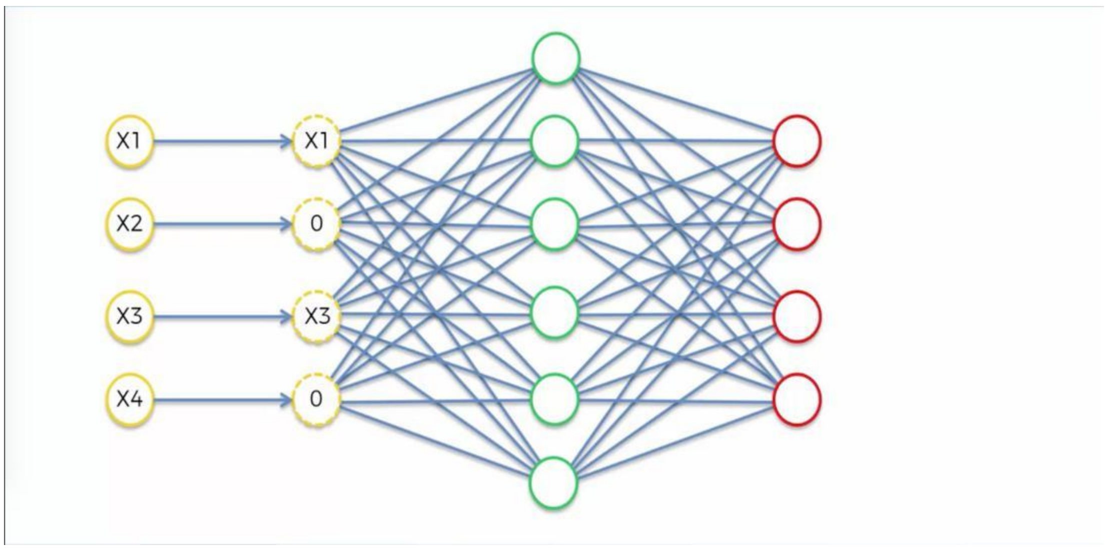
AUTOENCODERS:

Autoencoders are neural networks which are commonly used for feature selection and extraction. However, when there are more nodes in the hidden layer than there are inputs, the network is risking to learn the so called “Identity Function”,also called “Null Function”, meaning that the output equals the input, marking the autoencoder useless.

DENOISING AUTOENCODERS:

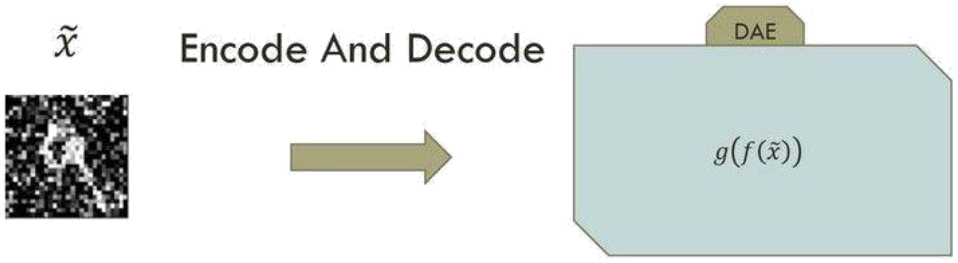
Denoising autoencoders are an extension of the basic autoencoder, and represent a stochastic version of it. Denoising autoencoders attempt to address identity-function risk by randomly corrupting input (i.e. introducing noise) that the autoencoder must then reconstruct, or denoise.



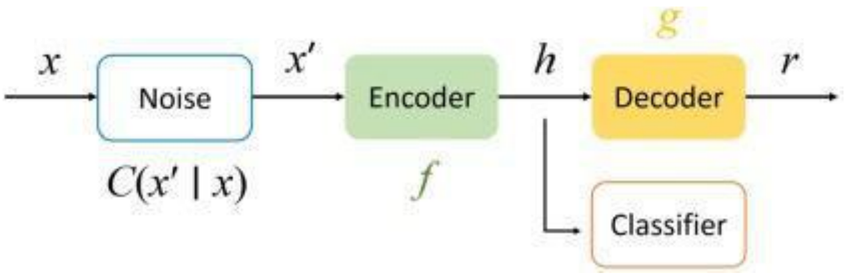
Denoising Autoencoders solve this problem by corrupting the data on purpo se by randomly turning some of the input values to zero. In general, the percentage of input nodes which are being set to zero is about 50%. Other sources suggest a lower count, such as 30%. It depends on the amount of data and input nodes you have.
When calculating the Loss function, it is important to compare the output values with the original input, not with the corrupted input. That way, the risk of learning the identity function instead of extracting features is eliminated.

