Autoencoder

Masoud Pourreza

Pourreza.masoud@gmail.com

2018

مرکز تحقیقات هوش پارت

Step one : Feature!

Cat vs Dog

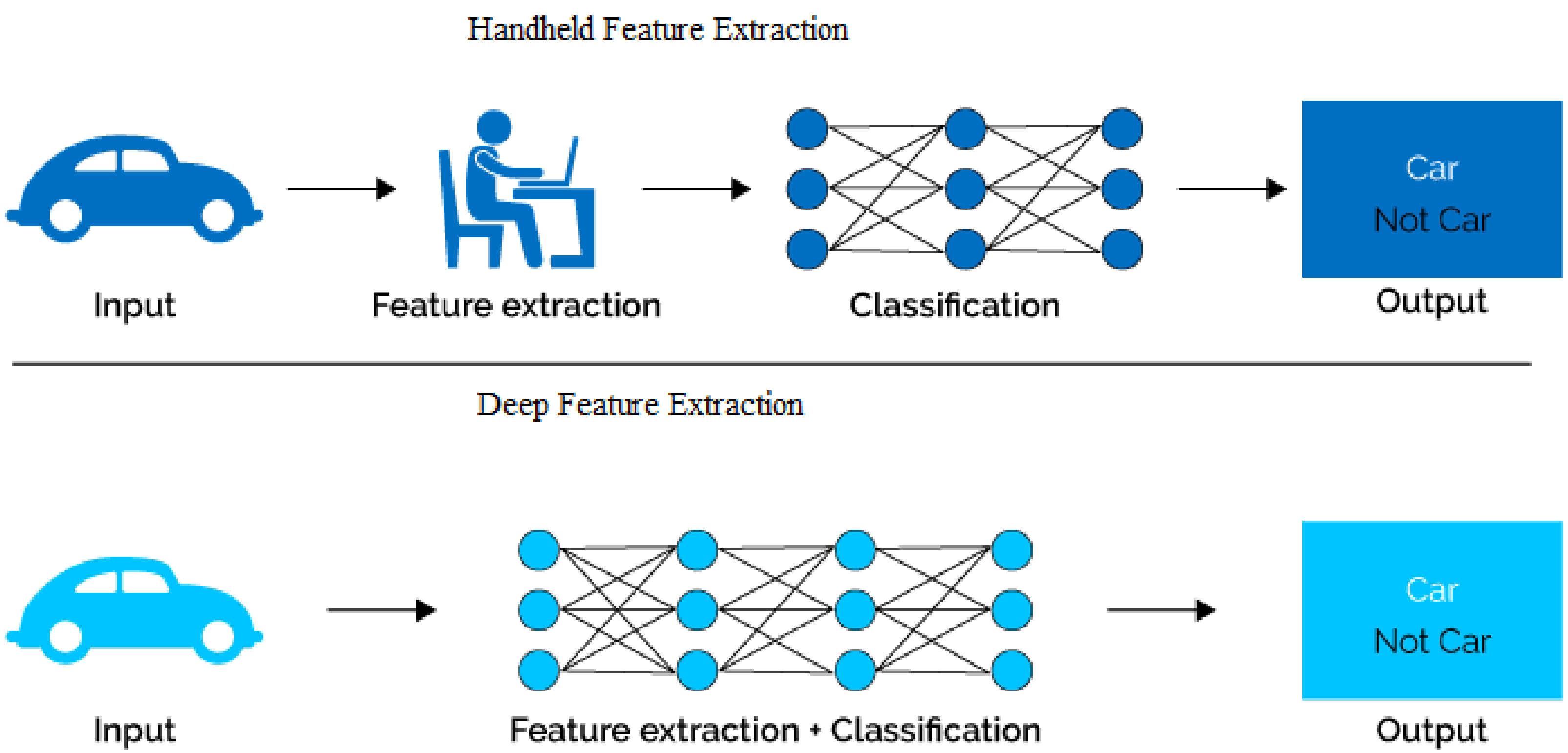
Easy?

cheetah vs leopard

Feature Engineering

- Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.
- Feature engineering turn your inputs into things the algorithm can understand.
- At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.

Feature Engineering (cont'd)



مرکز تحقیقات هوش پارت

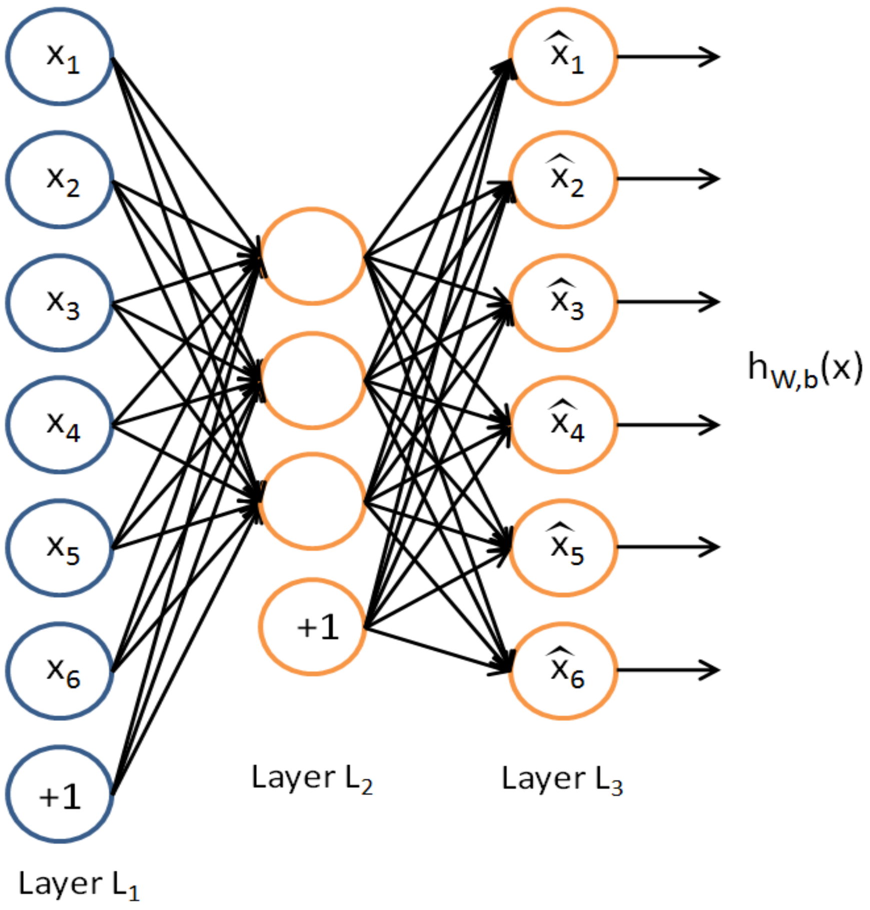
Unsupervised feature learning

- The unsupervised feature learning approach learns higher-level representation of the unlabeled data features by detecting patterns using various algorithms
- It is a self-taught learning framework developed to transfer knowledge from unlabeled data, which is much easier to obtain, to be used as preprocessing step to enhance the supervised inductive models.

