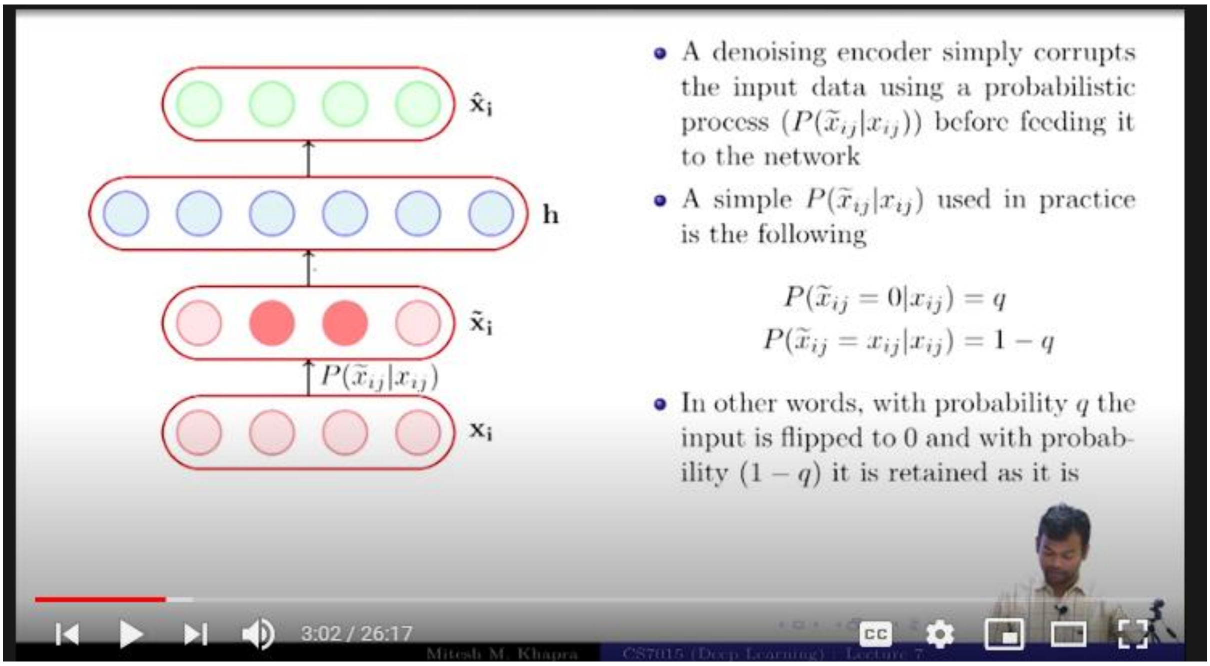
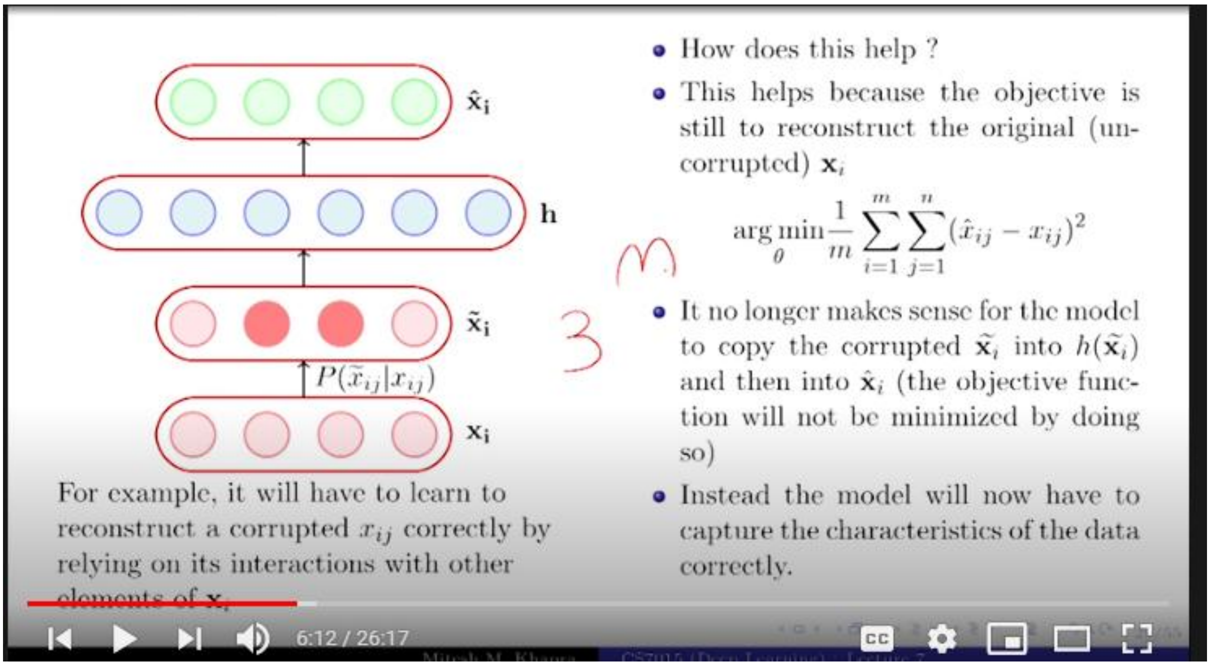
