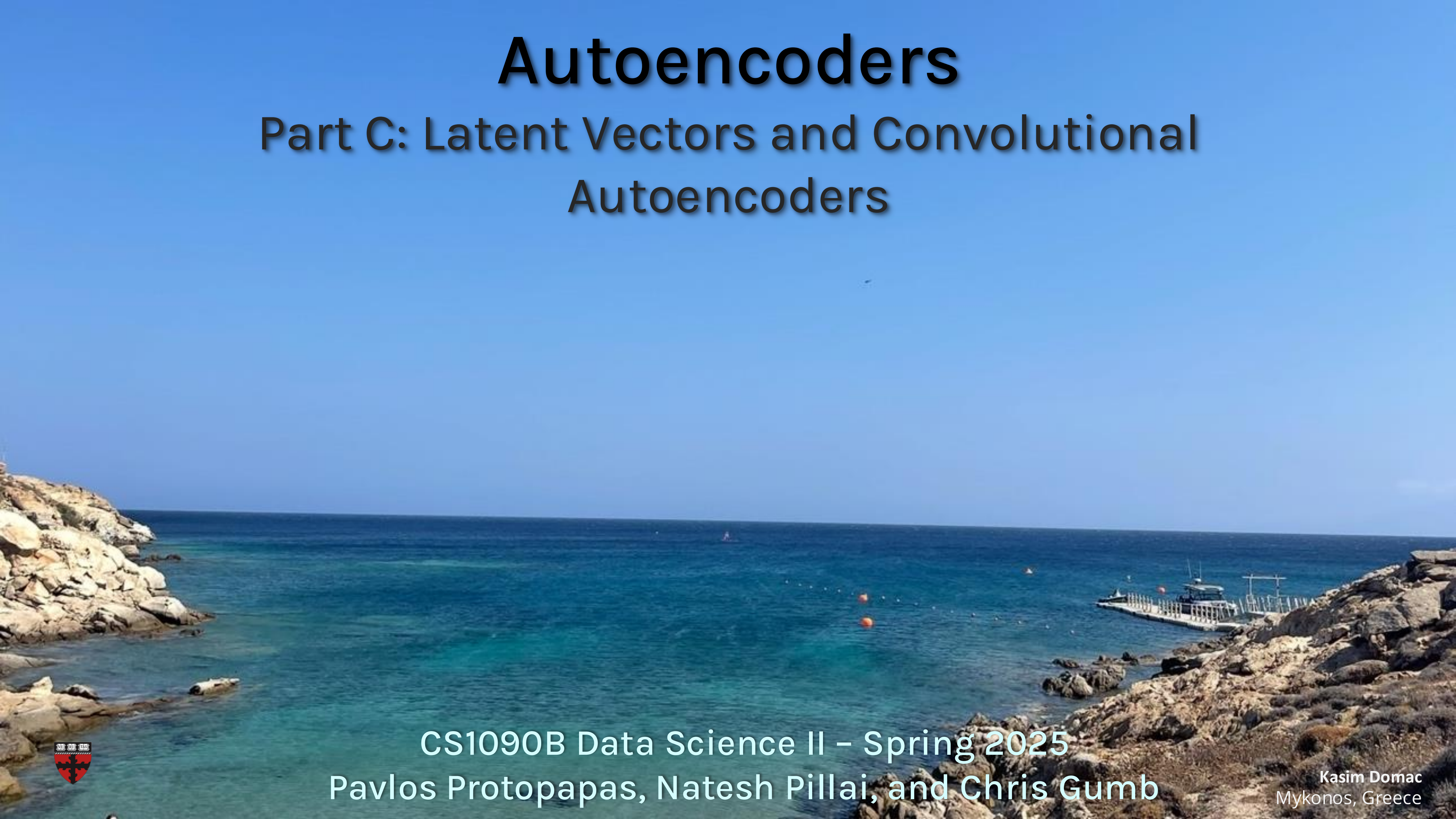


Autoencoders

Part C: Latent Vectors and Convolutional Autoencoders

CS1090B Data Science II – Spring 2025

Pavlos Protopapas, Natesh Pillai, and Chris Gumb

Kasim Domac
Mykonos, Greece

Regularized Autoencoders

- Sparse autoencoders
- Contractive autoencoders
- Denoising autoencoders

Sparse Autoencoders

This trade-off requires the model to maintain only the variations in the data required to reconstruct the input without holding on to redundancies within the input.