Applications of AE

- Dimensionality reduction
- Information Retrieval
- Denoising

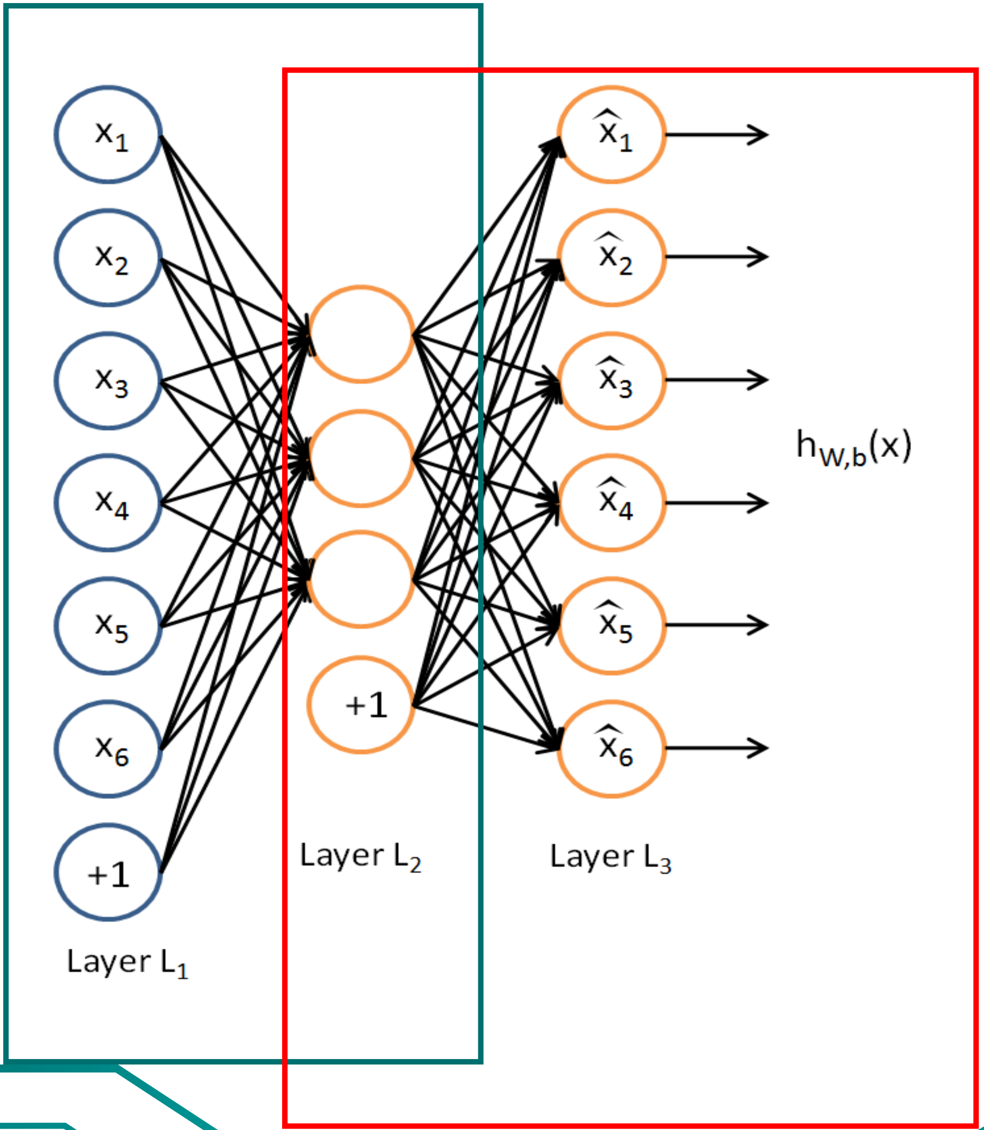
Simple Autoencoder



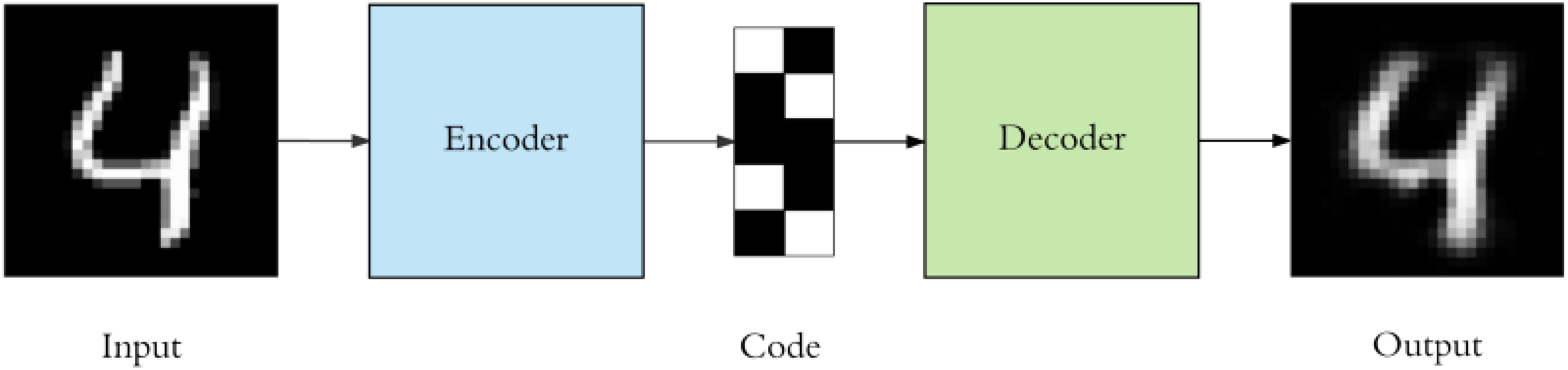
Encode-Decode

Encode

Decode

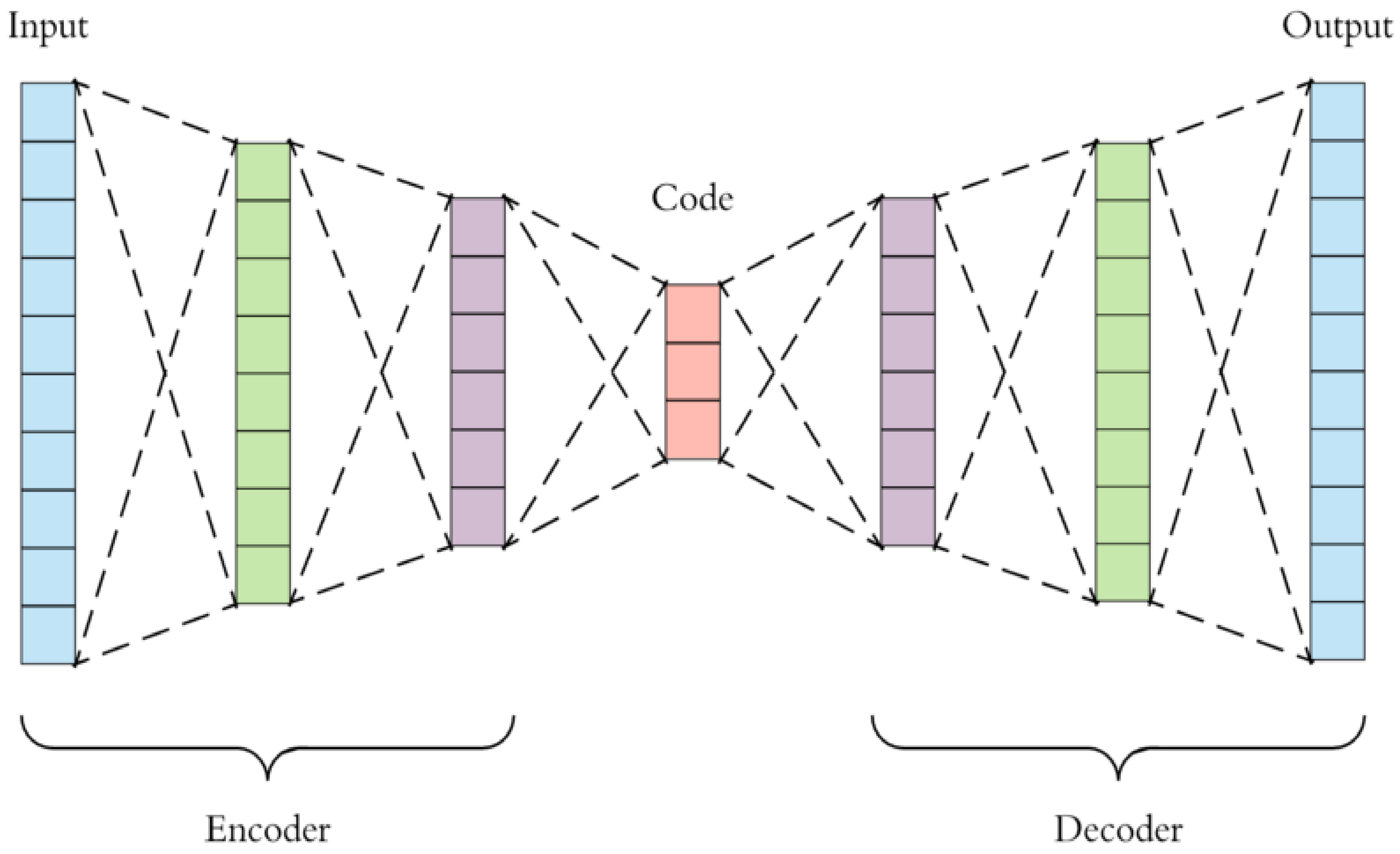


AE_Aim



Keras

Multilayer AE



Keras

Hyper parameters

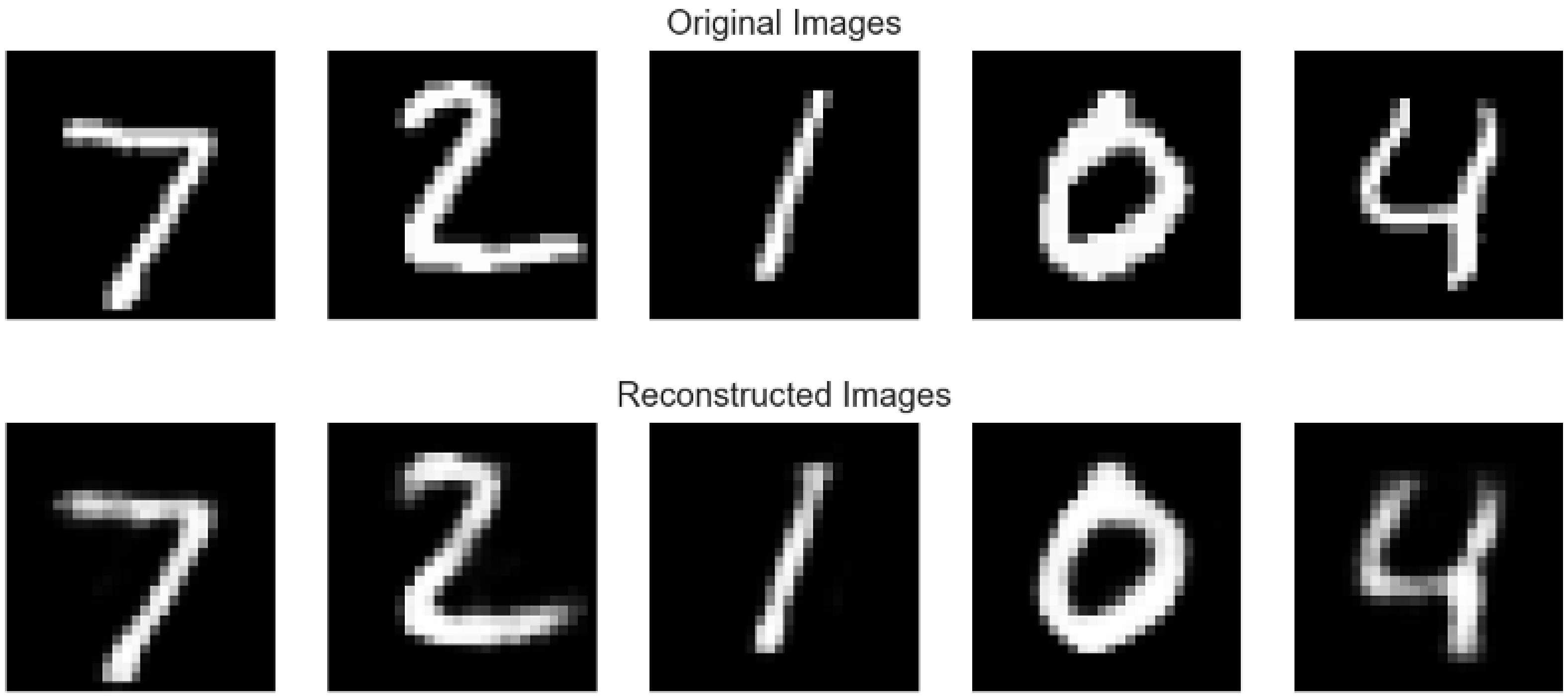
