Chen Data Science & AI for Neuroscience Summer School

CHEN TIANQIAO & CHRISSY INSTITUTE

Caltech

Autoencoders

Sabera Talukder

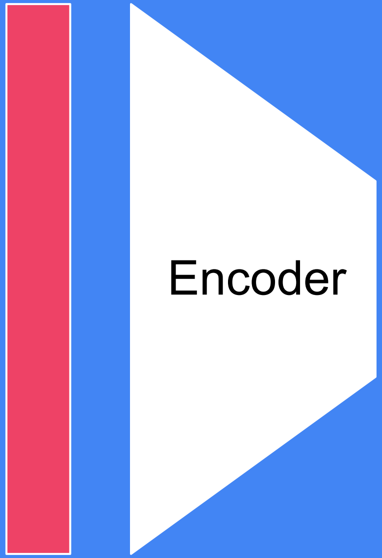
Representations in the context of architectures!

Principal Components Analysis

Principal Components Analysis

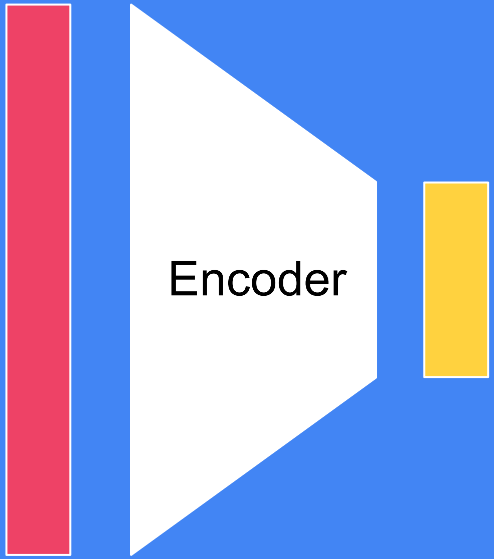
Input data

Principal Components Analysis



Input data

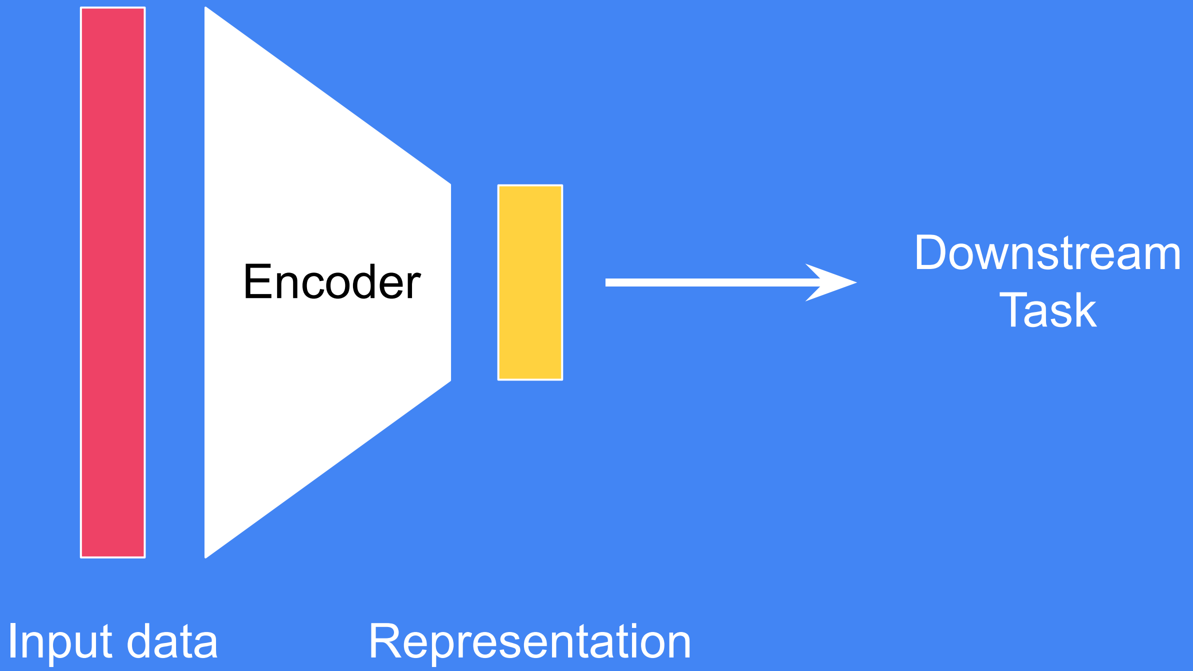
Principal Components Analysis



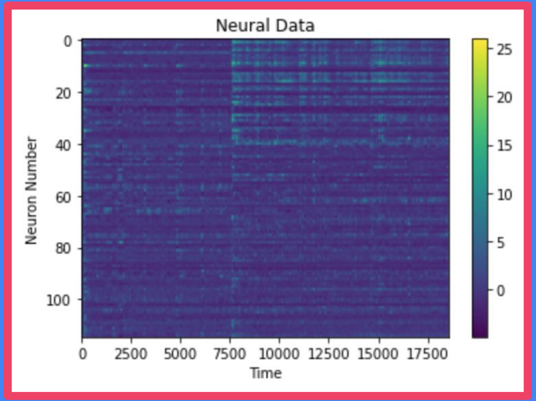
Input data

Representation

Principal Components Analysis

