NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas
Department : E & ECE, IIT Kharagpur

Topic

Lecture 26: Back propagation Learning – Examples II

CONCEPTS COVERED

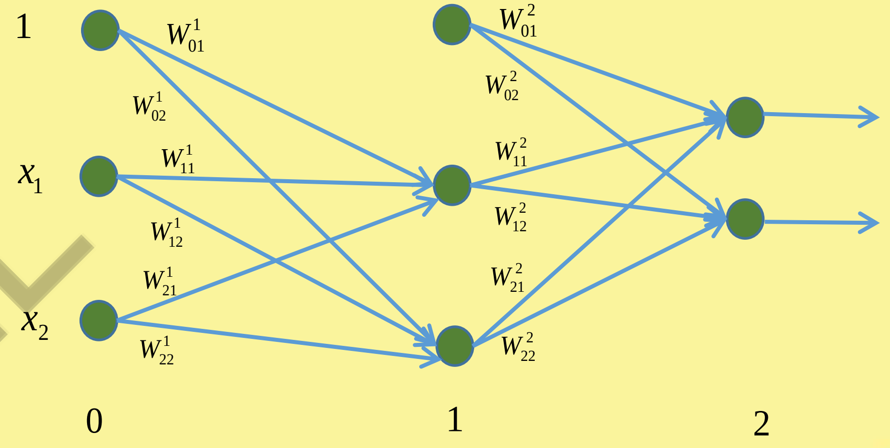
Concepts Covered:

- Back Propagation Learning in MLP
- Different Loss Functions
- Back Propagation Learning - Example
- Back Propagation – Node Level

Back Propagation Learning an Example

Back Propagation Learning:- Output Layer

$$E = \frac{1}{2}\sum_{j=1}^{2}\left(x_j^2 - t_j\right)^2 \qquad x_j^2 = \frac{1}{1+e^{-\theta_j^2}} \qquad \theta_j^2 = \sum_{i=0}^{2} W_{ij}^2 x_i^1$$



$$\frac{\partial E}{\partial W_{ij}^2} = \frac{\partial E}{\partial x_j^2}\cdot\frac{\partial x_j^2}{\partial \theta_j^2}\cdot\frac{\partial \theta_j^2}{\partial W_{ij}^2} = (x_j^2 - t_j)x_j^2(1-x_j^2)x_i^1$$

We set $\boxed{\delta_j^2 = x_j^2(1-x_j^2)(x_j^2 - t_j)} \Rightarrow \frac{\partial E}{\partial W_{ij}^2} = \delta_j^2 x_i^1$

$$\boxed{W_{ij}^2 \leftarrow W_{ij}^2 - \eta\frac{\partial E}{\partial W_{ij}^2}}$$

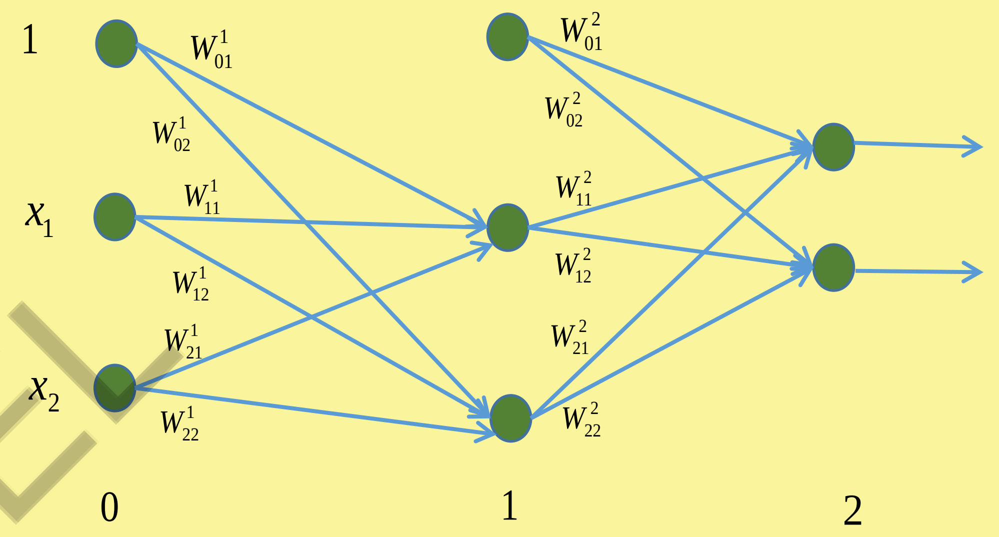
Feed Forward Pass

$$W^1 \qquad x_i^0 \qquad \theta_j^1 = \sum W_{ij}^1 x_i^0 \qquad x_j^1 = \frac{1}{1+e^{-\theta_j^1}}$$

$$\begin{bmatrix} 0.5 & 1.5 & 0.8 \\ 0.8 & 0.2 & -1.6 \end{bmatrix} \begin{bmatrix} 1 \\ 0.7 \\ 1.2 \end{bmatrix} = \begin{bmatrix} 2.51 \\ -9.8 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.92 \\ 0.27 \end{bmatrix}$$

$$W^2 \qquad x_i^1 \qquad \theta_j^2 = \sum_i W_{ij}^2 x_i^1 \qquad x_j^2 = \frac{1}{1+e^{-\theta_j^2}}$$

$$\begin{bmatrix} 0.9 & -1.7 & 1.6 \\ 1.2 & 2.1 & -1.-0.26 \end{bmatrix} \begin{bmatrix} 1 \\ 0.92 \\ 0.27 \end{bmatrix} = \begin{bmatrix} -0.232 \\ 3.057 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.44 \\ 0.95 \end{bmatrix}$$