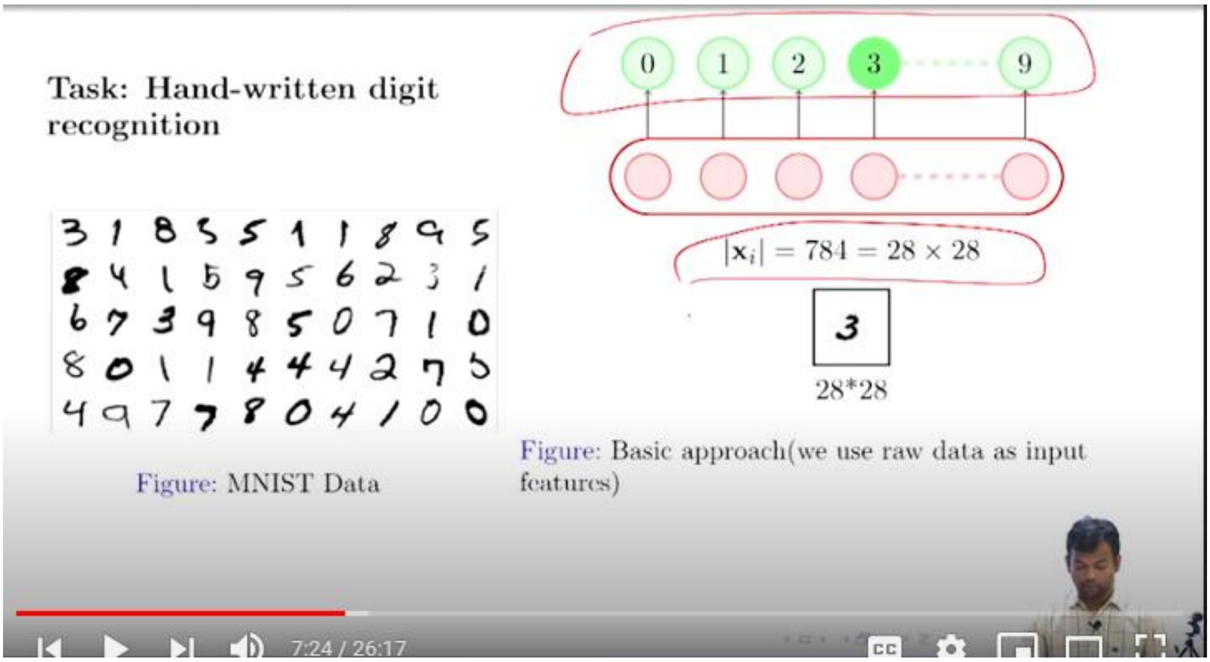
