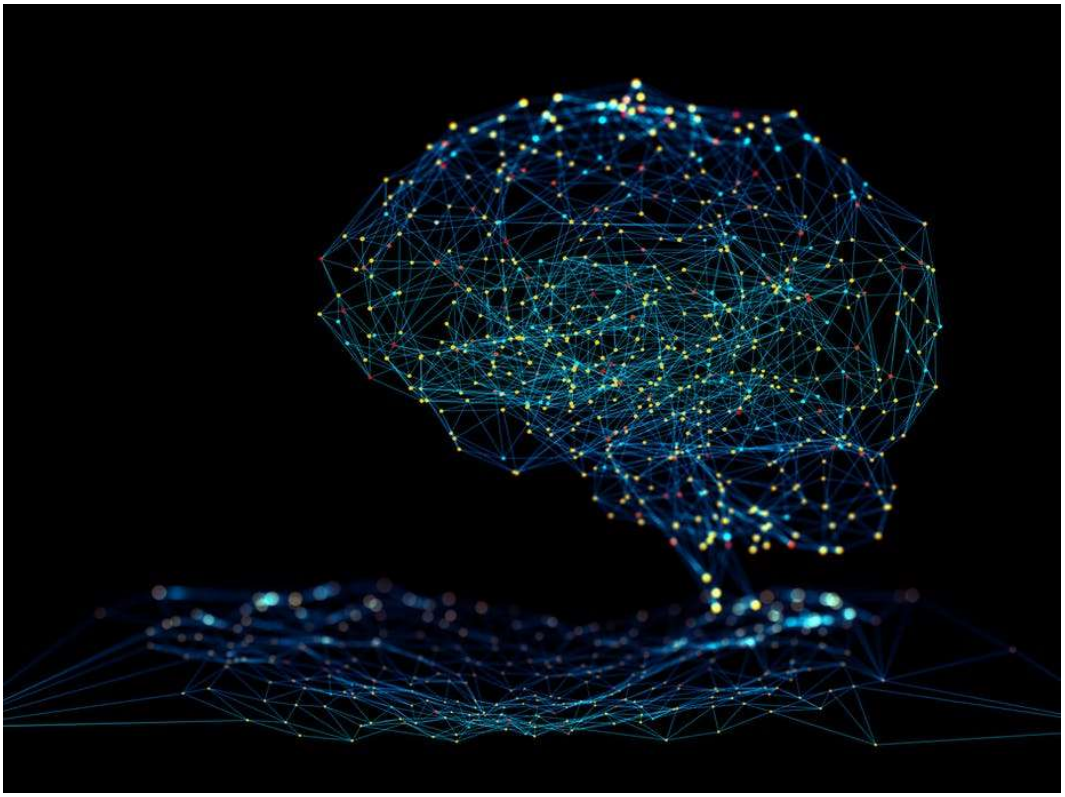


Understanding deep neural networks with clustering analysis and structural causal model

A study cracking the blackboxness of artificial neural networks with innovation

Master's thesis in Data Science and AI

KAVER EDWIN HUI

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2022

Understanding deep neural networks with clustering analysis and structural causal model

A study cracking the blackboxness
of artificial neural networks with innovation

Kaver Edwin Hui

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2022

Understanding deep neural networks with clustering analysis
and structural causal model
A study cracking the blackboxness
of artificial neural networks with innovation
KAVER EDWIN HUI

Supervisor: Viktor Rehnberg, Department of Physics
Examiner: Mats Granath, Department of Physics

Master's Thesis 2022
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Artificial neural networks (ANNs) are a subset of machine learning which mimicks human brain [1].

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2022

Understanding deep neural networks
with clustering analysis and structural causal model
A study cracking the blackboxness
of artificial neural networks with innovation

Kaver Edwin Hui
Department of Physics
Chalmers University of Technology

Abstract

The blackbox problem of machine learning algorithms has been limiting human trust in deep neural networks' decisions. While some researchers use clustering analysis to enhance networks' transparency, some use attribution method to improve the interpretability. This raises a question "how it will be if we apply both approaches on a deep neural network at once". The current study is an exploratory study that investigates the possibilities of relieving the blackbox problem by combining clustering analysis and a structural causal modelling-based attribution method. The study is developed on a conditional β variational autoencoder (β-VAE). It estimates the average causal effects (ACEs) of the decoder's inputs on the hidden layers and reconstruction layer, then conducts ACE-based and activation-based clustering analysis. We apply lesion test experiments to identify the clusters that are important to image reconstruction and answer our question "if there is a non-empty intersection of the two important clusters that contains neurons which are critical to image reconstruction". The results show that 1) there is one input neuron carrying the rotation and scaling roles in image reconstruction, 2) this neuron has positive ACEs on the stroke of the targeted digit, 3) both of the activation-based and ACE-based clustering analysis give us at least 5 clusters, 4) there is always one cluster that is important to image reconstruction, 5) the intersection of the two important clusters is not empty and contains neurons that are critical to image reconstruction and 6) among those critical neurons, there are only 4 to 21 hidden neurons in contrast to our decoder which has 896 hidden neurons. The findings suggest that there may exist the measure of chance of reducing the cost of training deep neural networks and protecting networks from being hacked by focusing on the critical hidden neurons.

Keywords: neural networks, conditional variational autoencoder, clustering, structural causal models

Acknowledgements

Till now I still remember how my memories and thoughts scattered around my room, liked a piece of paper being torn into thousand pieces and thrown everywhere. I didn't know it was PTSD until I talked to a psychologist working for refugees. Unfortunately she couldn't help, so I confronted it in my room alone while most of my group mates relied on my inputs so they could rest or enjoy their Scandinavian journey.

When the school was restored, I was lucky to meet Bernhard Mehlig who taught me artificial neural networks. His voice from the top of Gustaf Dahlénsalen relighted my passion in life and sped up my steps upstairs. I felt like arriving a new world when I put my step in the lecture hall. His teaching fulfilled a dream that I had never been aware.

In winter, all students tried hard to find their topic and supervisor. I had ideas in mind but I have seldom been agreed. A day I saw a proposal by Viktor Rehnberg, I saw some of his values in it and searched his profile online. I believed that we shared something in common, so I wanted him to be my supervisor. However I hesitated because I had only one class in clustering and Viktor's work was much more than that. I was worried and tried to talk with Viktor. Afterwards I believed that Viktor would be supportive and he proved my belief correct.

This project is difficult for several reasons. I am good at connecting dots, this makes the project covers a wide range of subjects. I also like using cutting edge techniques to make a breakthrough in existing problems. In machine learning this could be difficult as each of them has its limitations and the mechanism is difficult to be studied. Moreover, although my research background in medical sector helped a lot throughout the current project, the research methodology in engineering is much different. It took me much more time to examine and study each paper.

Fortunately Viktor was patient and open. He allowed me to learn with my pace and method, and demonstrated me personal qualities that make a machine learning engineer. I enjoyed our weekly meeting and appreciated every moment we had created in the project, for the whole journey gave me time to explore, try, learn, comfort my weariness and recover from my PTSD. The project is much more than it is described in this report.

