

♦ Member-only story



Search Medium



Write



Affects the Quality of Autoencoder Latent Representation

Hyperparameter tuning in autoencoders — Part 1



Rukshan Pramoditha · Follow

Published in Towards Data Science · 6 min read · Aug 23, 2022



42



...

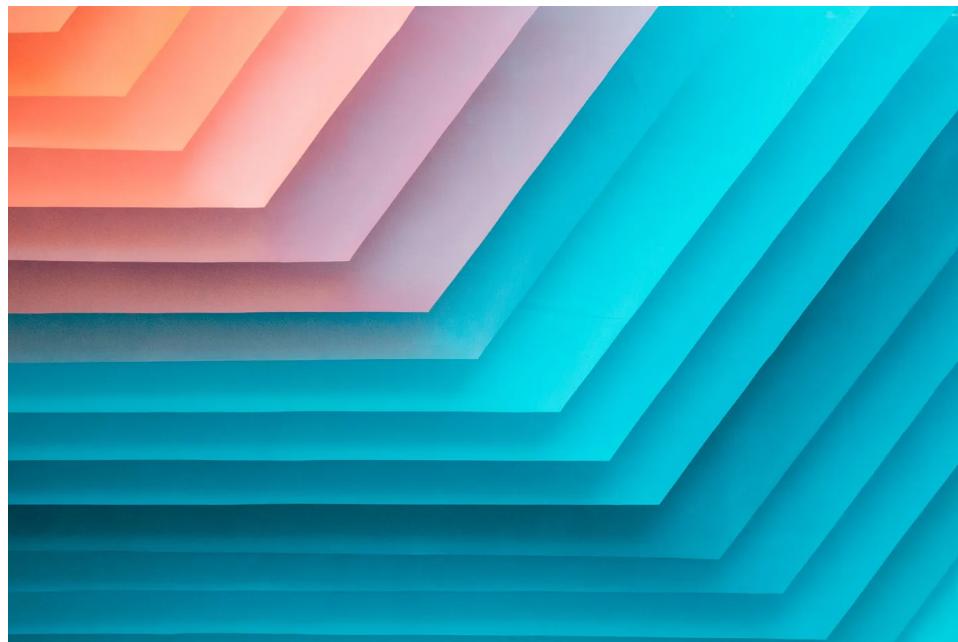


Photo by [Clark Van Der Beken](#) on [Unsplash](#)

Introduction

You may already know that autoencoder latent representation includes the most important features of the input data when its dimension is significantly lower than the dimension of the input data.

The quality of the autoencoder latent representation depends on so many factors such as *number of hidden layers*, *number of nodes in each layer*, *dimension of the latent vector*, *type of activation function in hidden layers*, *type of optimizer*, *learning rate*, *number of epochs*, *batch size*, etc. Technically, these factors are called autoencoder model hyperparameters.

Obtaining the best values for these hyperparameters is called *hyperparameter tuning*. There are different hyperparameter tuning techniques available in

How Number of Hidden Layers Affects the Quality of Autoencoder Latent Representation | by Rukshan Pramoditha | Toward...
machine learning. One simple technique is manually tuning one hyperparameter (here, number of hidden layers) while keeping other hyperparameter values unchanged.

Today, in this special episode, I will show you how the number of hidden layers affects the quality of autoencoder latent representation.

The dataset we use

We will use the [MNIST dataset](#) (see [Citation](#) at the end) to build the autoencoder models here.

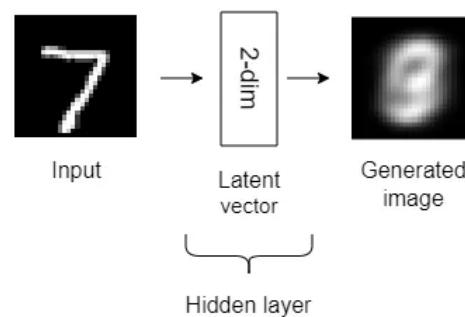
Acquire and prepare the MNIST dataset (Code by author)

Approach

We will build three autoencoder models with different architectures that only depend on the number of hidden layers while other hyperparameter values in the models remain unchanged. I will use several visualization techniques to verify the results.