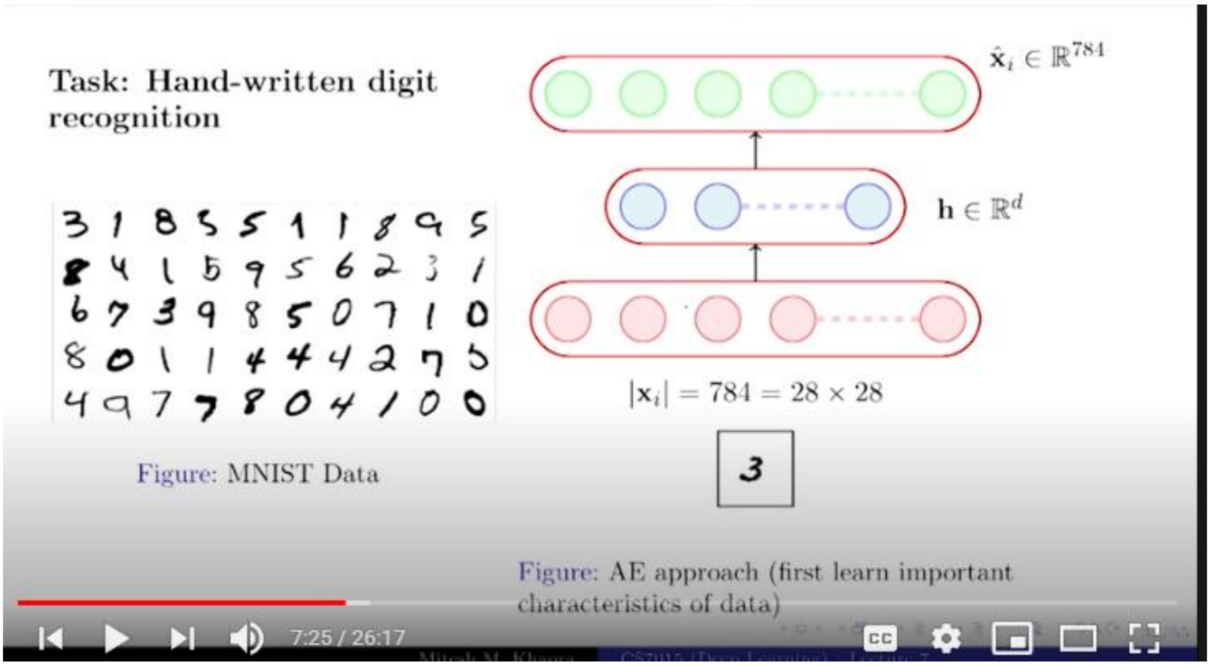
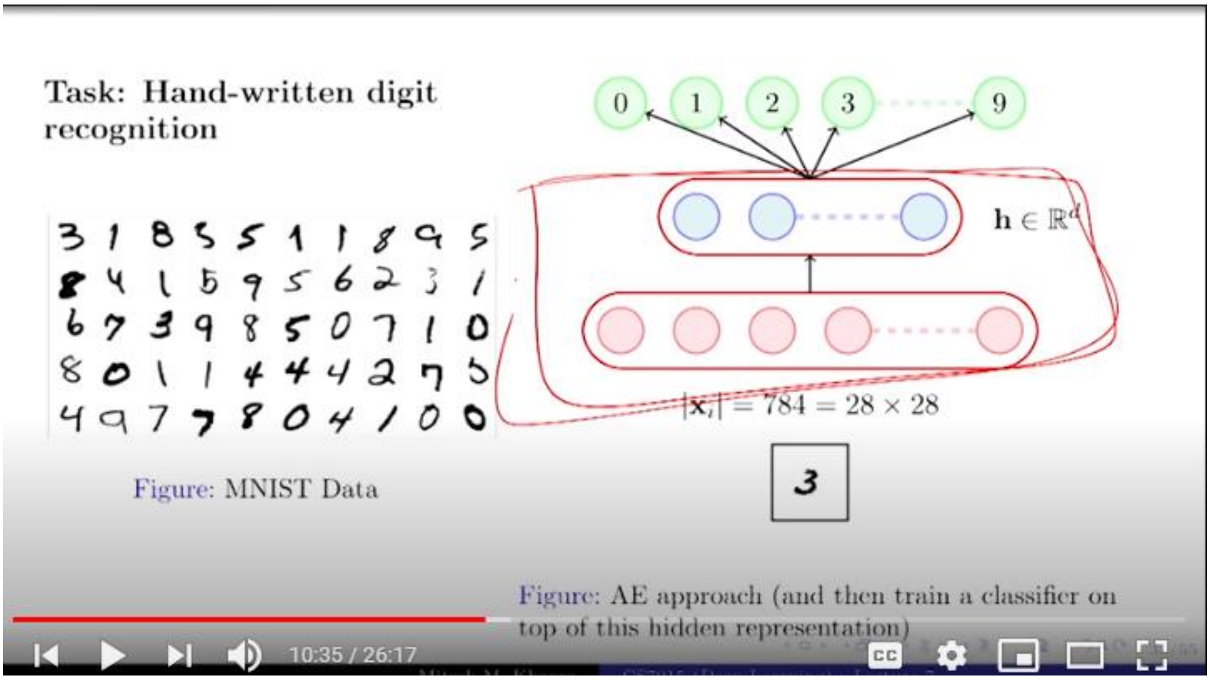