Back Propagation Learning:- Output Layer

$$\boxed{\delta_j^2 = x_j^2(1-x_j^2)(x_j^2 - t_j)}$$

$$\delta_1^2 = x_1^2(1-x_1^2)(x_1^2 - t_1)$$
$$= 0.44*(1-0.44)*(0.44-1)$$
$$= -0.138$$

$$\Rightarrow \frac{\partial E}{\partial W_{11}^2} = \delta_1^2 x_1^1 = -0.126$$

$$\delta_2^2 = x_2^2(1-x_2^2)(x_2^2 - t_2)$$
$$= 0.95*(1-0.95)*(0.95-0)$$
$$= 0.045$$

$$\Rightarrow \frac{\partial E}{\partial W_{12}^2} = \delta_2^2 x_1^1 = 0.04$$

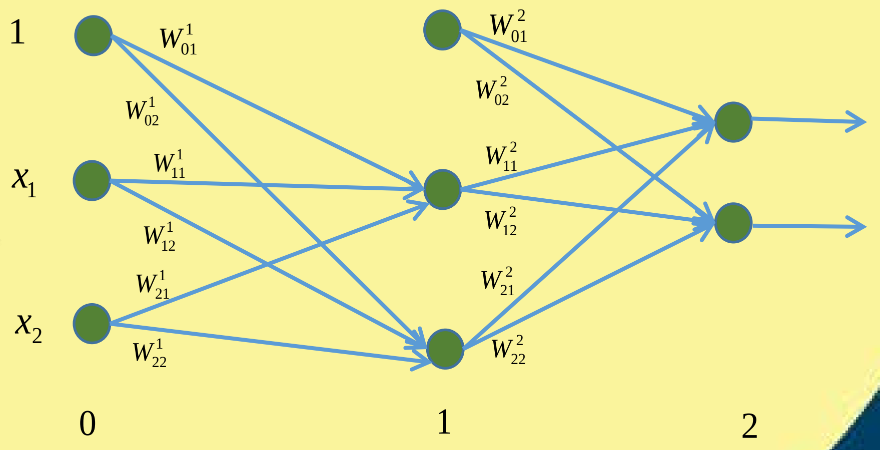
$$\boxed{W_{11}^2 \leftarrow W_{11}^2 + \eta * 0.126}$$

$$\boxed{W_{12}^2 \leftarrow W_{12}^2 - \eta * 0.04}$$

Back Propagation Learning:- Output Layer

$$\frac{\partial E}{\partial W_{21}^{2}} = \delta_1^2 x_2^1 = -0.037 \qquad \frac{\partial E}{\partial W_{22}^{2}} = \delta_2^2 x_2^1 = 0.012$$

$$\frac{\partial E}{\partial W_{01}^{2}} = \delta_1^2 x_0^1 = -1.38 \qquad \frac{\partial E}{\partial W_{02}^{2}} = \delta_2^2 x_0^1 = 0.045$$



Back Propagation Learning:- Hidden Layer

We set $\boxed{\delta_i^k = O_i^k(1-O_i^k)\sum_{j=1}^{M_{k+1}} \partial_j^{k+1} W_{ij}^{k+1}}$ $\Rightarrow \delta_i^1 = x_i^1(1-x_i^1)\sum_{j=1}^{2} \partial_j^2 W_{ij}^2$ $\Rightarrow \boxed{\frac{\partial E}{\partial W_{ij}^k} = \delta_i^k x_i^{k-1}}$

$$\delta_1^1 = x_1^1(1-x_1^1)\left[\delta_1^2 * W_{11}^2 + \delta_2^2 W_{12}^2\right]$$
$$= 0.92*(1-0.92)[(-0.137)*(-1.7)+0.045*2.1]$$
$$= 0.024$$

$$\delta_2^1 = x_2^1(1-x_2^1)\left[\delta_1^2 * W_{21}^2 + \delta_2^2 W_{22}^2\right]$$
$$= 0.27*(1-0.27)[(-0.137)*0.8+0.045*(-0.2)]$$
$$= -0.02$$

Back Propagation Learning:- Hidden Layer

$$\frac{\partial E}{\partial W_{11}^{1}} = \delta_1^1 * x_1^0 = 0.024 * 0.7 = 0.017$$

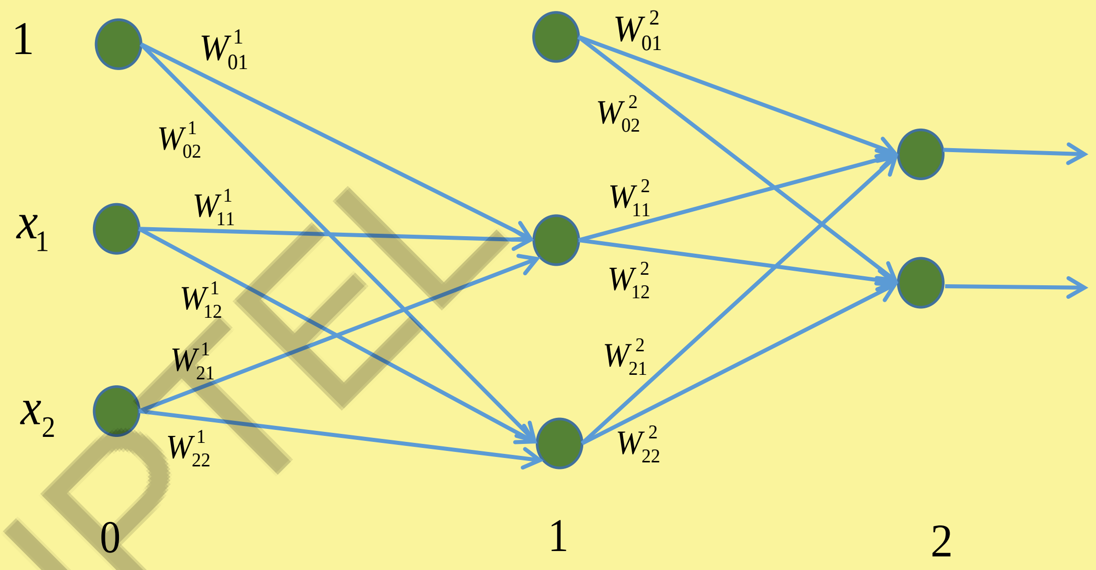
$$\frac{\partial E}{\partial W_{21}^{1}} = \delta_2^1 * x_1^0 = -0.02 * 0.7 = -0.014$$

$$\frac{\partial E}{\partial W_{21}^{1}} = \delta_1^1 * x_2^0 = 0.024 * 1.2 = 0.0288$$

$$\frac{\partial E}{\partial W_{22}^{1}} = \delta_2^1 * x_2^0 = -0.02 * 1.2 = -0.024$$

$$\frac{\partial E}{\partial W_{01}^{1}} = \delta_1^1 * x_0^0 = 0.024 * 1 = 0.024$$

$$\frac{\partial E}{\partial W_{02}^{1}} = \delta_2^1 * x_0^0 = -0.02 * 1 = -0.02$$



$$W_{ij}^1 \leftarrow W_{ij}^1 - \eta \frac{\partial E}{\partial W_{ij}^1}$$

NPTEL ONLINE CERTIFICATION COURSES

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas
Department : E & ECE, IIT Kharagpur

