Lightweighted Sparse Autoencoder based on Explainable Contribution

Joohong Rheey [1] **Hyunggon Park** [1]

Abstract

As deep learning models become heavier, developing lightweight models with the least performance degradation is paramount. In this paper, we propose an algorithm, SHAP-SAE (SHapley Additive exPlanations based Sparse AutoEncoder), that can explicitly measure the contribution of units and links and selectively activate only important units and links, leading to a lightweight sparse autoencoder. This allows us to explain how and why the sparse autoencoder is structured. We show that the SHAP-SAE outperforms other algorithms including a dense autoencoder. It is also confirmed that the SHAP-SAE is robust against the harsh sparsity of the autoencoder, as it shows remarkably limited performance degradation even with high sparsity levels.

1. Introduction

As deep learning approaches have tackled and solved an increasing number of real-world problems, the demand for improved performance has led to the development of heavier models (Baykal et al., 2022). However, these large and dense networks often require a significant number of floating operations (FLOPs) during inference. Consequently, it is essential to design lightweight models that enhance scalability and efficiency without compromising model quality. The importance of lightweight models becomes even more evident in scenarios where deep learning inference must adhere to stringent energy constraints. This is particularly evident when deploying models on battery-powered devices such as mobile devices and Internet of Things (IoT) devices. Additionally, lightweight models play a crucial role in distributed networks, particularly at the network edge, within the context of federated learning.

[1]Smart Factory Multidisciplinary Program, Department of Electronic and Electrical Engineering, Ewha Womans University, Seoul, Republic of Korea. Correspondence to: Hyunggon Park <hyunggon.park@ewha.ac.kr>.

Accepted after peer-review at the workshop on Neural Compression: From Information Theory to Applications at ICML 2023, Honolulu, Hawaii, USA. July, 2023.

Similar to other deep learning models, autoencoders have become heavier with the focus of *how the autoencoder can effectively compress input data* (Wang et al., 2014; 2016), rather than *how to effectively compress the autoencoder*. Autoencoder is generally (Hinton & Salakhutdinov, 2006) a dense network with fully-connected layers and most of the existing architectures are based on this structure, e.g., (Kaiser & Bengio, 2018; Nguyen et al., 2020; Chen et al., 2022). Due to the lack of research on lightweight autoencoders, we focus on the design of a lightweight autoencoder by imposing sparsity constraints on the hidden units. To make a compressed or sparse autoencoder, it is essential to *identify* which units and links are important in a trained autoencoder that is often dense and then selectively *activate* the units that are more important than other units. The sparsification has been conventionally performed by the combinations of activation functions, sampling steps, and different types of penalties (Makhzani & Frey, 2014). While this enables autoencoders to be sparse and efficient, the sparse autoencoders are often lack of explainability or interpretability (Makhzani & Frey, 2014; Srivastava et al., 2014; Pal & Baskar, 2015).

In this paper, we propose a novel SHAP-SAE (SHapely Additive exPlanations based Sparse AutoEncoder) algorithm that can make autoencoders sparse with explainability. Unlike prior works (Lundberg & Lee, 2017; Catav et al., 2021; Harris et al., 2022), where the Shapely value (Shapley, 1953) is used to measure the feature importance of input data, we use the Shapley value to explicitly quantify the importance of the units and links in an autoencoder. This enables us to identify the units or links that are with higher importance, and thus, the autoencoder can be sparsely represented by only activating the units and links with higher Shapley values. Note that this approach is providing not only a way of pruning the links but also a way of explaining how the sparse autoencoder works, i.e., the links marked as higher importance are only activated in the sparse autoencoder. Moreover, the proposed measure of unit and link importance can permit us to directly control the sparsity of the autoencoder, as units or links with low importance can be pruned to meet a target sparsity level. The proposed SHAP-SAE algorithm can completely remove the links with low importance by assigning zero weight to the pruned links in the sparse autoencoder. This property allows for a reduction

in computational complexity during inference.

