An Improved Just-in-time-learning Based Soft Sensor Method Using Dimensionality Reduction Analysis

Yu Chen, Tong Ye
College of Mathematics, Sichuan University, Chengdu, Sichuan, 610041
duoerxs@163.com

Abstract: In recent years, the data-driven soft sensor has achieved unprecedented development, which can update the model in real-time and save many costs. Most traditional methods are only suitable for single-mode processes, but the existing working conditions are generally multi-modal. In this paper, we propose an improved JITL method to modeling the complex industrial process. First, we apply AE to reduce the data dimensionality, exploring the intrinsic structure of data. Then a JITL strategy is adopted to select similar samples on the low-dimensional data and use them for the local model. Finally, PLS is developed for each model and predicts the variables. A three-phase flow case verifies the effectiveness of the proposed method. We find that compared with other methods, the proposed method has the highest accuracy.

1. INTRODUCTION
Nowadays, it is important to improve their production methods to improve production efficiency and ensure process safety and efficiency. It largely depends on the stable control of the production process of the product and real-time updates. However, in practice, some key variables are difficult to be measured, because of complex physical and chemical mechanisms and the uncertainty of the production process. To overcome these problems, soft sensor modeling is proposed by some research scholars. For soft sensors, it is necessary to extract quality-relevant features and make use of process data to build a mathematical model for quality prediction [1]. It can be mainly divided into two types of soft sensor models: mechanism-based models and data-driven models. Mechanism-based models rely on a deep understanding of the process mechanism in the production process. However, it is challenging to construct a mechanism-based model in complex processes. Instead, data-driven models do not require a precise understanding and description of the process technology but directly use the inherent information between the process data to establish a mathematical model. Data-driven soft measurement methods have received extensive attention in both industry and academia.

For data-driven models, we can build a mathematical model using traditional statistical methods. Least squares regression (LSR) is commonly used in the traditional statistics method. For example, Ling et al. used LSR to predict whether the ratio of organic aerosol to CO would change with age [2]. LSR finds the suitable parameters by minimizing the sum of squares of errors. However, it is not only easily influenced by outliers, but also it cannot deal with data with high collinearity. To address data collinearity, ridge regression is proposed. Ridge regression solves data collinearity by introducing quadratic penalty terms, and it has the effect of parameter shrinkage.

Nevertheless, it cannot extract features and reduce the dimension of data. To reduce the dimension of data, principal component regression (PCR) is then developed. It can realize the dimension reduction processing of multi-dimensional data, increase the running speed, and solve the problem of data
collinearity to a certain extent. For instance, JG Herman et al. [3] described a new method named methylation-specific PCR (MSP), which could rapidly assess the methylation status of virtually any group of CpG sites within a CpG island independent of the use of methylation-sensitive restriction enzymes. However, PCR is an unsupervised method, and it cannot express the principal components proposed reasonably. To solve this problem, partial least squares regression (PLS) is proposed, which includes the information of independent variables and considers the information of dependent variables. Therefore, it has strong stability. Huerta et al. [4] adopted PLS to diagnose and apply mining data. However, these methods are all for a single mode.

However, most industry processes show typical time-varying and multimodal characteristics, which the methods mentioned above cannot address. Process time-varying is that industry process often suffers from some problems that change process characteristics, such as equipment aging, catalyst deactivation, and change of raw materials. To handle these problems, soft sensor modeling is required to have the ability to update the model automatically. Therefore, the just-in-time learning (JITL) method, also called lazy learning, is proposed by some researchers. For JITL modeling, a local model is built from the historical data using the most relevant samples around the query data point when an estimated value is required. Compared to other methods, JITL has the following characteristics. First, only when the query sample needs to be predicted, the local modeling will be performed. Second, JITL uses some of the samples in the historical data that are most similar to the query samples, not all samples. Third, once the query sample prediction is completed, the model will be discarded, and JITL will create a new local model until the following sample arrives. In short, JITL is divided into three steps, including the selection of similar samples, local modeling, and output prediction. JITL is very popular in industry process modeling. For example, Zheng et al. [5] proposed an adaptive two degrees of freedom (2Dof) PI controller based on the just-in-time-learning (JITL) method for predictive speed control of permanent magnet synchronous linear motor (PMSLM). Pheng et al. [6] used JITL in the Chinese construction industry to improve its performance and competitiveness.