- Code size: number of nodes in the middle layer. Smaller size results in more compression.
- Number of layers: the Autoencoder can be as deep as we like.
- Number of nodes per layer : stacked structure
- Loss function: we either use *mean squared error (mse)*

Loss functions

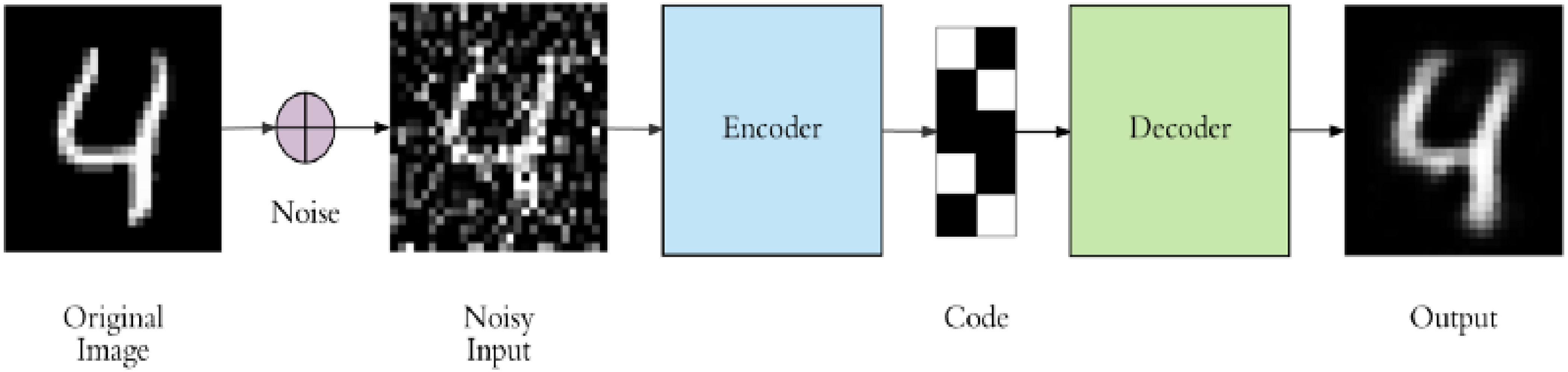
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