2. Shapley Value based Sparse Autoencoder

2.1. Overview of Autoencoder

Consider an autoencoder that consists of an encoder f and a decoder g, where both encoder and decoder have L layers, respectively. The encoder maps input $\mathbf{x}$ into a representation $\mathbf{z} = f(\mathbf{x})$. The number of units included in the l-th hidden layer is denoted by $n^{(l)}$. In the encoder, we assume that $n^{(l)} \leq n^{(l-1)}$, for $1 \leq l \leq L$, as dimensions are reduced over layers. $\mathbf{b}^{(l)} \in \mathbb{R}^{n^{(l)}}$ denotes a bias vector in the l-th hidden layer. A weight matrix $\mathbf{W}^{(l)} \in \mathbb{R}^{n^{(l)} \times n^{(l-1)}}$ can be expressed as

$$\mathbf{W}^{(l)} = \begin{bmatrix} -\mathbf{w}_1^{(l)^T} - \\ \vdots \\ -\mathbf{w}_{n^{(l)}}^{(l)^T} - \end{bmatrix} \tag{1}$$

where the weight vector for the k-th unit in the l-th layer $\mathbf{w}_k^{(l)} \in \mathbb{R}^{n^{(l-1)}}$ is given by

$$\mathbf{w}_k^{(l)} = \begin{bmatrix} w_{k1}^{(l)} & w_{k2}^{(l)} & \cdots & w_{kn^{(l-1)}}^{(l)} \end{bmatrix}^T. \tag{2}$$

The decoder of the autoencoder maps the representation $\mathbf{z} \in \mathbb{R}^{n^{(L)}}$ into its reconstruction $\hat{\mathbf{x}} \in \mathbb{R}^{n^{(0)}}$ of input $\mathbf{x} \in \mathbb{R}^{n^{(0)}}$, i.e., $\hat{\mathbf{x}} = g(\mathbf{z})$. The decoder has a symmetric structure to the encoder, so we assume that $n^{(2L-l)} = n^{(l)}$ for $0 \leq l \leq L$. Specifically, the decoder corresponds to the layers from the $(L+1)$th layer to the $2L$-th layer of the autoencoder. $\mathbf{W}'^{(l)} \in \mathbb{R}^{n^{(l)} \times n^{(l-1)}}$ and $\mathbf{b}'^{(l)} \in \mathbb{R}^{n^{(l)}}$ are weight matrix and bias vector in the l-th hidden layer of the autoencoder, respectively.

The goal is to determine the set of optimal parameters for the autoencoder, $\theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)} | \ 1 \leq l \leq L\}$ for the encoder and $\theta' = \{\mathbf{W}'^{(l)}, \mathbf{b}'^{(l)} | \ L+1 \leq l \leq 2L\}$ for the decoder, by minimizing the loss associated with the reconstruction error, i.e., $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$. Hence, the optimal parameters θ^* and θ'^* for the minimum reconstruction error are expressed as

$$\{\theta^*, \theta'^*\} = \underset{\theta, \theta'}{\operatorname{argmin}} \, \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \underset{\theta, \theta'}{\operatorname{argmin}} \, \mathcal{L}(\mathbf{x}, g(f(\mathbf{x}))).$$

2.2. Importance of Link based on Shapley Value

In order to estimate the contribution of each link in the autoencoder, we use the Kernel SHAP method, which approximate SHAP values based on LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al., 2016) and Shapley values (Shapley, 1953). To measure the importance of links based on their contributions to the output of a layer, where the output of the layer is computed by the weights of the links, we define the link importance (LI) based on the Shapley value. The LI of the link that connects the j-th unit in the $(l-1)$th layer and the i-th unit in the l-th layer is denoted by $\phi_{ij}^{(l)}$ and is defined as

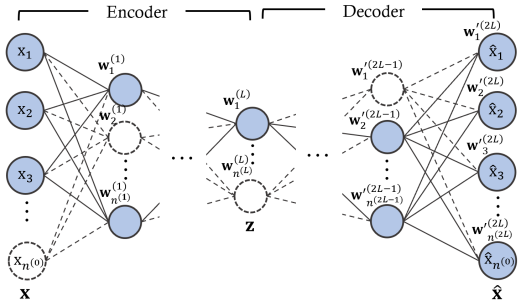


Figure 1. An illustration of sparse autoencoder based on SHAP-SAE algorithm with $2L$ layers. Dotted lines and white nodes indicate the removed links and nodes.

$$\phi_{ij}^{(l)} = \sum_{J \subseteq I \setminus \{j\}} \frac{|J|!(|I| - |J| - 1)!}{I!} (v(J \cup \{j\}) - v(J)), \tag{3}$$

where I is a set of links that are connected to i-th unit in the l-th layer for $1 \leq l \leq 2L$. $J(\subseteq I)$ denotes a subset excluding j-th link that is connected to the i-th unit in the l-th layer. In this paper, the Kernel SHAP method is used to compute $v(\cdot)$. Note that $l = 0$ means the input layer of the autoencoder, and thus, $\phi_{ij}^{(1)}$ is LI of the link that connects the j-th unit in the input layer and the i-th unit in the first layer.