Nevertheless, the prediction of JITL mainly depends on the selection of relevant samples used to build the local model. If the selection of relevant samples is not very accurate, the prediction precision of the model will also decrease. At the same time, when the data has high dimensionality, looking for relevant samples can be very time-consuming. In most JITL models, the selection of relevant samples is often based on Euclidean distance. However, it is not always accurate to use Euclidean distance to describe the similarity of samples since it does not consider the correlation of process variables, and in reality, the actual distance of these data points may not be Euclidean distance. Therefore, traditional Euclidean space metrics are difficult to use for real-world nonlinear data, such as the famous Swiss roll data set. To address these problems, isometric feature mapping (ISOMAP) [7] is proposed to measures the geodesic distance between samples instead of Euclidean distance, which is based on multi-dimensional scaling (MDS) [8]. However, one problem with ISOMAP is that it finds the optimal global solution for all samples. When the amount of data is large and the data dimension is high, the calculation is very time-consuming. Because of this problem, locally linear embedding (LLE) [9] is raised. It reduces the dimensionality by giving up the global optimum of all samples and only reduces the dimensionality by ensuring the local optimum. However, for these methods based on the above, the adjacency matrix needs to be recalculated when reducing the dimensionality, challenging to meet the real-time requirements of soft industrial sensing. So this limits the application of LLE.

Based on the above statements, we propose an improved JITL method for the soft industrial sensor. First, an autoencoders AE-based [10] dimensionality reduction method is applied. The AE passes the high-dimensional data through a multilayer neural network and encodes it into low-dimensional data to reconstruct the high-dimensional data. And then find similarities in the low-dimensional data we get, which makes the search faster and has a more apparent physical meaning. Secondly, we adopt a JITL strategy to find data samples that match the query sample from the historical accumulated data and use them for local modeling to achieve better modeling accuracy. Besides, through JITL, the model can be updated adaptively to overcome the problem of the time-varying process. Finally, for each local model,
we use PLS for modeling. PLS can capture the correlation between output variables and input variables and handle multicollinearity with stable performance.

The residual is organized as follows. In section II, preliminaries are briefly introduced. Section III shows the proposed method. In section IV, experiments will be introduced, and the method’s effectiveness will be verified through experiments. Some conclusions are shown in section V.

2. PRELIMINARIES

In this section, both Just-in-Time Learning and partial least squares regression are briefly revisited.

2.1. Just-in-Time Learning Modeling

JITL is a nonlinear modeling framework based on the idea of local modeling and database technology. The traditional global modeling method may not track the changes of the process state and describe the complex data structure, while the JITL method can establish a local model among the most relevant samples. In this way, a complex global model can be replaced by a simpler local model, and such a local model is instantaneous and updated. Generally speaking, JITL has three steps. First, when the query sample appears, the relevant samples that are most similar to the query sample are selected as the training samples for local modeling through a certain similarity measure. Second, build and train a local regression model on these relevant samples. Finally, the output of the query sample is predicted by the trained model to achieve the purpose of detecting some key variables.

2.2. Partial least squares regression

Partial least squares regression is a traditional regression method. First, n sets of data are observed for p independent variables and m dependent variables.

\[
X = (X_1, X_2, \ldots, X_p) = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \tag{1}
\]

\[
Y = (Y_1, Y_2, \ldots, Y_m) = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & & \vdots \\ y_{n1} & \cdots & y_{nm} \end{bmatrix} \tag{2}
\]

Then standardize the raw data. Denote \( F_0, E_0 \) as the standardized variable and matrix of the dependent variable \( Y \) and the independent variable \( X \). The second step is to extract comprehensive variables from \( X \).

\[
t_i = X \omega_i = \omega_{i1} x_1 + \omega_{i2} x_2 + \cdots + \omega_{ip} x_p \tag{3}
\]

where \( \omega_i (i=1,2,\ldots,p) \) denotes the characteristic coefficient of the i-th variable. The third step is to implement the regression of \( E_0 \) on \( t_i \) and the regression of \( F_0 \) on \( t_i \).

\[
E_0 = t_i p_i + E_i \tag{4}
\]

\[
F_0 = t_i F_i + F_i \tag{5}
\]

where \( E_i \) and \( F_i \) are the residual matrix of \( E_0 \) and \( F_0 \) respectively.