Autoencoder with one hidden layer

Autoencoder with one hidden layer



(Image by author)

The above autoencoder architecture can be coded as follows.

Defining autoencoder architecture with one hidden layer (Code by author)

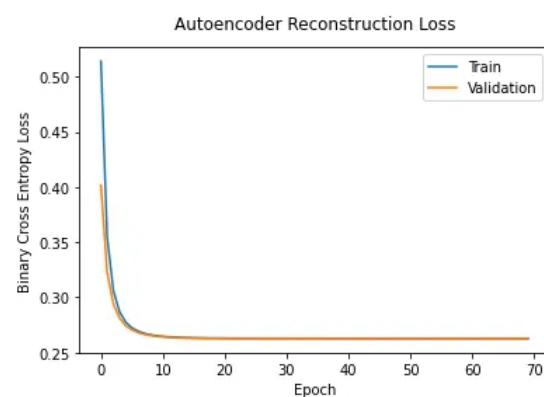
```
Model: "Autoencoder_1"
```

Layer (type)	Output Shape	Param #
<hr/>		
input_2 (InputLayer)	[(None, 784)]	0
dense_2 (Dense)	(None, 2)	1570
dense_3 (Dense)	(None, 784)	2352
<hr/>		
Total params:	3,922	
Trainable params:	3,922	
Non-trainable params:	0	

(Image by author)

Now, we can train the model.

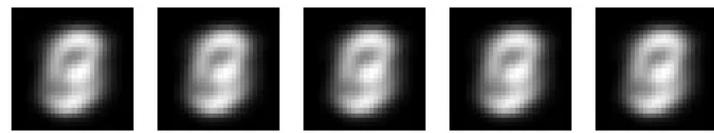
(Code by author)



(Image by author)

Now, we can visualize the compressed MNIST digits after encoding.

(Code by author)



The output of autoencoder with one hidden layer (Image by author)

This output is not clear enough to distinguish between each digit in the MNIST dataset because a single hidden layer is not enough to capture most of the complex non-linear patterns (relationships) in the MNIST data.

We can compare the above output with the original MNIST digits.

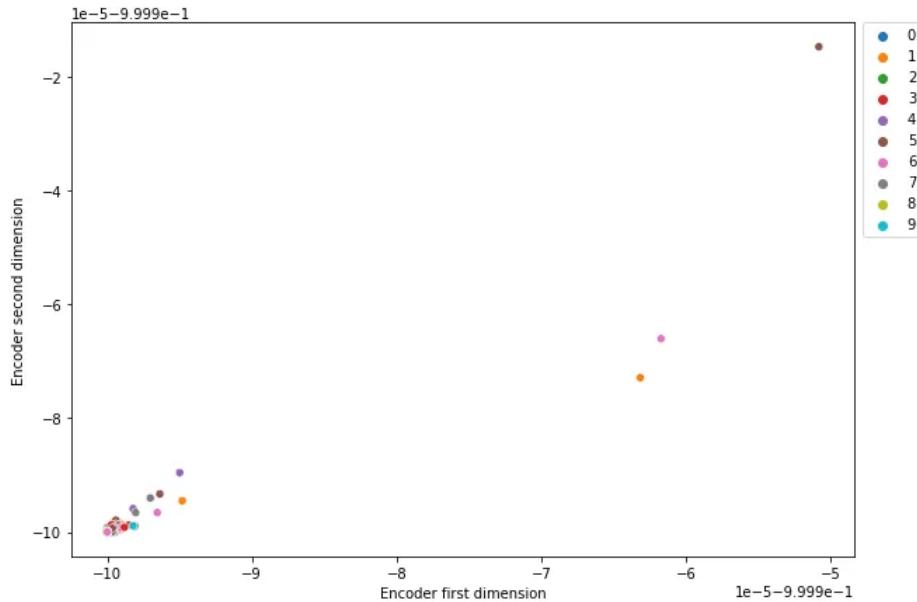
(Code by author)



Original MNIST digits (Image by author)

Now, we can visualize test MNIST data in the latent space to see how the autoencoder model with one hidden layer is capable of distinguishing between the nine digits.