DENOISING AUTOENCODER PROCESS:

Denoising autoencoder(DAE) attempts to obtain the robust latent representations by introducing a stochastic noise to the original data: $x \sim x'$.
The denoising autoencoder then reconstructs the original data from the corrupted input, which helps to discover the robust representations and prevent it from learning the less important identity.The noise is artificially introduced in DAE.



During the denoising procedure, the model needs to sample an observation *x*from the training set, which then generates a corresponding corrupted *x*0 according to the corruption process P(x'|x). then x′ is encoded so that a hidden representation *h*can be obtained, as seen in Fig.

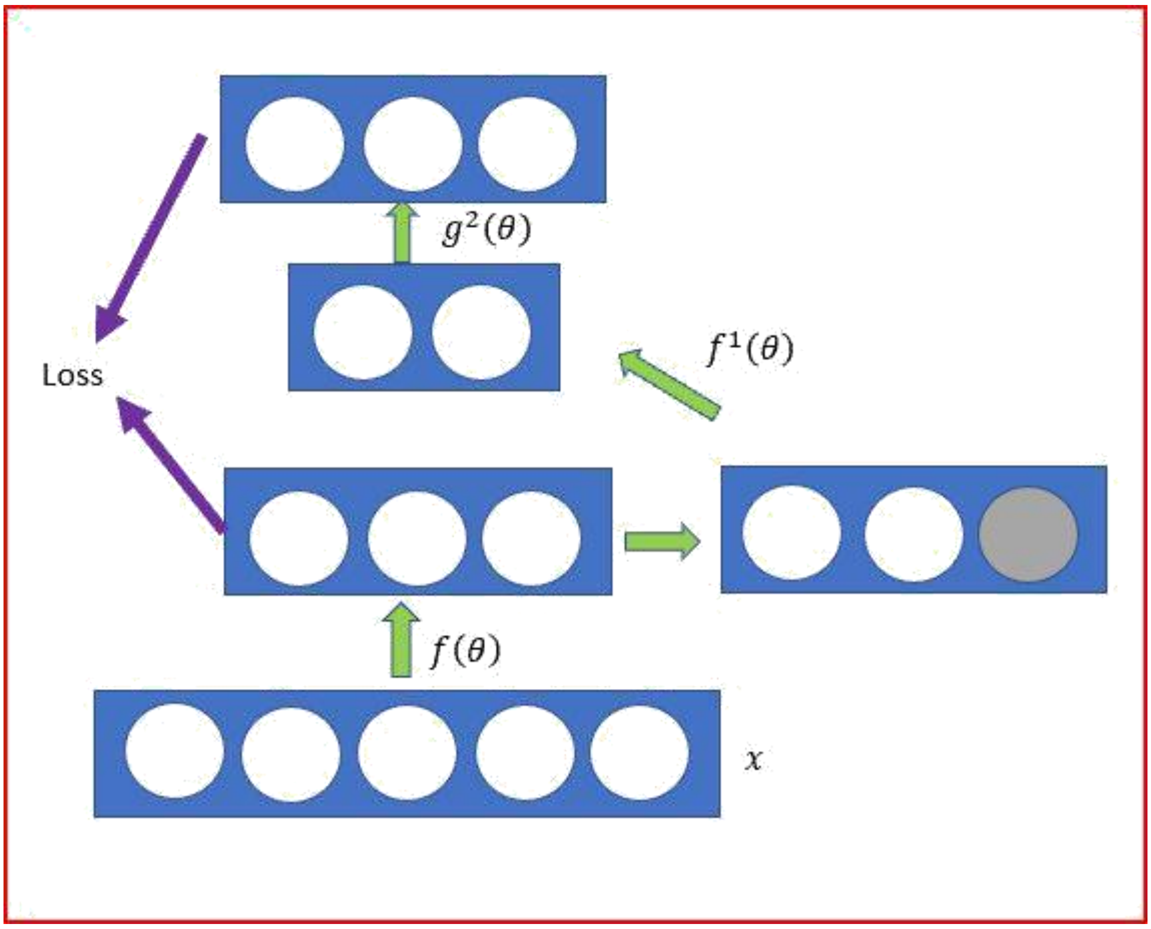


It is noted that in DAE, the loss function should be L(x,r) instead of L(x′,r). Since the core idea is that in order to let the decoder reconstruct the original uncorrupted input data from the corrupted one, the encoder has to generate the robust representations. The stochastic noise can be generated by randomly setting some of the input features to zero , or other more complex corruption process. For many unsupervised learning methods, probabilistic modelingand maximum likelihood estimation (MLE) are employed due to the efficiency and consistency.