Binary Cross entropy= -(y log(p) + (1-y) log(1-p))

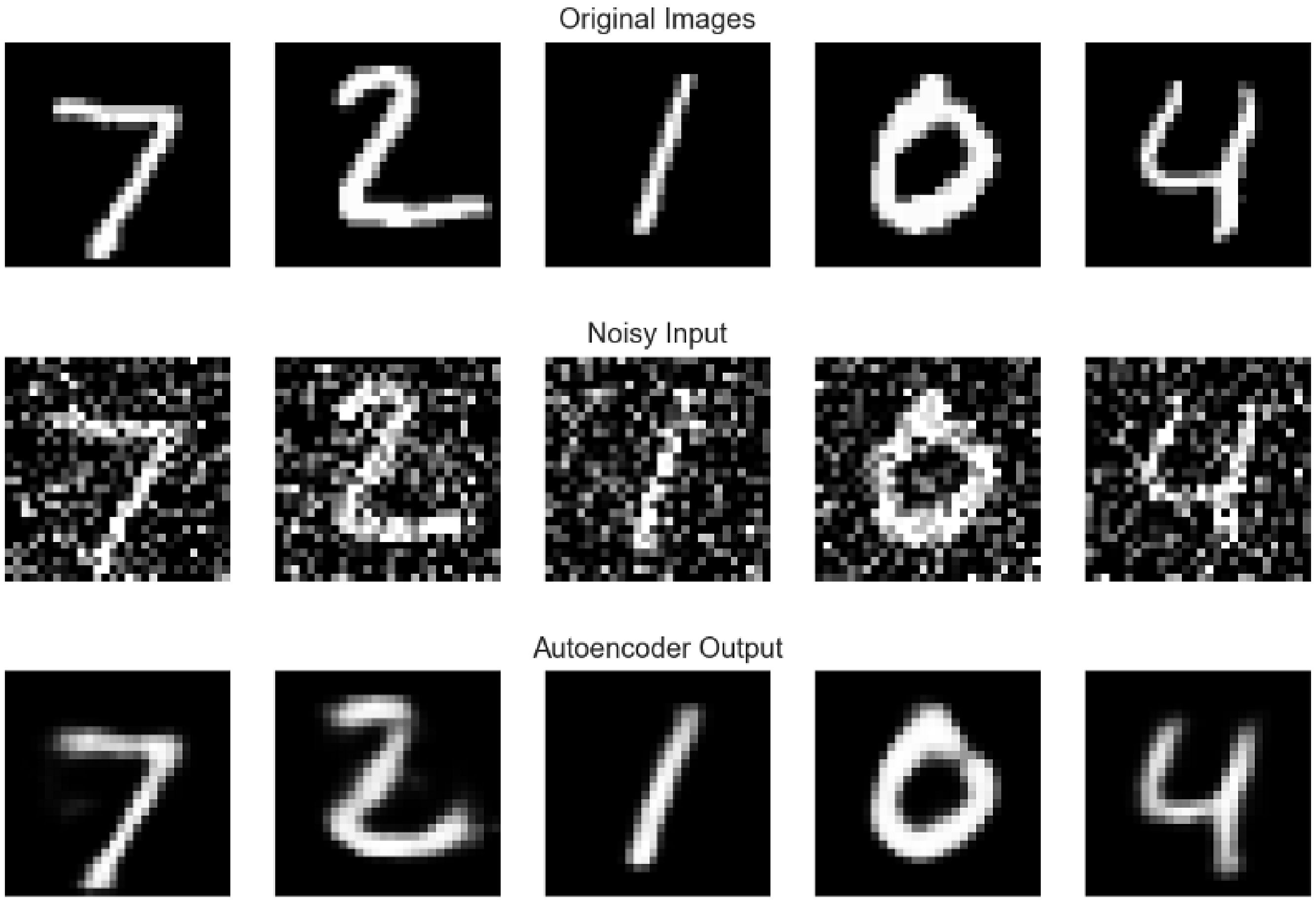
Reconstruction



Denoising Autoencoder



Output example

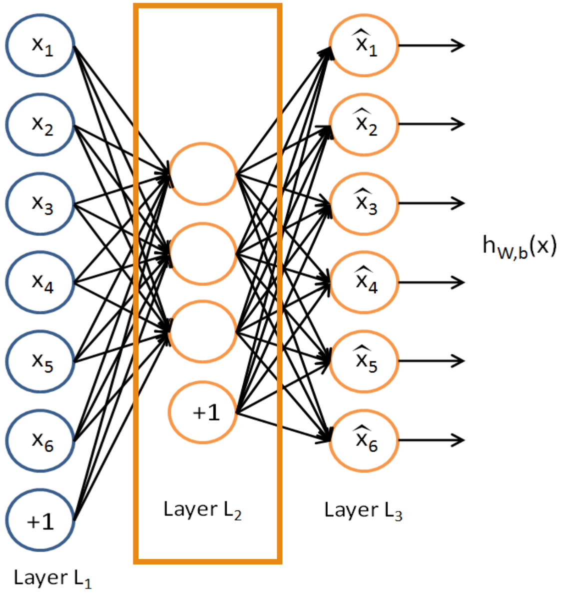


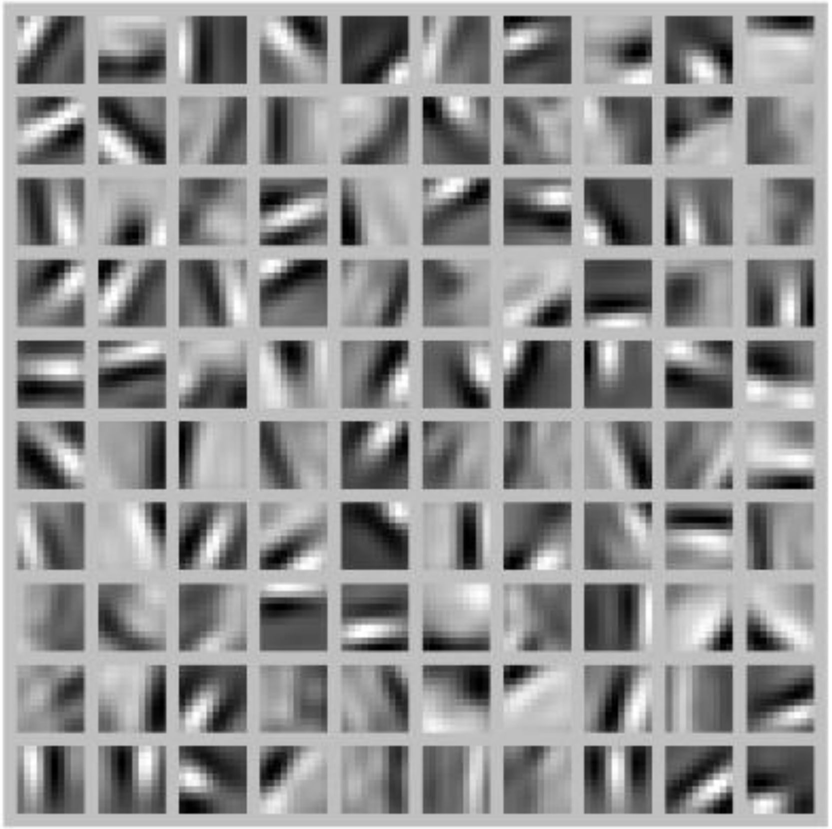
مرکز تحقیقات هوش پارت

Keras

AE representation

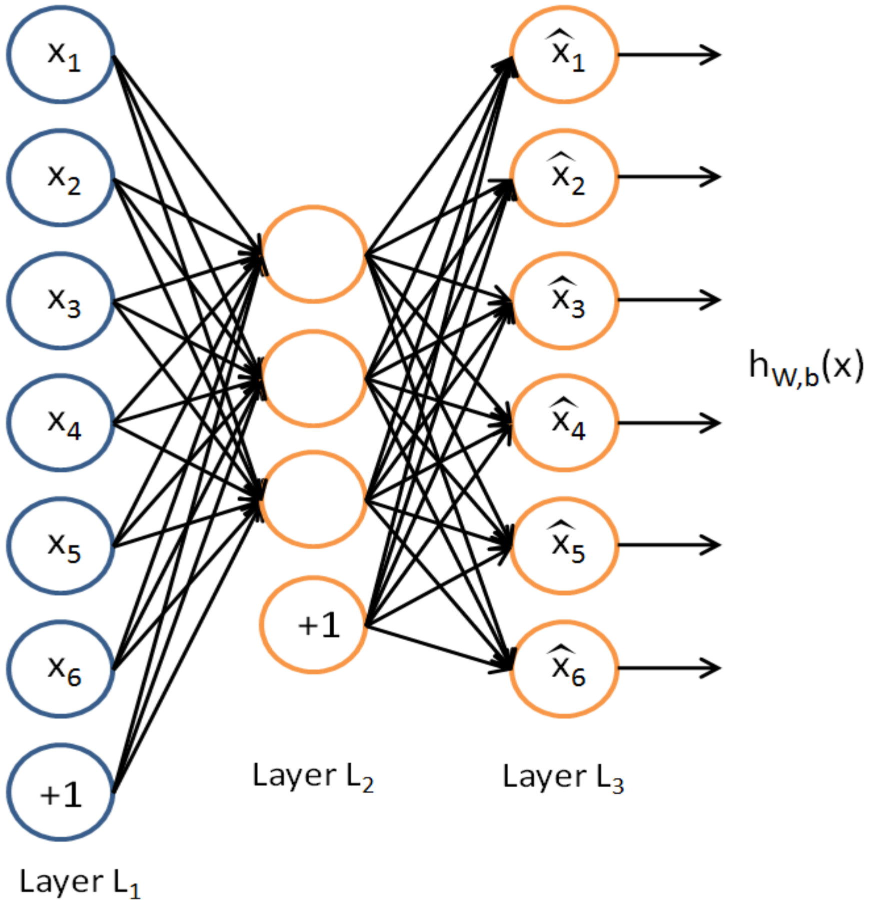
Number of neurons=100





مرکز تحقیقات هوش پارت

Sparse Autoencoder

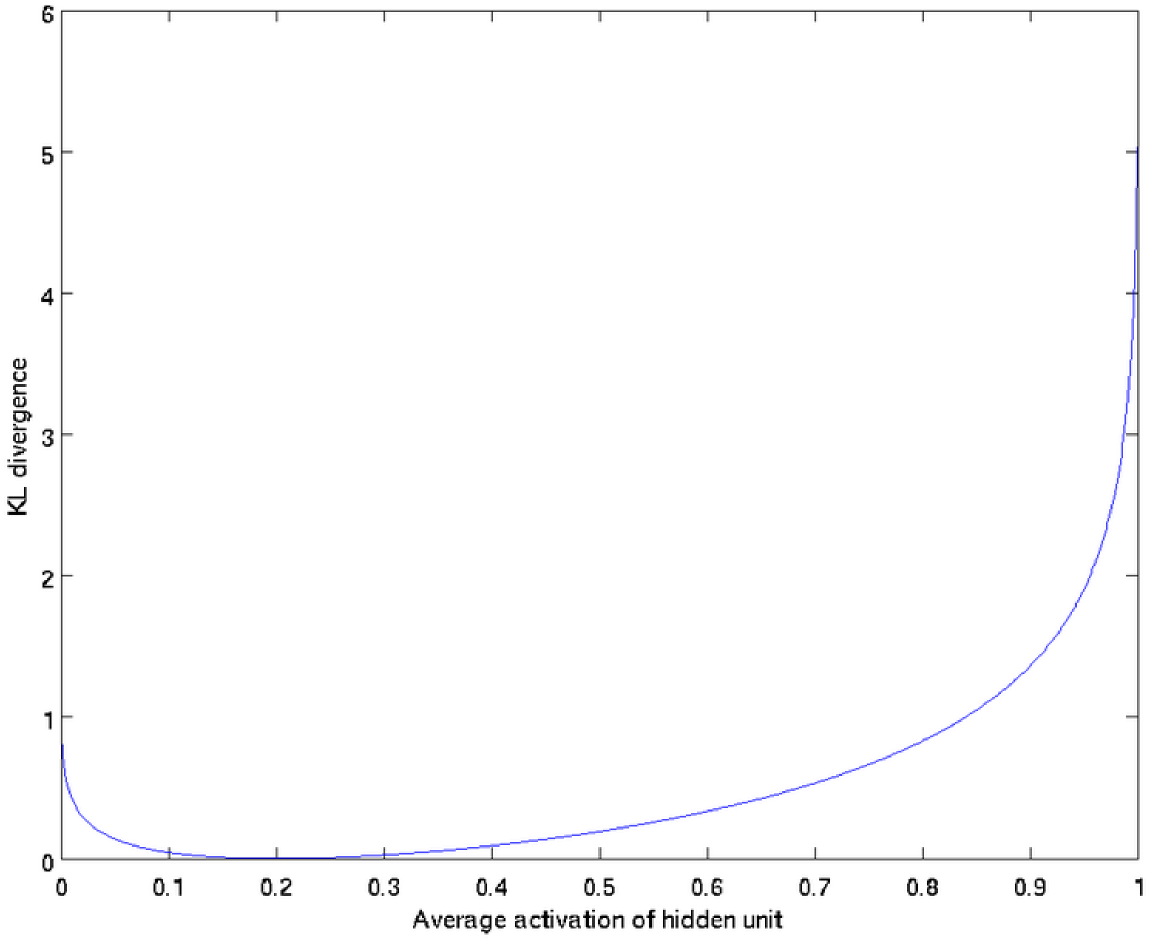


Sparsity Loss Function

$$J_{\text{sparse}}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} \mathrm{KL}(\rho || \hat{\rho}_j),$$

$$\mathrm{KL}(\rho || \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$$

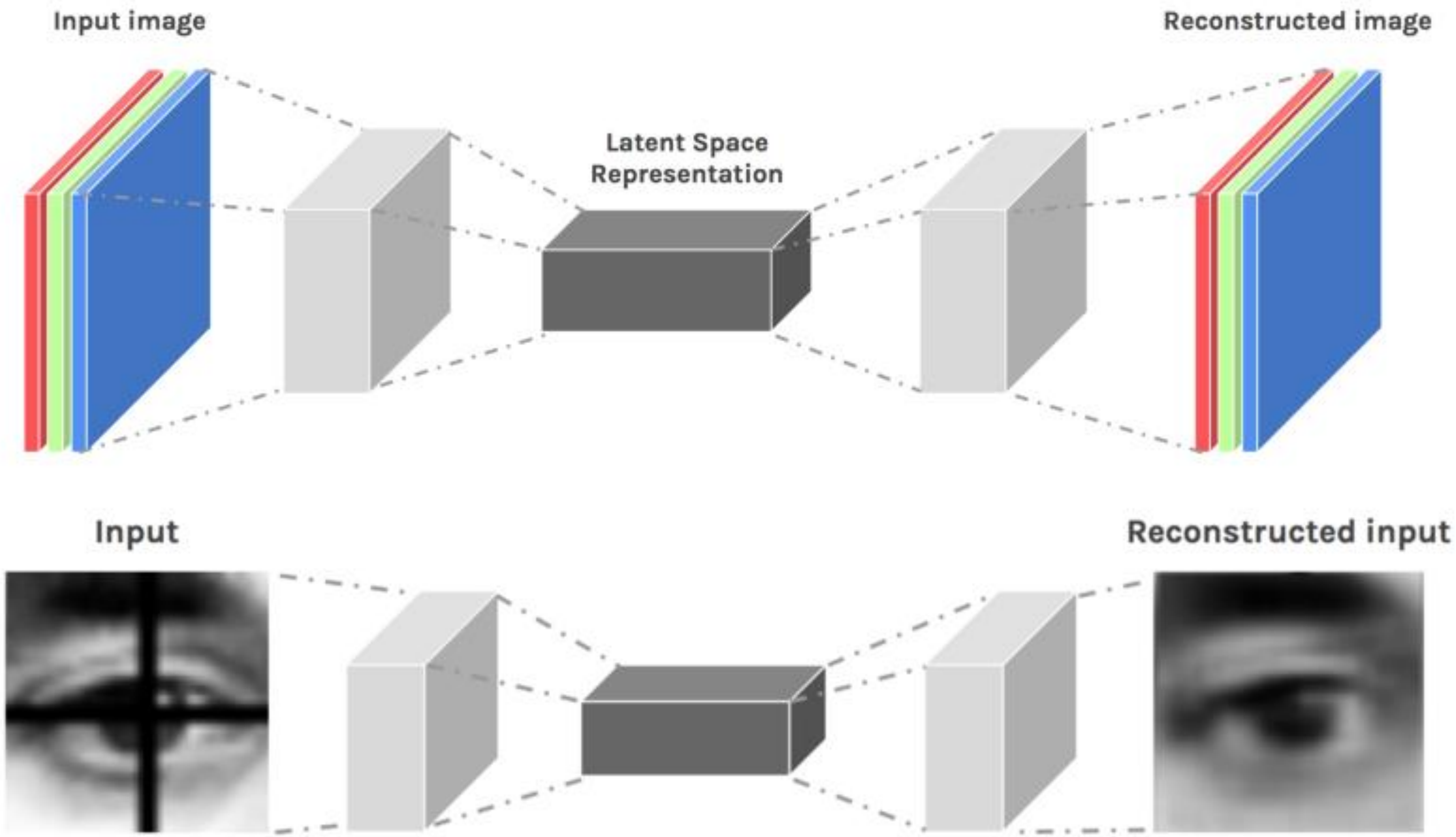
Kullback-Leibler (KL) divergence



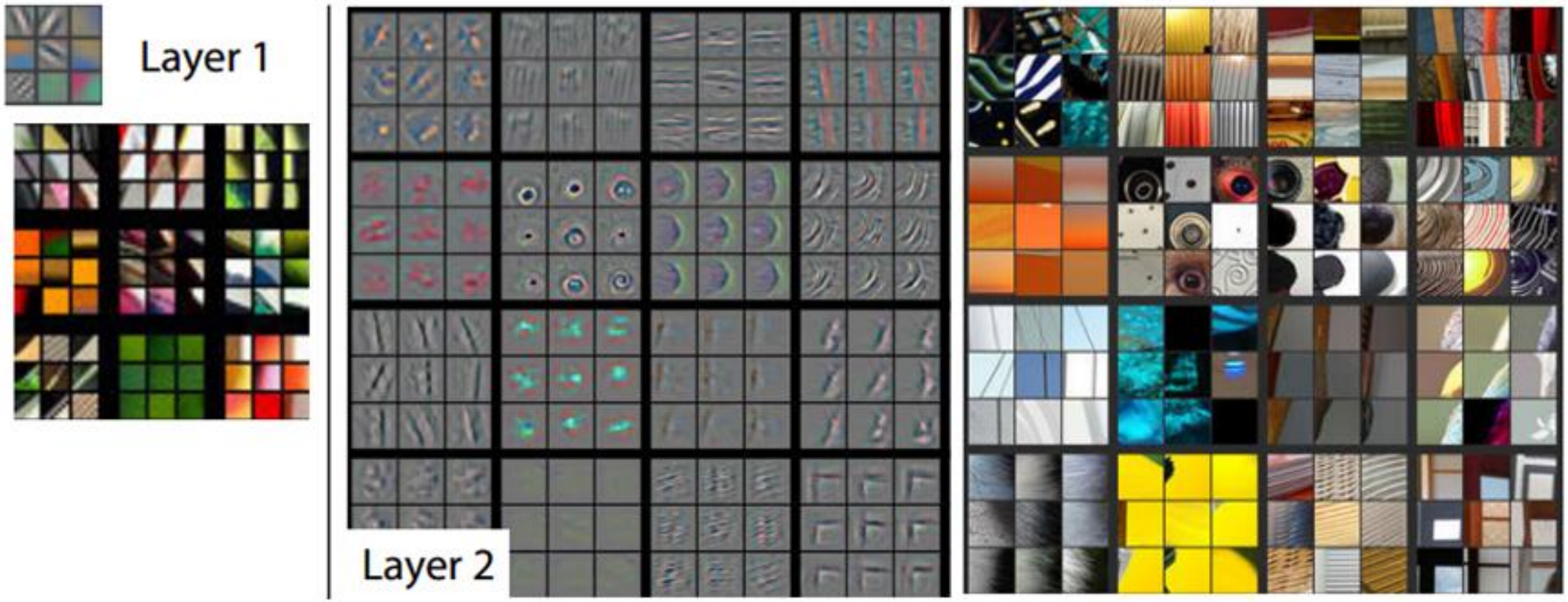
$\rho = 0.2$

Keras

Convolutional AE



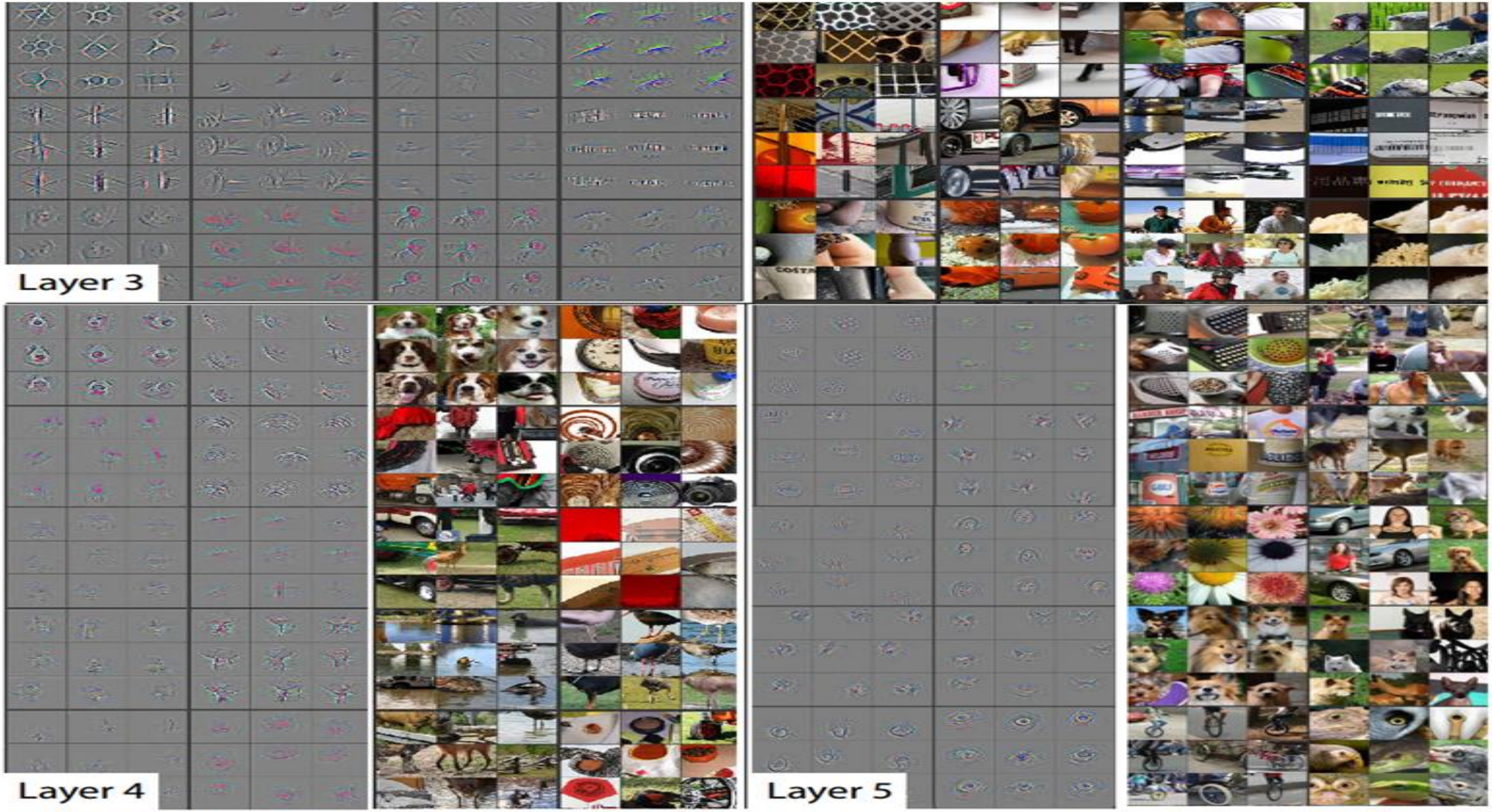
Deconvolution and unpooling



Visualizations of Layer 1 and 2. Each layer illustrates 2 pictures, one which shows the filters themselves and one that shows what part of the image are most strongly activated by the given filter. For example, in the space labled Layer 2, we have representations of the 16 different filters (on the left)

مرکز تحقیقات هوش پارت

Deconvolution and unpooling (cont'd)