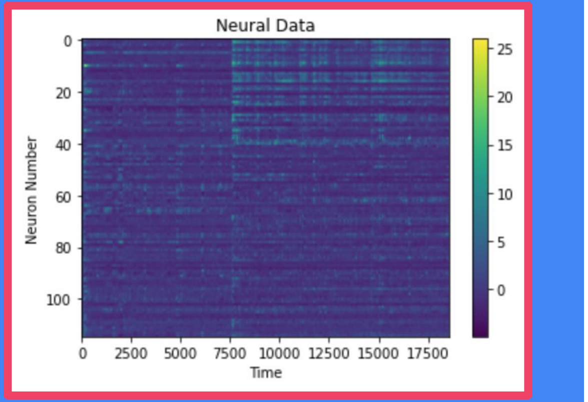


Principal Components Analysis

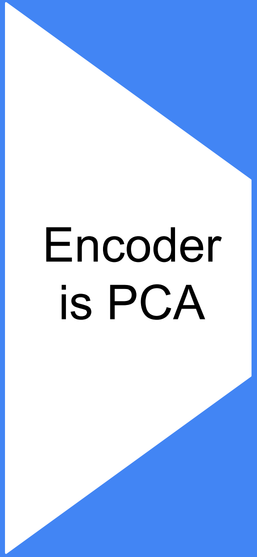


Input data

Principal Components Analysis

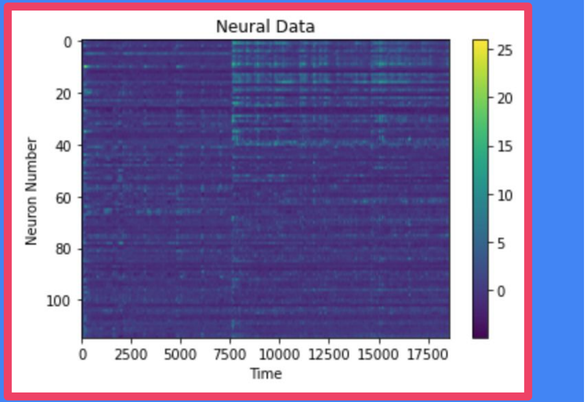


Input data



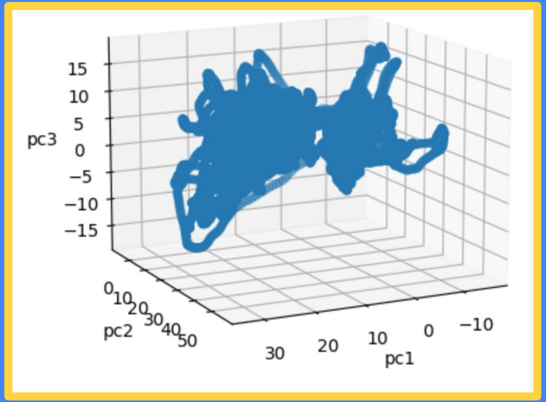
Encoder is PCA

Principal Components Analysis



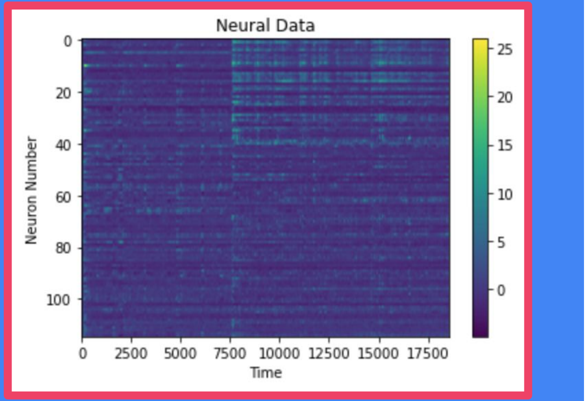
Input data

Encoder
is PCA



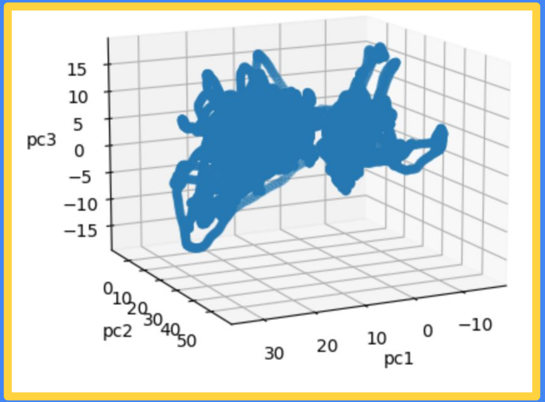
Representation

Principal Components Analysis

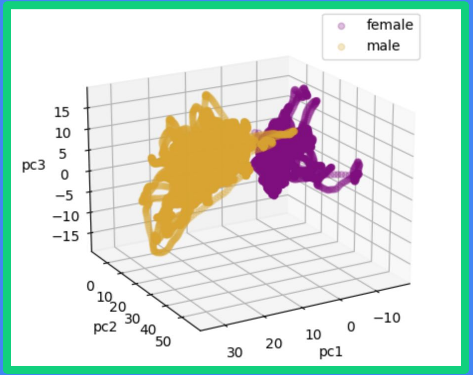


Input data

Encoder is PCA

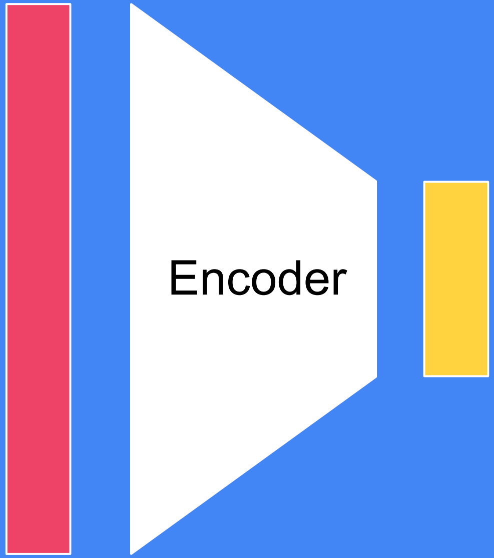


Representation



