Si K edge X-ray spectrum absorption interpretation using Neural Network

Yikun Huang,[1] Haoxuan Zhang,[1] Ye Zhang,[1] Hao Jiang,[1] and Chen Zheng[1]
Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, California 92093-0238, USA

(Dated: 16 June 2017)

I. ABSTRACT

In this project, a general x-ray absorption near-edge spectroscopy (XANES) spectra classification and interpretation analytical workflow is implemented using autoencoders neural network and feed forward neural network. We systematically investigated the effects of the layer number in autoencoders neural network. We show that the activation function and the dropout optimization has a strong influence on the convergence speed and accuracy of feed-forward neural network. Using the combination of autoencoders and feed forward neural network, the classification accuracy of Si K-edge XANES spectra reaches 92%.

II. INTRODUCTION

X-ray absorption spectroscopy (XAS) is one of the most important and widely used technique for determining the local structure and electronic structure of materials. Despite its popularity, only very few sparse XAS reference resources are available to the community. To our knowledge, no universal the XAS spectrum interpretation algorithm is available due to the shortage of reference XAS database. Typical spectrum interpretation procedure reported by the literatures usually include a proper comparison between an unknown spectrum and a group of reference spectra either from experimental measurement or computational simulation. Reference spectra close to the unknown spectrum in shape and edge position are than selected for further in depth study. The largest open XAS database is the EELS DB[1], which was created in the 1990s. The EELS DB only contains 21 XAS spectrums, which limited the capability and further usage of the database.

In our course project, we developed a universal XAS spectrum matching and interpretation platform using autoencoder and feed forward neural network using Si K-edge near-edge spectroscopy (XANES) as a model system. To overcome the shortage of standard reference data source. The FEFF software developed by Professor John Rehr, is used for ab initio multiple scattering calculations of X-ray spectra[2]. Typical signal processing technique such as spectra denoising, alignment and shift were tested and implemented during data preprocessing. The ultimate target of this universal workflow is automatical detection of a spectrums chemical space and potential elements, which also includes a XANES spectrum's absorbing species and crystal structural information of the source material. A rational classification framework built based on neural network is presented in this paper. Parameters utilized in machine learning algorithm could be customized by users. Supervised learning and interpretation of XANES spectra could be achieved using the the developed workflow.

III. PHYSICAL AND MATHEMATICAL FRAMEWORK

The entire classification process of XANES spectra could be divided into three stages.

1. Simulation and preprocessing of spectra.
2. Learning process of autoencoders.
3. Training of classification feed-forward neural network builds on top of trained autoencoders.

For data analysis and query convenience, the simulation and physical framework were setup using MongoDB as the database system. This architecture permits efficiency query and inserting of spectra data. The neural networks were implemented using Tensorflow. All trainings were done on Nvidia K80 GPU. The architecture summary of the framework is shown in Figure 1.

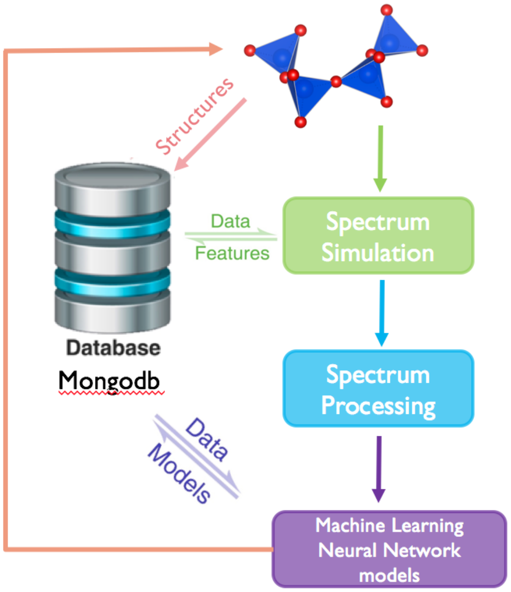


FIG. 1. Framework architecture of analysis workflow.