Countless thanks to Viktor for his support. Many thanks to Marina Axelson-Fisk for her approval to this innovative project and my appeal during admission. Thank you Mats Granath for his interest in this project and attendance of our weekly meeting. I also want to take this chance to thank our President Stefan Bengtsson who had approved my appeal and was responsive to his students.

Kaver Edwin HUI
Kålltorp, 2022 summer

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ACE	Average Causal Effect
ANN	Artificial neural network
SCM	Structural Causal Model
VAE	Variational Autoencoder

Contents

List of Figures

1
Introduction

1.1 Background

Artificial neural networks (ANNs) are a subset of machine learning, their name and structure are inspired by the human brain. They mimick the way that biological neurons signal to one another [1]. Deep learning, as a part of machine learning methods that is based on ANNs, inherits their characteristics [2]. They are powerful tools being used in information processing and pattern recognition which allow them to serve in various fields such as geotechnical engineering and economics [3, 4].

However, ANNs are also criticised of being opaque and hard to interpret due to the lack of transparency behind their behaviors [5]. For many domains this drawback limits the success in deploying deep learning systems. One example is the application in medical sector where end-user and healthcare workers must understand the reasoning behind the prediction models in order to accept or reject the suggested predictions [6], which is a challenge for ANNs since their algorithmic outcomes, as well as those from other machine learning methods, are probabilistic and do not reflect if any causal relationship exists [7].

This limitation is called "the blackbox problem" [8] and motivates research interests in its properties, interpretability and transparency, which will be introduced below.

1.2 Transparency

Zachary [9] has stated that transparency is the opposite of opacity or blackboxness that answers the question of "how does the model work". He considers transparency at three levels: the level of the entire model (simulatability), the level of individual components (decomposability) and the level of the learning algorithm (algorithmic transparency), of which modern deep learning methods that use heuristic optimisation procedures lack algorithmic transparency as it is difficult to figure out how they work and could not guarantee that they would work on unseen datasets.

Filan et al. [10] have mentioned that modular systems are desirable in terms of transparency as they allow those analyzing the system to inspect the function of individual modules so to understand the entire system. On the other hand, Hod et al. [11] have revealed groups of neurons that are important and coherent through graph-based partitioning. Both studies demonstrated the clusterability of deep neural networks, while the latter attested the former's statement.

Although clustering deep neural networks helps enhancing the transparency, the clustering is done with mathematical associations between neurons. The lack of interpretability becomes an obstacle in convincing stakeholders to trust the model's decisions, which is particularly important in medical sector [12].

1.3 Interpretability

It is difficult to give a formal technical definition for interpretability. In 2017, Zachary [9] has stated that post-hoc interpretability presents a distinct approach to extracting information from learnt models. Although this does do not explain how a model works, it might give useful information for machine learning engineers and end users.

Later, Mengnan et al. [5] have differentiated interpretability into two types: local interpretability which tries to figure out why the model makes the decision it makes and global interpretability which means that users can understand how the model works globally by inspecting the structures and parameters of a complex model. With these definitions, global interpretability can increase models' transparency, while local interpretability will help uncover the causal relations between a specific input and its corresponding model prediction.

To achieve interpretability in machine learning, Ancona et al. [13] have summarised three approaches. One of them builds explanation methods on top of existing models. This allows generating explanations for any existing black-box model without retraining the models and sacrificing their performances. Attribution methods belong to this approach and operate in the scope of local interpretability. One example is the new attribution method that has been proposed by Chattopadhyay et al. [14] for neural networks using first principles of causality.

1.4 Objective

Since studies either use clustering analysis or attribution methods to relieve the blackbox problem. This raises a question "how it will be if we apply both approaches on a deep neural network at once". Therefore, the current study is an exploratory study that aims to investigate the possibilities of relieving the blackbox problem by the combining the application of clustering analysis and the attribution method that is based on structural causal modelling [14].

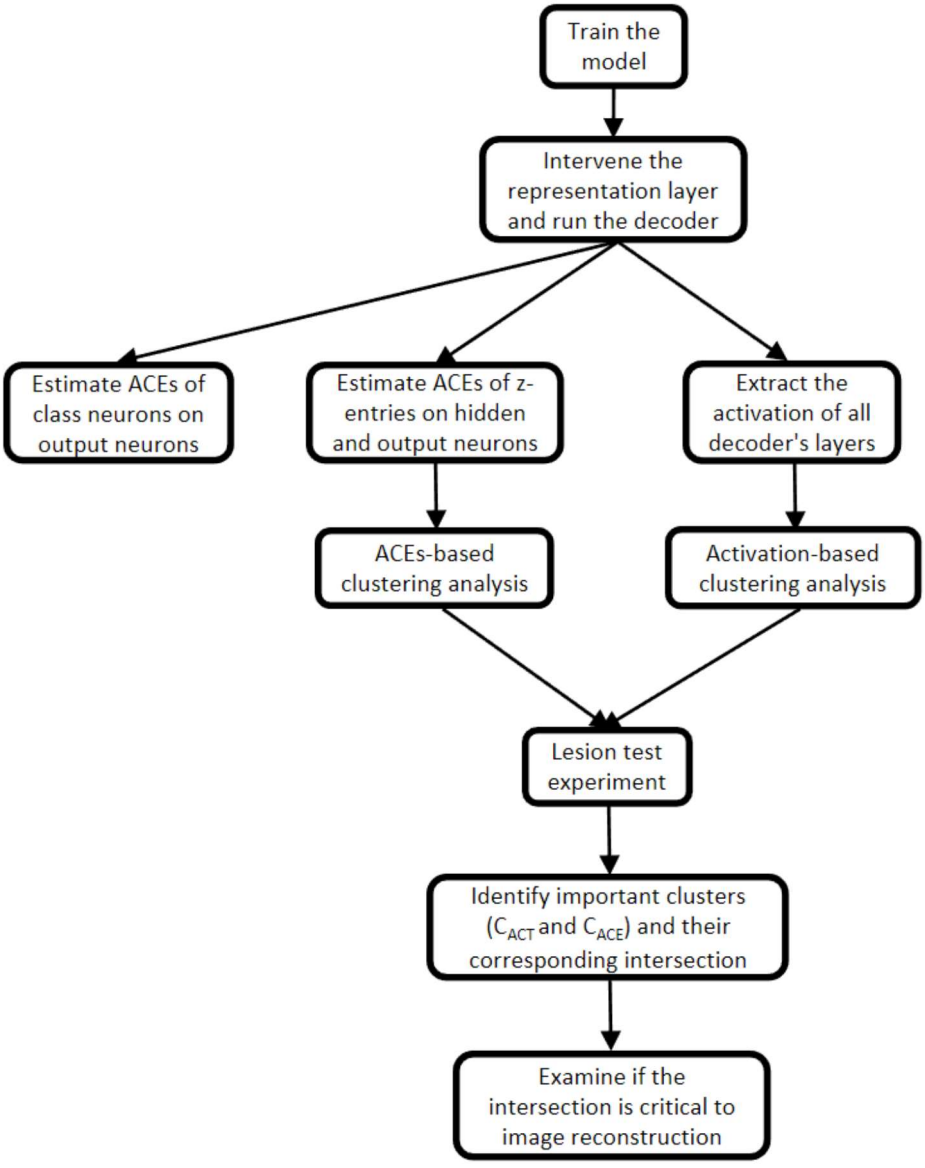


Figure 1.1: Flowchart of the study design

2

Methods

2.1 Model and Dataset

In this study, we use a 10-layer conditional Beta variational autoencoder (β-VAE) as our deep neural network [15]. β-VAE is a generative model that contains an encoder, a representation layer and a decoder (fig. 2.1). Conditional β-VAE allows us to give additional information to the model for better learning.