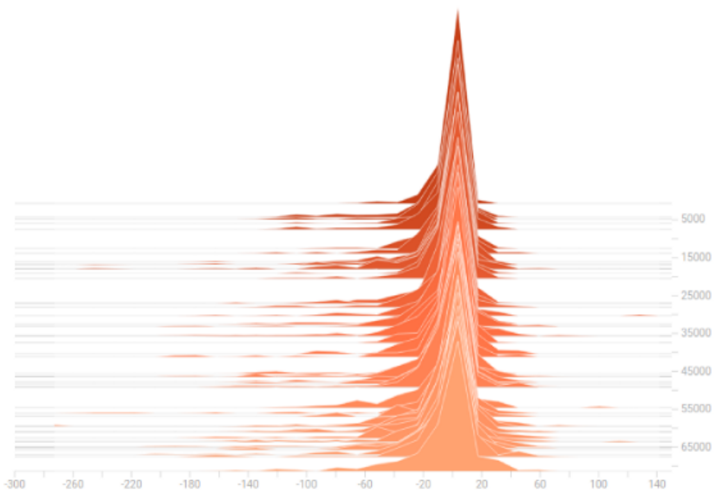


Visualizations of Layers 3, 4, and 5

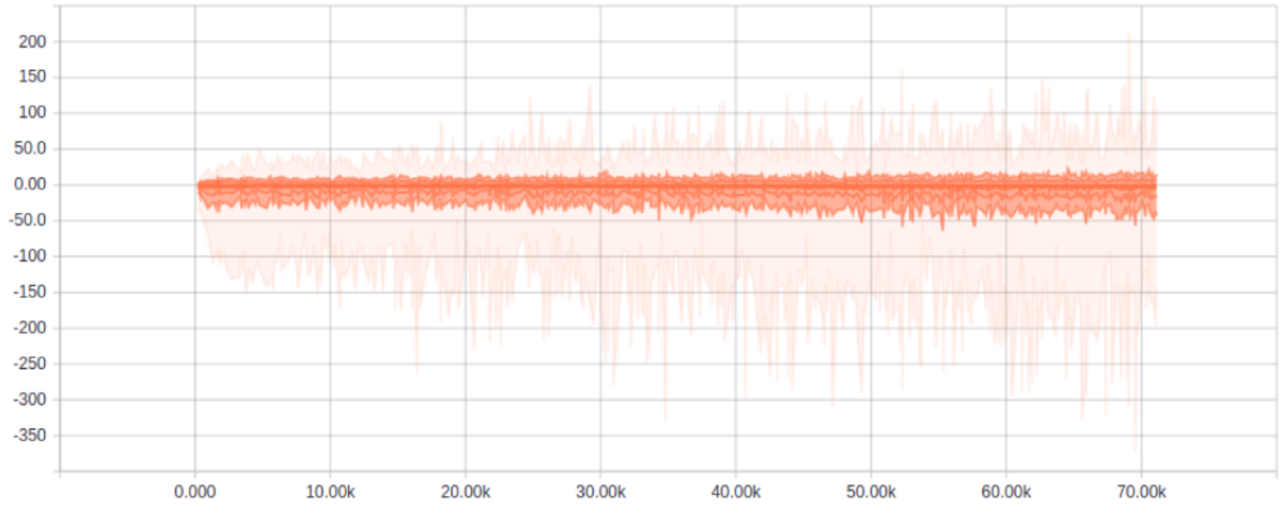
Keras

Adversarial AE

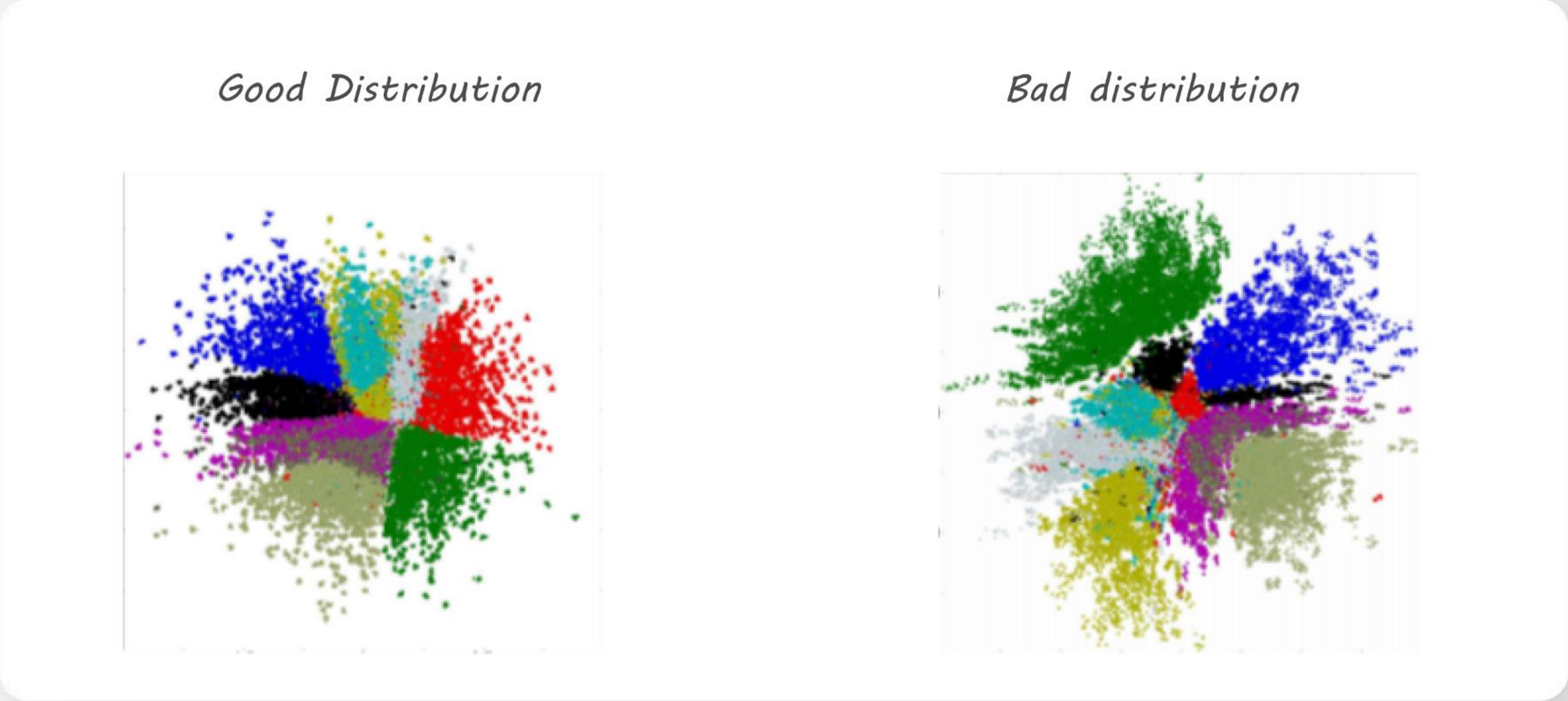
Encoder histogram



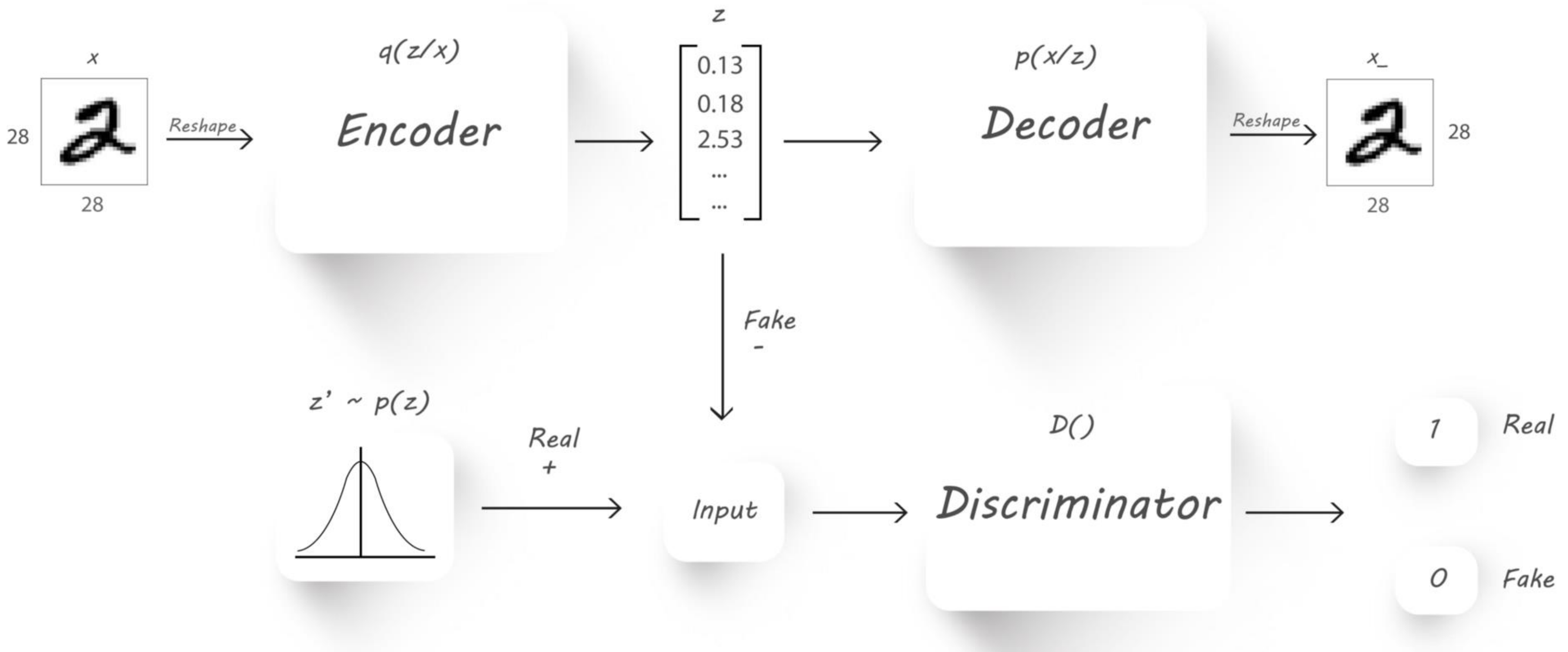
Encoder Distribution



Adversarial AE (cont’d)



Adversarial AE (cont’d)



مرکز تحقیقات هوش پارت

Keras

Other videos

- https://www.aparat.com/partdpai