Topic

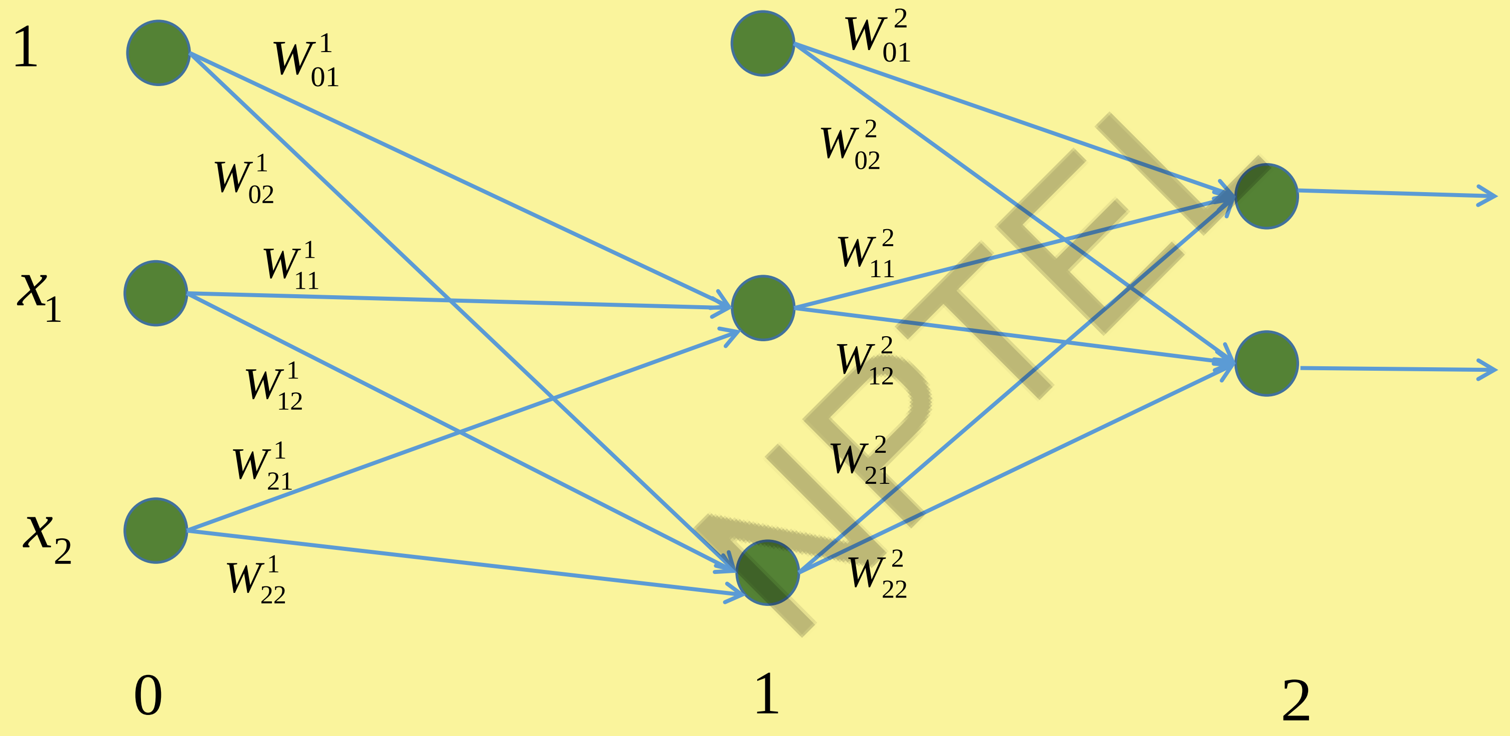
Lecture 27: Back propagation Learning – Examples

CONCEPTS COVERED

Concepts Covered:

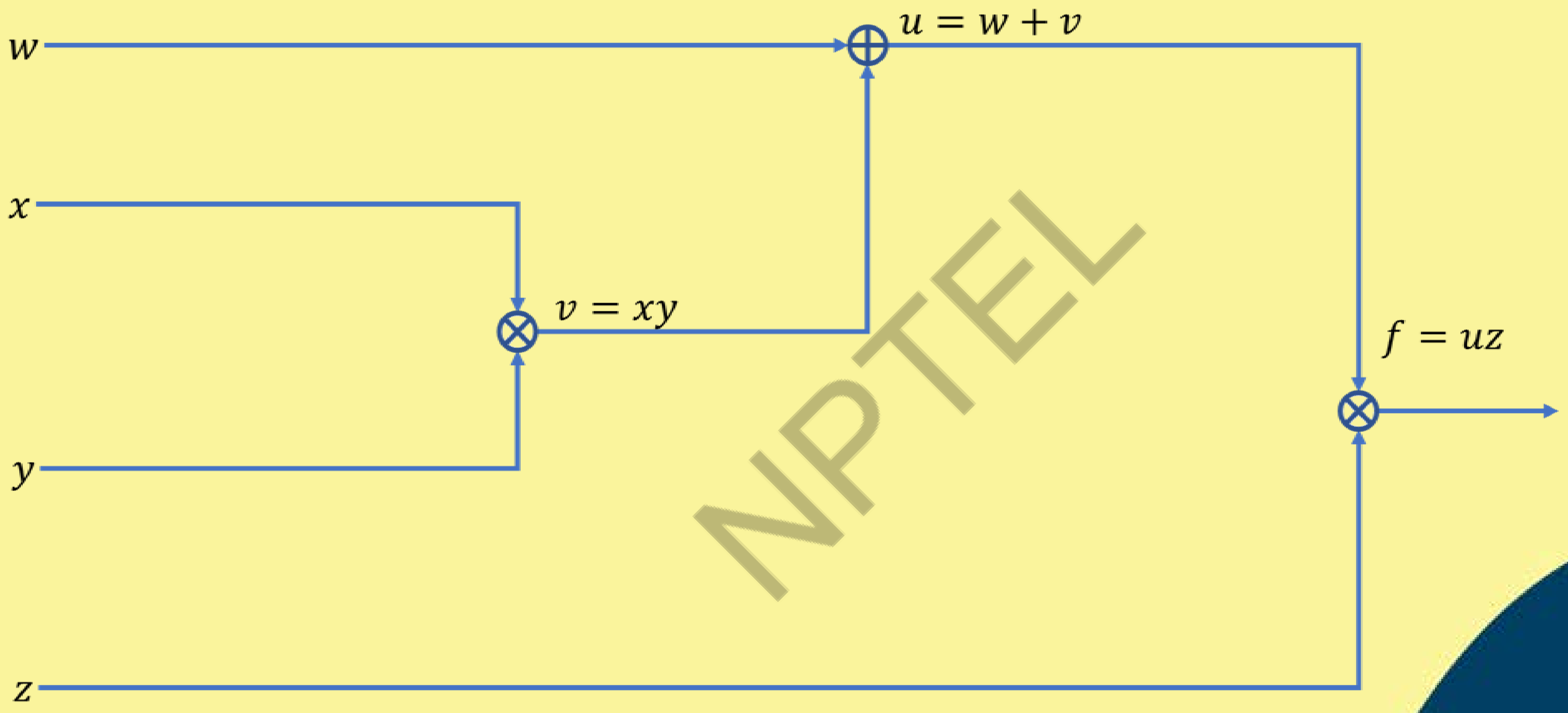
- Back Propagation Learning in MLP
- Back Propagation Learning – Network Level
- Back Propagation – Node Level