.

STACKED DENOISING AUTOENCODERS:

- Stacked Autoencoders is a neural network with multiple layers of sparse autoencoders
- When we add more hidden layers than just one hidden layer to an autoencoder, it helps to reduce a high dimensional data to a smaller code representing important features

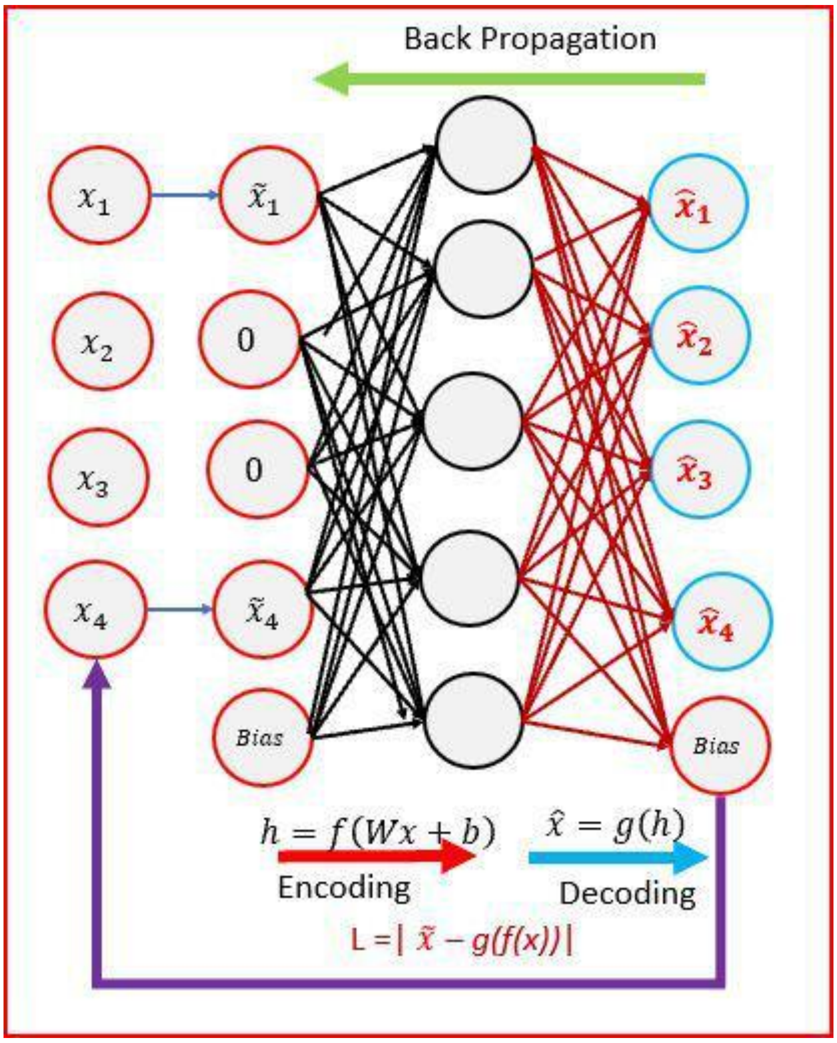


- Each hidden layer is a more compact representation than the last hidden layer
- We can also denoise the input and then pass the data through the stacked autoencoders called as **stacked denoising autoencoders**
- In Stacked Denoising Autoencoders, input corruption is used only for initial denoising. This helps learn important features present in the data. Once the mapping function f(θ) has been learnt. For further layers we use uncorrupted input from the previous layers.
- After training a stack of encoders as explained above, we can use the output of the stacked denoising autoencoders as an input to a stand alone supervised machine learning like support vector machines or multi class logistics regression.

WHICH AUTOENCODER?