(Code by author)

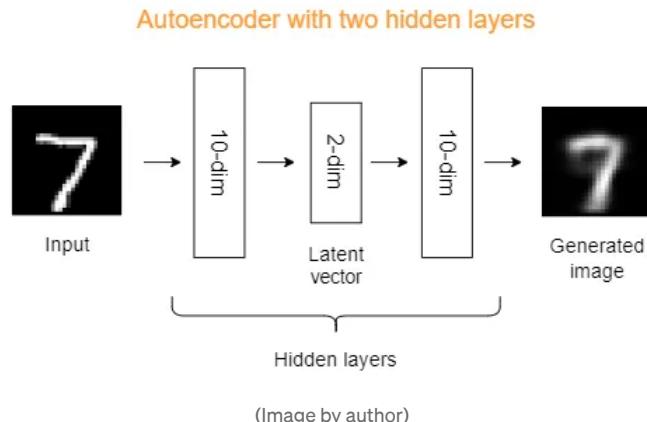


Visualize the test MNIST data in the latent space (Image by author)

The digits do not have clear separate clusters in the latent space. It means that the autoencoder model with only one hidden layer cannot clearly distinguish between the nine digits in the test MNIST data.

As a solution for this, we can increase the number of hidden layers in the autoencoder model.

Autoencoder with two hidden layers



The above autoencoder architecture can be coded as follows.

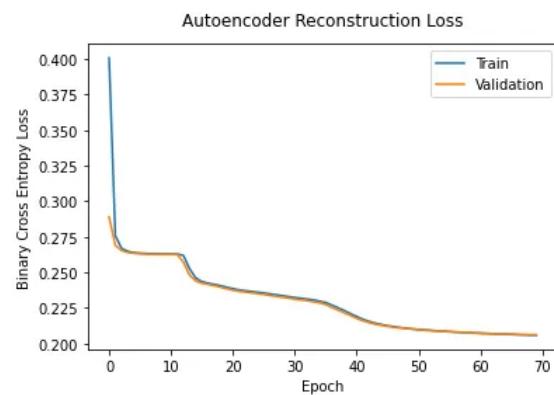
Defining autoencoder architecture with two hidden layers (Code by author)

```
Model: "Autoencoder_2"
```

Layer (type)	Output Shape	Param #
<hr/>		
input_3 (InputLayer)	[(None, 784)]	0
dense_4 (Dense)	(None, 10)	7850
dense_5 (Dense)	(None, 2)	22
dense_6 (Dense)	(None, 10)	30
dense_7 (Dense)	(None, 784)	8624
<hr/>		
Total params: 16,526		
Trainable params: 16,526		
Non-trainable params: 0		

(Image by author)

Now, we can train the model as previously.



(Image by author)

Now, we can visualize the compressed MNIST digits after encoding.