The fourth step is to replace \( E_i \) and \( F_i \) with residual matrix \( E_0 \) and \( F_0 \), and use the same method to extract \( t_i \) to extract the comprehensive variable \( t_2 \), and so on. Until h comprehensive variables \( t_1, t_2, \ldots, t_h \) are extracted, use the cross-validity test to determine the value of \( h \). The final PLS model:

\[
y_j = \beta_{j1} x_1 + \beta_{j2} x_2 + \cdots + \beta_{jp} x_p + y_h \tag{6}
\]

3. JUST-IN-TIME LEARNING MODEL BASED ON AUTOENCODERS

3.1. Motivation

In modern industrial processes, the data we collect is typically nonlinear and has a high dimension. If the data are processed directly, the computer will slow down, consuming plenty of time and will take up
a lot of storage space. Therefore, we need to reduce the dimensionality of the data. However, traditional dimensionality reduction methods such as PCA are based on Euclidean distance for dimensionality reduction, a linear dimensionality reduction method. However, sometimes, the actual distance between data points is not described by Euclidean distance. As shown in Figure 1, where Fig.1 a represents the display of the original data set in a high-dimensional space, if the data uses Euclidean distance as the actual distance between two points, this is not correct. Moreover, Fig.1 b shows the original data display using PCA linear dimensionality reduction on a two-dimensional plane. Therefore, linear dimensionality reduction methods cannot reflect the intrinsic properties of high-dimensional data. We use nonlinear dimensionality reduction methods, the most typical of which is manifold learning. However, manifold learning [11] is a dimensionality reduction method based on the overall situation. When adding new data points, the graphs used for reducing dimensionality must be reconstructed, and the original graphs cannot be used for calculations. Therefore, manifold learning cannot achieve real-time dimensionality reduction. Furthermore, we use a neural network method to do dimensionality reduction learning, which guarantees nonlinear dimensionality reduction and meets the real-time requirements of industrial processes.

3.2. The method of reducing the dimensionality

The dimensionality reduction of the data can obtain computational advantages and significantly improve the intelligibility of the data. Especially in JITL, dimensionality-reduced data for model construction can improve the prediction accuracy of the model. Meanwhile, it can eliminate redundancy to solve the problem of multicollinearity. Therefore, it is necessary to use a method that can perform non-linear dimensionality reduction and make the model adaptively update. AE is a typically NN-based method, which is composed of an encoder and a decoder. AE can achieve the above advantages because AE uses the backpropagation (BP) [12] algorithm, which consists of two processes: the forward propagation of the signal and the backward propagation of the error to make the output value equal to the input value. AE first compresses the input into a latent space representation and then uses this representation to reconstruct the output. The autoencoder consists of two parts: encoder and decoder. The specific process is as follows.

First, we use the encoder to compress the input into a latent space representation. It can be expressed by the encoding function, where $f(x)$ represents the obtained low-dimensional data features

$$h = f(x)$$  \quad (7)

Then, the input of the decoder comes from the input of the latent space representation, and the decoding function can be used.

$$\hat{x} = g(h)$$  \quad (8)

where $g(h)$ denotes the function of the decoder. Therefore, the entire AE can be described by the function $\hat{x} = g(f(x))$, where the output $\hat{x}$ is close to the original $x$. The learning process of the AE is to minimize the reconstruction error

$$L(x, g(f(z))) = \|x - \hat{x}\|^2$$  \quad (9)

where $L$ is a loss function so that the difference between $g(f(x))$ and $x$ is minimized. When the learning is completed, the encoder is retained, and the query samples are input to the encoder to obtain the reconstructed data. The whole process will not change the original reconstructed data set. Therefore, real-time dimensionality reduction can be achieved. So AE is used for data dimensionality reduction to make our model have the ability to update adaptively.
3.3. Just-in-Time learning model based on partial least squares regression

When the n-dimensional query sample \( x_{new} \) appears, we need to find the most similar sample to the newly obtained query sample \( x_{new} \) from the existing data set. We first reduce the dimensionality of the sample \( x_{new} \) to m-dimensionality feature \( h_{new} \) through the encoder, which can be expressed as

\[
h_{new} = f(x_{new})
\]  