There are two types of autoencoders:

1. **Denoising Autoencoders(DAE)**

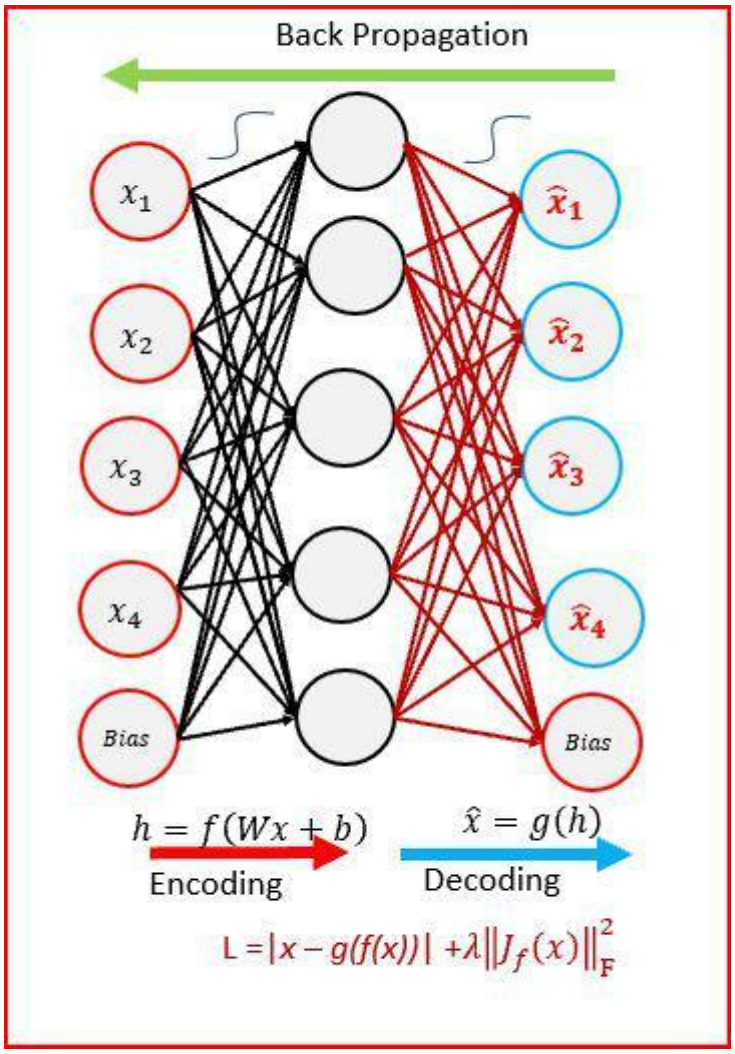


- Denoising refers to intentionally adding noise to the raw input before providing it to the network. Denoising can be achieved using stochastic mapping.
- Denoising autoencoders create a corrupted copy of the input by introducing some noise. This helps to avoid the autoencoders to copy the input to the output without learning features about the data.
- Corruption of the input can be done randomly by making some of the input as zero. Remaining nodes copy the input to the noised input.
- Denoising autoencoders must remove the corruption to generate an output that is similar to the input. Output is compared with input and not with noised input. To minimize the loss function we continue until convergence.
- Denoising autoencoders minimizes the loss function between the output node and the corrupted input.

$$\boxed{L = | \tilde{x} - g(f(x)) |}$$

- Denoising helps the autoencoders to learn the latent representation present in the data. Denoising autoencoders ensures a good representation is one that can be derived robustly from a corrupted input and that will be useful for recovering the corresponding clean input.
- Denoising is a stochastic autoencoder as we use a stochastic corruption process to set some of the inputs to zero.

2. Contractive Autoencoders(CAE)



- Contractive autoencoder(CAE) objective is to have a robust learned representation which is less sensitive to small variation in the data.
- Robustness of the representation for the data is done by applying a penalty term to the loss function. The penalty term is **Frobenius norm of the Jacobianmatrix.** Frobenius norm of

theJacobian matrix forthe hidden layer is calculated with respect to input. Frobenius norm of the Jacobian matrix is the sum of square of all elements.

$$L = |x - g(f(x))| + \lambda \|J_f(x)\|_F^2$$

$$\|J_f(x)\|_F^2 = \sum_{ij} \left(\frac{\partial h_j(x)}{\partial x_i} \right)^2$$

- Contractive autoencoder is another regularization technique like sparse autoencoders and denoising autoencoders.
- CAE surpasses results obtained by regularizing autoencoder using weight decay or by denoising. CAE is a better choice than denoising autoencoder to learn useful feature extraction.
- Penalty term generates mapping which are strongly contracting the data and hence the name contractive autoencoder.

Lecture-11

Name: Sapna

Roll No: 15MI448