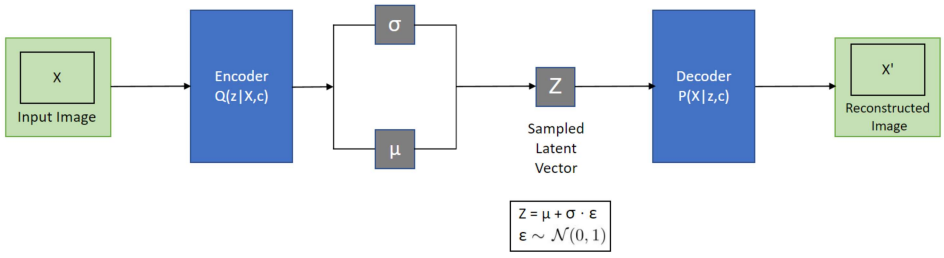


Figure 2.1: Structure of a variational autoencoder

The additional information in this study is the class label of the input image. We denote it as c, a one-hot vector. X is the image being fed into the model, while z is the latent variable in the representation layer. The encoder learns $Q(z|X,c)$ while the decoder learns $P(X|z,c)$, where $Q(z|X,c)$ is a function that is used to infer the prior $P(z|c)$.

Equation 1 shows the loss function of a conditional β-VAE. The Kullback-Leiber Divergence in the second term measures the difference between $Q(z|X,c)$ and the prior $P(z|c)$. The smaller it is, the better the encoder learns.

The hyperparameter β is a regularisation coefficient that balances the reconstruction accuracy and the degree of disentanglement in latent representations that are learnt [15]. When $\beta = 1$, it is a regular conditional VAE. Note that if no additional information is provided to the model, the loss function is changed simply by removing the condition c from the equation.

$$-\max \mathbb{E}[\log \underbrace{P(X|z,c)}_{\text{decoder}}] + \beta(D_{KL}[\underbrace{Q(z|X,c)}_{\text{encoder}} || \underbrace{P(z|c)}_{\sim\mathcal{N}(0,1)}]) \tag{1}$$

In the current study, we reference the study conducted by Chattopadhyay et al. [14] and use its loss function (eq. 2) with $\beta = 10$ and MNIST as our dataset. After training our model, we use the decoder for our experiments. In other words, we no longer involve the encoder in the remaining procedures.

$$\frac{10}{\text{batch size}} \sum_{\text{batch size}} \Big\{ -\max \mathbb{E}[\log P(X|z,c)] + \beta(D_{KL}[Q(z|X,c)||P(z|c)]) \Big\} + D_{KL}[Q(z|X,c)||\underbrace{P(z|X,c)}_{\text{encoder's goal}}] \quad (2)$$

2.2 Intervening the Representation Layer

The decoder's input layer contains 20 neurons. We set the first ten as our latent variables z_i for $i \in \{0, 1, ..., 9\}$ and the last ten as the ten entries of an one-hot vector c_k for $k \in \{0, 1, ..., 9\}$ representing the class label of the digit that we want to reconstruct. To examine the reproducibility of the reference study [14], specific values are assigned to the latent variables. In other words, the representation layer is intervened.

In each experiment we choose $c_5 = 1$ since digit 5 contains horizontal, vertical and turning strokes that appear in all other digits. In the meantime, we assign a value $\alpha \in \{-3, -2, ..., 3\}$ to z_i such that $z_i \sim \mathcal{N}(0, 1)$ ranges across the three standard deviations of its distribution [14]. The remaining z-entries are randomised with a standard Gaussian distribution.

We run the decoder to obtain the reconstructed images. Since the images are static, we use them to build a GIF file for each z-entry. The GIF file depicts changes in the reconstructed images when z_i is assigned with different α-values.

2.3 Average Causal Effects Estimation

For the estimation of average causal effects of each input neuron on each output pixel, we reference the attribution method introduced by Chattopadhyay et al. [14]. Since a neural network's architecture could be represented as a graph, Chattopadhyay et al. viewed the neural network as a Structural Causal Model (SCM). SCM is a mathematical framework that helps discovering causality between objects of interest [16]. In their study, they developed a new attribution method that could identify the causal influence of an input neuron on an output neuron. The authors modified the original ACE formula (eq. 3) such that it fits the neural network's setting in which x is not binary but usually continuous (eq. 4). In the current study, we use this modified equation to estimate the causal relationship.

$$ACE^{y}_{do(x=1)} = \mathbb{E}[y|do(x = 1)] - \mathbb{E}[y|do(x = 0)] \quad (3)$$

$$ACE^{y}_{do(x_i=\alpha)} = \mathbb{E}[y|do(x_i = \alpha)] - \text{baseline}_{x_i} \tag{4}$$

where x_i here represents either z_i or c_k, while α is a value assigned to x_i. Recalling our input layer which is a concatenation of a 10-dimensional real vector z and a 10-dimensional one-hot vector c, their corresponding baseline is computed differently.

The baseline for c is simply $\mathbb{E}[y|do(c_k \neq \alpha)]$. This is done by excluding all images that belong to the selected class label from the training set, such that we obtain an updated distribution of data points for further sampling. For z, we take the average of the expectation of the reconstructed images with $z_i = \{\alpha \in \mathbb{Z} | -3 \leq \alpha \leq 3\}$ as its baseline.

We estimate the ACEs of c_k on the reconstructed images of our decoder for the reproducibility of our reference study. For z_i, we estimate their ACEs on the reconstructed images and each hidden layer for clustering analysis as described in the following section. We show the results by plotting the matrices that hold the ACE values as a heat map.

2.4 Clustering Analysis

We divide the clustering analysis into two parts. The first is to cluster the decoder by the estimated ACE of z_i on each hidden layer. The second is to cluster the decoder by the activation of its neurons.

We use Spearman coefficient as the similarity matrix, as it assesses how well the relationship between two variables can be described using a monotonic function, and linear relationships between neurons' activation values are not expected [17]. The same applies on the ACE values. Then we use affinity propagation as our clustering algorithm. Both of the clustering results are used for lesion test experiment.

Affinity propagation is a clustering algorithm that performs clustering according to message passing between data points [18]. It can use different general similarity notions such as negative Euclidean distance between data points (e.g. neurons' activation values) as the similarity $s(i, k)$ [19]. If data point k has a large value of $s(k, k)$, it is more likely to be an examplar of others. Thus $s(k, k)$ alters the number of identified examplars. Since the authors calls $s(k, k)$ as "preference" [18], we will use this term instead of $s(k, k)$ in the following sections.

Different from other popular clustering algorithms such as k-means clustering, affinity propagation doesn't require a pre-specified number of clusters as it considers all data points as candidate centers at once and gradually identifies clusters. This makes affinity propagation capable to avoid many of the poor solutions that are caused by unlucky initialisation and hard decisions [18].

