Parameterization of a Convolutional Autoencoder for Reconstruction of Small Images

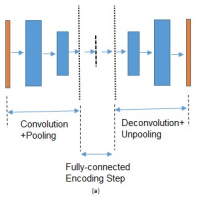
Lai Meng Tang, Li Hong Lim and Paul Siebert
University of Glasgow

***Abstract*— A convolutional autoencoder (CAE) is formed by combining a convolutional neural network and an autoencoder, to take both their advantages in reconstructing the output from a compact, latent representation of the input. However, to our best knowledge, there is no exact recommendation for parameterizing a CAE, such as deciding the number of neurons in the hidden bottleneck layer of a CAE to avoid "underfitting" and "overfitting" of the network. Hence, a framework for deriving an optimum set of CAE parameters for the reconstruction of input images based on the standard MNIST data set is presented in this paper. The robustness of the parameters on a different image size's data set, like the SVHN, is then verified. Our results show that for small (28 x 28) and (32 x 32) pixels' input images, having 2560 neurons at the hidden bottleneck layer and 32 convolutional feature maps can result in optimum reconstruction performance for the CAEs. In addition, the quantitative Mean-Square-Error and the qualitative (2D visualization of the neurons' activation, the histogram statistics and estimated source entropy at the hidden layers) analysis methodology provided by this work can provide a good framework for deciding the parameter values of the CAEs to provide good representations of the input image.**

I. Introduction

In recent years, convolutional neural network (CNN) based approaches have demonstrated significant improvements over previous conventional image processing methods in almost all computer vision related tasks they have been applied to. By combining the CNN and the autoencoder, we can create an unsupervised, hierarchical feature-based representation learner [1] to reconstruct an input image using the latent representation at its hidden bottleneck layer. However, the research on determining the optimum parameters of the combined CAEs, in particular the optimum number of neurons in the hidden bottleneck layer to represent the latent representation of the input, are naturally lacking to this date, to our best knowledge. Although there exists some methods which are used for determining the number of neurons in the hidden nodes of a conventional neural network [2], still the research on deciding the optimum parameters such as the number of neurons in the hidden bottleneck layer is lacking to this date, which is a challenging issue for a CAE to avoid "under-fitting" and "over-fitting" issues. Even the most recent development of autoencoders ([3], [4]) did not addresss such parametrization concern. Thus the main motivation behind this work is to derive an optimum set of CAE parameters for the reconstruction of the input image, which is the fundamental purpose of the autoencoders.

Section II walks the reader along the convolutional autoencoders model architecture derived for this study. Then it leads us to Section III to discuss the methodology used in our experiments to quantitatively and qualitatively analyze the effectiveness of a CAE in preserving good feature information and reconstructing input images. Section IV presents experimental results of our convolutional autoencoder models using the standard MNIST data set for reconstruction of small images. Section V compares the reconstruction performance of the selected model with its parameters optimized, with other similar input data set, the Street View House Numbers (SVHN), to check the robustness of the model on different image sizes. Finally, Section VI summarizes our findings and concludes our work.



	Model 1
layer_name	Size
0 input	Start with 1 input image of size 28x28
1 conv	To get 32 feature maps of size 26x26 (filter size 3x3, stride 1, pad 0)
2 pool	To get 32 feature maps of size 13x13
3 flatten	5408 neurons
4 encode_layer	Number of neurons (Nn=40), fully-connected
5 hidden	7200 neurons, fully-connected
6 unflatten	To get 32 feature maps of size 15x15
7 unpool	To get 32 feature maps of size 30x30
8 deconv	End with 1 output image of size 28x28 (filter size 3x3)
9 output_layer	

(b)

Fig. 1: Basic Convolutional Autoencoder Model (Model 1).

	Model 1
layer_name	Size
0 input	Start with 1 input image of size 28x28
1 conv	To get 32 feature maps of size 26x26 (filter size 3x3, stride 1, pad 0)
2 pool	To get 32 feature maps of size 13x13
3 flatten	5408 neurons
4 encode_layer	Number of neurons (Nn=40), fully-connected
5 hidden	7200 neurons, fully-connected
6 unflatten	To get 32 feature maps of size 15x15
7 unpool	To get 32 feature maps of size 30x30
8 deconv	End with 1 output image of size 28x28 (filter size 3x3)
9 output_layer	

Fig. 2: Model 1's architecture.