Question: How to achieve this?

For most cases, this involves constructing a loss function where one term encourages our model to be sensitive to the inputs (ie. reconstruction loss $\mathcal{L}(x, \hat{x})$ and a second term discourages memorization/overfitting (ie. an added regularizer).

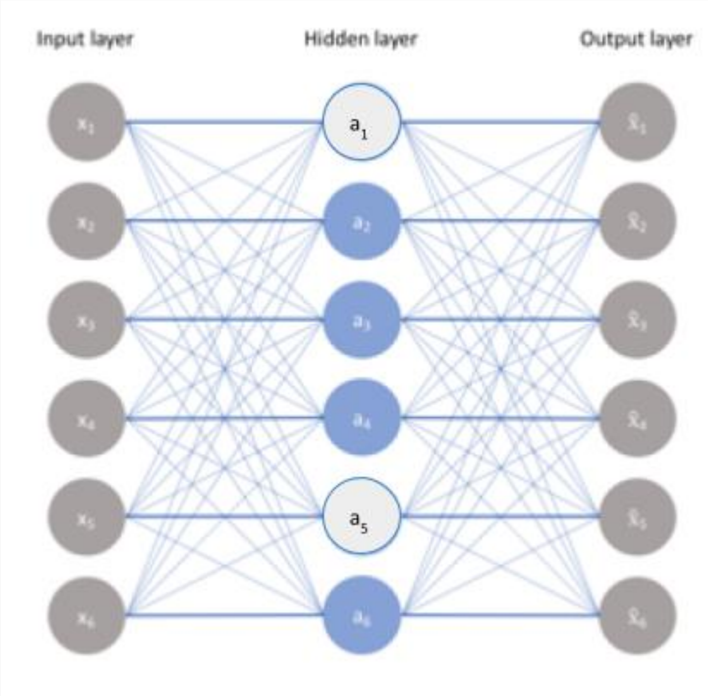
$$\mathcal{L}\left(x, g\left(f(x)\right)\right) + \Omega(z)$$

Regularization on output of encoder (latent space), not on network parameters

Sparse Autoencoders

- We **allow our network to sensitize individual hidden layer nodes toward specific attributes of the input data**.
- A sparse autoencoder is selectively activate regions of the network depending on the input data.
- Limiting the network's capacity to memorize the input data without limiting the networks capability to extract features from the data.

$$\mathcal{L}(x, \hat{x}) + \lambda \sum_i |z_i|$$



https://arxiv.org/pdf/2011.07346.pdf

Contractive Autoencoders

> One would expect that **for very similar inputs, the learned encoding would also be very similar**.

We can explicitly train our model for this to be the case by requiring that the *derivative of the hidden layer activations are small* with respect to the input.

Question: How do we find how much the encoded space would change if the input changes?