2.5 Lesion Test Experiment

After obtaining the results from clustering analysis, we proceed with lesion test experiment [**?**]. Lesion test is done by dropping out a cluster or subcluster of neurons that is of interest. In each test, we remove one cluster of neurons from the model by setting their activation values to zero. Then we run the decoder with $z_i = \{\alpha \in \mathbb{Z} | -3 \leq \alpha \leq 3\}$ and $c_5 = 1$ to obtain the reconstructed images.

The reconstructed images are used to generate the described animation. Then we check if the reconstruction succeeds or fails. Here we define failed reconstruction as images which digit cannot be recognised by visual inspection or are reconstructed as digit other than digit 5. The failed reconstruction indicates that the removed cluster is important to the reconstruction.

Since neurons that are important to image reconstruction have activation values and ACE values that change significantly with different α-values, they should be similar and clustered in the same cluster. We expect that in each of the two results of the clustering analysis, there is one cluster that fails the reconstruction task. We call it "important cluster" with a notation C_{ACE} and C_{ACT}, of which C_{ACE} represents the important cluster from the clustering analysis on the ACEs, while C_{ACT} is the important cluster from the clustering analysis on the activation.

We know that if the clustering analysis is perfectly optimised, C_{ACE} and C_{ACT} should only contain neurons that are important to image reconstruction. However we do not conduct optimisation in this study, so each of them contains both the neurons that are important to the reconstruction task and those that are not.

Since the important neurons in the two sets are correctly clustered by the algorithm, we want to know if the two sets share common important neurons. To answer our question, one way is to find out the intersection of C_{ACE} and C_{ACT}, then conduct lesion test experiment on it.

If the model fails to reconstruct images in the corresponding lesion test experiment, the intersection of the two sets contains neurons that are critical to image reconstruction. We name these neurons as "critical neurons" and this intersection as "critical cluster".

3

Results

3.1 Reconstructed Images

Figure 3.1 shows the reconstructed images generated by the decoder. The images are a bit blurry, because the decoder serves for image reconstruction and we set $\beta = 10$ such that the model weighs the encoder's loss more than the decoder's loss during the training phase.

Looking into each column of figure 3.1, all grids except z_1 appear to be static. When we use these images to build a GIF file, we could see that z_1 shows a digit rotating clockwise. In addition, the reconstructed digit in $z_1 = -3$ is bigger than that in $z_1 = 3$. This suggests that, in our trained model, the neuron z_1 carries both rotation and scaling roles in image reconstruction.

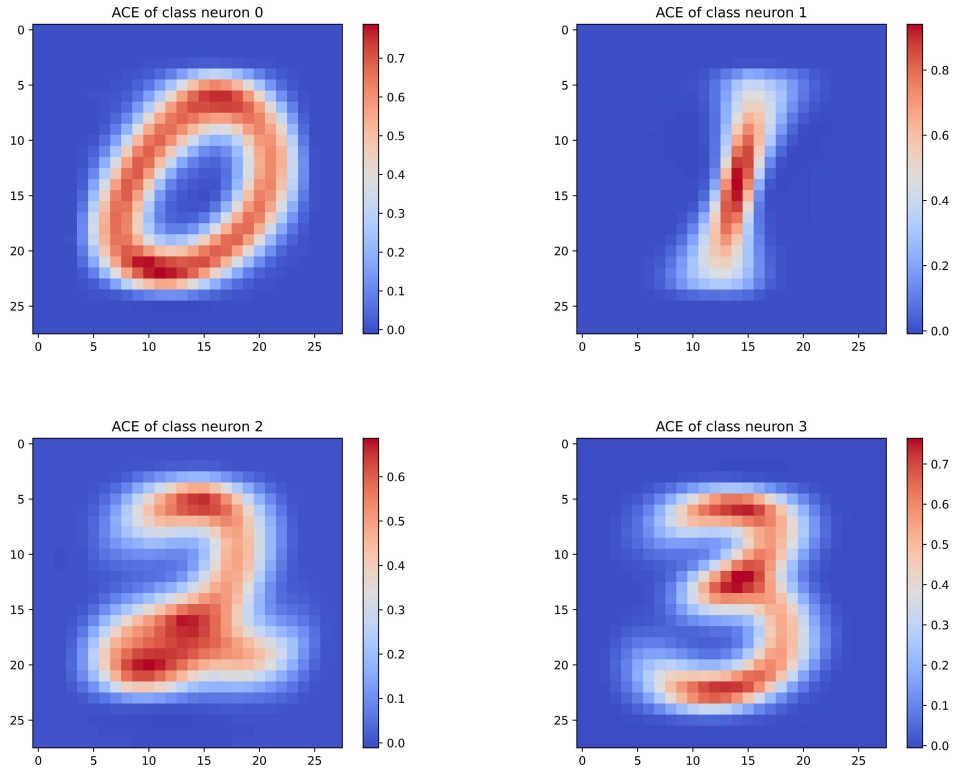
Figure 3.1: Reconstructed images generated by the decoder. Each column refers to z_i for $i \in \{0, 1, ..., 9\}$. Each row has an α-value ranges between -3 and 3 with an interval of 1

3.2 ACE Estimations

Recalling section 2.3, after running decoder and obtaining reconstructed images, the average causal effects of c_k and z_i are estimated as our next step.

3.2.1 ACE of c_k

Figure 3.2 illustrates the heat maps of the ACE matrices with $c_k = 1$. All average causal effects are positive. At each k-value, the ACE matrix forms an image that looks like digit k. This indicates that c_k has higher ACE on the strokes of digit k and controls which digit the model is going to reconstruct. This matches our setting for the current β-VAE model in which we provide the class label such that the model learns better.