A. Input data simulation and preprocessing

X-ray absorption fine structure (XAFS) refers to the details of how x-ray are absorbed by an atom at energies near and above the core-level binding energies of that atom (Figure 2). XAFS spectra are very sensitive to the chemical and physical states of atoms. A x-ray absorption spectrum is usually divided into two regimes: x-ray absorption near-edge spectroscopy (XANES) and extended x-ray absorption fine-structure (XAFS). The XANES is more crucial for the analysis of oxidation states and coordination chemistry of a material (Figure 3). The signal intensity of XANES spectra is multiple orders of magnitude higher than XAFS spectra. Therefore, the XANES spectra are more robust to noise at the lower specimen concentrations.

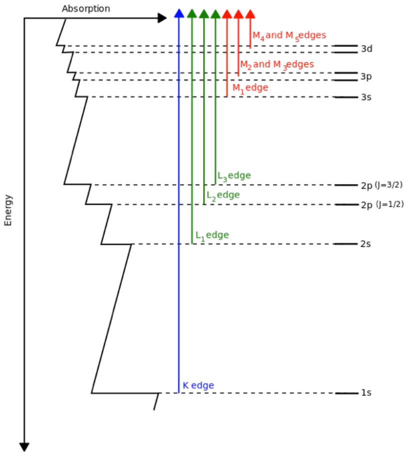


FIG. 2. Relationship between x-ray absorption spectrum and the electronic structure of a material.

The traditional approach of XANES spectra interpretation is mainly based on a straightforward comparison between the collected spectrum and reference spectra collected. This method highly relies on the chemical intuition of researchers. So far, no simple analytic description of XANES has been derived. Clearly, as shown in Figure 3, the edge position and shape of XANES are very sensitive to the valence state of the XANES absorption atom. The shape and edge position of a spectrum could be useful in identifying the phases of a material.

In our project, we collected more than 500 Si K-edge XANES spectra with respect to different chemical compositions using the FEFF software[2]. We divided the entire collection of spectra into 12 subgroups according their compositions. Each subgroup's spectra has a unique chemical system id that represents elements included in the materials. To avoid the inconsistency on absorption energy between different spectra, we applied interpolation using the Scipy.interp1d package. The sampling point density was set to 500 for each spectrum in order to minimize information loss during interpolation. After spectra interpolation, all spectra share the same absorption energy range. In later training, we reduced each XANES spectra to a one dimension vector corresponding to x-ray absorption values. A neural network is then setup for classification task of the entire spetra collection. The neural network structure is shown in Figure 4.

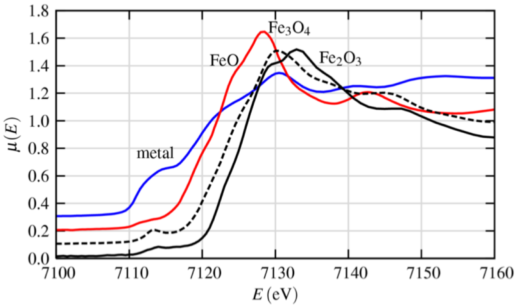


FIG. 3. Fe K-edge XANES spectrum of Fe metal and different FE oxides with different Fe oxidation states.[3]

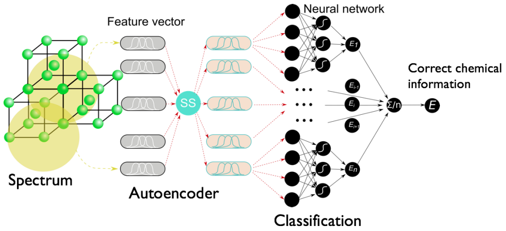


FIG. 4. Structure of spectrum simulation, autoencoders and classification feed-forward neural network

B. Autoencoders

An autoencoder is a network with three or more layers, where the input layer and the output have the same number of neurons, and those intermediate (hidden) layers have a lower number of neurons. The network is trained to reproduce the input data. The same pattern of activity in the input layer will be revealed in the output layer.