(10)

Use Euclidean distance as the similarity measure. Calculate the similarity \( d_k \) between \( h_{new} \) and the k-th sample \( h_k \) in the existing data set:

\[
d_k = \sqrt{\sum_{i=1}^{m} (h_{new,i} - h_{k,i})^2}
\]  

(11)

where \( h_{new,i} \) denotes as the i-th component of \( h_{new} \). The more minor the \( d_k \), the stronger the correlation between the query sample and the k-th existing sample; the more significant, the \( d_k \) weaker the correlation between the query sample and the k-th existing sample. The number of similar samples selected affects the prediction accuracy of our model. If too few samples are selected, the model constructed may not be accurate. If too many samples are selected, it will take too long to find similar samples and select some samples that are not relevant to the query samples. Thus we select \( s \) samples that are most relevant to the query sample. Denote them as \( h_1, h_2, \ldots, h_s \). Here we should not set the value \( s \) too large. If the value \( s \) is too large, the locality of the model will be weakened, and the prediction error will be enormous; If the value \( s \) is too small, that is, the number of samples selected is small, which will lead to poor generalization ability of the model, and also make the model’s prediction error of variables significant. In this way, we get \( s \times m \) design matrix \( H \):

\[
H = (H_1, H_2, \ldots, H_m)
\]  

(12)

And its corresponding dependent variable matrix \( Y \):

\[
Y = (Y_1, Y_2, \ldots, Y_p)
\]  

(13)

Separately extract the first pair of components of the two variable groups and maximize the correlation. Suppose the first pair of components are extracted from the two sets of variables as \( T_i \) and \( U_1 \):

\[
T_i = w_{i1}H_1 + \cdots + w_{im}H_m = w_i^T H
\]  

(14)

\[
U_1 = v_{i1}Y_1 + \cdots + v_{ip}Y_p = v_i^T Y
\]  

(15)
For the needs of regression analysis, each of $T_i$ and $U_i$ is required to extract as much variation information as possible for the variable group in which they are located and to maximize the degree of correlation. From the standardized observation data matrices $H$ and $Y$ of the two sets of variables, the score vector of the first pair of components can be calculated, denoted as $t_i$ and $u_i$, respectively:

$$t_i = Hw_i = \begin{pmatrix} h_{k1} & h_{k2} & \cdots & h_{kn} \\ w_{11} & w_{12} & \cdots & w_{1n} \end{pmatrix} = \begin{pmatrix} t_{i1} \\ \vdots \end{pmatrix} \quad (16)$$

$$u_i = Yv_i = \begin{pmatrix} y_{k1} & y_{k2} & \cdots & y_{kn} \\ v_{11} & v_{12} & \cdots & v_{1n} \end{pmatrix} = \begin{pmatrix} u_{i1} \\ \vdots \end{pmatrix} \quad (17)$$

The covariance $Cov(T_i, U_i)$ of the first pair of components $T_i$ and $U_i$ can be calculated by the inner product of the score vectors $t_i$ and $u_i$ of the first pair of components. Then we get:

$$\{t_i, u_i\} = \langle Hw_i, Yv_i \rangle = w_i ^ T H Y v_i$$

$$w_i ^ T w_i = \|w_i\|^2 = 1, \quad v_i ^ T v_i = \|v_i\|^2 = 1 \quad (18)$$

Using the Lagrange multiplier method, the above problem can be transformed into finding the unit vectors $w_i$ and $v_i$ to maximize $\theta \equiv w_i ^ T H Y v_i$. We just need to calculate the eigenvalues and eigenvectors of the $m$-order matrix $M = H Y Y H$. The maximum eigenvalue of $M$ is $\theta^*$, and the corresponding unit eigenvector is the solution $w_i$ and $v_i$ can be calculated by $w_i$:

$$v_i = \frac{1}{\theta^*} Y H w_i \quad (19)$$

Establish the regression of $Y_1, \cdots, Y_r$ to $T_i$, and the regression of $H_1, \cdots, H_m$ to $T_i$:

$$H = t_i \alpha_i + E_i$$
$$Y = t_i \beta_i + F_i$$

The least-square estimate of regression coefficients $\alpha_i, \beta_i$ is:

$$\alpha_i = (t_i t_i)^{-1} t_i H = \frac{H t_i}{| t_i | ^2}$$
$$\beta_i = (t_i t_i)^{-1} t_i Y = \frac{Y t_i}{| t_i | ^2} \quad (21)$$