Backpropagation at Network Level

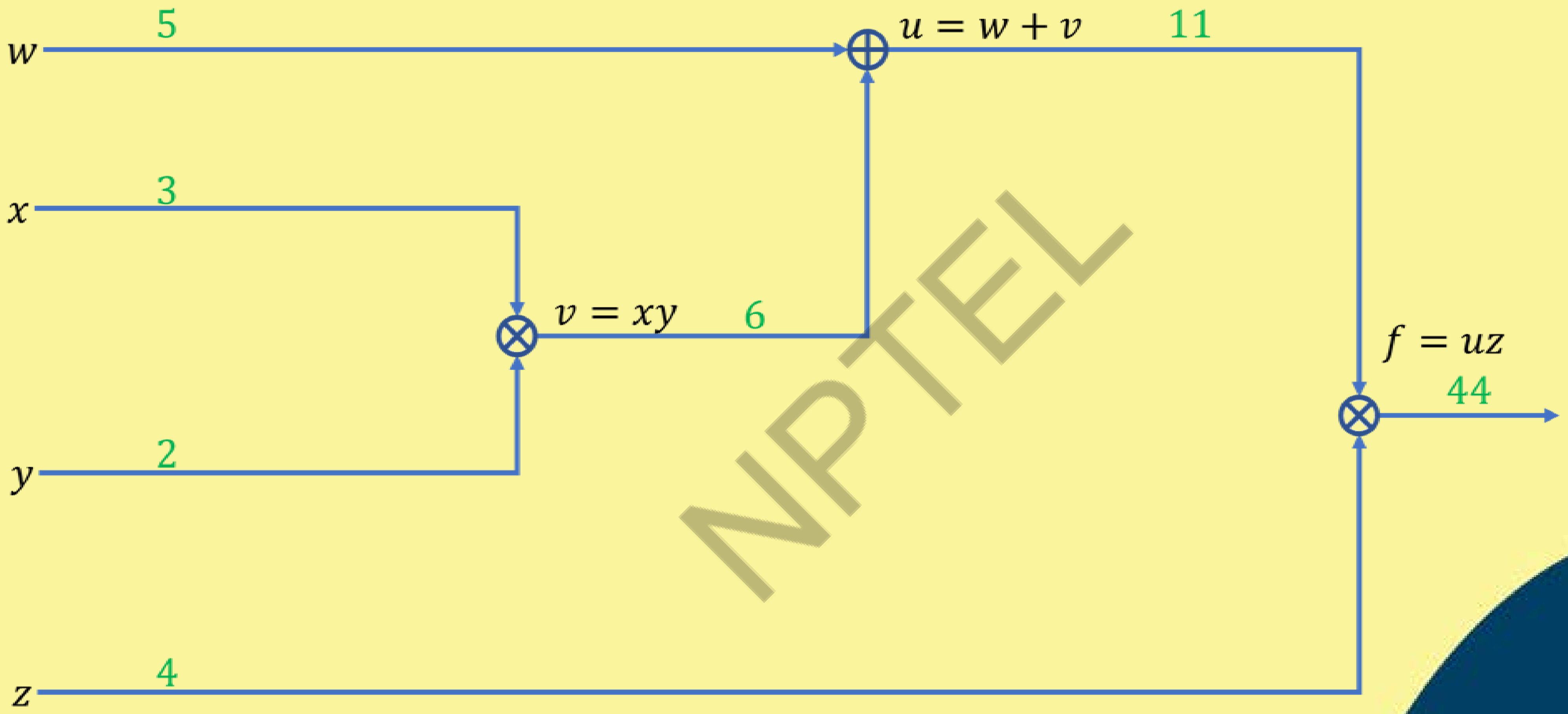


Back Propagation Learning at Node Level

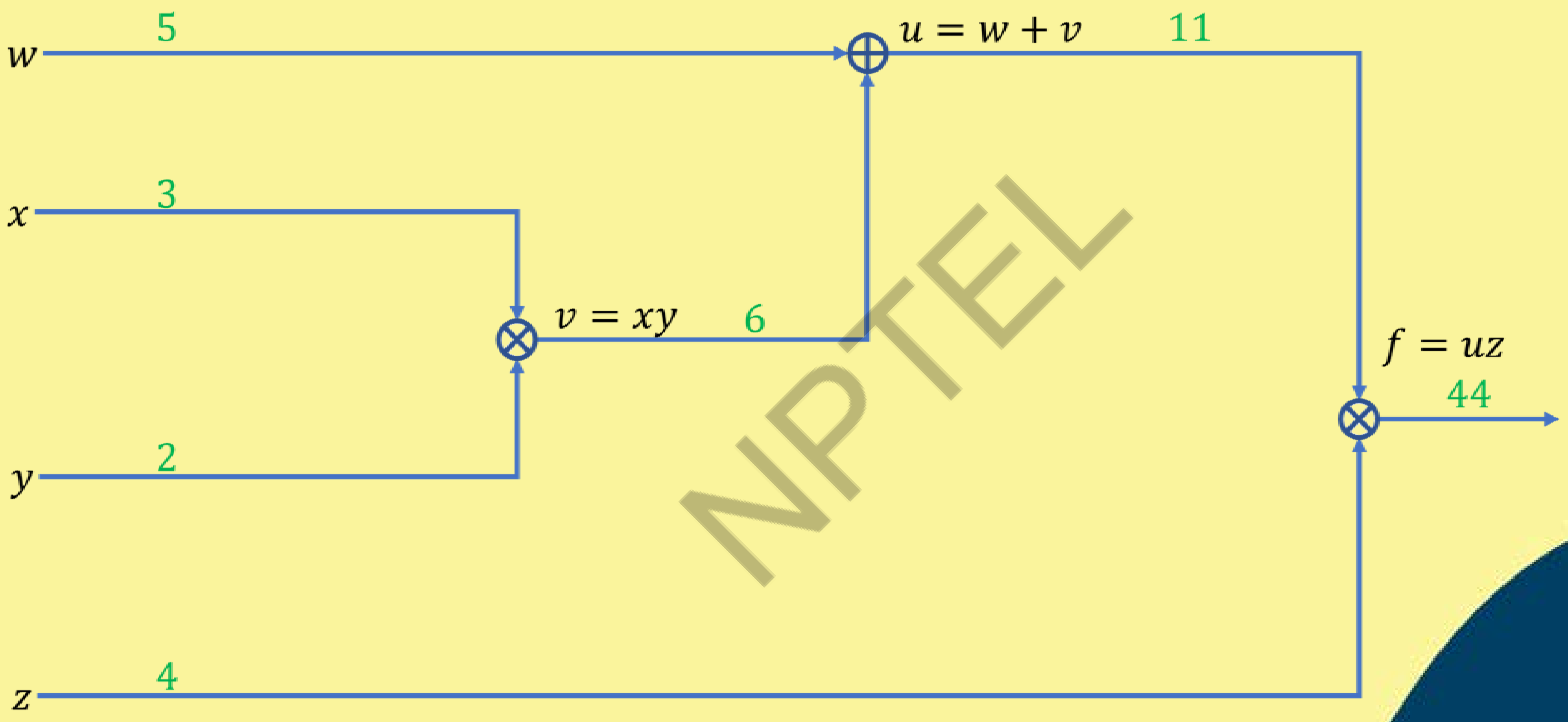
Example: Node Architecture



Example: Forward Pass



Example: Backpropagation



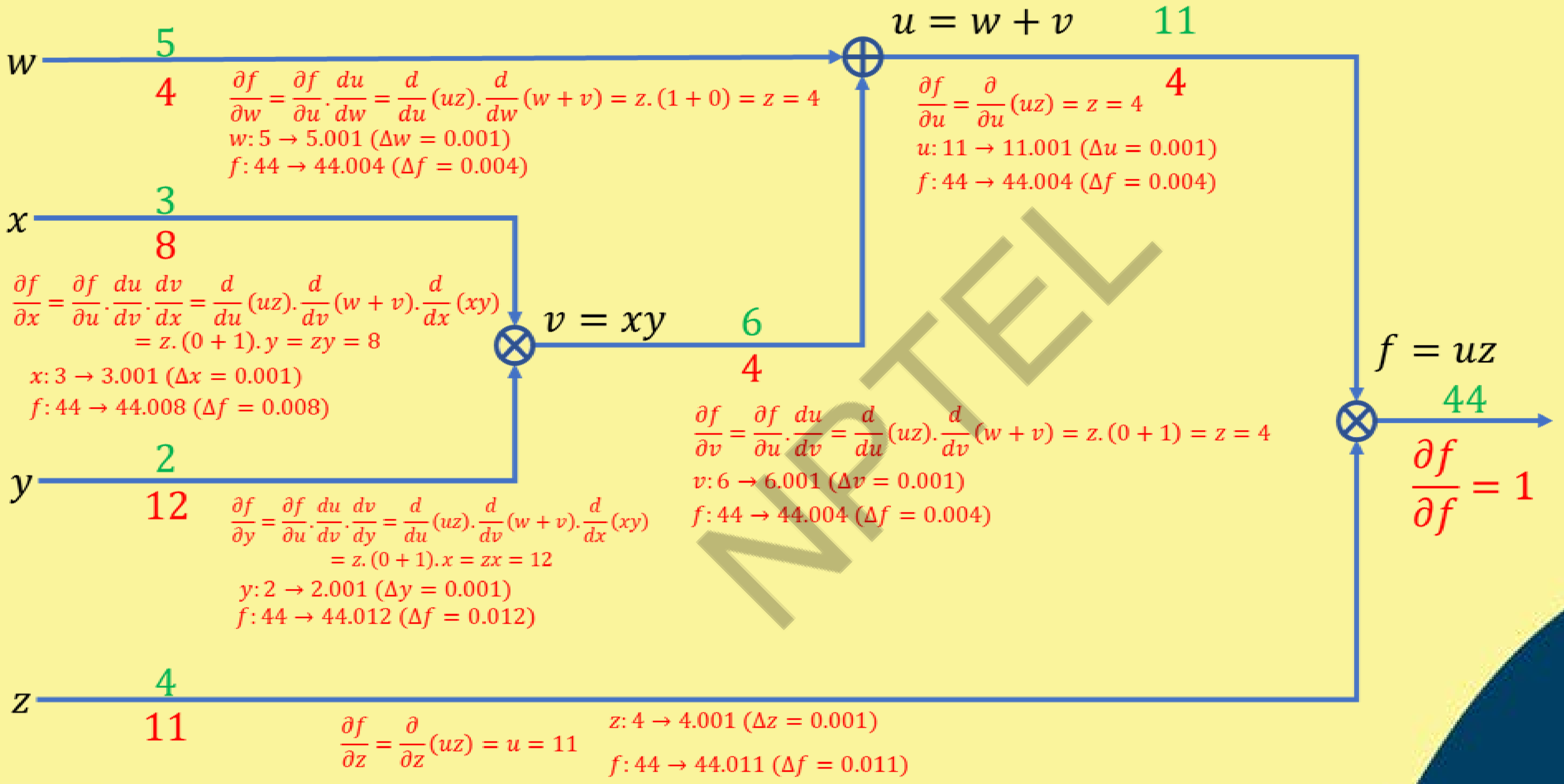
Back propagation: Pseudo Code

```
# Set Input
  w=5; x=3; y=2; z=4

# Forward Pass
  v = x*y
  u = w+v
  f = u+z

# Backward Pass
  dfdu = z
  dfdz = u
  dfdw = 1*dfdu   # dudw = 1
  dfdv  = 1*dfdu  # dudv = 1
  dfdx  = y*dfdv  # dvdx = y
  dfdy  = x*dfdv  # dvdy = x
```