Downstream Task

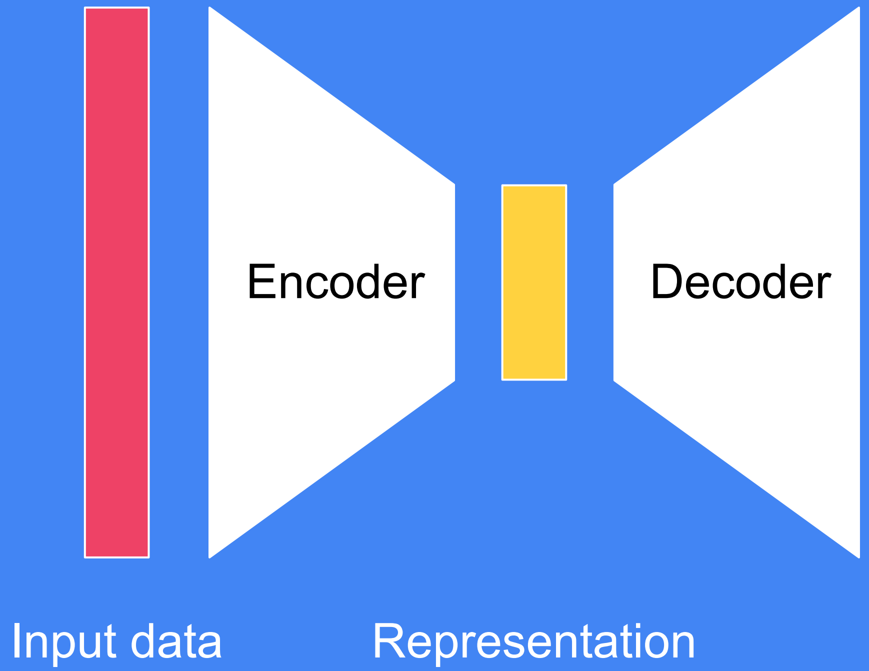
Autoencoder



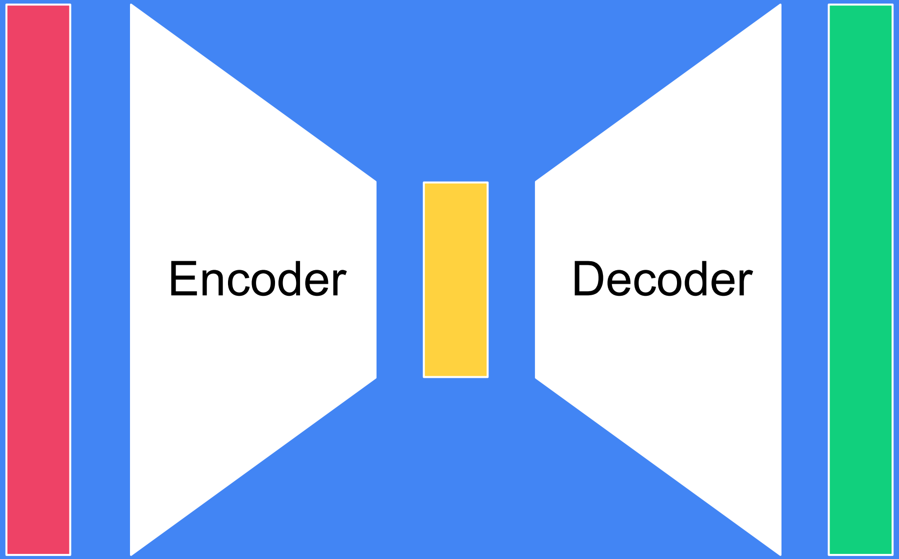
Input data

Representation

Autoencoder



Autoencoder



Encoder

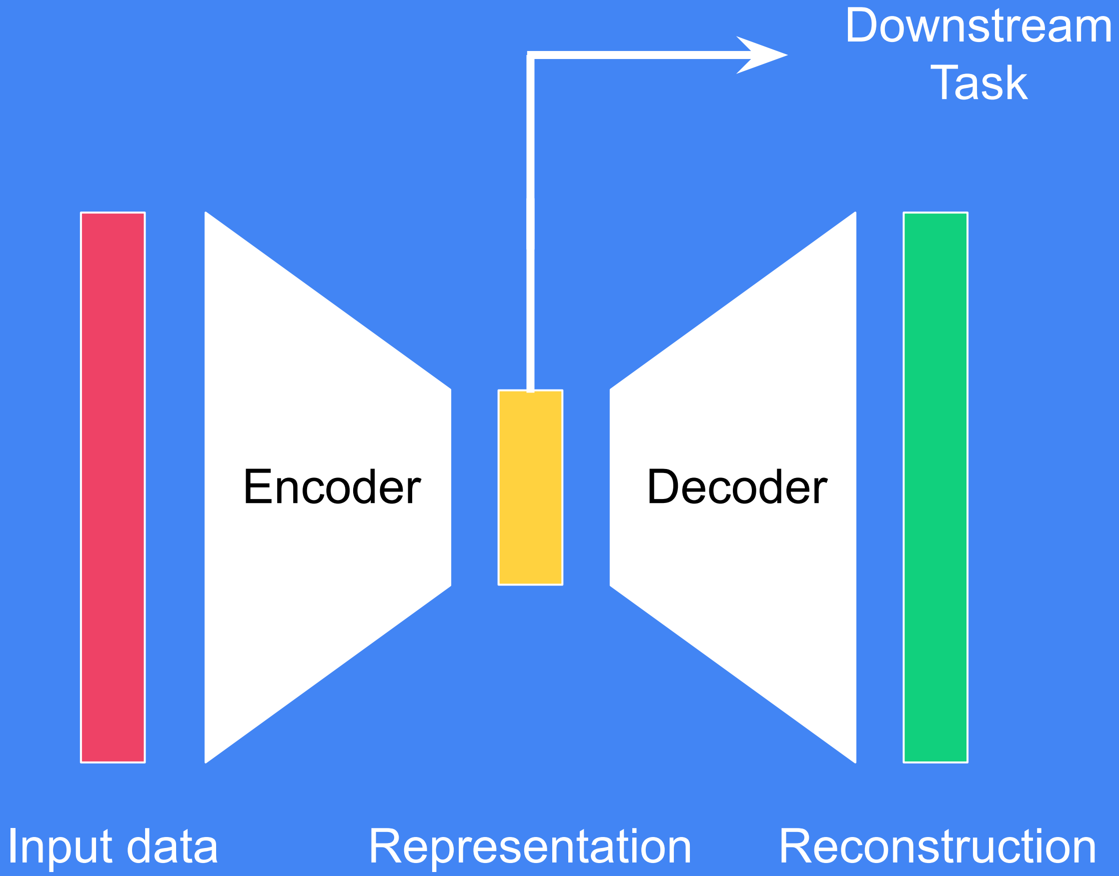
Decoder

Input data

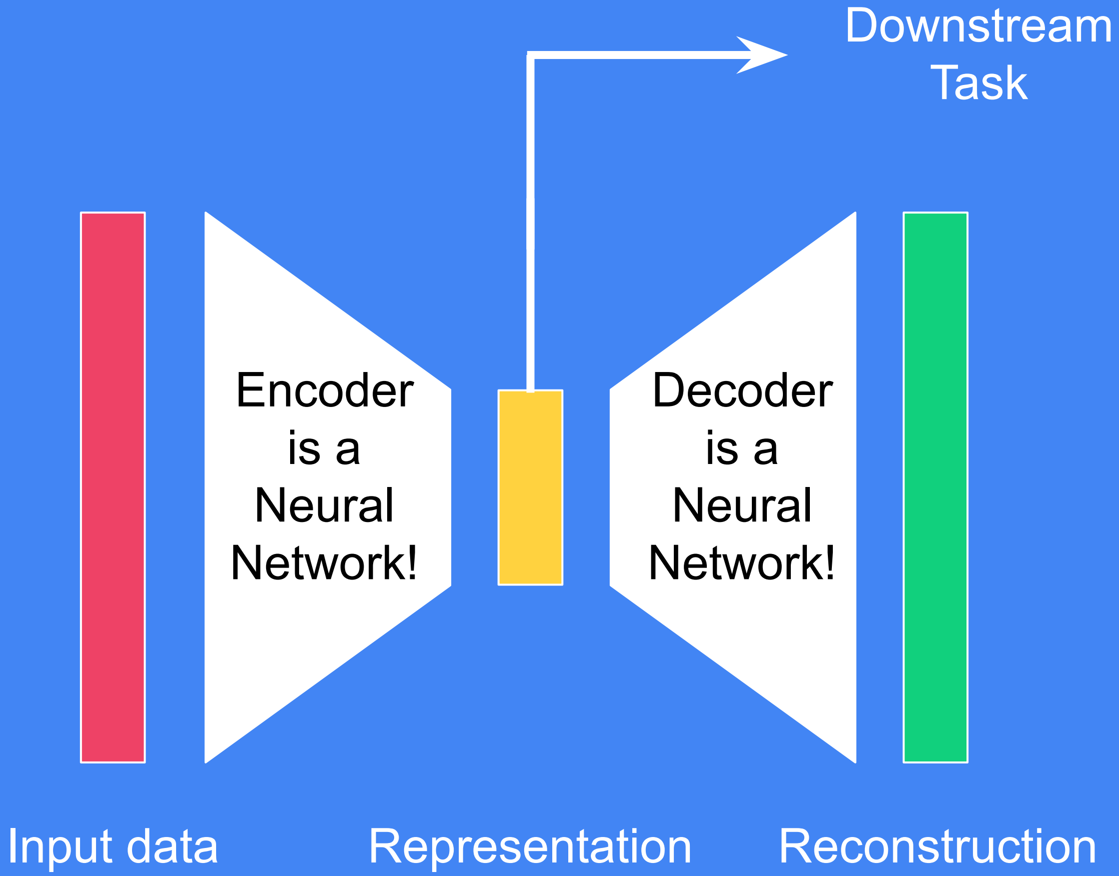
Representation

Reconstruction

Autoencoder



Autoencoder



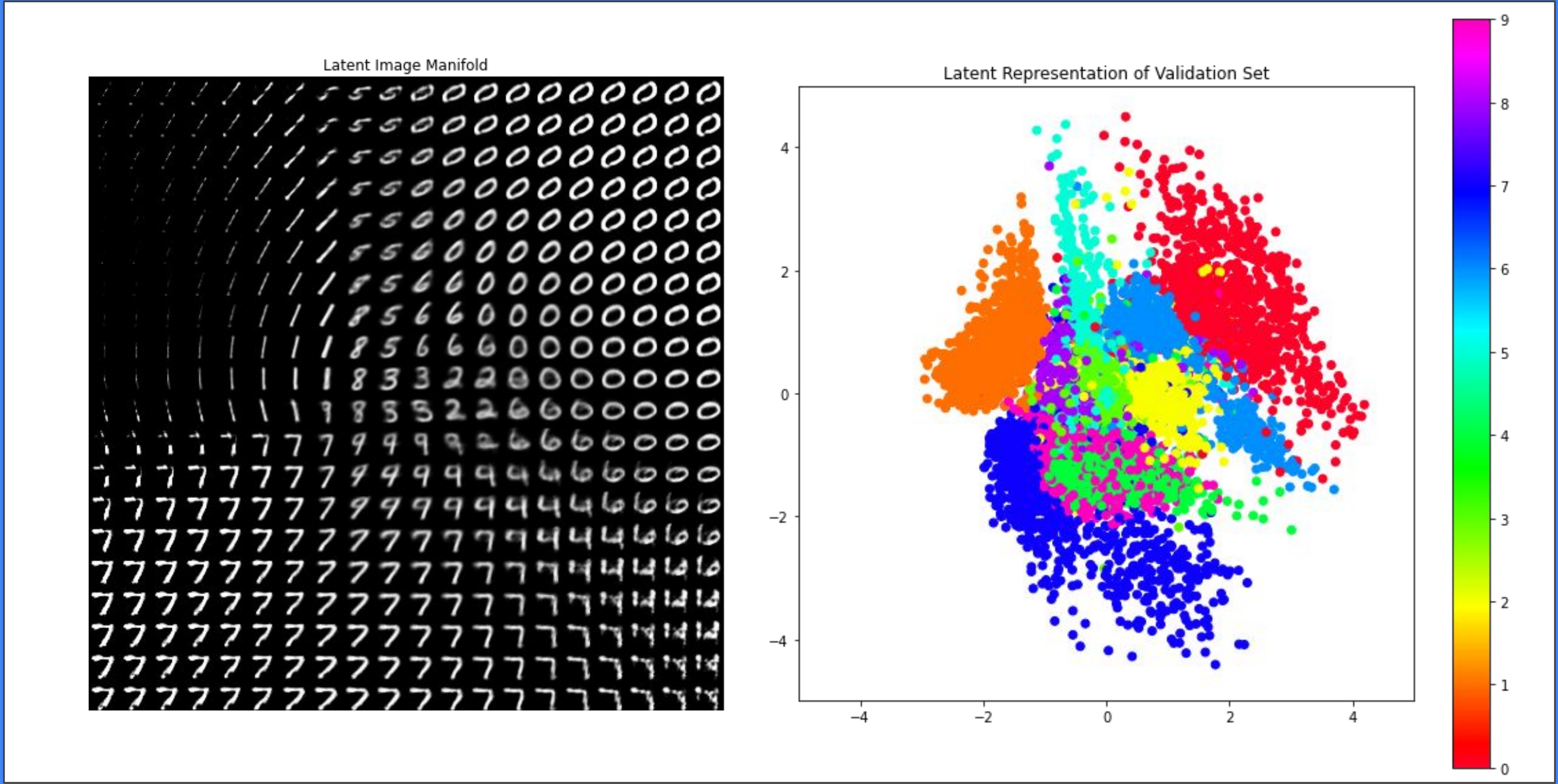
MNIST Example: Dataset

MNIST Example: Dataset

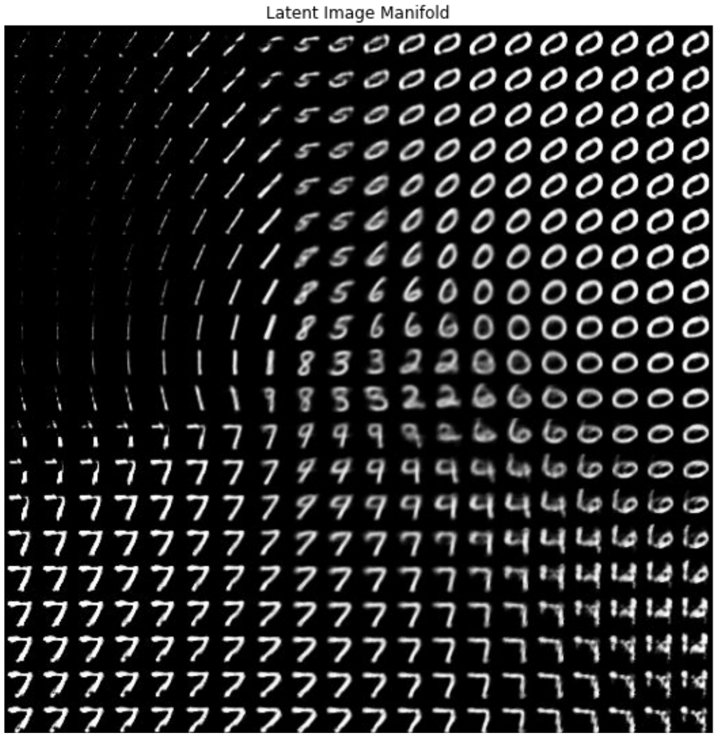
MNIST Example: Dataset

Goal of using an autoencoder on MNIST is to generate handwritten digits!