In our project, we want to use the autoencoder neural network to improve the learning process. As the overall dataset's size is relatively small. A well-trained autoencoder neural network will be able to denoise our input spectra and keep the most critical information of each spectrum for further classification problem. An autoencoder typically works through two phases:

1. An encoder phase that corresponds to a dimensional reduction for the original input.

2. A decoding phase that is capable of reconstructing

the original input from the encoded (compressed) representation.

During training, both the input and the output of the autoencoders are the 500 spectra. Copying the input to the output may sound useless as we are not interested in the output of the autoencoders. While we hope that training the autoencoders could help us retrieve useful features of the spectra data. The trained autoencoders neural network could project the spectra onto a subspace with lower dimensions. Learning an undercomplete representation, i.e. code dimension of an autoencoder is less than the input dimension, forces the autoencoder to capture the latent features of the training spectra[4]. The mean squared error function[5] was chosen as the cost function (equation 1). As the point of the autoencoder is to create a reduction matrix that is capable of reproducing the original data, i.e. spectra, the input data is identical to the decoded data.

$$MSE = \frac{1}{n}\sum_{n=1}^{N}(\hat{Y}_i - Y_i)^2, \qquad (1)$$

where $\hat{Y}_i$ is the spectra generated by autoencoders and Y_i is the ground truth spectra, i.e. the input spectra.

C. Feed-forward neural networks

The feed-forward neural network (FNN), also known as multi-layer perceptron, is constructed separately. The input of the FNN is the output of the encoder part of the autoencoders. FNN is widely used for supervised pattern classification.[6,7]

We use a five layers model to construct the FNN. The input layer, L_1, is comprised of D input variables, where D is the batch size. In our study, the training of FNN is conducted at $D = 16$. There are three hidden layers in our FNN. Hidden layer L_2 has 100 neurons. Hidden layer L_3 and L_4 has 60 neurons and 30 neurons respectively. In the output layer, L_5, there are 12 neurons. The softmax function is used as the activation function in the output layer L_5. The probability of a spectrum belongs to a class is calculated using the following expression 2:

$$y_k(\mathbf{x},\mathbf{w}) = \frac{\exp(a_k(\mathbf{x},\mathbf{w}))}{\sum_{j=1}^{K}\exp(a_j(\mathbf{x},\mathbf{w}))}, \quad k = 1,\cdots,K \qquad (2)$$

, where $y_k(\mathbf{x},\mathbf{w})$ represents the probability of x from class k. The class label with highest probability is used as the predicted label of the spectrum. We use the cross-entropy loss function[8] in FNN. The loss function of FNN is defined in equation 3:

$$L(w) = \frac{1}{n}\sum_{n=1}^{N}[y_n log\hat{y_n} + (1 - y_n)log(1 - \hat{y_n})], \qquad (3)$$

where N represents the number of samples. $\hat{y}_n$ is the predicted label of sample n. y_n is the true label of the sample.

To test the effect of different activation functions and dropout, we implement four different kinds of configuration of FNN in our study:

1. FNN using logistic sigmoid as the intermediate activation function.

2. FNN using ReLU as the intermediate activation function.

3. FNN using ReLU activation function with the dropout ratio equals 0.8.

4. FNN using a mixture of ReLU and sigmoid activation functions in different layers with the dropout ratio set to 0.8.

.

The logistic sigmoid function is given by the following equation:

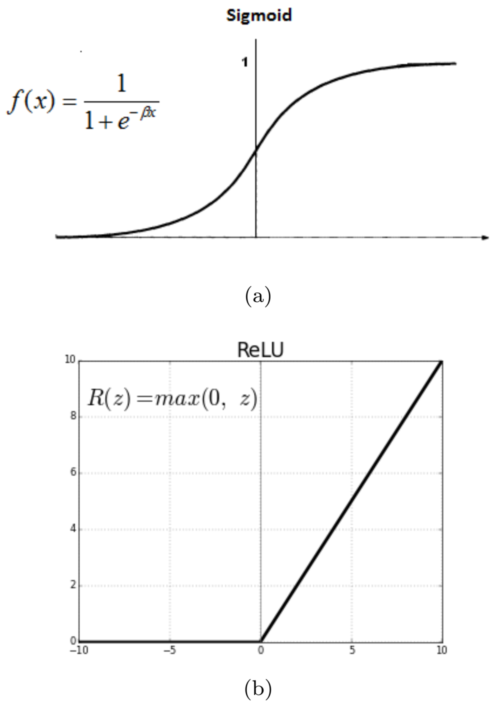
$$f(a) = \sigma(a) = \frac{1}{1 + e^{-a}}, \qquad (4)$$

where a_j is the jth linear combination of the input variable calculated using equation 5.[7]

$$a_j = \sum_{i=1}^{D} w_{ji}x_i + w_{j0}, \qquad j = 1,\cdots,M. \qquad (5)$$

In equation 5, M is the number of neurons in layer j. i is the index number of a neuron in the layer. The parameter w_{j0} is called the biases. Figure 5 shows a comparison between the Sigmoid and the ReLU activation function. A Rectified Linear Unit (ReLU) unit computes the function $f(x) = max(0, x)$. Theoretically, ReLU is computationally more efficient because it does not require any exponential computations, such as those required in Sigmoid or tanh activations. Furthermore ReLU was found to greatly accelerate the convergence of stochastic gradient descent compared to the Sigmoid/tanh functions.[9]

An effective strategy researchers take to improve the model's robustness is introducing a dropout optimization during the training process of FNN.[10]During the learning phase, the connections with the next layer can be limited to a subset of neurons to reduce the weights to be updated, this learning optimization technique is called dropout. This technique decreases the overfitting within a neural network with many layers and/or neurons. In



(a)

(b)

FIG. 5. (a) Diagram of sigmoid activation function (b) ReLU function.

general, the dropout layers are positioned after the layers that possess a lot of trainable neurons. In our study, we test the dropout ratio ranges from 0.4 to 1 with 0.2 as interval.

Several optimization methods are provided in the TensorFlow software. In this paper, Adam[11](Adaptive Moment estimation) is used.

IV. RESULTS

A. Structure and configuration of Autoencoders

Figure 6 shows the decreasing of training cost of autoencoders of different configuration during 200 epochs training. We observe that the performance of NN 2 is the best among all 4 different configurations we tested. As shown in Figure 6, when we reduced the autoencoder layer to one, it takes much longer time for the autoencoder neural network to converge. The training costs of NN 1, NN 2 and NN 3 converge within first 100 epochs, while NN 4 converges after 180 epochs. The neural network configuration information is summarized in Table I. Based on our test, we decide to use the configuration 2 as the autoencoders structure in further classification problem.

To visualize the effectiveness of autoencoders, we directly output the results of encoder part of the NN 1 autoencoders. We project the data on two axes as shown in Figure 7. We notice that on two dimensions the autoencoder successfully captures latent factors of spectra. Even though the performance of configuration 1 is not the best as observed from our previously mentioned tests, the scattering plot indicates that the autoencoder works effectively in separating the dataset linearly in lower dimension space.

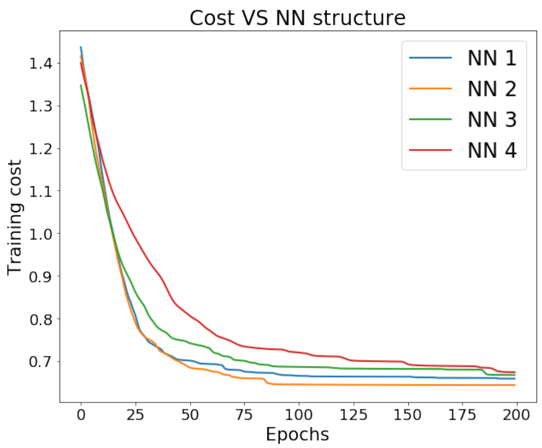


FIG. 6. Mean square root training cost of autoencoders with different configurations.