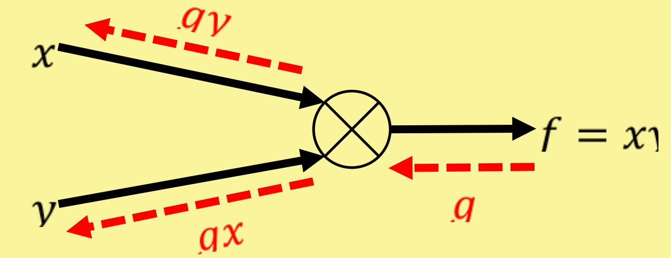
Back propagation: Pseudo Code

```
# Set Input
  w=5; x=3; y=2; z=4

# Forward Pass

  v = x*y
  u = w+v
  f = u+z

# Backward Pass

  dfdu = z
  dfdz = u
  dfdw = 1*dfdu   # dudw = 1
  dfdv  = 1*dfdu  # dudv = 1
  dfdx  = y*dfdv  # dvdx = y
  dfdy  = x*dfdv  # dvdy = x
```

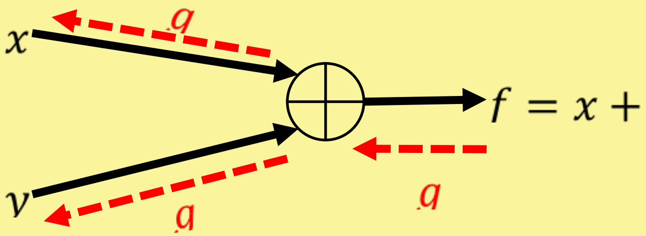
Example: Calculate Gradients



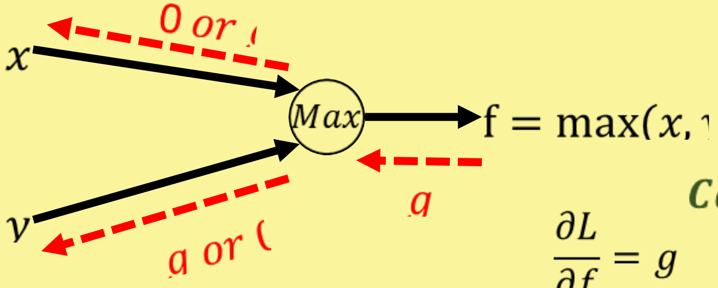
Understanding Gradient Backward



$$\frac{\partial L}{\partial f} = g;\quad \frac{\partial f}{\partial x} = y;\ \frac{\partial f}{\partial y} = x;\ \frac{\partial L}{\partial x} = \frac{\partial L}{\partial f}\cdot\frac{\partial f}{\partial x} = gy;\ \frac{\partial L}{\partial y} = \frac{\partial L}{\partial f}\cdot\frac{\partial f}{\partial y} = gx$$



$$\frac{\partial L}{\partial f} = g;\quad \frac{\partial f}{\partial x} = 1;\ \frac{\partial f}{\partial y} = 1;\ \frac{\partial L}{\partial x} = \frac{\partial L}{\partial f}\cdot\frac{\partial f}{\partial x} = g;\ \frac{\partial L}{\partial y} = \frac{\partial L}{\partial f}\cdot\frac{\partial f}{\partial y} =$$