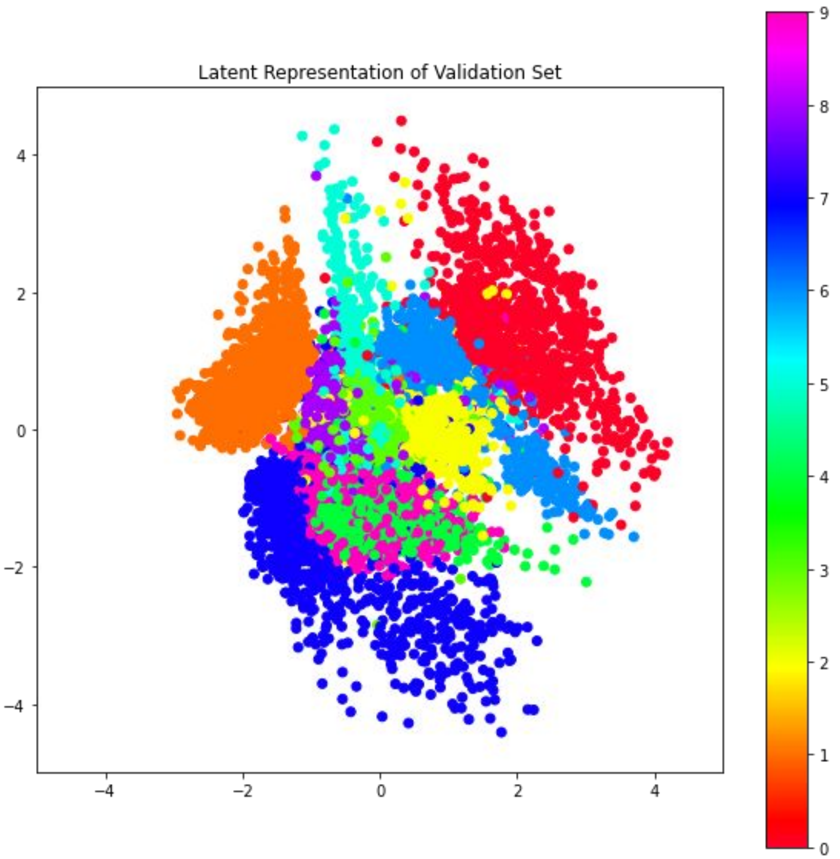
MNIST Example: 2D Latent Space



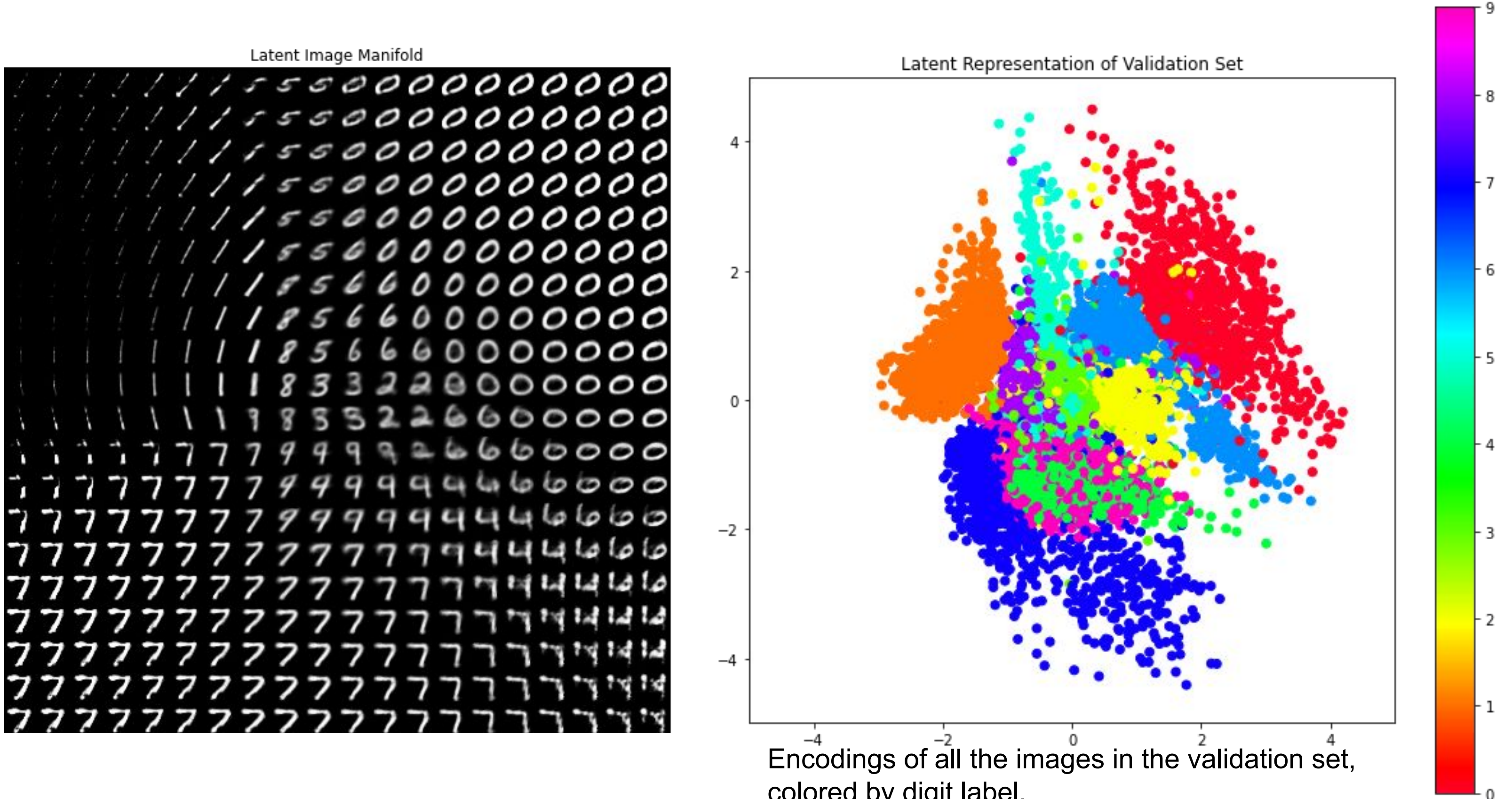
MNIST Example: 2D Latent Space



Sample images generated by picking points in the 2D latent space on a grid and then decoding them.

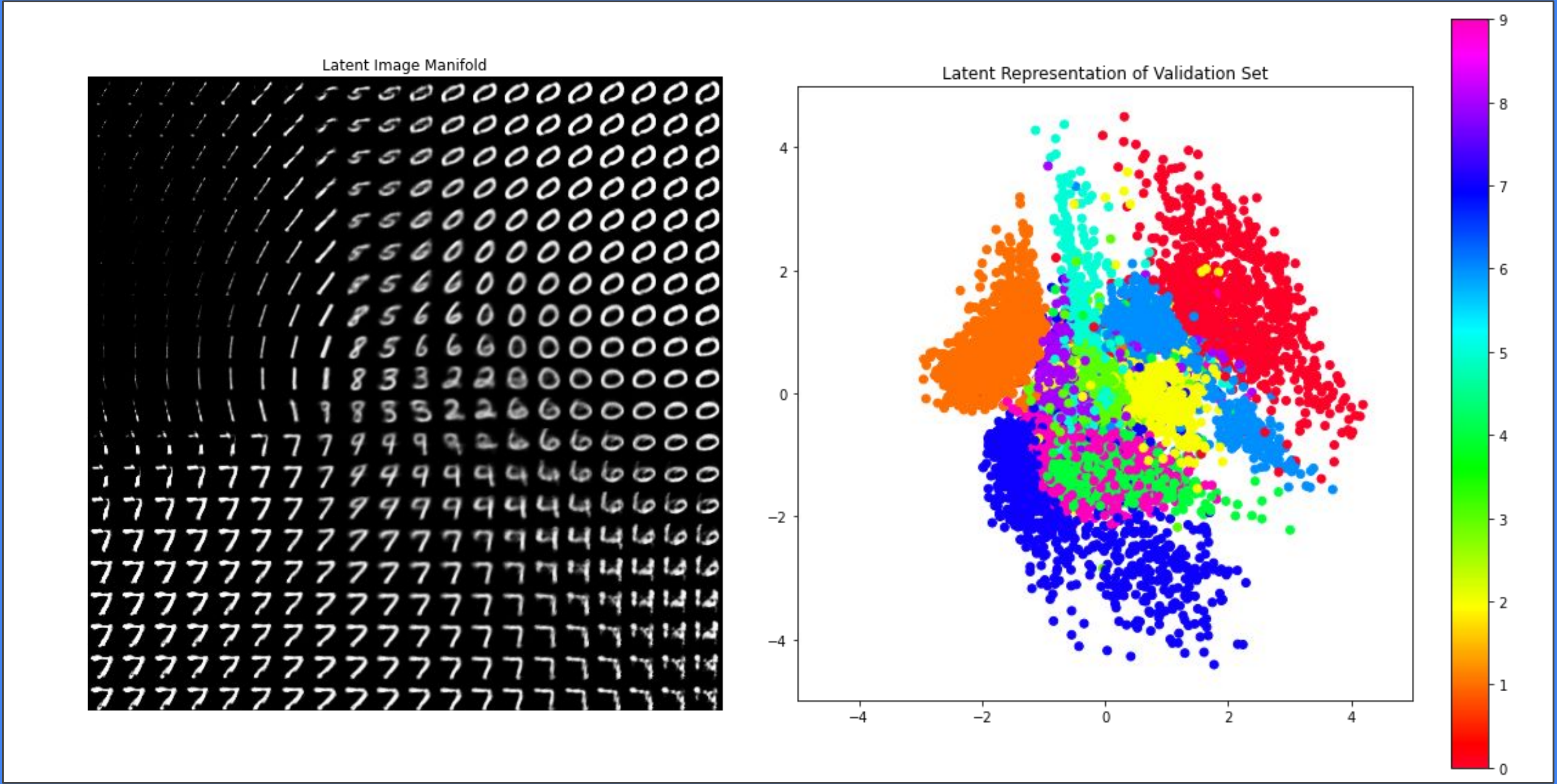


MNIST Example: 2D Latent Space

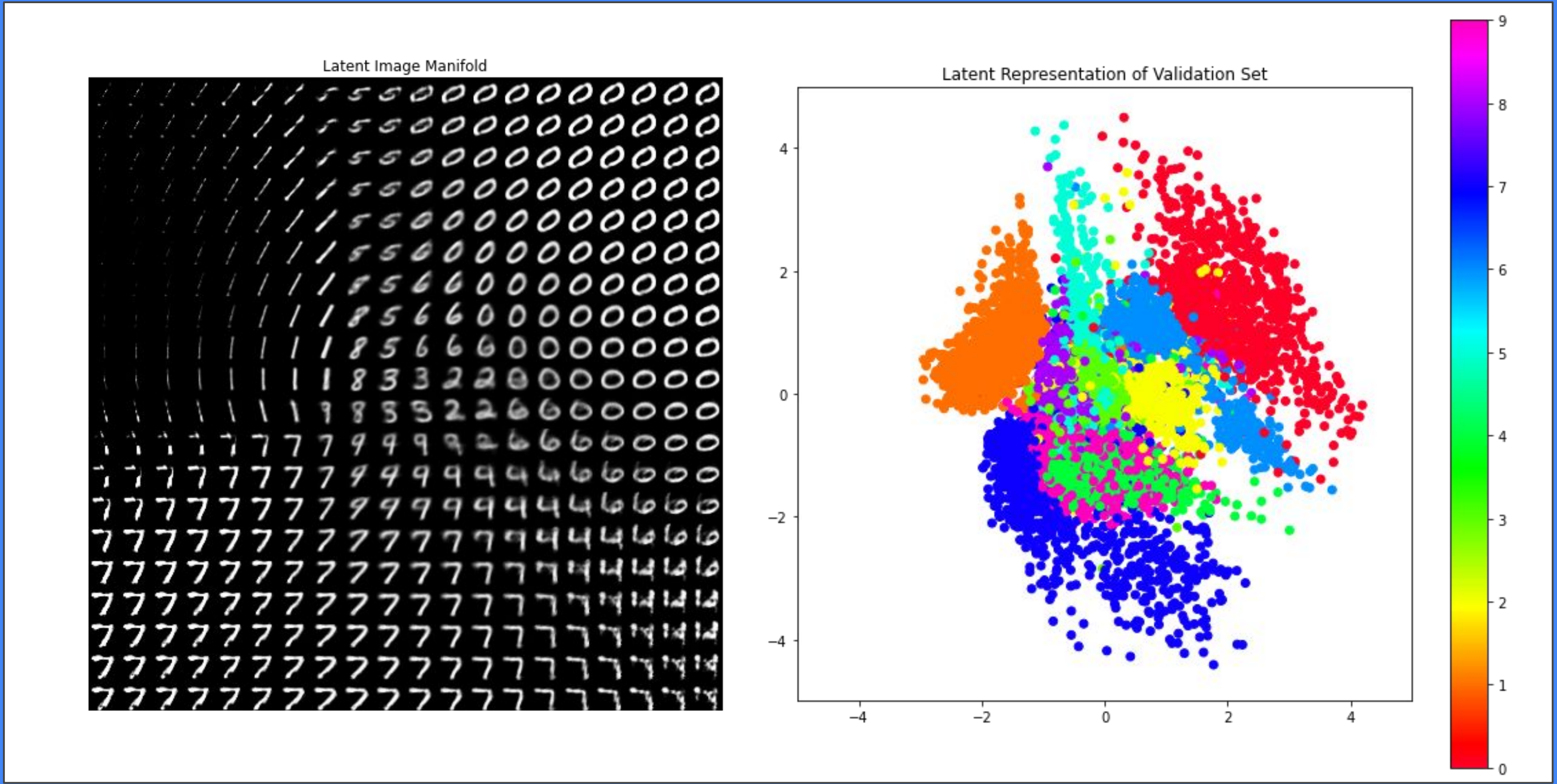


Encodings of all the images in the validation set, colored by digit label.