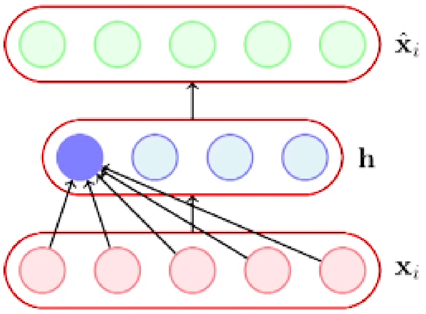
We will now see a practical application in which AEs are used and then compare Denoising Autoencoders with regular autoencoders


Task: Hand-written digit recognition
0
1
2
3
9
$|\mathbf{x}_i| = 784 = 28 \times 28$
3
28*28
Figure: MNIST Data
Figure: Basic approach(we use raw data as input features)
7:24 / 26:17


Task: Hand-written digit recognition
$\hat{\mathbf{x}}_i \in \mathbb{R}^{784}$
$\mathbf{h} \in \mathbb{R}^d$
$|\mathbf{x}_i| = 784 = 28 \times 28$
3
Figure: MNIST Data
Figure: AE approach (first learn important characteristics of data)
7:25 / 26:17


Task: Hand-written digit recognition
0 1 2 3 9
h ∈ ℝ^d
|x_i| = 784 = 28 × 28
3
Figure: MNIST Data
Figure: AE approach (and then train a classifier on top of this hidden representation)
10:35 / 26:17


We will now see a way of visualizing AEs and use this visualization to compare different AEs
12:15 / 26:17
Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 7



$$\max_{\mathbf{x}_i} \ \{w_1^T \mathbf{x}_i\}$$

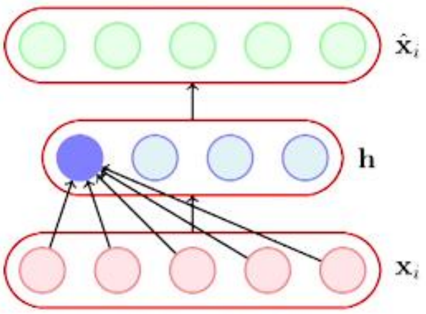
$$s.t. \ \ ||\mathbf{x}_i||^2 = \mathbf{x}_i^T \mathbf{x}_i = 1$$