II. Convolutional Autoencoder Models

For a CAE, we have to build a network which takes an image as an input, and tries to reconstruct the same image as an output. The architecture of this skeleton model (model 1) is shown in Fig. 1.

For this model 1, we will start with a network with one convolutional/pooling (Conv/Pool) layer and one deconvolution/unpooling (Deconv/Depool) layer, with filter sizes of 3×3. The narrow encoded (bottleneck) layer starts with 40 neurons (N_n), with the number of epochs, N_e, equals 20 and the number of feature maps, N_m, equals 32, as shown in Fig. 2. We then compare Model 1 with Model 2 which comprises three Conv/Pool layers and three Deconv/Depool layers, as shown in Fig. 3.

III. Methodology

A. 2-D Neuron Activation, Histogram Statistics and Entropy Visualization

For Model 1, the information captured by the 5408 neurons at the flatten layer (Layer 3) are compressed to 40 or more neurons at the fully-connected bottleneck layer (layer 4). Since it is difficult to visualize them in 1-D, naturally it will be easier to visualize them in a 2-D form as shown in Fig. 4. There are 169 neurons arranged in each row and they correspond to the 169 neurons (or 13×13) in each of the 32 feature map. Since each feature map at the flatten layer (layer 3) is the result of the convolution using different filters learnt followed by a subsequent pooling operation, any variation seen along the row should show the information captured by each map, which is represented by the neurons' activation states. Any variation along the column should be irrelevant for visualization of the information captured by each convolution and pooling operations at each of the feature map, as they may not show strong image structures.

To further examine the utilization of the 5408 neurons for the flatten layer (Layer 3), we will also look at the entropy values of the neighboring 169 neurons for each neuron shown in each row of Fig. 4. The entropy values will be calculated and presented in a 2-D form as well. Also, although the neurons at the bottleneck layer (layer 4) can't be arranged by feature maps row-by-row, we will visualize and examine the encoded neurons with the assumption that the variations across neurons arranged in a 2-D form will still show some structure of activation as the neurons are still in the right order of neighboring sequence.

Finally, the histogram visualization of the stacked feature maps at both layers under study is conducted. By looking at the statistics and histograms of each of the visualized 2-D maps and entropy images, we can then verify our conclusion on the effectiveness of the latent representation of input by our CAE.

B. Accuracy Study of Image Reconstruction using MSE

By using the MNIST data set as the training and validation input images, the training and validation results of our network can be measured by the Mean-Square-Error (MSE) values which are the square of the error values between the reconstructed output and the training and validation input images, serves to compare the acuracy of image reconstruction of the two network models for us to decide which network

	Model 1	Model 2
layer_name	Size	Size
0_input	Start with 1 input image of size 28x28	Start with 1 input image of size 28x28
1_conv1	32 feature maps of size 26x26 (filter size 3x3, stride 1, pad 0)	32 feature maps of size 24x24 (filter size 5x5, stride 1, pad 0)
2_pool1	32 feature maps of size 13x13	32 feature maps of size 12x12
3_conv2		32 feature maps of size 8x8 (filter size 5x5, stride 1, pad 0)
4_pool2		32 feature maps of size 4x4
5_conv3		32 feature maps of size 2x2 (filter size 3x3, stride 1, pad 0)
6_pool3		32 feature maps of size 1x1
7_flatten	5408 neurons	32 neurons
8_encode_layer	Number of neurons (Nn=40), fully-connected	Number of neurons (Nn=40), fully-connected
9_hidden	7200 neurons, fully-connected	1152 neurons, fully-connected
10_unflatten	32 feature maps of size 15x15	32 feature maps of size 6x6
11_unpool3		32 feature maps of size 12x12
12_deconv3		32 feature maps of size 10x10 (filter size 3x3, stride 1, pad 0)
13_unpool2		32 feature maps of size 20x20
14_deconv2		32 feature maps of size 16x16 (filter size 5x5, stride 1, pad 0)
15_unpool 1	32 feature maps of size 30x30	32 feature maps of size 32x32
16_deconv1	End with 1 output image of size 28x28 (filter size 3x3)	End with 1 output image of size 28x28 (filter size 5x5, stride 1, pad 0)
17_output_layer		

Fig. 3: Comparison of Model 1 and Model 2.

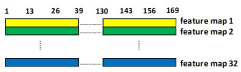


Fig. 4: Visualization of the neurons' arrangement at layer 3 in a 2-D form.

model is better. We can then use the selected model for our qualitative study in the subsequent stages.