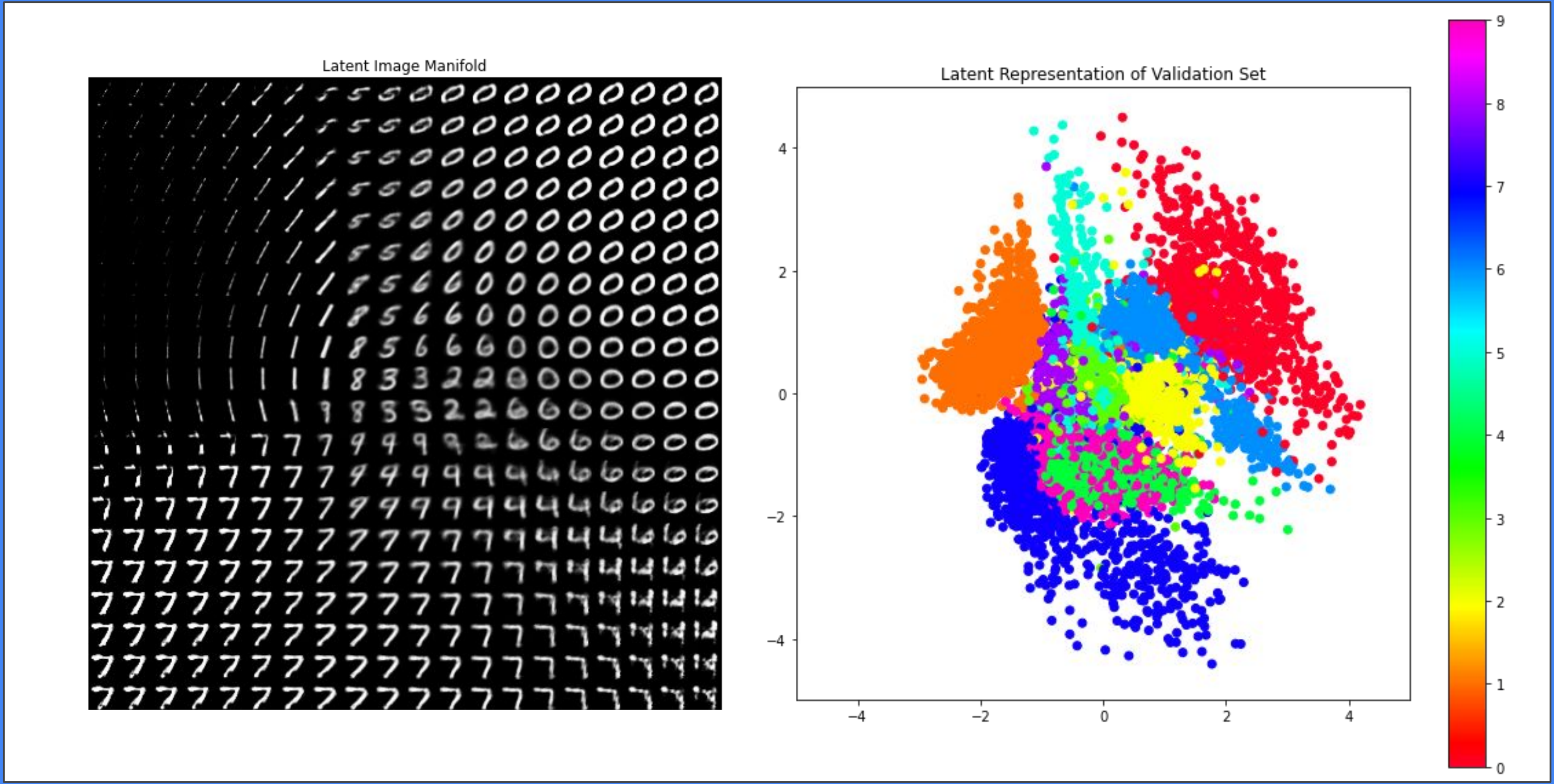
MNIST Example: 2D Latent Space



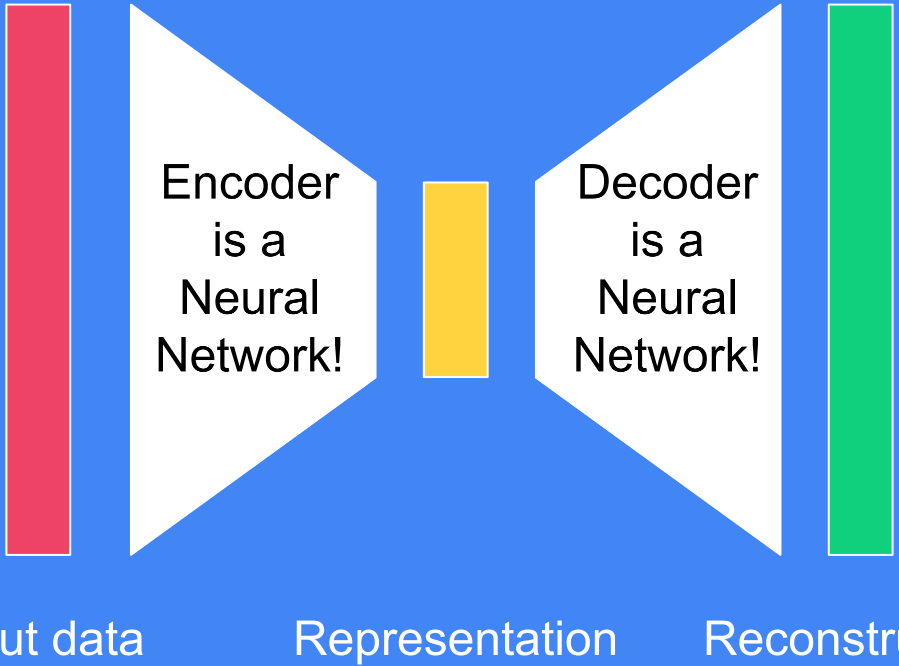
Takeaway 1: There's no "meaning" to the latent space in any strict sense. But, we can get a "qualitative" sense of what happens as we traverse axes. What features do you see in the space?



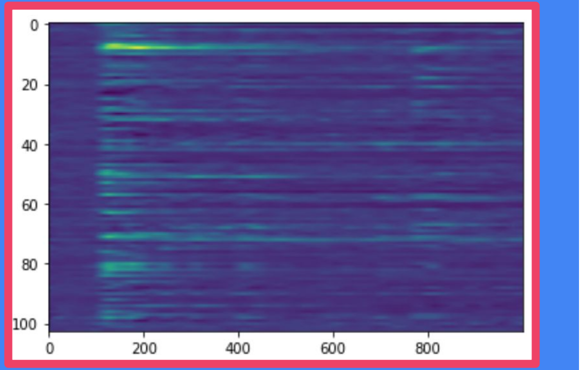
Takeaway 2: This model has no prior concept of what each digit is. However, it learns to essentially embed digit types in localized regions of the latent space. Why? Because this representation is a useful one that it learns that helps with the task of reconstruction, which is what we want from good features.



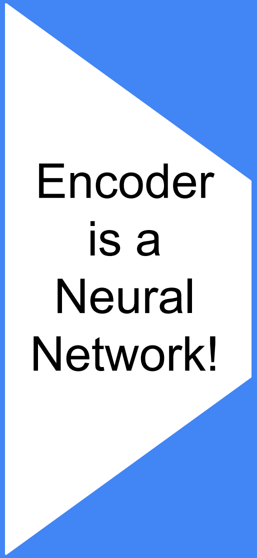
Autoencoder



Autoencoder

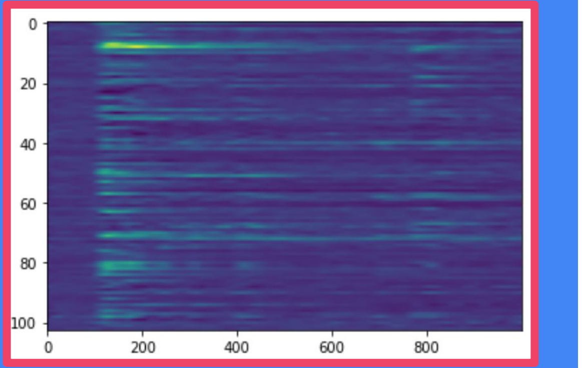


Input data

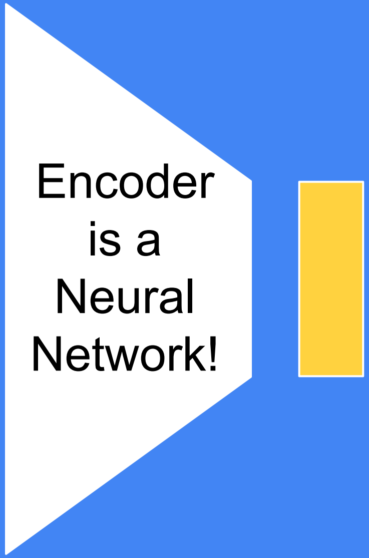


Encoder is a Neural Network!