In order to measure the impact of each unit on its next layer, we define the unit importance (UI). The UI of the j-th unit in the $(l-1)$th layer is expressed as

$$\overline{v}_j^{(l-1)} = \frac{1}{n^{(l)}} \sum_{i=1}^{n^{(l)}} |\phi_{ij}^{(l)}|, \text{ for } 1 \leq l \leq 2L, \tag{4}$$

which represents an average impact on the computation of output in the l-th layer. Hence, a larger value implies a greater impact on its next layer.

2.3. SHAP-SAE

Let $\{\mathbf{W}, \mathbf{W}'\}$ be the set of parameters in a trained autoencoder that is a fully connected network. In order to make a lightweight sparse autoencoder, we design a mask function $\mathcal{M}$ that can activate only important links in the trained autoencoder. The importance of each link is measured by Equation (3) and the masking process is performed in the sparsification stage. The results of the masking process in the encoder and decoder are denoted by

$$\mathbf{W}^* = \mathcal{M}(\mathbf{W}), \text{and } \mathbf{W}'^* = \mathcal{M}(\mathbf{W}'), \tag{5}$$

respectively, where the elements in $\mathbf{W}^*$ or $\mathbf{W}'^*$ become zero if they are considered as unimportant by $\mathcal{M}$. An illustration of a sparse autoencoder based on SHAP-SAE is shown in Figure 1.

We define the total LI $\phi_T^{(l)}$ of the l-th layer as the sum of individual LIs in its layer, i.e.,

$$\phi_T^{(l)} = \sum_{i=1}^{n^{(l)}} \sum_{j=1}^{n^{(l-1)}} \phi_{ij}^{(l)}, \; 1 \leq l \leq 2L. \tag{6}$$

Moreover, the set of descending ordered Shapely values in the l-th layer is expressed as

$$\mathbf{\Phi}^{(l)} = \left[\Phi^{(l)}(1), \Phi^{(l)}(2), \ldots, \Phi^{(l)}(n^{(l-1)}n^{(l)})\right] \tag{7}$$

where $\Phi^{(l)}(k) \geq \Phi^{(l)}(k+1)$ for integer k $(1 \leq k < n^{(l-1)}n^{(l)})$. With an *importance level* denoted by m $(0 < m \leq 1)$, the support set $\Gamma^{(l)}$ is constructed as

$$\Gamma^{(l)} = \left\{ (i,j) \,\middle|\, \sum_{k=1}^{k^*} \Phi^{(l)}(k) \geq m \cdot \phi_T^{(l)} \right\}, \tag{8}$$

which is the set of the pairs (i,j) of the units i and j that have the k^* largest contribution. In other words, since each $\Phi^{(l)}$ corresponds to the value of $\phi_{ij}^{(l)}$ in the descending order, the pairs (i,j) of the link that has the k^* largest LIs in the layer are elements of $\Gamma^{(l)}$. The set of other elements that are not included in $\Gamma^{(l)}$ is denoted by $\Gamma^{(l)\mathsf{c}}$. Correspondingly, the *sparsity level* η can be computed as

$$\eta = \frac{\sum_{l=1}^{2L} |\Gamma^{(l)\mathsf{c}}|}{\sum_{l=1}^{2L} |\Gamma^{(l)} \cup \Gamma^{(l)\mathsf{c}}|}. \tag{9}$$

The mask function $\mathcal{M}$ is a simple mapping for an element $w_{ij}^{*(l)}$ or $w_{ij}^{\prime *(l)}$ such that

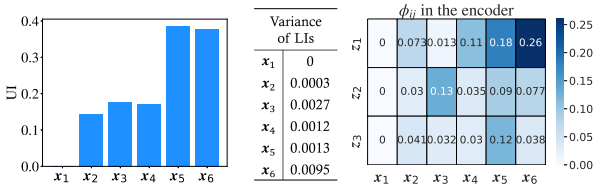
$$\begin{cases} w_{ij}^{*(l)} = 0, & \text{if } (i,j) \in \Gamma^{(l)\mathsf{c}} \\ w_{ij}^{\prime *(l)} = 0, & \text{if } (i,j) \in \Gamma^{(l)\mathsf{c}} \end{cases} \tag{10}$$

and the rest of the weights remain unchanged. Note that the unit should be removed if all links connected from the unit are deactivated, regardless of the mask function.