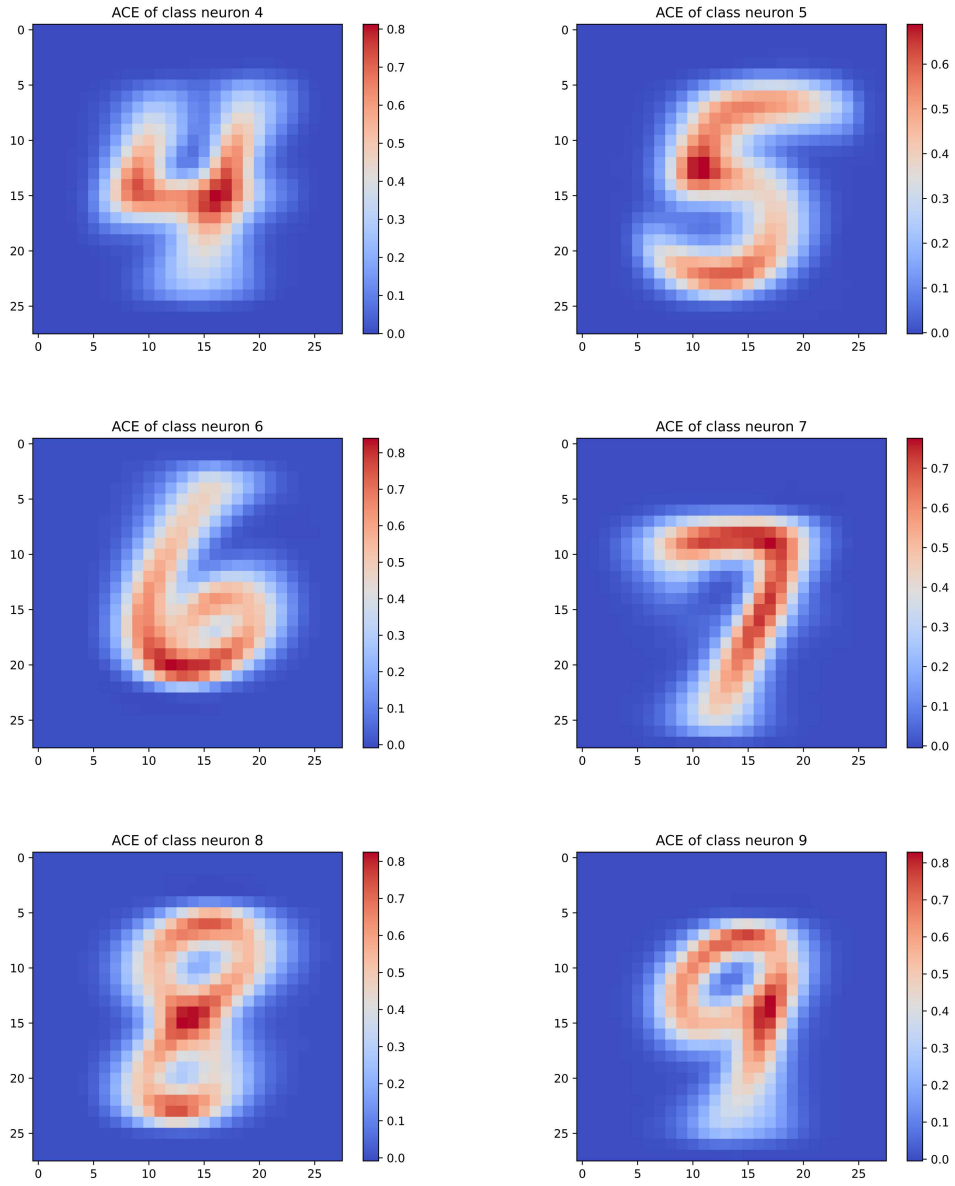


Figure 3.2: Heat maps of the ACE matrices with $c_k = 1$ for $k \in \{0, 1, ..., 9\}$

3.2.2 ACE of z_i

According to the results above, we consider the z-entries to be two types: the one that carries the rotation and scaling roles on reconstructed images (i.e. z_1) and the one that doesn't show any significant role (e.g. z_0). In this study, we estimate the ACE of z_1 and z_0 instead of all z-entries due to time limitation.

Figure 3.3 shows the heat maps of the ACE matrices when z_1 is assigned with different α-values. Same as the ACE matrices of c_5, z_1 has causal effects on the stroke of the digits. The corresponding ACE values range between -0.4 and 0.5. The positive ACE values form the digit alike that reconstructed by the decoder. When we use these static images to build a GIF animation, we find that the positive ACE values construct a rotating digit just like the decoder.

However, the ACE matrices of z_0 do not guarantee to construct a proper digit. As shown in figure 3.4, the ACE values do not form a digit at $z_0 = 2$. Its range of ACE values varies when it is assigned with different α-values.

On the other hand, it is also difficult for us to find a sequence of changes along the change of the α-value in z_0. All of these show that when the z-entry does not carry a rotation or scaling role on image reconstruction, it has causal effects on the stroke of the digit only when it is assigned with particular values.

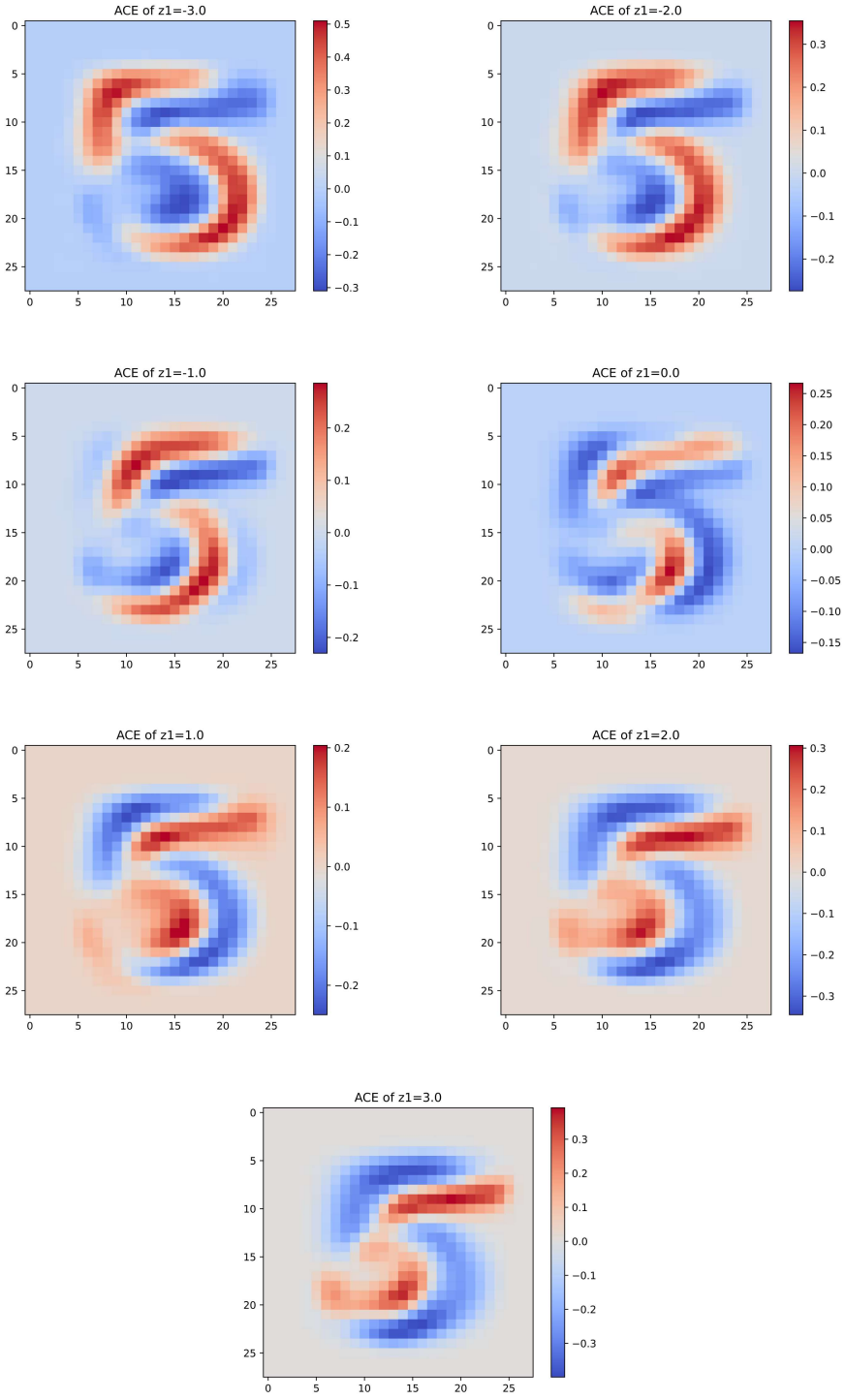


Figure 3.3: Heat maps of the ACE matrices with $z_1 = \{\alpha \in \mathbb{Z} | -3 \leq \alpha \leq 3\}$