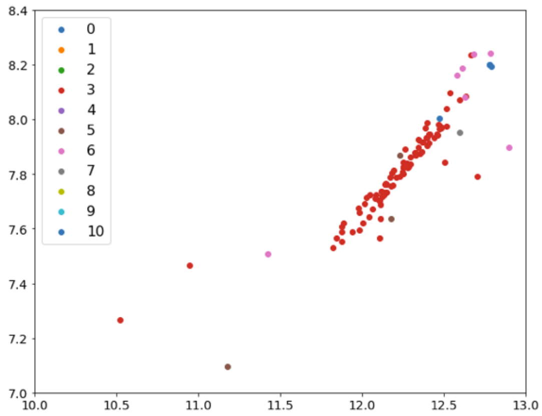


FIG. 7. Scattering plot of spectrum after encoder using autoencoders with configuration 1.

B. Structure and configuration of FNN

From Figure 8, we notice that introduction of dropout optimization decreases the classification accuracy the FNN from 95% to 90% when the dropout ratio was set to 0.8. Instead, the dropout optimization speeds up the convergence of FNN. No significant accuracy fluctuation is observed after 100 training epochs when we drop 20% neuron during training phase by using a dropout ratio equals 0.8. Further increase in dropout ratio has a detri-

Configuration	NN 1	NN 2	NN 3	NN 4
Encoder layer 1	128	128	128	128
Encoder layer 2	64	64	64	-
Encoder layer 3	32	32	-	-
Encoder layer 4	2	-	-	-
Decoder layer 1	2	-	-	-
Decoder layer 2	32	32	-	-
Decoder layer 3	64	64	64	-
Decoder layer 4	128	128	128	128

TABLE I. Configuration of autoencoders neural network tested

mental effect on the classification performance of FNN. The fluctuation on the classification accuracy reappears when the dropout ratio equals 0.6 or 0.4. The classification accuracy does not get affected significantly with respect to the change of dropout ratio. Over 89% classification accuracy is obtained at dropout ratio equals 0.4. The main reason of probably we do not have enough training data for our FNN. Comparing our dataset with the MNIST dataset[12], the size of our dataset is 100 smaller. It is quite surprising that the performance of FNN does not degrade to large extend on a dataset with such small size. The investigation provides an indirect evidence that supports the robustness of FNN in the classification task. In the later investigation regarding the effect of different activation functions, we will keep the dropout ratio at 0.8.

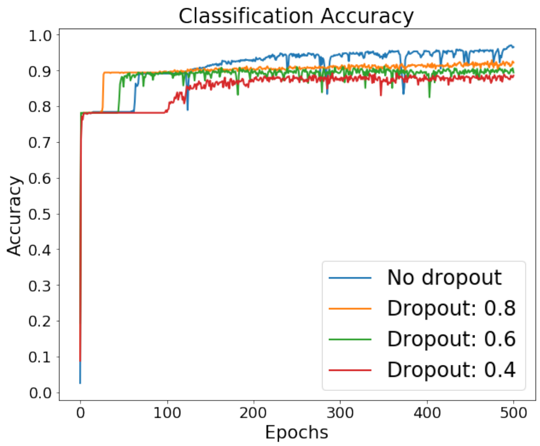


FIG. 8. Effect of dropout in preventing overfitting of FNN.

To obtain a more quantitative assessment of the activation function, we tested four different kinds of FNN architectures. Table II summarized the architectures we implemented in our study. The FNN1 configuration is the default neural network structure we used in our dropout ratio screening. In FNN2, we replaced all four layers' activation function to Sigmoid function. We mixed Relu activation function with Sigmoid activation function in FNN3 and FNN4. The first two layers' activation function of FNN3 is Sigmoid, the last two layers' activation function of FNN4 is ReLU. While in FNN4, the first two layers' activation function is ReLU. To ensure the convergence of training, we trained all models for 500 epochs.

Configuration	FNN1	FNN2	FNN3	FNN4
Layer 1	ReLU	Sigmoid	Sigmoid	ReLU
Layer 2	ReLU	Sigmoid	Sigmoid	ReLU
Layer 3	ReLU	Sigmoid	ReLU	Sigmoid
Layer 4	ReLU	Sigmoid	ReLU	Sigmoid

TABLE II. Configuration of autoencoders tested, the dropout ratio of all 4 configuration equals 0.8.