Contractive Autoencoders

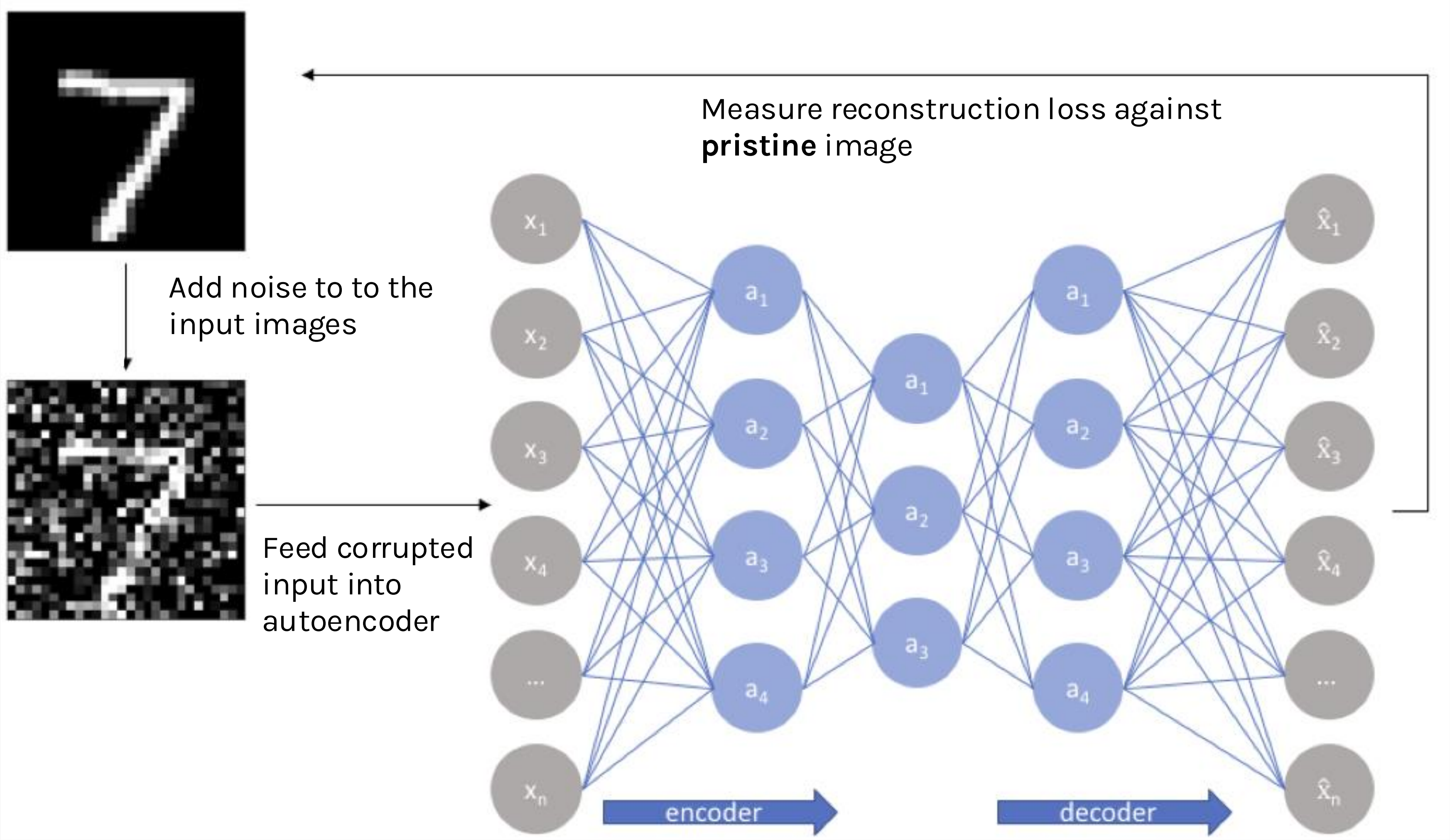
Derivatives

$$\mathcal{L}\left(x, g\left(f(x)\right)\right) + \lambda \sum_i ||\nabla_x z_i||^2$$

This forces the model to learn a function that does not change much when x changes slightly. Because this penalty is applied only at training examples, it forces the autoencoder to learn features that capture information about the training distribution.

Denoising

A popular use of autoencoders is to remove noise from samples.



Infilling

Claim is that AE learns the contextual information of the images. That would mean if some parts of the image is missing then


HAMA
Pavlos
Carabao