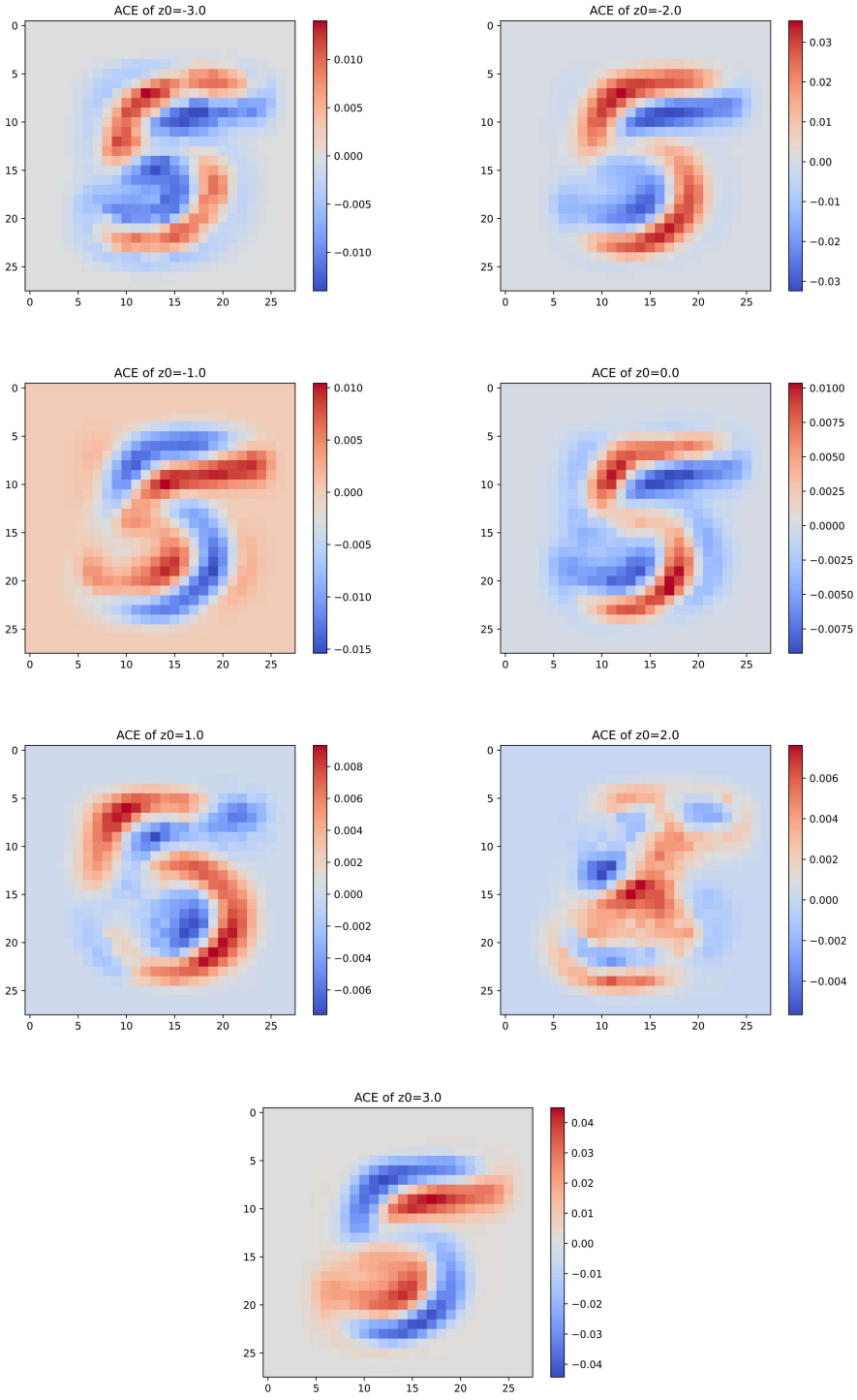


Figure 3.4: Heat maps of the ACE matrices with $z_0 = \{\alpha \in \mathbb{Z} | -3 \leq \alpha \leq 3\}$

3.3 Clustering Analysis

For each α-value assigned to z_1, the number of clusters is stable when the preference value falls in particular range, so we run the algorithm multiple times for the preference value. Figure 3.5 shows the number of clusters identified by the algorithm with different preference values. The preference value for this sample should be lower than 0.1 in order to obtain a stable clustering result.

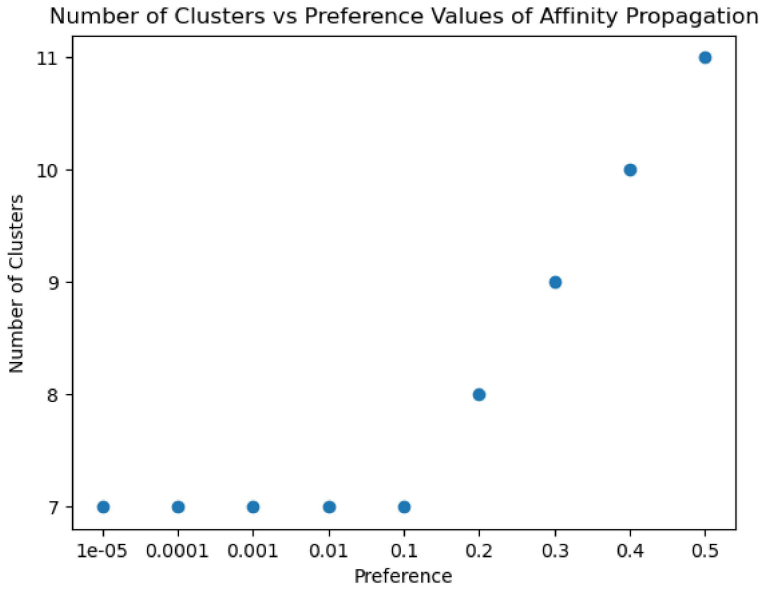


Figure 3.5: Number of clusters is determined when the algorithm suggests a minimum number of clusters over a range of preference

Generally, the clustering results based on activation values and ACE values are similar. For each α-value being assigned to z_1, both of them have at least 5 clusters. While the clustering that is based on activation values has at most 11 clusters, the clustering that is based on ACE values has at most 9 clusters (table 3.1).

z1	Activation	ACE
-3	8	9
-2	8	8
-1	9	9
0	11	6
1	10	9
2	5	8
3	7	5

Table 3.1: Number of clusters suggested by the clustering analysis at different α-values being assigned to z_1

3.4 Lesion Test Experiment

3.4.1 Important Cluster

The results of the lesion test experiment show that each set of clustering results has at least one cluster that significantly alters the reconstructed images. Among those clusters, one is significantly more influential. If it is removed, the reconstructed images become hard to be recognised by visual inspection or a wrong digit is reconstructed.

Each row of figure 3.6 illustrates the reconstructed images when a cluster is removed from the decoder. When cluster 2 (row 3) is removed, the network fails to reconstruct digit 5 and appears to reconstruct digit 3. Therefore we identify cluster 2 as an important cluster.