C. Robustness Study using another data set

The final part of our methodology of image reconstruction qualitative and quantitative analysis is to verify the formulation of the network parameterization guideline of the CAE using a more complex small image data set, the SVHN, with different image sizes of (32 x 32). This will help to check the robustness of our hypothesis that the convolutional autoencoder is capturing useful information and being utilized in an efficient manner by setting the appropriate N_e, N_n and N_m numbers to parameterize our CAE. The results are presented in the next section.

IV. Results and Discussions

In this section we will first present the MSE results for varying N_e of the two models discussed in the previous section, Model 1 and Model 2. Beside using this results for selecting the better architecture out of them, we will also compare the sensitivity of the MSE results to N_n for both models, to find the optimum range of N_e and N_n for our subsequent CAE experiments. The purpose of the subsequent experiments are to optimize the network structure and to understand how well the neurons are utilized in the convolutional autoencoders for the selected model, as N_n and N_m are varied. We will use the methodology discussed to analyze these results.

A. Model Selection and Number of Epochs, N_e

The effect of varying N_e on both models 1 and 2, is shown in Fig. 5. We observe that the convergence rate of the MSE for both models is an exponential function over Ne, i.e. ke^{-aN_e}, where a is a constant rate of reduction and k is the initial MSE value. The values of k and a, found by performing a best-fit function for both models, are about the same, in the range of < 0.4 and < 0.1 respectively. But the initial MSE for Model 2 is found higher than Model 1 quite significantly, as shown in Fig. 5. Note that N_n, is fixed at 40 for the experiment. As their MSE values become smaller when N_e increases, they reach their steady state MSE values at $N_e > 15$. Hence, for all subsequent experiments, we have fixed $N_e = 20$. Note that N_m is fixed at 32 throughout our experiments.

We observe that having more neurons (and hence more parameters) are better for image reconstruction by a CAE when comparing Model 1 and 2 architecture. The flatten layer (layer 3, before the bottleneck layer) of Model 1 has a much larger number of neurons (5408, and hence more number of parameters), vs 32 neurons only for Model 2. Model 2 may have too much information loss compared to

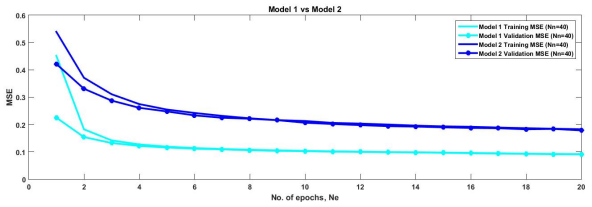


Fig. 5: Comparison of convergence rate between model 1 and model 2 for $N_n = 40$ at the bottleneck layer.

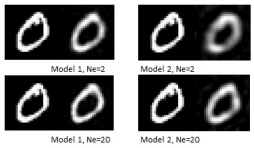


Fig. 6: Visual results of input (left half) and reconstructed (right half) images of model 1 and model 2 for $N_e = 2$ and 20, with fixed $N_n = 40$ at the bottleneck layer.

Model 1 before being compressed. Figure 5's results have indeed shown that by increasing the number of neurons in the hidden layer just before data compression plays an important role in reducing both training error and generalization error measured in MSE for a small MNIST image data set.

These quantitative results are further supported by the qualitative evaluation results as shown in Figure 6. The input and reconstructed images for model 1 and 2 are compared visually at $N_e = 2$ and 20, with N_n fixed at 40. As shown by the results, model 2's reconstructed image has less sharper appearance compared to model 1.

From the results, we conclude that Model 1 which has 1 Conv/Pool and 1 Deconv/Depool layers is better than Model 2 or other models which has more Donv/Pool and Deconv/Depool layers for small images like the MNIST data set. Hence Model 1 is selected for all our subsequent experiments.

B. Number of Neurons, N_n

In this section, we present the comparison results of the 2-D visualization of neurons' activation, the mean and standard deviation values of the 2-D neurons' activation maps, the 2-D visualization of the entropy, as well as the image reconstruction results (measured by MSE), as N_n, is varied from 40 to 2560. The N_e and N_m are fixed at 20 and 32 respectively throughout our experiments.