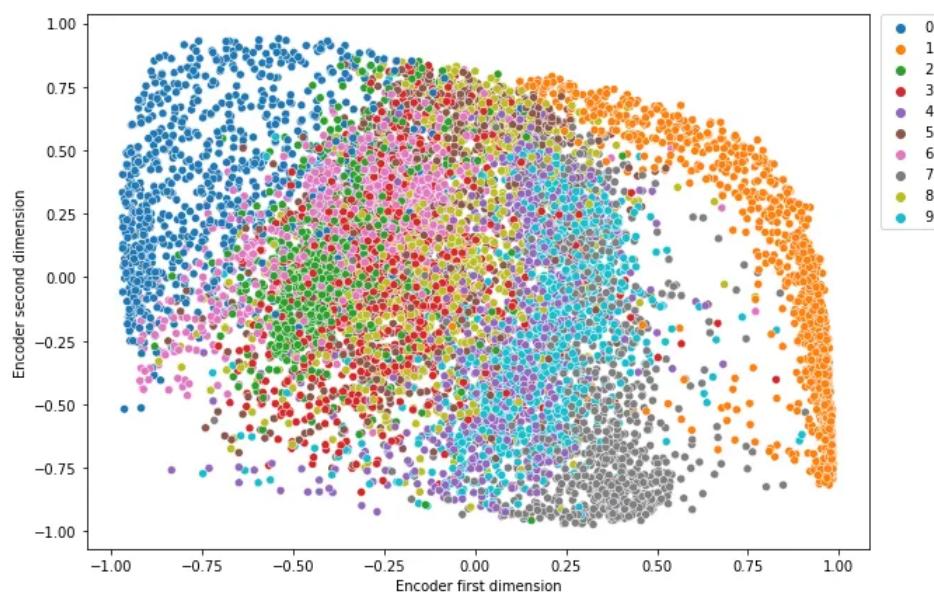
The output of autoencoder with two hidden layers (Image by author)

This output is much better than the previous output because two hidden layers can capture a significant amount of complex non-linear patterns (relationships) in the MNIST data. But, the result is still far away from the original MNIST digits.



Original MNIST digits (Image by author)

Now, we can visualize test MNIST data in the latent space to see how the autoencoder model with two hidden layers is capable of distinguishing between the nine digits.

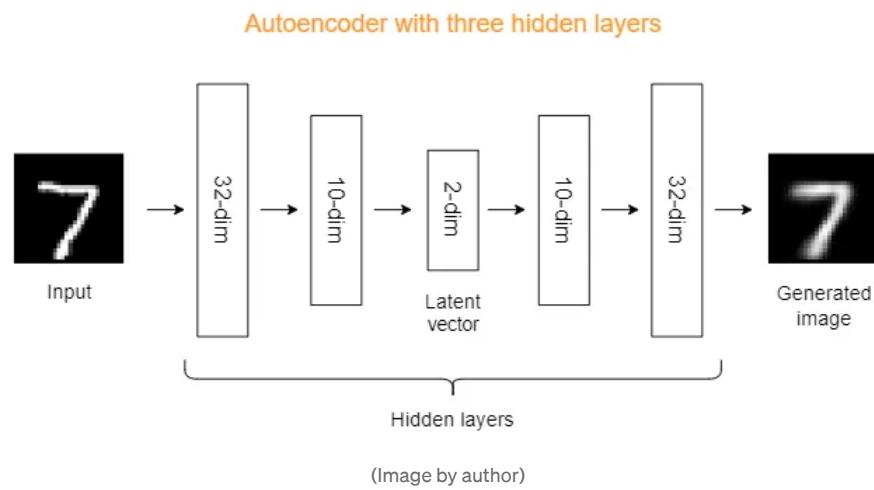


Visualize the test MNIST data in the latent space (Image by author)

As compared to the previous output, some digits have clear separate clusters in the latent space and some do not. It means that the autoencoder model with two hidden layers can distinguish between the nine digits in the test MNIST data to some extent, but not perfectly!

As a solution for this, we can further increase the number of hidden layers in the autoencoder model.

Autoencoder with three hidden layers



The above autoencoder architecture can be coded as follows.