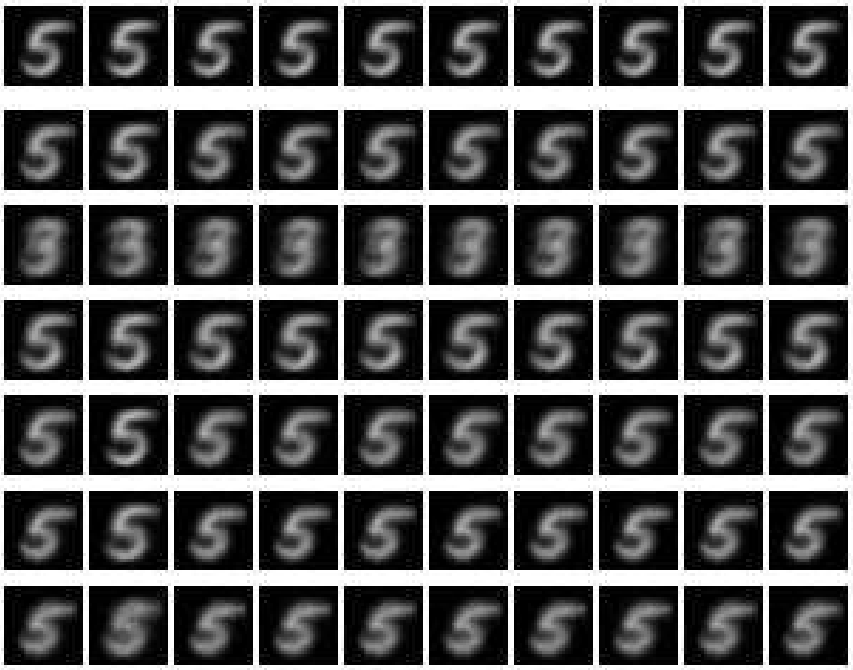


Figure 3.6: Visualisation of the results of lesion test experiments. Each row represents a cluster, while each column represents an z-entry

3.4.2 Critical Cluster

For each α-value being assigned to z_1, we identify the common neurons that exist in C_{ACE} and C_{ACT}. This new set of neurons is our critical cluster. We remove it from the network and conduct the lesion test experiment as described.

The results are shown in figure 3.7. When the neurons in the critical cluster are removed from the network, the network is unable to reconstruct digit 5 as well as a complete decoder regardless of the α-value being assigned to z_1. This proves that the critical cluster contains neurons that are essential for reconstructing the digits correctly with good quality.

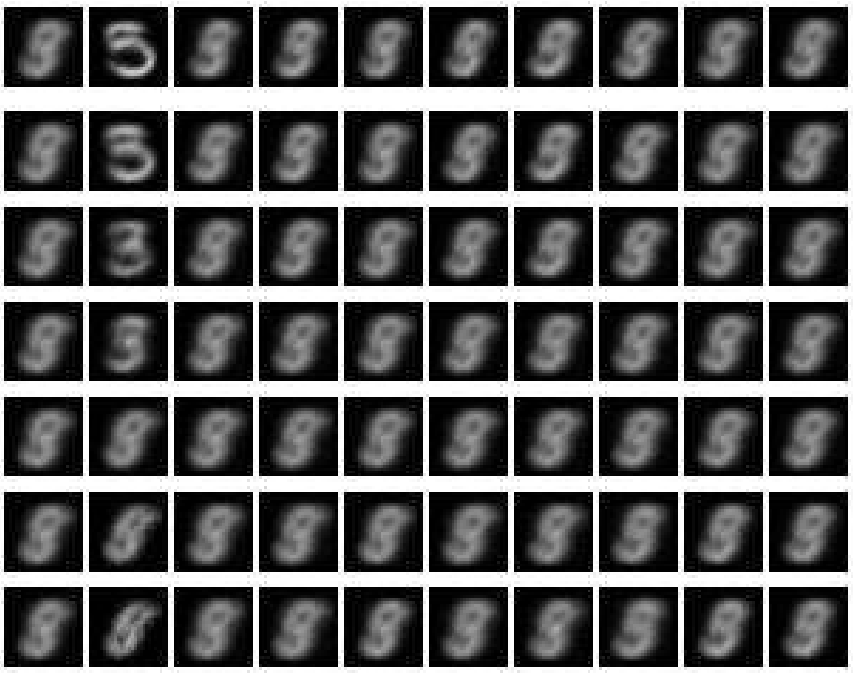


Figure 3.7: Results of lesion test experiment on the critical cluster. Each row represents $z_1 = \{\alpha \in \mathbb{Z} | -3 \leq \alpha \leq 3\}$, while each column represents an z-entry

4

Conclusion

The current study produces results that to some degree match the reference study conducted by Chattopadhyay et al. [14] since both models have neurons carrying out the rotation and scaling roles in image reconstruction. While there are two input neurons being responsible for the rotation and scaling roles in the reference study, our model has only one input neuron carrying the roles.

For the ACE estimation, our study reproduces same results for the latent neurons z_i and the class neurons c_k. In addition, we find that for z-entries that carry rotation and scaling roles, their ACE heat maps reconstruct recognisable digits in sequence, such that the corresponding GIF animation shows a rotating digit similar to the reconstructed images. In contrast, the ACE heat maps of z-entries that do not carry any role in image reconstruction, cannot reconstruct a recognizable digit at particular α-values. It is also difficult to find a sequence of changes along their ACE heat maps.

The results of clustering analysis based on activation values and ACE values are similar. The decoder could be clustered into at least 5 clusters. While the number of clusters is similar between the activation-based and ACE-based clustering analysis, it differs significantly at $z_1 = 0$.

The lesion test experiments show that there is at least one cluster being influential to image reconstruction in both of the activation-based and ACE-based clustering analysis. We believe that this is due to the lack of optimisation in the clustering analysis, so some important neurons are clustered wrongly. Among those influential clusters, there is always one cluster reconstructing images that appear to be digit 3. This matches our expectation that there is only one important cluster in both clustering analysis.

Since the intersection of the two important clusters C_{ACE} and C_{ACT} is not empty and the model fails to reconstruct images in the corresponding lesion test experiment, we successfully identify the critical cluster and its critical neurons.

We also compare the number of neurons in the important clusters and critical cluster. The number of important neurons identified in activation-based clustering analysis ranges from 65 to 650, while that in ACE-based clustering analysis ranges from 114 to 895.

Nonetheless, our study shows that the number could further be lowered by identifying the critical cluster. The orange line in figure 4.1 represents the number of critical neurons. It is significantly lower than the two important clusters.

A highlight is that the critical neurons include input neurons, output neurons and the hidden neurons based on our setting. If we exclude input and output neurons from the critical clusters, we will find that the number of hidden neurons that we are interested ranges between 4 and 21 inclusively.

These findings show that the combined application of clustering analysis and the SCM-based attribution method helps enhancing a network's transparency and interpretability, so makes contributions to the problem of blackboxness. They suggest that if we want to study deep neural networks, we could focus on hidden neurons that are critical to the tasks instead of the entire networks.