3. Experiment Results

3.1. SHAP-SAE with Synthetic Dataset

Dataset. To confirm the performance of the proposed SHAP-SAE, we first consider the synthetic dataset that consists of 15,000 instances. $\boldsymbol{x}_1$ is a set of values that are constant. $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are generated from the uniform distribution $U(a,b)$ of the interval $[a,b]$. The data in $\boldsymbol{x}_2$ is sampled from $U(0,1)$.



(a) (b) (c)

Figure 2. Statistics of importance value, including **(a)** The UIs, **(b)** the variance of LI, and **(c)** the LIs, in the encoder for the synthetic dataset.

$\boldsymbol{x}_3$ contains the data equally sampled from two classes, one from $U(0,1/4)$ and the other from $U(3/4,1)$. Similarly, $\boldsymbol{x}_4$ contains the data sampled from two classes, one from $\mathcal{N}(1/4,0.1)$ and the other from $\mathcal{N}(3/4,0.1)$, where $\mathcal{N}(\mu,\sigma^2)$ denotes Gaussian distribution. The other two data sets $\boldsymbol{x}_5$ and $\boldsymbol{x}_6$ are generated by the sum of other data sets, i.e., $\boldsymbol{x}_5 = \boldsymbol{x}_1 + \boldsymbol{x}_2$ and $\boldsymbol{x}_6 = \boldsymbol{x}_3 + \boldsymbol{x}_4$.

Sparsification. We consider a simple autoencoder structure that has an input layer with six units, one hidden layer with three units, and an output layer with six units, i.e., $L = 1$. Given the synthetic data sets, the weight matrices $\{\mathbf{W}, \mathbf{W}'\}$ of the autoencoder are determined in the training stage.

For the SHAP-SAE, some of the activated links in $\{\mathbf{W}, \mathbf{W}'\}$ of the trained autoencoder can be deactivated in the sparsification stage. In the experiments, we set the importance level $m = 0.8$. The support sets $\Gamma^{(1)}, \Gamma^{(2)}$ at the first layer and second layer are constructed as

$$\Gamma^{(1)} = \{(1,2),(1,4),(1,5),(1,6),(3,5)\},$$

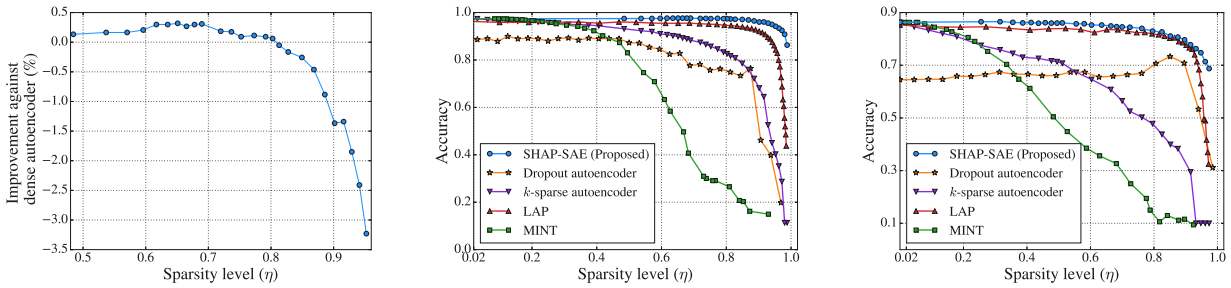
and

$$\Gamma^{(2)} = \{(2,1),(2,3),(3,1),(3,3),(4,1),(5,3),(6,1)\}$$

with the parameters of $k^* = 8$ and $\Phi^{(l)}(8) = 0.073$ for $l = 0$. Note that $|\Gamma^{(1)}| = 5$ because three pairs of $(2,3),(2,5),(2,6)$ that were included in the eight elements in $\Gamma^{(1)}$ are excluded as unit z_2 is not activated.

Explainability. To discuss the explainability of the proposed SHAP-SAE, we quantify how much the UIs of feature $\boldsymbol{x}_j$ attribute to the change of the representation in the hidden layer. $\overline{v}_j^{(0)}$ represents an average impact of $\boldsymbol{x}_j$ on the computation of $\mathbf{z}$ in the hidden layer.