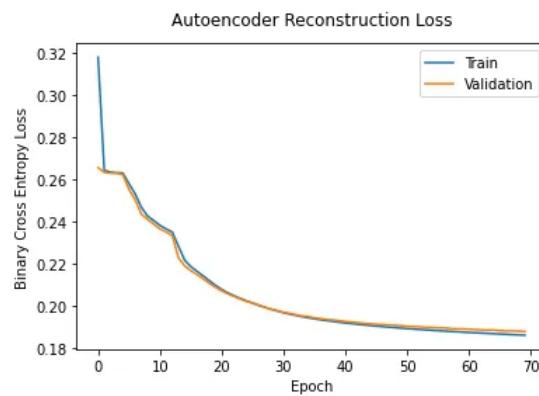
Defining autoencoder architecture with three hidden layers (Code by author)

```
Model: "Autoencoder_3"
```

Layer (type)	Output Shape	Param #
<hr/>		
input_5 (InputLayer)	[(None, 784)]	0
dense_14 (Dense)	(None, 32)	25120
dense_15 (Dense)	(None, 10)	330
dense_16 (Dense)	(None, 2)	22
dense_17 (Dense)	(None, 10)	30
dense_18 (Dense)	(None, 32)	352
dense_19 (Dense)	(None, 784)	25872
<hr/>		
Total params: 51,726		
Trainable params: 51,726		
Non-trainable params: 0		

(Image by author)

Now, we can train the model as previously.



(Image by author)

Now, we can visualize the compressed MNIST digits after encoding.



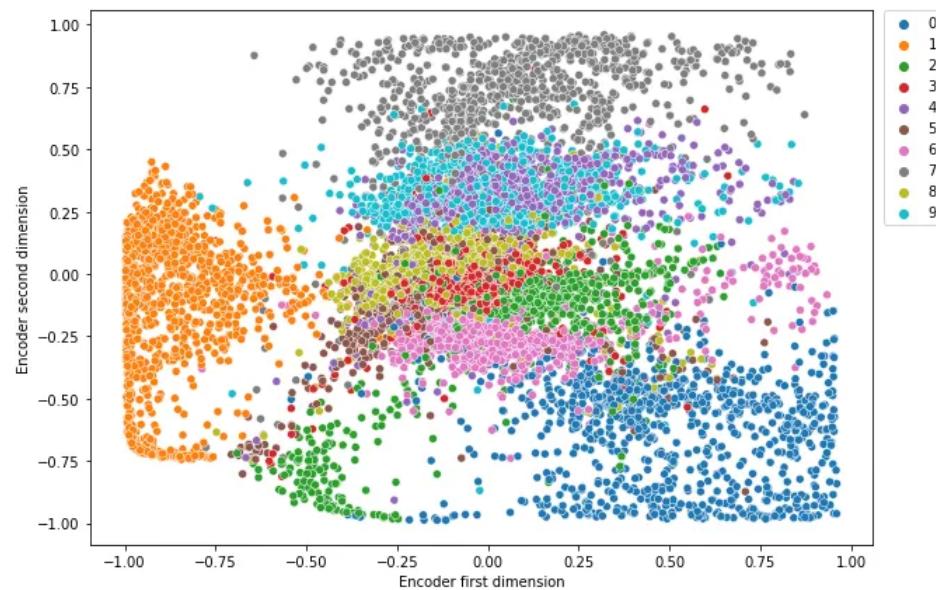
The output of autoencoder with three hidden layers (Image by author)

This output is much better than the previous two outputs because three hidden layers can capture most of the complex non-linear patterns (relationships) in the MNIST data. In addition to that, this output is close enough to the original MNIST digits, but it is not perfect!



Original MNIST digits (Image by author)

Now, we can visualize test MNIST data in the latent space to see how the autoencoder model with three hidden layers is capable of distinguishing between the nine digits.