$$\frac{\partial L}{\partial f} = g$$

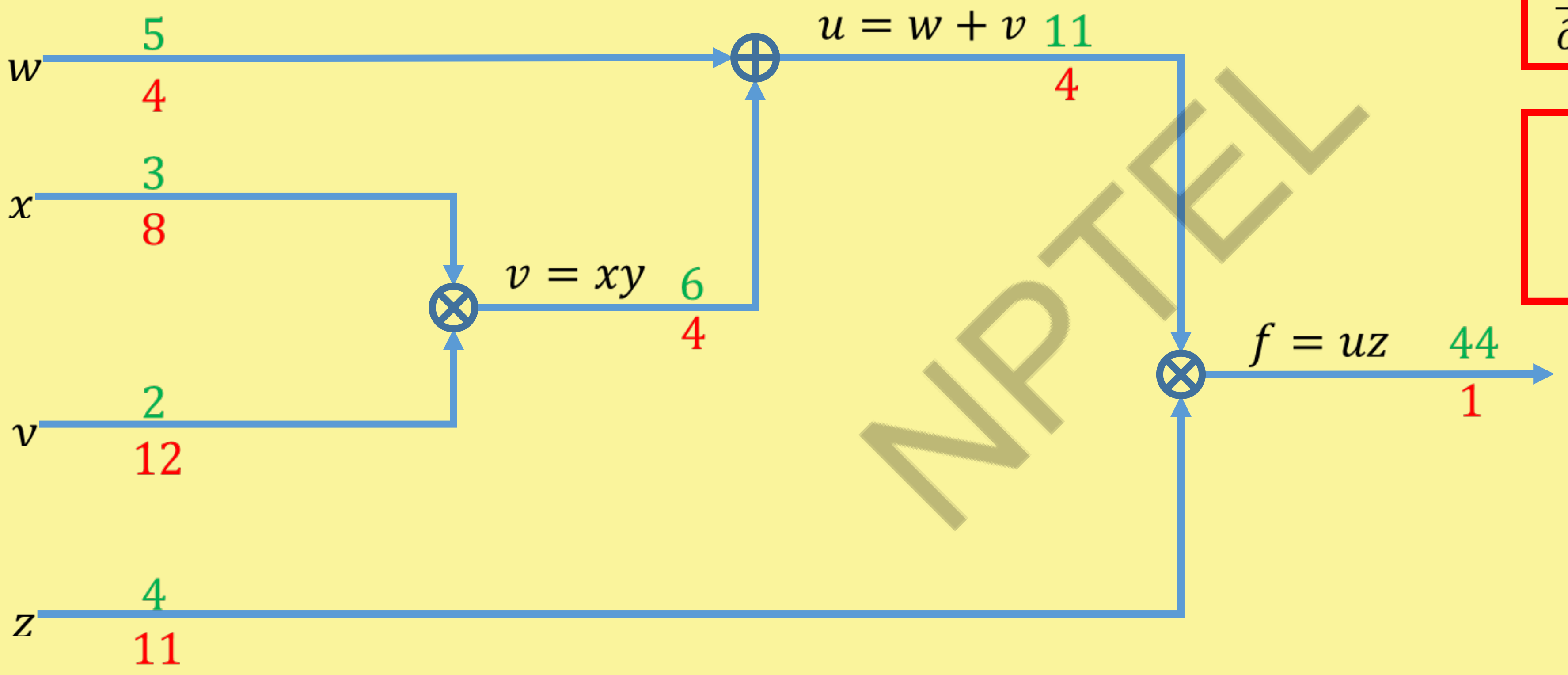
$\boldsymbol{Case - I: x > y; f = max(x, y) = x}$

$$\frac{\partial f}{\partial x} = 1;\ \frac{\partial f}{\partial y} = 0;\ \frac{\partial L}{\partial x} = \frac{\partial L}{\partial f}\cdot\frac{\partial f}{\partial x} = g\quad \frac{\partial L}{\partial y} = \frac{\partial L}{\partial f}\cdot\frac{\partial f}{\partial y} = 0$$

$\boldsymbol{Case - II: x < y; f = max(x, y) = y}$

$$\frac{\partial f}{\partial x} = 0;\ \frac{\partial f}{\partial y} = 1;\ \frac{\partial L}{\partial x} = \frac{\partial L}{\partial f}\cdot\frac{\partial f}{\partial x} = 0;\ \frac{\partial L}{\partial y} = \frac{\partial L}{\partial f}\cdot\frac{\partial f}{\partial y} = g$$

Previous example: different approach



$$f = x +$$

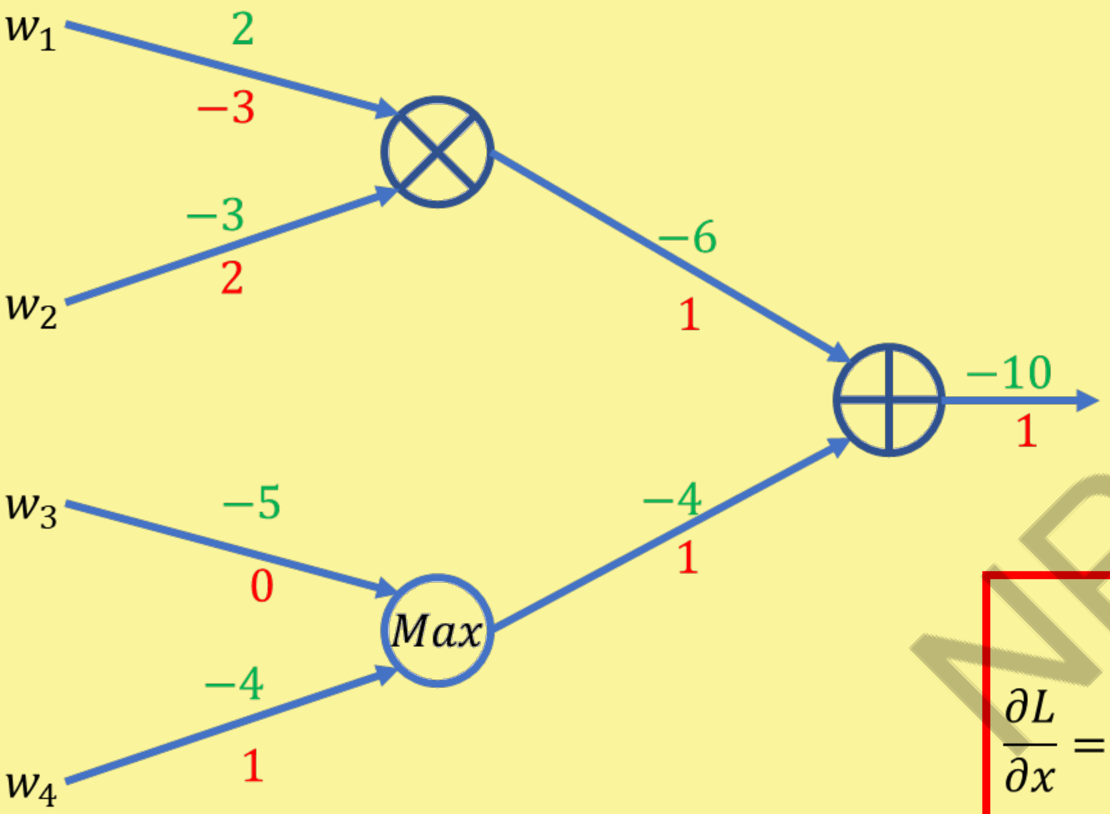
$$\frac{\partial L}{\partial x} = g \qquad \frac{\partial L}{\partial y} = g$$