Fig. 7 shows the visualization of the stacked feature maps neuron activation states and Fig. 8 shows the visualization of the histogram of the stacked feature maps at the flatten layer (layer 3) with different number of neurons at the bottleneck layer (layer 4). As shown in Fig. 7, there are more activated neuron's structure/ patterns as N_n increases from 40 to 2560. Fig. 8(a) to (c) show there are many inactivated neurons concentrated at the zero grey level cluster. There is no significant difference seen among their distributions, although the distributions are more evenly spread out as N_n increases from 40 to 640. But for Fig. 8(d), we can see that

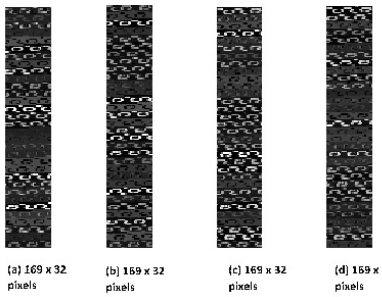


Fig. 7: Visualization of the stacked feature maps neuron activation states at layer 3 with different number of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

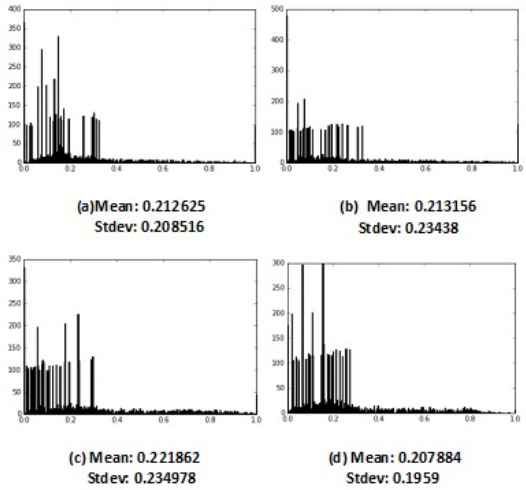


Fig. 8: Visualization of the histogram of the stacked feature maps at layer 3 with different number of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

there are lesser inactivated neurons concentrated at the zero grey level cluster. Hence we may conclude that information representation ability at $N_n = 2560$ is better in information representation for our convolutional autoencoder model.

Next, we will look at the first estimate of the source's entropy in the neighboring 169 neurons for each neuron in the same feature map at layer 3 according to a 2-D stacked feature map, as shown in Fig. 9. The results shows the number of high entropy values growing as N_n increases from 40 to 2560 at the bottleneck layer, as shown from Fig. 9(a) to (d). Fig. 9(a) shows least number of high entropy values due to a large number of inactivated neurons with a probability $p_r(r_k)$ values close to 1. Hence from the many high entropy values as shown in Fig. 9(d), we may conclude that $N_n = 2560$ will have better information representation ability.

Similarly, we can employ the same techniques to visualize the activation of the neurons at the bottleneck layer (layer 4). The results are shown at Figs. 10 to 12. Our results show that almost all neurons are fully activated to represent the

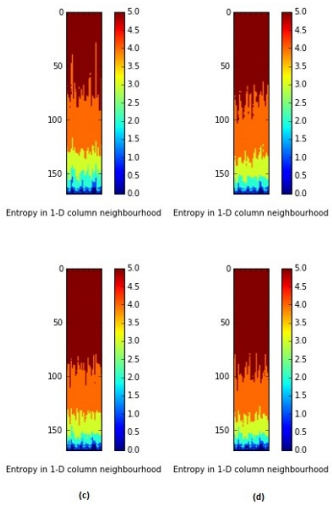


Fig. 9: Visualization of the 169 neighborhood's first estimated entropy values of the stacked feature maps' neurons at layer 3 with different number of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

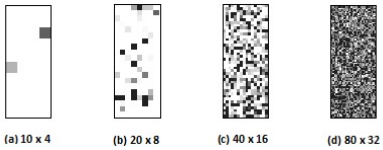


Fig. 10: Visualization of the stacked feature maps neuron activation states at layer 4 with different number of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

bottleneck information, with $N_n = 40$. As N_n increases from 40 to 2560, there are more neurons activation with lesser saturation observed, in particular so for $N_n = 2560$ which shows very few highly saturated neurons at the rightmost of the grey level distribution. This may explain the improvement in training and generalization (validation) losses (measured in MSE) shown in Fig. 5 discussed previously as N_n is increased from 40 to 2560 for Model 1.