Autoencoder



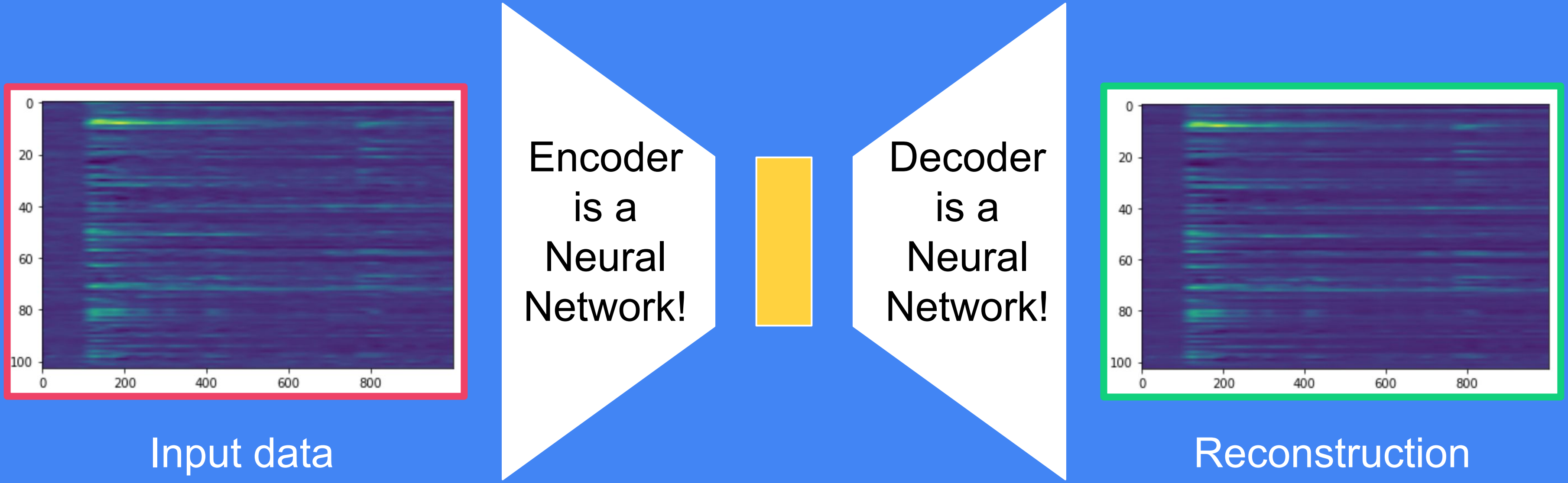
Input data



Encoder is a Neural Network!

Representation

Autoencoder



Encoder is a Neural Network!

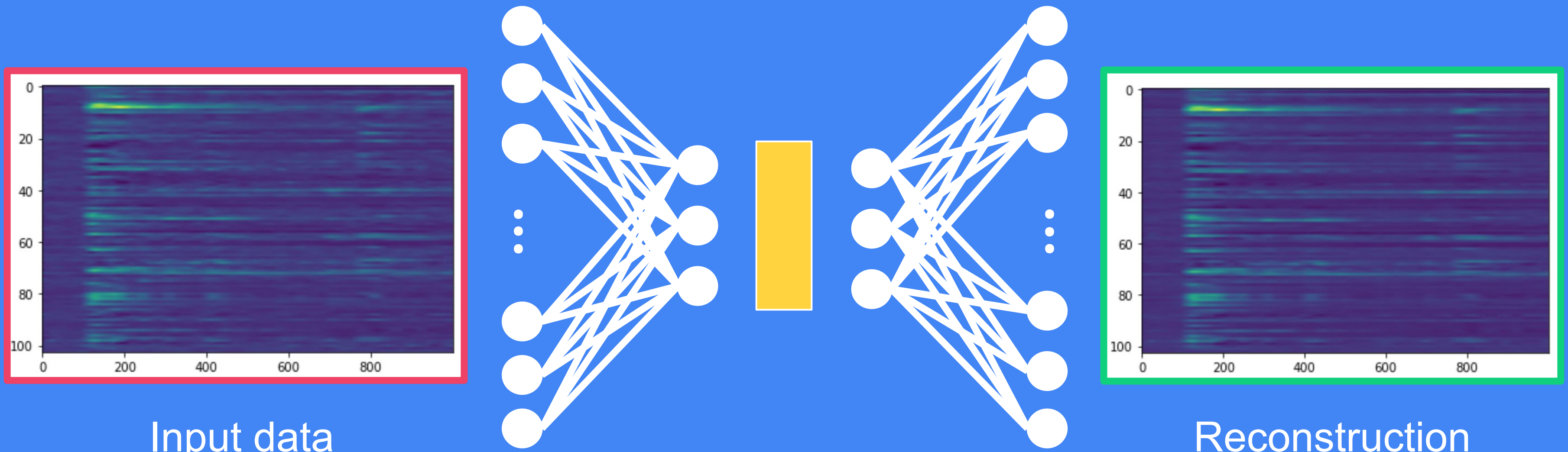
Decoder is a Neural Network!

Input data

Reconstruction

Representation

Autoencoder

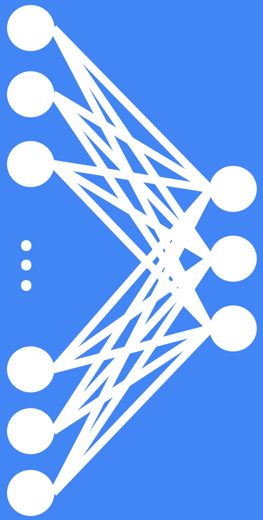


Input data

Reconstruction

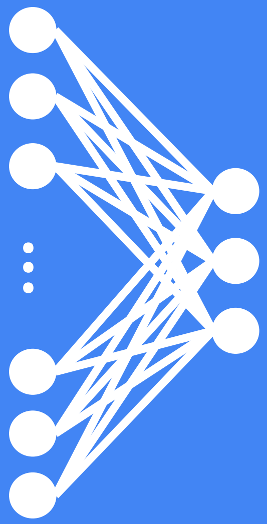
Representation

Linear Layer



Linear Layer

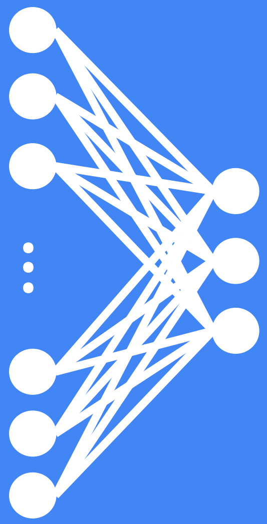
A “linear layer” is just a matrix that multiplies the input and produces some output.



Linear Layer

A “linear layer” is just a matrix that multiplies the input and produces some output.

Fact: the composition of two linear layers is going to be linear. Why is this important? If we only use linear operations in our model, then the expressiveness of our model is going to be restricted significantly.

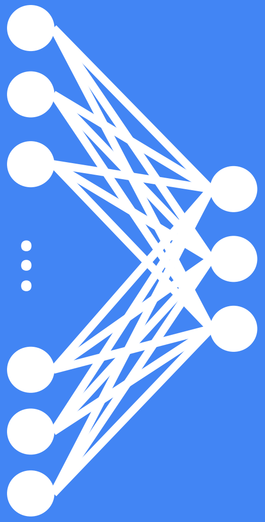


Linear Layer

A “linear layer” is just a matrix that multiplies the input and produces some output.

Fact: the composition of two linear layers is going to be linear. Why is this important? If we only use linear operations in our model, then the expressiveness of our model is going to be restricted significantly.

Nonlinearities go between the linear layers (functions that go between the layers of a neural network are called activation functions) so that when we compose all the layers together, we get a nonlinear function. This allows us to build universal function approximators!

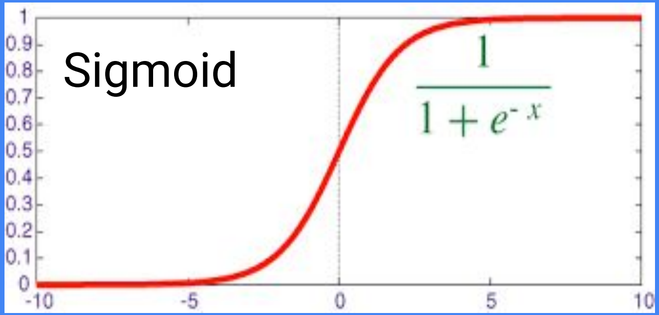


Nonlinearities

We introduce nonlinearities in our model in the form of activation functions.

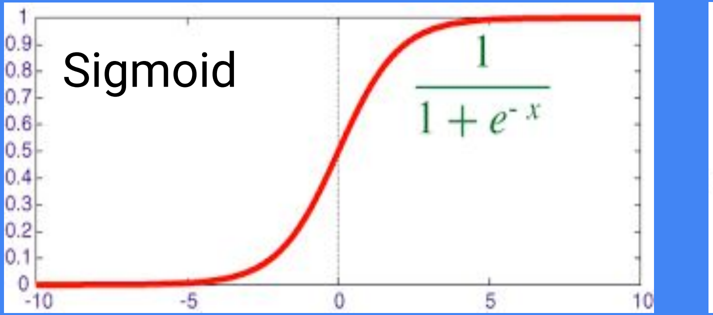
Nonlinearities

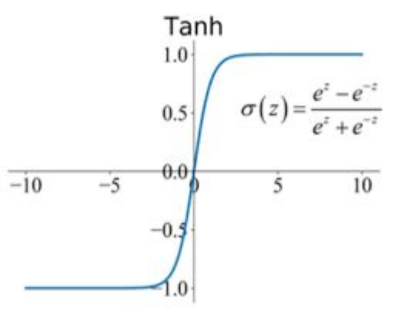
We introduce nonlinearities in our model in the form of activation functions.



Nonlinearities

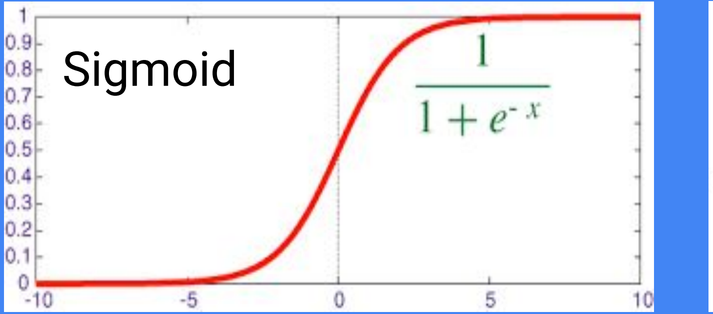
We introduce nonlinearities in our model in the form of activation functions.

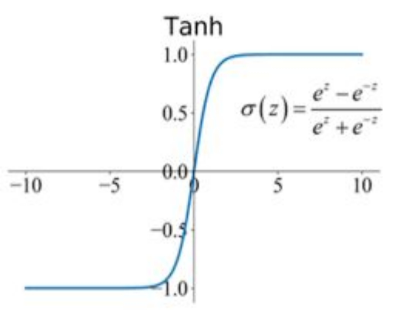


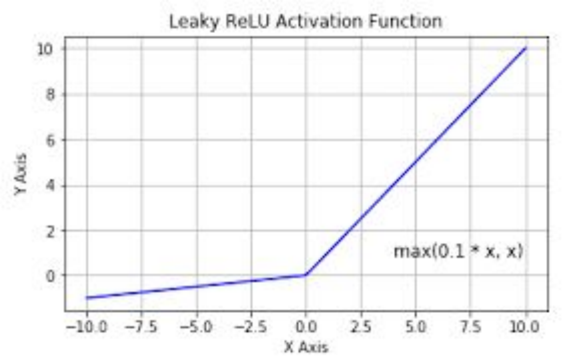


Nonlinearities

We introduce nonlinearities in our model in the form of activation functions.