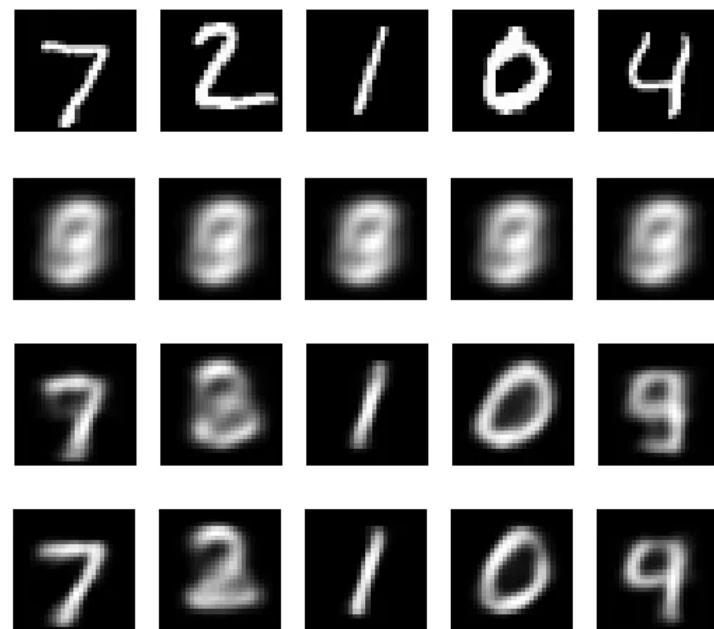
Visualize the test MNIST data in the latent space (Image by author)

As compared to the previous two outputs, most of the digits have clear separate clusters in the latent space. It means that the autoencoder model with three hidden layers can clearly distinguish between the nine digits in the test MNIST data.

Conclusion

You may try to further increase the number of hidden layers in the autoencoder model. When you do so, as the model becomes more complex with many hidden layers, it may result in overfitting that significantly reduces the quality of autoencoder latent representation. To mitigate overfitting in autoencoder models, you can try out different regularization techniques such as dropout or early stopping.

To conclude this article, I'd like to show you the outputs of the three autoencoder models in one picture as compared to the original MNIST data.



Autoencoder outputs in one picture (Image by author)

The top row represents the original MNIST data. As we increase the number of hidden layers in the autoencoder model one by one, the quality of autoencoder latent representation increases! We can consider further increasing the number of hidden layers, but that may result in overfitting as I mentioned earlier.

This is the end of today's post.

Please let me know if you've any questions or feedback.

Read next (Recommended)

- An Introduction to Autoencoders in Deep Learning

An Introduction to Autoencoders in Deep Learning

Understanding the principles behind autoencoders

[rukshanpramoditha.medium.com](https://rukshanpramoditha.medium.com/an-introduction-to-autoencoders-in-deep-learning-181215c8e7d1)

- How the Dimension of Autoencoder Latent Vector Affects the Quality of Latent Representation

How the Dimension of Autoencoder Latent Vector Affects the Quality of Latent Representation

Hyperparameter tuning in autoencoders — Part 2

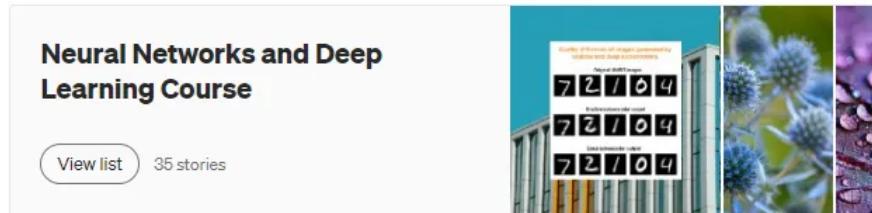
[rukshanpramoditha.medium.com](https://rukshanpramoditha.medium.com/how-the-dimension-of-autoencoder-latent-vector-affects-the-quality-of-latent-representation-181215c8e7d1)

- How Autoencoders Outperform PCA in Dimensionality Reduction

How Autoencoders Outperform PCA in Dimensionality Reduction

Dimensionality reduction with autoencoders using non-linear data
[towardsdatascience.com](https://towardsdatascience.com/how-number-of-hidden-layers-affects-the-quality-of-autoencoder-latent-representation-181215c8e7d1)

- All episodes of my “[Neural Networks and Deep Learning Course](#)”



Click on the image to access my Neural Networks and Deep Learning Course (Screenshot by author)

Support me as a writer

I hope you enjoyed reading this article. If you'd like to support me as a writer, kindly consider [signing up for a membership](#) to get unlimited access to Medium. It only costs \$5 per month and I will receive a portion of your membership fee.