In other words, there may exist the possibility of reducing the cost of deep neural networks, enhancing networks' performance by tuning the parameters of the critical hidden neurons and encrypting critical hidden neurons for protecting the networks from being hacked [20].

However, since the lesion test experiment for the intersection uses only one set of samples, there is no sufficient evidence to prove its repeatability and consistency. Therefore, more studies are needed for the findings and suggestions above.

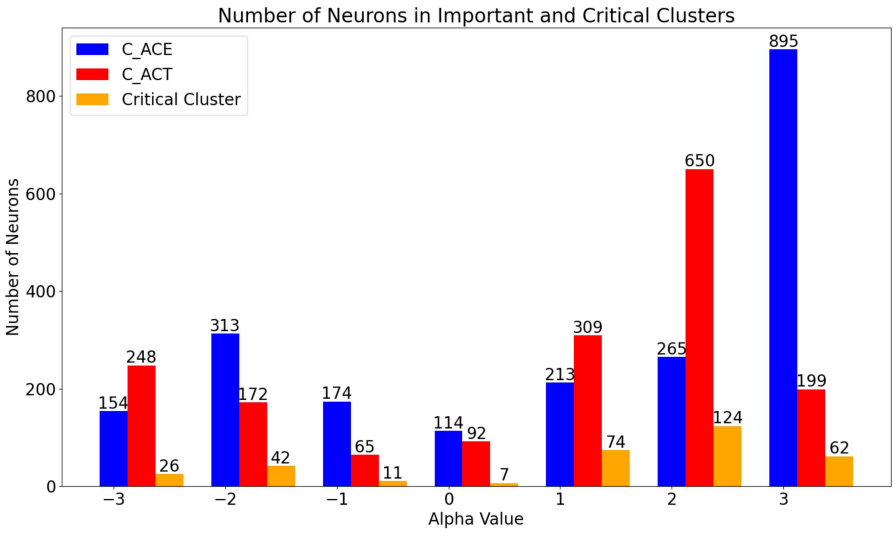


Figure 4.1: Number of neurons identified in important/critical clusters

4.0.1 Future Work

Due to time limitation, it has two drawbacks that are anticipated to be handled in future studies. The first is the lack of optimisation in clustering analysis. In the current study, the preference is selected through repeated experiments. The selection of this parameter could be improved by different algorithms such as the improved fruit fly optimisation suggested by Zhou et al. [21]. Another drawback is the insufficient amount of samples used in the lesion test experiment, so the current study is incapable to show the consistency of the results.

As an exploratory study, the current study answers our research questions. It also reveals findings that need further investigations. We anticipate more studies could be conducted on the modification and the observations of the current study.

Bibliography

[1] IBM Cloud Education. "Neural Networks". IBM.com. https://www.ibm.com/cloud/learn/neural-networks#:~:text=Neural%20networks%2C%20also%20known%20as,neurons%20signal%20to%20one%20another (accessed Apr. 23, 2022).

[2] Wikipedia, the free encyclopedia. "Deep Learning". Wikipedia. https://en.wikipedia.org/wiki/Deep_learning (accessed 2022-05-01).

[3] Y. C. Wu and J. W. Feng, "Development and Application of Artificial Neural Network" *Wireless Pers. Commun.*, vol. 102, pp. 1645-1656, Sept. 2018, doi: 10.1007/s11277-017-5224-x.

[4] H. Moayedi, M. Mosallanezhad, and A. S. A. Rashid, "A systematic review and meta-analysis of artificial neural network application in geotechnical engineering: theory and applications" *Neural Comput. Applic.*, vol. 32, pp. 495–518, Jan. 2020. doi: 10.1007/s00521-019-04109-9.

[5] M. Du, N. Liu, and X. Hu, "Techniques for Interpretable Machine Learning" *Commun. ACM.*, vol. 63, no. 1, pp. 68-77, Jan. 2020. doi: 10.1145/3359786.

[6] G. Stiglic, P. Kocbek, N. Fijacko, M. Zitnik, K. Verbert, L. Cilar, "Interpretability of machine learning-based prediction models in healthcare" *WIREs Data Min. Knowl. Discov.*, vol. 10, no. 5, May 2020. doi: 10.1002/widm.1379.

[7] J. Morley, C. C. V. Machado, C. Burr, J. Cowls, I. Joshi, M. Taddeo, and L. Floridi, "The Ethics of AI in Health Care: A Mapping Review" *Soc. Sci. Med.*, vol. 260, Jun. 2020. doi: 10.1016/j.socscimed.2020.113172.

[8] Y. Bathaee, "The Artificial Intelligence Black Box and the Failure of Intent and Causation" *Harv. J. Law Technol.*, vol. 31, no. 2, 2018.

[9] Z. C. Lipton, "The Mythos of Model Interpretability" *Queue*, vol. 16, no. 3, pp. 31-57, Jun. 2018. doi: 10.1145/3236386.3241340.

[10] D. Filan, S. Casper, S. Hod, C. Wild, A. Critch, and S. Russell, "Clusterability in Neural Networks" Mar. 2021. *arXiv: 2103.03386v1.*

[11] S. Hod, S. Casper, D. Filan, C. Wild, A. Critch, and S. Russell. "Detecting Modularity in Deep Neural Networks" Sept. 2021. *arXiv: 2110.08058v1.*

[12] M. A. Ahmad, A. Teredesai and C. Eckert, "Interpretable Machine Learning in Healthcare" *IEEE Intell. Inform. Bull.*, vol. 19, no. 1. Aug. 2018.

[13] M. Ancona, E. Ceolini, C. Öztireli, and M.Gross, "Gradient-Based Attribution Methods" in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, vol. 11700. W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K. R. Müller, Eds., Springer Cham, 2019, pp. 169–191.

[14] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian, "Neural Network Attributions: A Causal Perspective" in *36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, Jun. 9-15, 2019.

[15] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework" presented at *ICLR 2017 Conf.*, Toulon, France, Apr. 24-26, 2017.

[16] J. Pearl, "The Seven Tools of Causal Inference with Reflections on Machine Learning", Univ. California, Los Angeles, CA, USA, Tech. Rep. R-481, Nov. 2018.

[17] V. A. C. Horta, I. Tiddi, S. Little, A. Mileo, "Extracting knowledge from Deep Neural Networks through graph analysis" *Future Gener. Comput. Syst.*, vol. 120, pp. 109-118. 2021, doi: 10.1016/j.future.2021.02.009.

[18] B. J. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points" *Science*, vol. 315. no. 5814, pp. 972-976, Feb. 2007, doi: 10.1126/science.1136800.

[19] Probabilistic and Statistical Inference Group. "Affinity Propagation FAQ". Frey Lab.
`https://genes.toronto.edu/affinitypropagation/faq.html#legal`
(accessed Apr. 23, 2022).

[20] M. D. Kissner, "Hacking Neural Networks: A Short Introduction" 2019. [Online]. Available: `https://arxiv.org/abs/1911.07658`

[21] R. Zhou, Q. Liu, J. Wang, X. Han, and L. Wang "Modified Semi-supervised Affinity Propagation Clustering with Fuzzy Density Fruit Fly Optimization" *Neural Comput. Applic.*, vol. 33, pp. 4695–4712, May 2021. doi: 10.1007/s00521-020-05431-3.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY