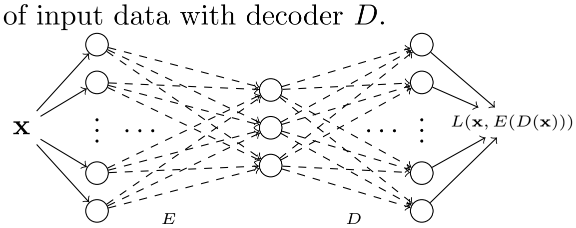
A Survey On Autoencoders

BILICI, M. Şafak
safakk.bilici.2112@gmail.com

Abstract– **Autoencoders are an unsupervised learning architectures in neural networks. Theys are commonly used in Deep Learning tasks; such as generative models, anomaly detection, dimensionality reduction. In this article, we will evaluate theoretical approaches of Autoencoders and see it's extensions.**

1 Introduction

Autoencoders are an unsupervised learning method. They map the input data into lower dimensional space with encoder E, and then maps into same space that have same dimension of input data with decoder D.



The main idea behind Autoencoders is to attempt to copy its input to its output. The input layer is fed with input vector $\mathbf{x}$ and the loss is calculated at output layer between $\mathbf{x}$ and $E(D(\mathbf{x}))$, in other words the loss is $L(\mathbf{x}, E(D(\mathbf{x})))$. It measures difference between our original input and the consequent reconstruction. We named the middle layer, that is connection between encoder E and decoder D, as the "bottleneck". We can denote our output of bottleneck as $\mathbf{h} = E(\mathbf{x})$ and denote our output as $\hat{\mathbf{x}} = D(\mathbf{h}) = D(E(\mathbf{x}))$. We can define our encoder and decoder as conditional probability density function that are $p_{encoder}(\mathbf{h}|\mathbf{x})$ and $p_{decoder}(\hat{\mathbf{x}}|\mathbf{h})$.
The loss function is named reconstruction loss which is $L(\hat{\mathbf{x}}, \mathbf{x})$. We can treat the process as a feedforward networks; the loss can be minimized via mini-batch statistics following gradients computed by backpropagation algorithm,

$$\min_{\theta} L = \nabla_{\theta} L(\mathbf{x}, E(D(\mathbf{x}))) = \nabla_{\theta} L(\mathbf{x}, \hat{\mathbf{x}})$$

The bottleneck is the key of the effectiveness of Autoencoders. We map our input vector to bottleneck: the bottleneck keeps the 'latent informations' of input $\mathbf{x}$. The network represents input but in lower dimensions. In other words, it behaves like a approximative compression algorithm. The encoding parameters are learned in training process. Then we map bottleneck information $\mathbf{h}$ into same dimension as input $\mathbf{x}$. Then, this procedure can be seen as approximative extracting compressed latent information.

2 Undercomplete Autoencoders