- We can think of each neuron as a filter which will fire (or get maximally) activated for a certain input configuration $\mathbf{x}_i$
- For example,

$$\mathbf{h}_1 = \sigma(W_1^T \mathbf{x}_i) \ [ignoring \ bias \ b]$$

Where W_1 is the trained vector of weights connecting the input to the first hidden neuron

- What values of $\mathbf{x}_i$ will cause $\mathbf{h}_1$ to be maximum (or maximally activated)
- Suppose we assume that our inputs are normalized so that $\|\mathbf{x}_i\| = 1$



$$\max_{\mathbf{x}_i} \ \{w_1^T \mathbf{x}_i\}$$

$$s.t. \ \ ||\mathbf{x}_i||^2 = \mathbf{x}_i^T \mathbf{x}_i = 1$$

$$\text{Solution:} \ \ \mathbf{x}_i = \frac{w_1}{\sqrt{w_1^T w_1}}$$

- Thus the inputs

$$\mathbf{x}_i = \frac{W_1}{\sqrt{W_1^T W_1}}, \frac{W_2}{\sqrt{W_2^T W_2}}, \ldots \frac{W_n}{\sqrt{W_n^T W_n}}$$

will respectively cause hidden neurons 1 to n to maximally fire

- Let us plot these images ($\mathbf{x}_i$'s) which maximally activate the first k neurons of the hidden representations learned by a vanilla autoencoder and different denoising autoencoders
- These $\mathbf{x}_i$'s are computed by the above formula using the weights $(W_1, W_2 \ldots W_k)$ learned by the respective autoencoders