Use residual matrix $E_i$ and $F_i$ instead of $H$ and $Y$ and repeat the above steps, get $r$ components $t_1, t_2, \cdots, t_r$. Then we get:

$$H = t_1 \alpha_1 + \cdots + t_r \alpha_r + E_r$$
$$Y = t_1 \beta_1 + \cdots + t_r \beta_r + F_r \quad (22)$$

Let $H_i^* (i = 1, \cdots, m)$, $Y_i^* (i = 1, \cdots, p)$ denote standardized variables. Then we have:

$$T_i = w_i H_1^* + \cdots + w_{im} H_m^* \quad (23)$$
$$Y_i^* = \beta_1 T_i + \beta_2 T_i + \cdots + \beta_j T_j \quad (j = 1, 2, \cdots, p) \quad (24)$$

Substitute (19) into (20). The partial least squares regression equation of the standardized dependent variable is obtained:

$$\hat{Y}_i^* = a_1^* H_1^* + \cdots + a_m^* H_m^* \quad (j = 1, \cdots, p) \quad (25)$$
Then revert to the partial least squares regression equation of the original variable:

$$\hat{Y}_j = a_{j0} + a_{j1}H_1 + \cdots + a_{jm}H_m \quad (j = 1, \cdots, p) \quad (26)$$

4. STUDY CASE

4.1. Three-phase flow experiment

We use the data obtained in the three-phase flow experiment done at Cranfield University to validate the effectiveness and flexibility of the proposed dimension reduction method. The three-phase flow experiment is to inject oil, water, and air at different flow rates and then separate them after a series of operations and equipment. The purpose of the three-phase flow experiment is to study the impact of multiphase flow supply on small industrial equipment. Our sampling time for data is every 1 second, and the pressure of the three-phase separator is always 0.1 MPa. Here we use our method to predict air delivery pressure and pressure in the bottom of the riser through the data of other variables. All the variables we use are shown in Statistical process monitoring of a multiphase flow facility.

4.2. Experiments

We use the above-mentioned three-phase flow experiment to illustrate the effectiveness of the method. The data we selected here include two working conditions: high wind speed and low water speed, and the other is low wind speed and high water speed.

We use AE to reduce the dimensionality of the original experimental data. Here we set up a total of 9 hidden layers. Our AE network structure is expanded from 21 dimensions to 128 dimensions and then compressed to 10 dimensions. In order to illustrate the effectiveness of AE dimensionality reduction, we draw a scatter plot of any two features of the original data for comparison with the scatter plot of any two features after dimensionality reduction.

![Fig. 2 Original data of feature scatter plot](image)

**Fig. 2** Original data of feature scatter plot

Fig 2 shows the two feature scatter plots randomly selected by the original data. We can see that the features of the two working conditions are not well separated, but the features of the two working conditions should be different, which shows the original data does not explain the intrinsic structure well.

![Fig. 3 Feature scatter plot after AE dimensionality reduction](image)

**Fig. 3** Feature scatter plot after AE dimensionality reduction

Fig 3 shows the characteristic scatter plots of any two features of the two working conditions after AE dimensionality reduction. We can see that the two characteristics of the two working conditions
are separated, which shows that the low-dimensional space can better reflect the intrinsic structure of the data.

After AE dimensionality reduction, the two working conditions can be well separated, which will help us find similar samples later. We use the JITL method based on the Euclidean distance similarity measure to find samples similar to the query samples. We found that selecting 10 samples that were similar to the query samples was the best through cross-validation. Therefore, we established a PLS model for these samples. In this process, we retained 4 latent variables for modeling, and the regression equation obtained was used to predict the query samples. We predicted the PT312 and PT401 of 100 query samples and judged the accuracy of the prediction by the MSE between the actual value and the predicted value. In order to compare the superiority of our method, we also involved the following experiments for comparison, one of which is to directly perform PLS on all data to establish a global prediction model. The other is to use the JITL method to directly measure the similarity of the original data and find the 10 most similar samples to the query sample to establish a PLS model to predict its dependent variables.