This is further supported by the first estimate of the source's entropy in the neighboring neurons for each neuron in the same feature map at layer 4 presented in a 2-D form in Fig. 12. It shows many low entropy values due to many highly saturated neurons with $p_r(r_k)$ close to one, with the first estimated entropy values close to zero. As the entropy values get larger from Fig. 12(a) to (d), there are lesser low entropy values seen and the information captured is more clustered. This results is consistent with the conclusion from Fig. 9, which shows that the larger N_n will result in better information representation ability. Hence, from the many high entropy values as shown in Fig. 12(d), we may conclude that $N_n = 2560$ will have better information representation ability.

Fig. 13 shows the comparison of the reconstruction results

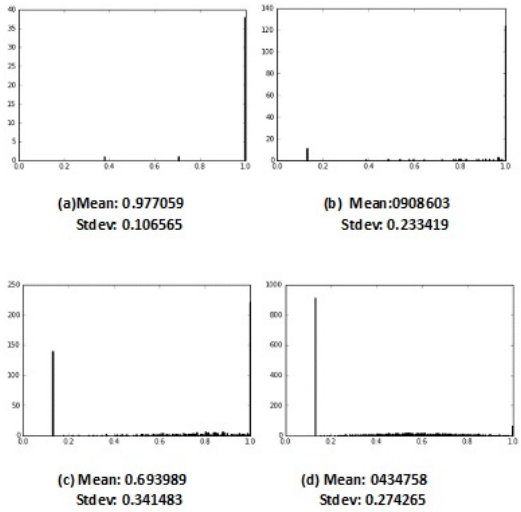


Fig. 11: Visualization of the histogram of the stacked feature maps at layer 4 with different number of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

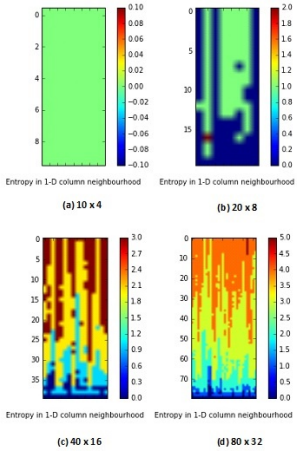


Fig. 12: Visualization of the neighborhood's first estimated entropy values of the stacked feature maps' neurons at layer 4 with different number of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

(measured in MSE) and the mean and standard deviation values of the 2-D neurons' activation maps, as N_n is varied from 40 to 2560, at both layer 3 and 4. The reduction in the MSE values as N_n increases as shown in Fig. 13 suggests that information representation ability at $N_n = 2560$ is better for our convolutional autoencoder model.

C. Number of Feature Maps, N_m

Inline with our results of varying N_n, we believe that as N_m increases, we should also see better information representation by our CAE. Hence, the same techniques were used to examine the neurons' activation state for the flatten layer and bottleneck layer (Layers 3 and 4) with N_n fixed at 1200, and varying N_m (N_m=8, 16, 24, 32 and 40).

As shown by the results in Figs. 14 to 20, the visualization

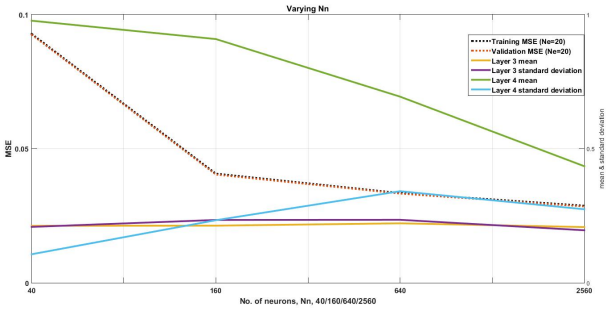


Fig. 13: Comparison of the image reconstruction results (measured by MSE), and the mean and standard deviation values of the 2-D neurons' activation maps, as N_n is varied from 40 to 2560, at both layer 3 and 4.

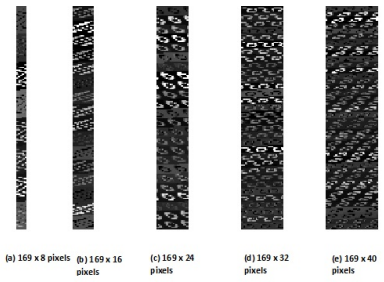


Fig. 14: Visualization of the stacked feature maps neuron activation states at layer 3 with different number of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; and (d) $N_m = 32$;and (e) $N_m = 40$.

of the stacked feature maps' neuron activation states, the histograms of the stacked feature maps, the entropy maps and the MSE results at layers 3 and 4 (for different number of feature maps) have all shown that N_m=32 or more will have better information representation ability. The results are consistent with our previous discussions.