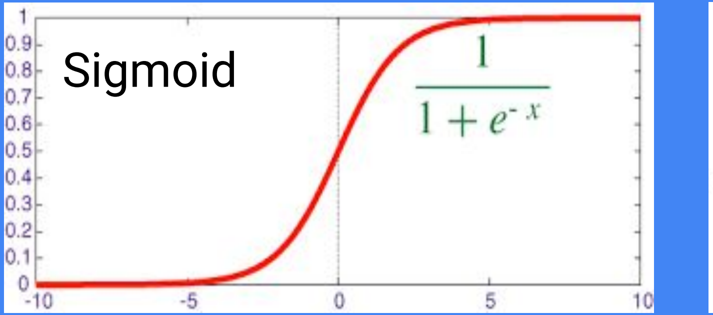


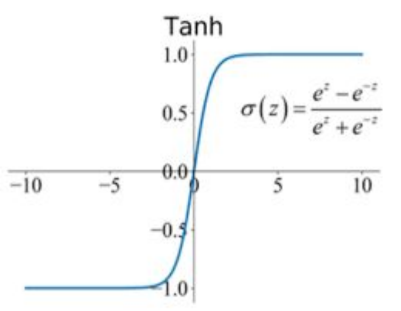


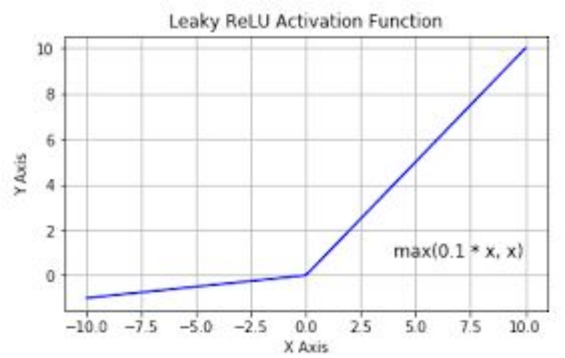


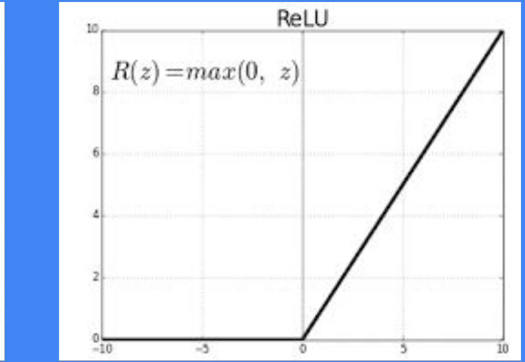
Nonlinearities

We introduce nonlinearities in our model in the form of activation functions.









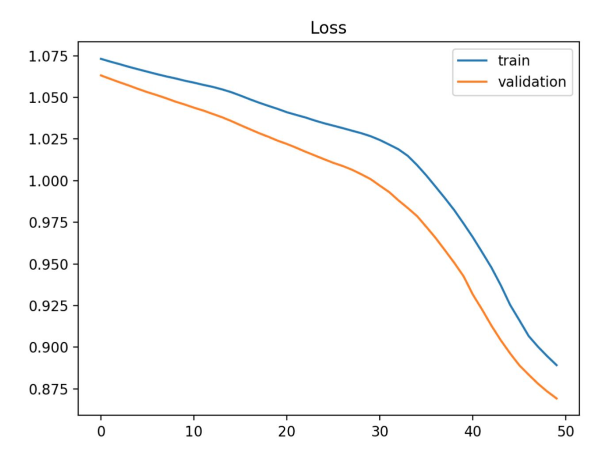
Jump ahead to training!

Jump ahead to training!

Specifically underfitting and overfitting, and how to read training and validation curves!

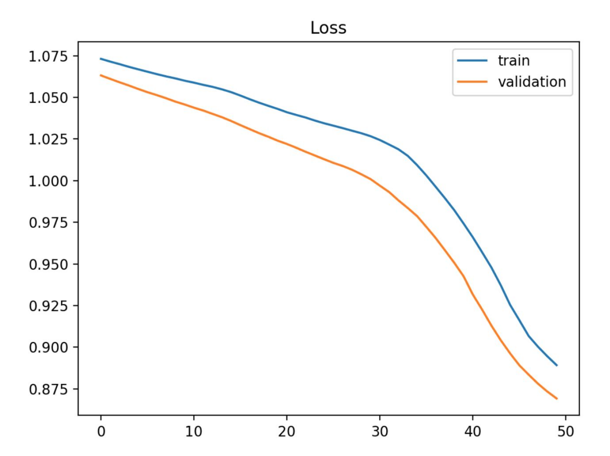
What is the training set versus the validation set?

Underfitting = when the model doesn’t have a low loss on the training set



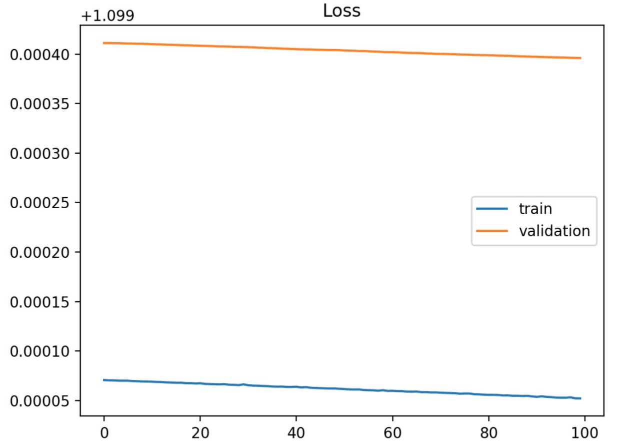
Underfitting = when the model doesn’t have a low loss on the training set

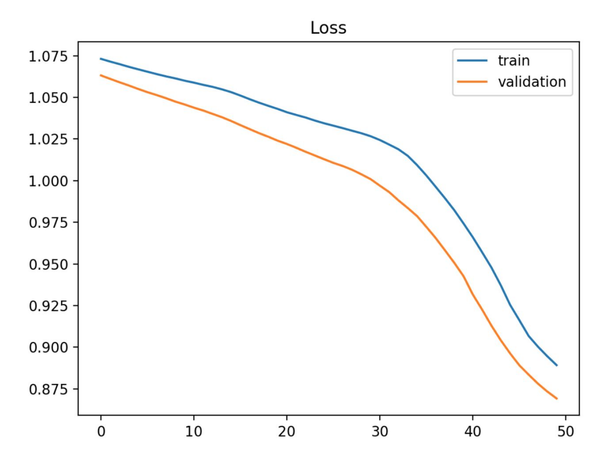
Overfitting = when the model starts to memorize the training set



Underfitting = when the model doesn’t have a low loss on the training set

Overfitting = when the model starts to memorize the training set

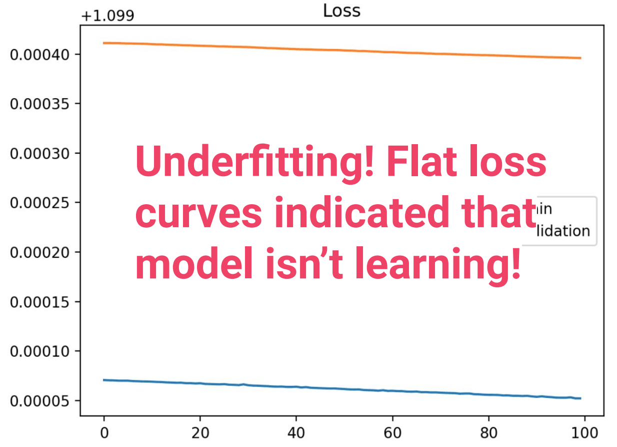




Jason Brownlee

Underfitting = when the model doesn’t have a low loss on the training set

Overfitting = when the model starts to memorize the training set

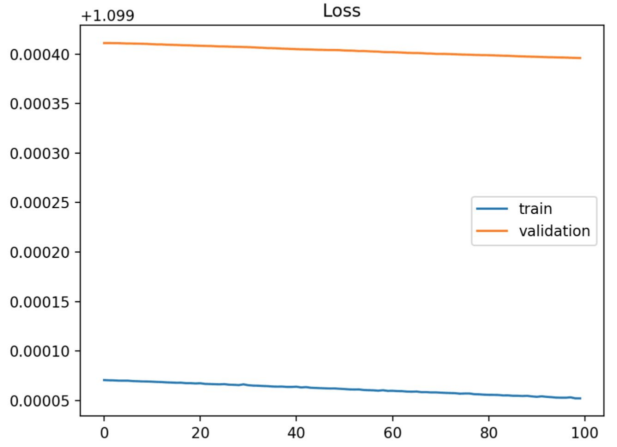


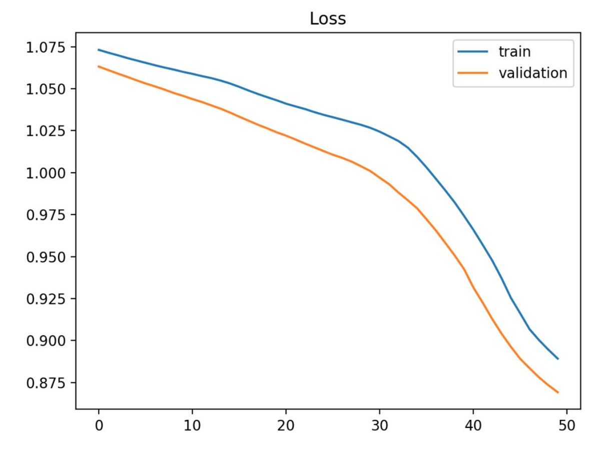
Underfitting! Flat loss curves indicated that model isn’t learning!