From the training results, as shown in Figure 9, we have established that same classification accuracy could be achieved with different activation functions. Replacing the ReLU activation function with Sigmoid activation function smooths the accuracy fluctuation occurs after 200 epochs in FNN1.

From Figure 9 and Figure 8, there exists a plateau within first 100 epochs in general. This indicates a saturation of training. It is because the ReLU is activated only above 0 and the gradient is 0 whenever the unit is not active[13]. The vanishing of gradient occurs below 0 in ReLU layer causes slow optimization convergence as shown in Figure 9. When we replaced the ReLU layers partially with Sigmoid layers, we see a shortening of the training saturation plateau. The FNN4 structure performs the best among all four FNN architectures. The FNN4 is able to achieve the highest classification accuracy and second-best convergence speed using a combined ReLU, Sigmoid and dropout approach.

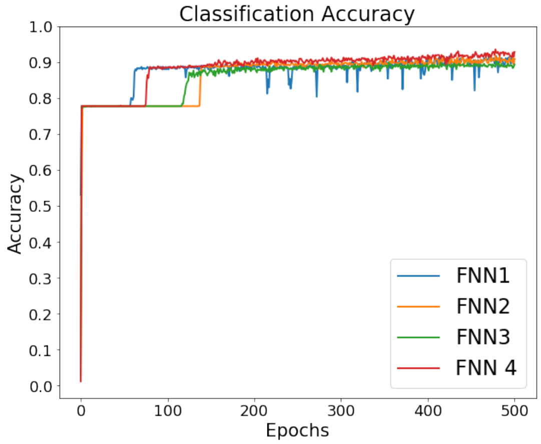


FIG. 9. Effect of activation function in FNN.

V. CONCLUSION

To conclude, we have preformed classfication of Si K-edge XANES spectra using a combination of autoencoders and FNN. The 2D scatter plot suggested that the autoencoder has learned the principle subspace of the training data. Using FNN, we show that the classification of simulated Si K-edge XANES could be achieved with a classification accuracy exceeds 90%. The presence of dropout optimization and a mixture of ReLU/Sigmoid activation function leads to an increase in the robustness and convergence speed of FNN. The FNN architecture with first two layers using ReLU activation function and last two layers using Sigmoid activation function is the recommended one based on our investigation.

VI. REFERENCES

[1] P. Ewels, T. Sikora, V. Serin, C. P. Ewels, and L. Lajaunie, "A Complete Overhaul of the Electron Energy-Loss Spectroscopy and X-Ray Absorption Spectroscopy Database : eelsdb . eu," Microscopy and Microanalysis , 717–724 (2017).

[2] J. J. Rehr, "Theoretical approaches to x-ray absorption fine structure," Reviews of Modern Physics **72**, 621–654 (2000).

[3] M. Newville, "Fundamentals of XAFS," Reviews in Mineralogy and Geochemistry **78**, 33–74 (2014).

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) http://www.deeplearningbook.org.

[5] "Mean squared error," https://en.wikipedia.org/wiki/Mean_squared_error, accessed: 2017-06-15.

[6] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012) pp. 3642–3649.

[7] H. Niu, E. Reeves, and P. Gerstoft, "Source localization in an ocean waveguide using supervised machine learning," ArXiv e-prints (2017), arXiv:1701.08431 [physics.ao-ph].

[8] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (The MIT Press, 2012).

[9] T. M. Breuel, "The effects of hyperparameters on SGD training of neural networks," CoRR **abs/1508.02788** (2015).

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research **15**, 1929–1958 (2014), arXiv:1102.4807.

[11] D. P. Kingma and J. L. Ba, "Adam: a Method for Stochastic Optimization," International Conference on Learning Representations 2015 , 1–15 (2015), arXiv:1412.6980.

[12] Y. LeCun and C. Cortes, "MNIST handwritten digit database," (2010).

[13] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," Proceedings of the 30 th International Conference on Machine Learning **28**, 6 (2013).