D. Robustness Check

Fig. 21 shows our CAE's image reconstruction's visual results based on the SVHN data set. Though the quantitative results on SVHN data set based on our methodology are not shown here as they are repeating similar patterns as the MNIST. Hence we arrive at the same conclusion when varying the N_n and N_m.

In conclusion, both the quantitative MSE and the qualitative visualization of neurons' activation and entropy at layer 3 and 4 of our CAE model have proven that having N_e=20, N_n=2560 and N_m=32 will represent information well for both small input image sizes of (28 x 28) and (32 x 32) pixels.

V. Conclusion and Future Work

In this paper, we present the results of our experiments in the parameterization study of our CAE model, which consists of single Conv/Pool and Deconv/Depool layers. The

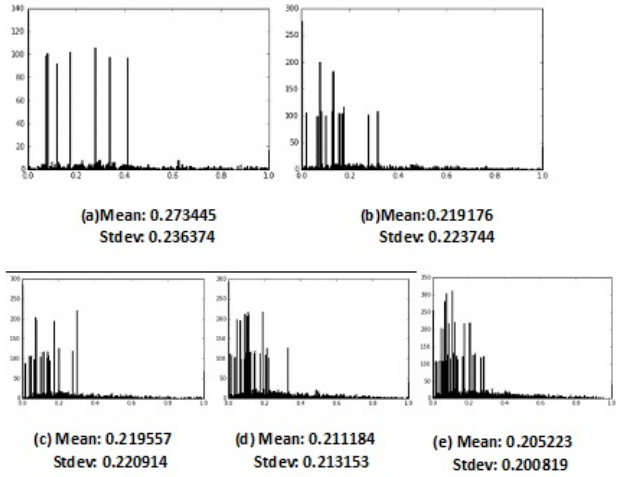


Fig. 15: Visualization of the histogram of the stacked feature maps at layer 3 with different number of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; and (d) $N_m = 32$;and (e) $N_m = 40$.

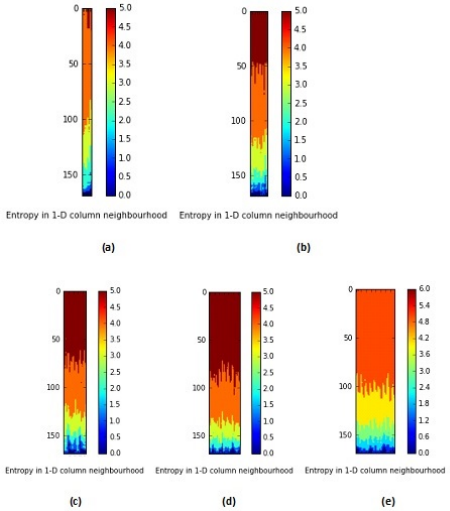


Fig. 16: Visualization of the 169 neighborhood's first estimated entropy values of the stacked feature maps' neurons at layer 3 with different number of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; and (d) $N_m = 32$;and (e) $N_m = 40$.

Fig. 17: Visualization of the stacked feature maps neuron activation states at layer 4 with different number of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; and (d) $N_m = 32$;and (e) $N_m = 40$.

effectiveness of our proposed methodology on this selected CAE model is checked both qualitatively and quantitatively using small input MNIST (28 x 28 pixels' images) data set,

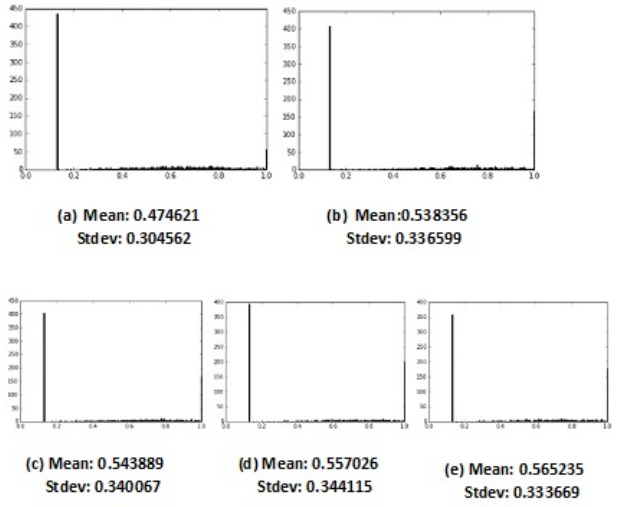


Fig. 18: Visualization of the histogram of the stacked feature maps at layer 4 with different number of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; and (d) $N_m = 32$;and (e) $N_m = 40$.