The UIs for data sets are shown in Figure 2(a). It is clearly observed that $\overline{v}_5^{(0)}$ and $\overline{v}_6^{(0)}$ are larger than other UIs of data sets. This is because $\boldsymbol{x}_5 = \boldsymbol{x}_1 + \boldsymbol{x}_2$ and $\boldsymbol{x}_6 = \boldsymbol{x}_3 + \boldsymbol{x}_4$, so that they can include the information of other data sets. Hence, larger contributions can be made by $\boldsymbol{x}_5$ and $\boldsymbol{x}_6$ to the training of the autoencoder. This also means that the contribution of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_4$ to the training of autoencoder could be marginal, as they can be considered as redundant



(a) (b) (c)

Figure 3. **(a)** Performance improvement of SHAP-SAE against dense autoencoder on the MNIST dataset. Performance of lightweighted sparse autoencoders based on SHAP-SAE algorithm and other pruning algorithms on **(b)** the MNIST dataset and **(c)** the Fashion-MNIST dataset.

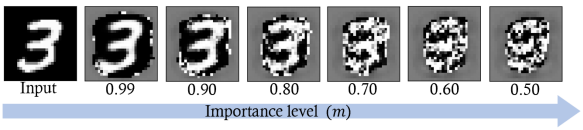


Figure 4. Outputs of SHAP-SAE depending on importance levels on the MNIST dataset.

to $\boldsymbol{x}_5$ or $\boldsymbol{x}_6$. The UIs also confirm this explanation, i.e.,

$$\overline{v}_5^{(0)} > \overline{v}_2^{(0)}, \quad \overline{v}_6^{(0)} > \overline{v}_3^{(0)} \text{ and } \quad \overline{v}_6^{(0)} > \overline{v}_4^{(0)}.$$

Note that $\overline{v}_1^{(0)} = 0$ because $\boldsymbol{x}_1$ is the set of constant values, which is obviously irrelevant to $\mathbf{z}$, i.e., $\boldsymbol{x}_1$ can be considered dummy so that no contributions are made.

In order to analyze the impact of the distribution of input data on the UIs and LIs, we study the variance of LIs. It is observed from Figure 2(b) that the data sets with similar UIs may have different variances. For example, $\boldsymbol{x}_5$ and $\boldsymbol{x}_6$ have similar UIs, but the variance of LIs related to $\boldsymbol{x}_5$ is significantly low. This is because $\boldsymbol{x}_5$ is uniformly distributed over the entire input range so that it can evenly affect all units in the next layer. This is similar to $\boldsymbol{x}_2$, which is also uniformly distributed. However, other data sets, such as $\boldsymbol{x}_3$, $\boldsymbol{x}_4$ and $\boldsymbol{x}_6$, show larger variances, meaning that their impact on the next units is more focused as shown in Figure 2(c).

3.2. SHAP-SAE with Real-World Dataset

Datasets. We consider MNIST (LeCun et al., 2010) and Fashion MNIST datasets (Xiao et al., 2017). Each image in the datasets is reshaped into a column vector and the pixel values are normalized in the range of $[0, 1]$.

Performance Analysis. Figure 3(a) shows the performance improvement of the SHAP-SAE in terms of accuracy compared to the dense autoencoder. While it can be expected that the performance degrades as the autoencoder becomes sparser, interestingly, we can observe that the SHAP-SAE outperforms the dense autoencoder up to the sparsity level $\eta = 0.8$. This is because the initial pruning may lead to the reduction of learned noise following the principle of Occam's hill (Rasmussen & Ghahramani, 2000). Intuitively, the smaller model may enforce the learning process to *focus* on more important and general aspects of the models. Figure 4 visualizes the outputs of SHAP-SAE with different importance levels m, where gray pixels represent the locations where the weights are zero. As importance level m increases, the mask function $\mathcal{M}$ removes less important weights, so that edges of the images are removed first. Since the impact of the edges on the classification could be marginal, the performance degradation is limited (e.g. only 1.37% performance degradation with 0.90 sparsity level).

Performance Comparisons. Figure 3(b) and Figure 3(c) show the experimental results comparing the performance of the SHAP-SAE with other autoencoder pruning algorithms, namely the k-sparse autoencoder (Makhzani & Frey, 2014) and Dropout autoencoder (Srivastava et al., 2014), as well as other neural network pruning algorithms, namely LAP (Look Ahead Pruning) (Park et al., 2020) and MINT (Mutual Information-based Neuron Trimming) (Ganesh et al., 2021). It is clearly shown that the SHAP-SAE outperforms all other benchmarks over the range of sparsity levels. Note that the SHAP-SAE is remarkably robust against the sparsity of the autoencoder.