The following table shows the MSE between the predicted value and the actual value of the variables PT312 and PT401 for the three methods.

| Variables | PLS       | JITL+PLS  | AE+JITL+PLS |
|-----------|-----------|-----------|-------------|
| PT312     | 0.012229  | 0.011841  | 0.010124    |
| PT401     | 0.009060  | 0.008125  | 0.008004    |

From the above table, we can know that the effect of PLS modeling directly on the original data is not as good as the effect of using the JITL method to perform PLS local modeling. It was shown the importance of introducing similarity measures and establishing local models. After introducing AE to reduce the dimensionality of the data and then using JITL to build a local PLS model, the effect is better than the previous two methods, which shows the necessity of data dimensionality reduction. It can better reflect the similarity between data to reduce errors and improve the accuracy of variable prediction.

In order to see the difference between the predicted value and the actual value of these three methods more clearly and intuitively, we selected 100 test sample points from these two working conditions to predict the variable PT312 and variable PT401. The first fifty sample points are the sample points collected from the first working condition. Moreover, the following fifty sample points are the sample points collected from the second working condition. Here, the following are the predicted line graphs of the three methods for the variable PT312, PT401, and the corresponding actual value line graphs.

Fig 4 Line chart of the actual value of PT312 and the predicted value of the three methods
We can see from Fig. 4 and Fig. 5 that it is more accurate to use the JITL method to establish a PLS model to predict the variables after using AE dimensionality reduction. The predicted values are only small fluctuations near the actual value. The other two methods that do not use AE dimensionality reduction will have large deviations on some query sample points, and the prediction effect is not very good.

5. CONCLUSION

In this work, we have proposed an improved dimensionality reduction method to reduce the dimensionality of the data. AE is adopted to find the intrinsic structure of the data. Then we use JITL to make our model can adapt to multi-modal working conditions. Meanwhile, using PLS allows us to predict the variables that need to be estimated. A three-phase flow case verifies the effectiveness of the proposed method, and the accuracy is improved compared to other methods. In future work, we can also make an adaptive model for selecting the number of similar samples to improve our method because this allows the model to select the best number of similar samples each time which helps to improve the accuracy of the model.

Reference

[1] Abeykoon, Chamil. "A novel soft sensor for real-time monitoring of the die melt temperature profile in polymer extrusion." IEEE Transactions on Industrial Electronics 61.12 (2014): 7113-7123.

[2] Leng L, Zhang T, Kleinman L, et al. Ordinary least square regression, orthogonal regression, geometric mean regression and their applications in aerosol science[C]//Journal of Physics: Conference Series. IOP Publishing, 2007, 78(1): 012084.

[3] Luping, Z. H. A. O., Z. H. A. O. Chunhui, and G. A. O. Furong. "Phase transition analysis based quality prediction for multi-phase batch processes." Chinese Journal of Chemical Engineering 20.6 (2012): 1191-1197.

[4] Huerta, Mauricio, et al. "A beta partial least squares regression model: diagnostics and application to mining industry data." Applied Stochastic Models in Business and Industry 34.3 (2018): 305-321.

[5] Zhang, Xinmin, Yuan Li, and Manabu Kano. "Quality prediction in complex batch processes with just-in-time learning model based on non-gaussian dissimilarity measure." Industrial & Engineering Chemistry Research 54.31 (2015): 7694-7705.

[6] Pheng, Low Sui, and Gao Shang. "The application of the just-in-time philosophy in the Chinese construction industry." Journal of construction in developing countries 16.1 (2011): 91-111.

[7] Huang, Zhenhua, Xin Xu, and Lei Zuo. "Reinforcement learning with automatic basis construction based on isometric feature mapping." Information Sciences 286 (2014): 209-227.

[8] Kruskal, Joseph B. Multidimensional scaling. No. 11. Sage, 1978.

[9] Zhao, Deli. "Formulating LLE using alignment technique." Pattern Recognition 39.11 (2006): 2233-2235.
[10] Baldi, Pierre. "Autoencoders, unsupervised learning, and deep architectures." *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings, 2012.

[11] Cayton, Lawrence. "Algorithms for manifold learning." *Univ. of California at San Diego Tech. Rep* 12.1-17 (2005): 1.

[12] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." *nature* 323.6088 (1986): 533-536.