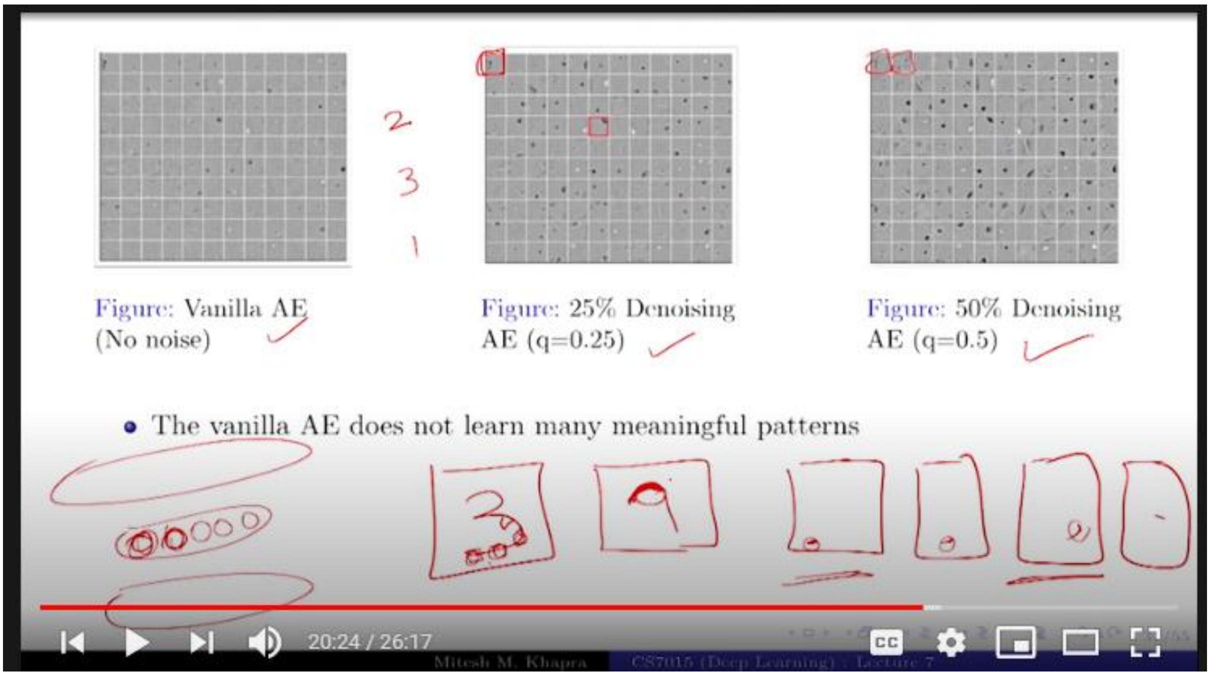
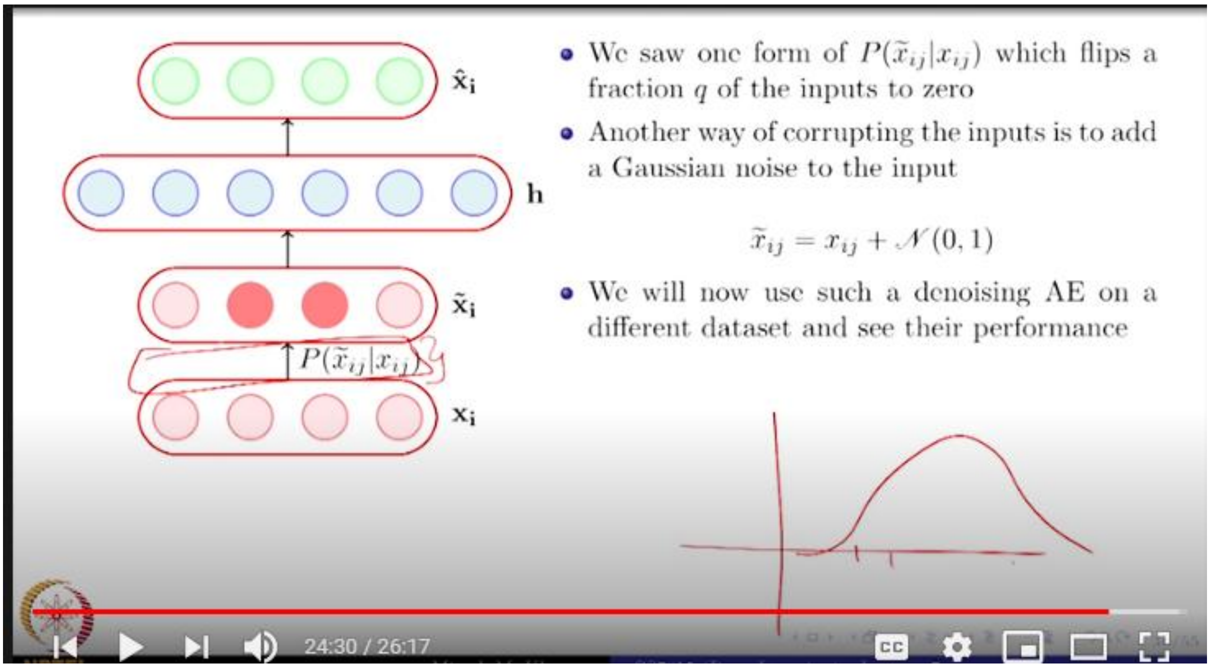


Figure: Vanilla AE (No noise)
Figure: 25% Denoising AE (q=0.25)
Figure: 50% Denoising AE (q=0.5)
2
3
1
• The vanilla AE does not learn many meaningful patterns
20:24 / 26:17
Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 7


$\hat{\mathbf{x}}_\mathbf{i}$
$\mathbf{h}$
$\tilde{\mathbf{x}}_\mathbf{i}$
$P(\tilde{x}_{ij}|x_{ij})$
$\mathbf{x}_\mathbf{i}$
• We saw one form of $P(\tilde{x}_{ij}|x_{ij})$ which flips a fraction q of the inputs to zero
• Another way of corrupting the inputs is to add a Gaussian noise to the input
$\tilde{x}_{ij} = x_{ij} + \mathcal{N}(0, 1)$
• We will now use such a denoising AE on a different dataset and see their performance
24:30 / 26:17