4. Conclusion

In this paper, we propose the SHAP-SAE algorithm to design a lightweight autoencoder. The SHAP-SAE algorithm can explicitly measure the unit and link importance of an autoencoder based on the Shapely value so that only important units and links can be activated. This allows the sparse autoencoder to be explainable and robust against high sparsity levels. Experimental results show that SHAP-SAE outperforms the other pruning algorithms.

Acknowledgements

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0- 00739, Development of Distributed/Cooperative AI based 5G+ Network Data Analytics Functions and Control Technology), and in part by the Korea Foundation for Women In Science, Engineering and Technology (WISET) grant funded by the Ministry of Science and ICT (MSIT) under the team research program for female engineering students.

References

Aas, K., Jullum, M., and Løland, A. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.

Baykal, C., Dikkala, N., Panigrahy, R., Rashtchian, C., and Wang, X. A theoretical view on sparsely activated networks. In *Advances in Neural Information Processing Systems*, 2022.

Blalock, D., Gonzalez Ortiz, J. J., Frankle, J., and Guttag, J. What is the state of neural network pruning? *Proceedings of Machine Learning and Systems*, 2:129–146, 2020.

Catav, A., Fu, B., Zoabi, Y., Meilik, A. L. W., Shomron, N., Ernst, J., Sankararaman, S., and Gilad-Bachrach, R. Marginal contribution feature importance-an axiomatic approach for explaining data. In *International Conference on Machine Learning*, pp. 1324–1335. PMLR, 2021.

Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., and Wang, J. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

Cohen, S., Dror, G., and Ruppin, E. Feature selection via coalitional game theory. *Neural Computation*, 19(7): 1939–1961, 2007.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021.

Ganesh, M. R., Corso, J. J., and Sekeh, S. Y. Mint: Deep network compression via mutual information-based neuron trimming. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8251–8258. IEEE, 2021.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems*, 28, 2015.

Harris, C., Pymar, R., and Rowat, C. Joint shapley values: a measure of joint feature importance. In *International Conference on Learning Representations*, 2022.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science*, 313 (5786):504–507, 2006.

Kaiser, Ł. and Bengio, S. Discrete autoencoders for sequence models. *arXiv preprint arXiv:1801.09797*, 2018.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Lee, N., Ajanthan, T., and Torr, P. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2018.

Lin, Z., Liu, J. Z., Yang, Z., Hua, N., and Roth, D. Pruning redundant mappings in transformer models via spectral-normalized identity prior. In *Findings of the Association for Computational Linguistics: EMNLP*, 2020.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017.

Makhzani, A. and Frey, B. K-sparse autoencoders. In *International Conference on Learning Representations*, 2014.

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017.

Molchanov, P., Mallya, A., Tyree, S., Frosio, I., and Kautz, J. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, 2019.

Ng, A. Sparse autoencoder. *CS294A Lecture notes*, 72 (2011):1–19, 2011.

Nguyen, K., Nguyen, S., Ho, N., Pham, T., and Bui, H. Improving relational regularized autoencoders with spherical sliced fused gromov wasserstein. In *International Conference on Learning Representations*, 2020.

Pal, A. and Baskar, S. Speech emotion recognition using deep dropout autoencoders. In *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, pp. 1–6, 2015.

Park, S., Lee, J., Mo, S., and Shin, J. Lookahead: A far-sighted alternative of magnitude-based pruning. In *International Conference on Learning Representations*, 2020.

Rasmussen, C. and Ghahramani, Z. Occam's razor. *Advances in Neural Information Processing Systems*, 13, 2000.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?" explaining the predictions of any classifier. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Shapley, L. S. A value for n-person games. In Kuhn, H. W. and Tucker, A. W. (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Sundararajan, M. and Najmi, A. The many shapley values for model explanation. In *International Conference on Machine Learning*, pp. 9269–9278. PMLR, 2020.

Wang, H., Qin, C., Bai, Y., Zhang, Y., and Fu, Y. Recent advances on neural network pruning at initialization. In *International Joint Conference on Artificial Intelligence*, 2021.

Wang, W., Huang, Y., Wang, Y., and Wang, L. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 490–497, 2014.

Wang, Y., Yao, H., and Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

Williamson, B. and Feng, J. Efficient nonparametric statistical inference on population feature importance using shapley values. In *International Conference on Machine Learning*, pp. 10282–10291. PMLR, 2020.

Wortsman, M., Farhadi, A., and Rastegari, M. Discovering neural wirings. *Advances in Neural Information Processing Systems*, 32, 2019.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.