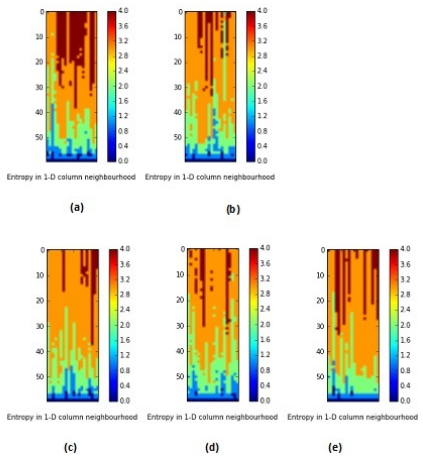


Fig. 19: Visualization of the neighborhood's first estimated entropy values of the stacked feature maps' neurons at layer 4 with different number of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; and (d) $N_m = 32$;and (e) $N_m = 40$.

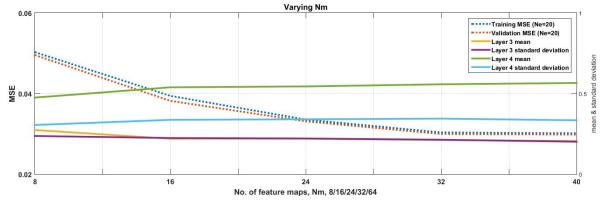


Fig. 20: Comparison of the image reconstruction results (measured by MSE), and the mean and standard deviation values of the 2-D neurons' activation maps, as N_m is varied from 8 to 40, at both layer 3 and 4.

and also verified by the SVHN (32 x 32 pixels' images) data set with different image sizes.

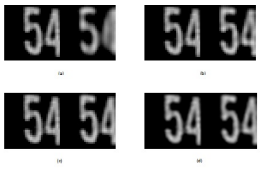


Fig. 21: Input (left half) image and reconstructed (right half) 32×32 results of the SVHN data set with different number of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

We conclude from our results of parameterization study that for small input images, having 2560 neurons at the hidden bottleneck layer (N_n) and 32 convolutional feature maps (N_m) can result in optimum reconstruction performance for our CAE model. We believe that this parameterization results could potentially be extended to more complex or larger images to generalize the results. In fact, a similar study has been conducted on the more complex CIFAR10 data set to check this hypothesis. With the same N_n used, the parameters increased were only the number of epochs (N_e, which has increased proportionally to 400) and the number of feature maps (N_m, which has increased proportionally to 256) for an optimum image reconstruction task, which are expected as higher dimensionality images would require more training epochs and feature maps. Because the number of neurons in the hidden bottleneck layer of a CAE remain unchanged, we believe that such recommended optimum N_n setting could be extended to larger input image sizes and complexity.

Our results also show that using both the quantitative (MSE) and the qualitative (2D visualization of the neurons' activation, histogram statistics and estimated source entropy at the layers just before and at the bottleneck layer) analysis methodology, as proposed by this work, it can provide a good framework for deciding the optimum parameter values of the convolutional autoencoders to provide good representation of the input image.

Moving forward, we may extend the study of our CAE to larger image data set such as the Caltech 101 (about 300 x 200 pixels) to verify our hypothesis of the possible generalization of our parameterization study methodology and results to larger image sizes for the CAE.

REFERENCES

[1] Leng, B., Guo, S., Zhang, X., and Xiong, Z., *3d object retrieval with stacked local convolutional autoencoder*, Signal Processing, 112:119-128, 2015.

[2] K. Ghana Sheila and S. N. Deepak, *Review on Methods to Fix Number of Hidden Neurons in Neural networks*, Mathematical Problems in Engineering, vol. 2013, Article ID 425740, 11 pages, 2013. doi:10.1155/2013/425740

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Book in preparation for MIT Press, http://www.deeplearningbook.org, 2016.

[4] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, *Adversarial autoencoders*, in: International Conference on Learning Representations (ICLR), arXiv:1511.05644, San Juan, 2016.