Join Medium with my referral link - Rukshan Pramoditha

Read every story from Rukshan Pramoditha (and thousands of other writers on Medium). Your membership fee directly...

rukshanpramoditha.medium.com

Thank you so much for your continuous support! See you in the next article.
 Happy learning to everyone!

MNIST dataset info

- **Citation:** Deng, L., 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6), pp. 141–142.
- **Source:** <http://yann.lecun.com/exdb/mnist/>
- **License:** Yann LeCun (Courant Institute, NYU) and Corinna Cortes (Google Labs, New York) hold the copyright of the MNIST dataset which is

Rukshan Pramoditha

2022-08-23

Artificial Intelligence

Deep Learning

Autoencoder

Data Science

Hyperparameter Tuning



Written by Rukshan Pramoditha

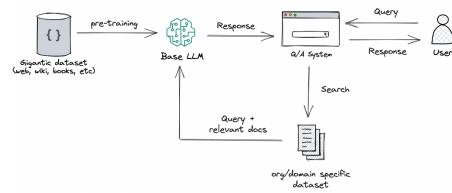
6.1K Followers · Writer for Towards Data Science

Follow



2,000,000+ Views | BSc in Stats | Top 50 Data Science, AI/ML Technical Writer on Medium | Data Science Masterclass:
<https://datasciencemasterclass.substack.com/>

More from Rukshan Pramoditha and Towards Data Science



Rukshan Pramoditha in Towards Data Science

Why do we set a random state in machine learning models?

random_state=42

◆ · 6 min read · Apr 30, 2022

217

4

+

...

2K

16

+

...



Giuseppe Scalamogna in Towards Data Science

New ChatGPT Prompt Engineering Technique: Program Simulation

A potentially novel technique for turning a ChatGPT prompt into a mini-app.

9 min read · Sep 4

1.2K

12

+

...

1.1K

10

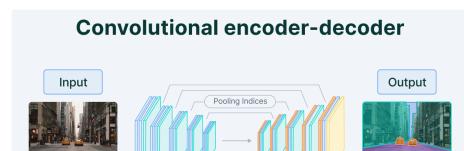
+

...

[See all from Rukshan Pramoditha](#)

[See all from Towards Data Science](#)

Recommended from Medium



Piero Paialunga in Towards Data Science

Learn AI

Hands-on Generative Adversarial Networks (GAN) for Signal...

Here's how to build a generative Deep Learning model for Signal Processing in a fe...

8 min read · Dec 28, 2022

344

6



...

Autoencoders

What is an autoencoder?

12 min read · Mar 26

5



...

Lists

Predictive Modeling w/ Python

20 stories · 417 saves

ChatGPT

21 stories · 162 saves

ChatGPT prompts

24 stories · 413 saves

Natural Language Processing

640 stories · 245 saves

Frederik vl

Understanding Bias and Variance in Machine Learning

The terms bias and variance describe how well the model fits the actual unknown data...

3 min read · Sep 16

24



...

Hayden LaBrie in GoPenAI

Understanding Autoencoders—Part VIII: The Decoder

The decoder is the second main part of the autoencoder. The decoder, at a high level,...

4 min read · Aug 30

15



...

 AL Anany 

The ChatGPT Hype Is Over—Now Watch How Google Will Kill...

It never happens instantly. The business game is longer than you know.

◆ · 6 min read · Sep 2



9.1K



278



...

 Sadaf Saleem

Neural Networks in 10mins. Simply Explained!

What are Neural Networks?

9 min read · May 15



122



1



...

[See more recommendations](#)