Jason Brownlee

Underfitting = when the model doesn’t have a low loss on the training set

Overfitting = when the model starts to memorize the training set

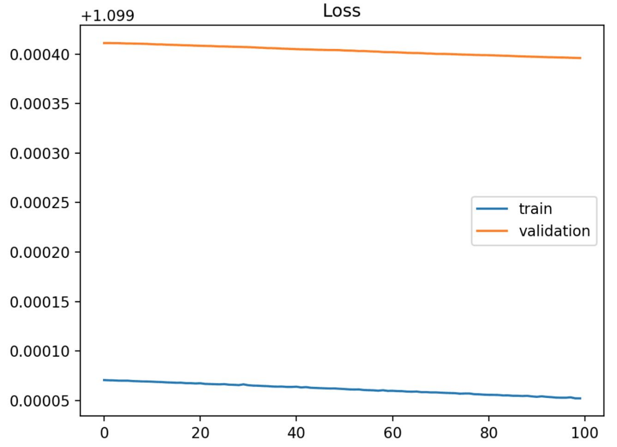


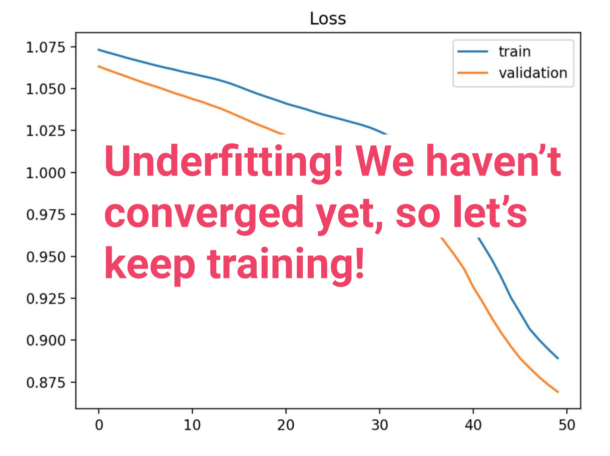


Jason Brownlee

Underfitting = when the model doesn't have a low loss on the training set

Overfitting = when the model starts to memorize the training set



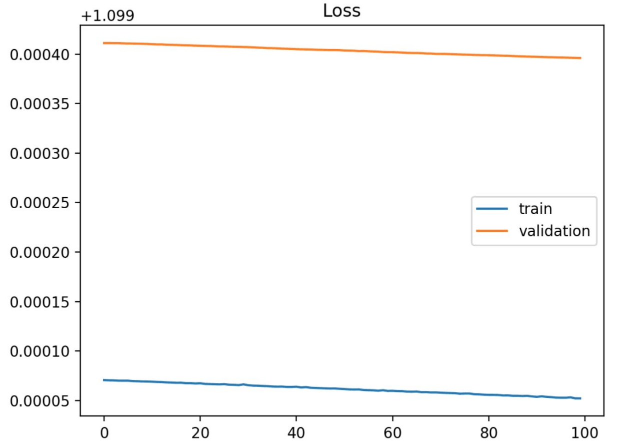


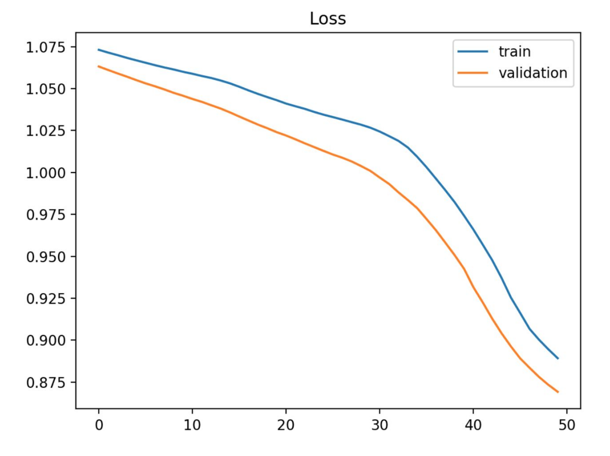
Underfitting! We haven't converged yet, so let's keep training!

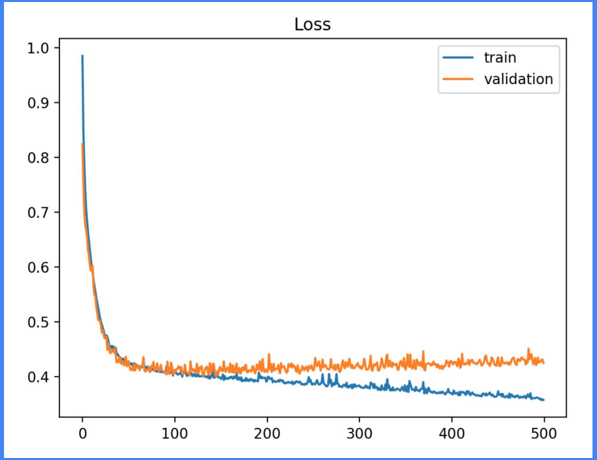
Jason Brownlee

Underfitting = when the model doesn’t have a low loss on the training set

Overfitting = when the model starts to memorize the training set



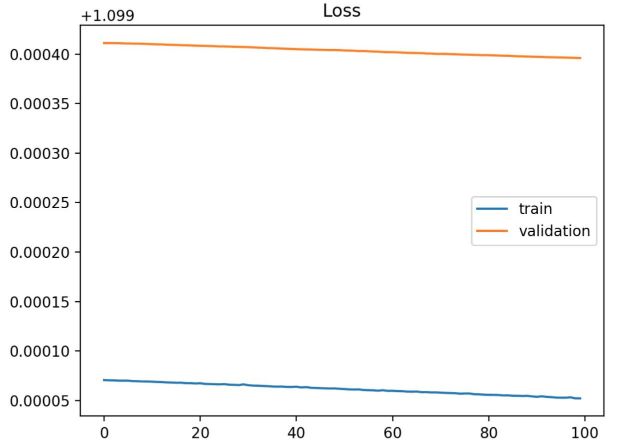


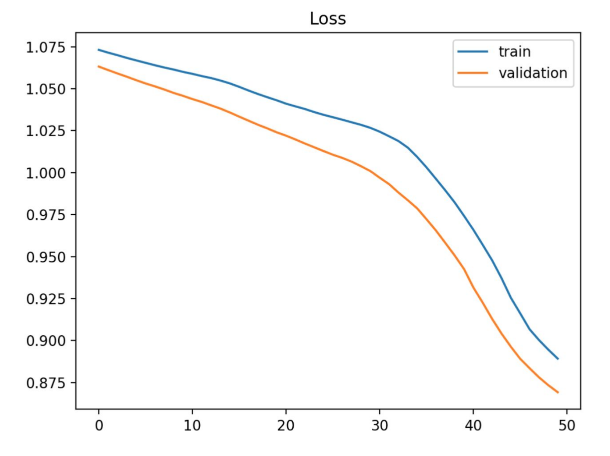


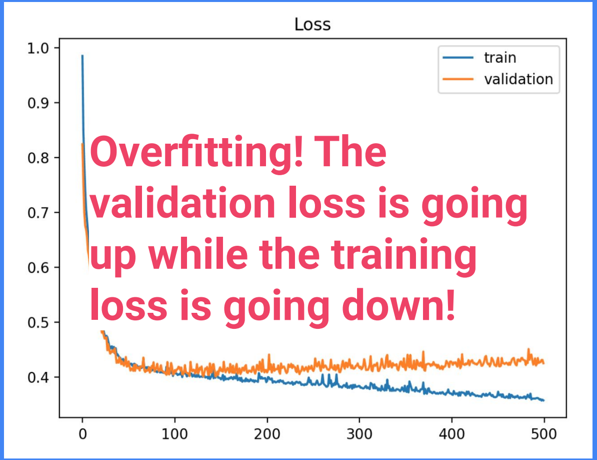
Jason Brownlee

Underfitting = when the model doesn’t have a low loss on the training set

Overfitting = when the model starts to memorize the training set





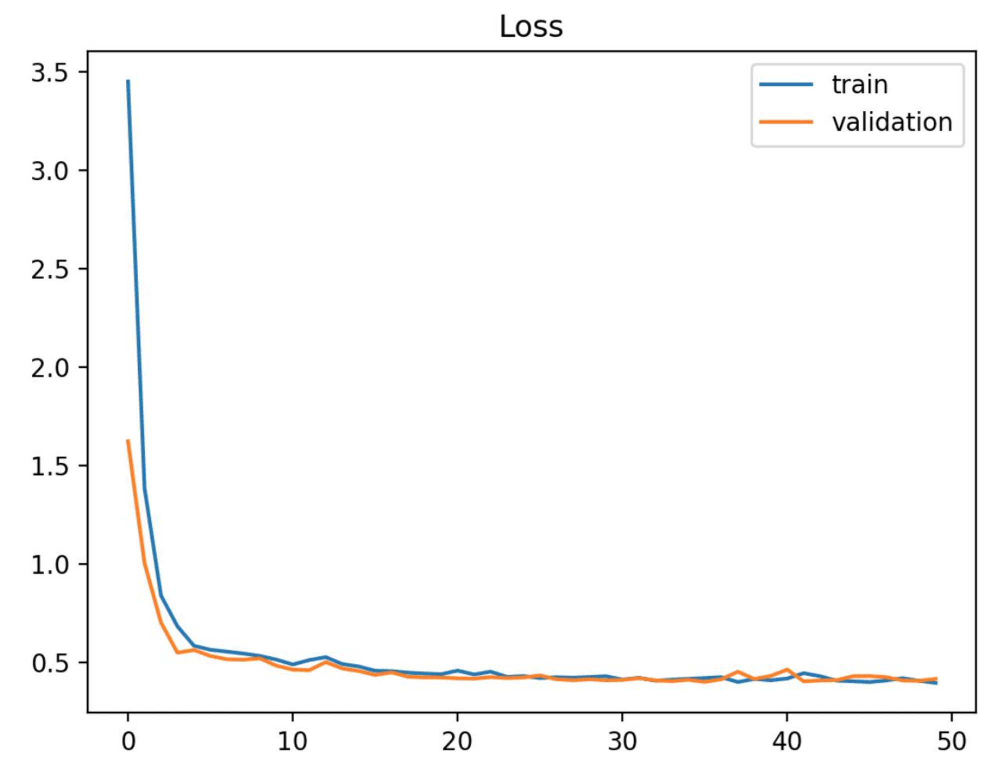


Overfitting! The validation loss is going up while the training loss is going down!

Jason Brownlee

What do you think a well trained model will look like with respect to the training and validation losses?

Just Right!



Jason Brownlee