, C.M., 1979. Nonlinear Programming: Theory and Algorithms. Wiley, Washington, DC.

Chen, C.-W., Chen, D.-Z., Hu, S.-X., 2000. Feedforward networks based on prior knowledge and its application in modeling the true boiling point curve of the crude oil. Journal of Chemical Engineering of Chinese Universities, in press.

Gallant, A.R., White, H., 1982. On learning the derivatives of an unknown mapping with multilayer feedforward networks. Manuscript, Department of Economics, University of California at San Diego.

Hagan, M.T., Menhaj, M., 1994. IEEE Transactions on Neural Networks 5 (6), 989–993.

Hornik, K., Stinchcombe, M., White, H., 1989. Neural Networks 2, 359–366.

Hornik, K., Stinchcombe, M., White, H., 1990. Neural Networks 3, 551–560.

Hu, S.-X., 1983. Journal of East China Petroleum Institute 1, 1–18.

Hu, S.-X., Tang, J.-N., 1987. Petroleum Processing 18 (2), 1–10.

Joerding, W.H., Meador, J.L, 1991. Neural Networks 4, 847.

Li, Y.-z., Le, C.-x., Zhou, L.-m., 1991. Systematology. Press of Central China Normal University, pp. 114–133.

Martens, H.A., Dardenne, P., 1998. Chemometrics and Intelligent Laboratory Systems 44, 99–121.

Martens, H.A., Naes, T., 1989. Multivariate Calibration. Wiley, Chichester.

Thompson, M.L., Kramer, M.A., 1994. AIChE Journal 40 (8), 1328–1340.