$$f = x$$

$$\frac{\partial L}{\partial x} = gy \qquad \frac{\partial L}{\partial y} = gx$$

$$\frac{\partial L}{\partial f} =$$

Another Example



$$f = x +$$

$$\frac{\partial L}{\partial x} = g \qquad \frac{\partial L}{\partial y} = g$$

$$\frac{\partial L}{\partial f} =$$

$$f = x$$

$$\frac{\partial L}{\partial x} = gy \qquad \frac{\partial L}{\partial y} = gx$$

$$f = \max(x, y)$$

$$\frac{\partial L}{\partial x} = g \; if \; x > y \qquad \frac{\partial L}{\partial y} = g \; if \; y > x$$

$$= 0 \; otherwise \qquad = 0 \; otherwise$$

NPTEL ONLINE CERTIFICATION COURSES

Thank you

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur

Topic

Lecture 28: Autoencoder

CONCEPTS COVERED

Concepts Covered:

- Back Propagation Learning in MLP
- Autoencoder
  - Undercomplete Autoencoder
  - Autoencoder vs. PCA
  - Sparse Autoencoder
  - Denoising Autoencoder
  - Contractive Autoencoder
  - Convolution Autoencoder

Autoencoder

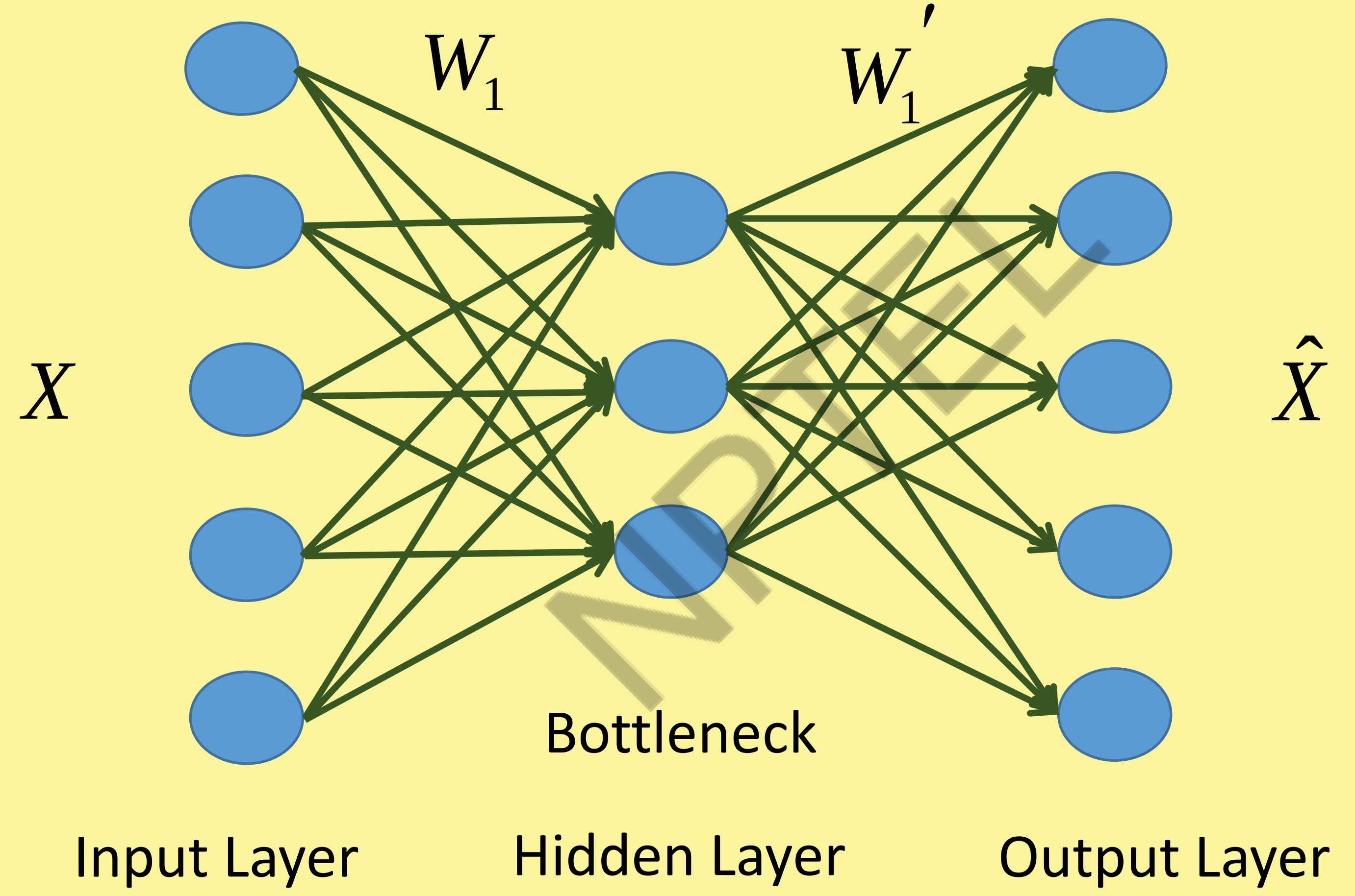
- Unsupervised Learning where Neural Networks are subject to the task of representation learning.
- Impose a bottleneck in the network
- The bottleneck forces a compressed knowledge representation of the input.

Autoencoder

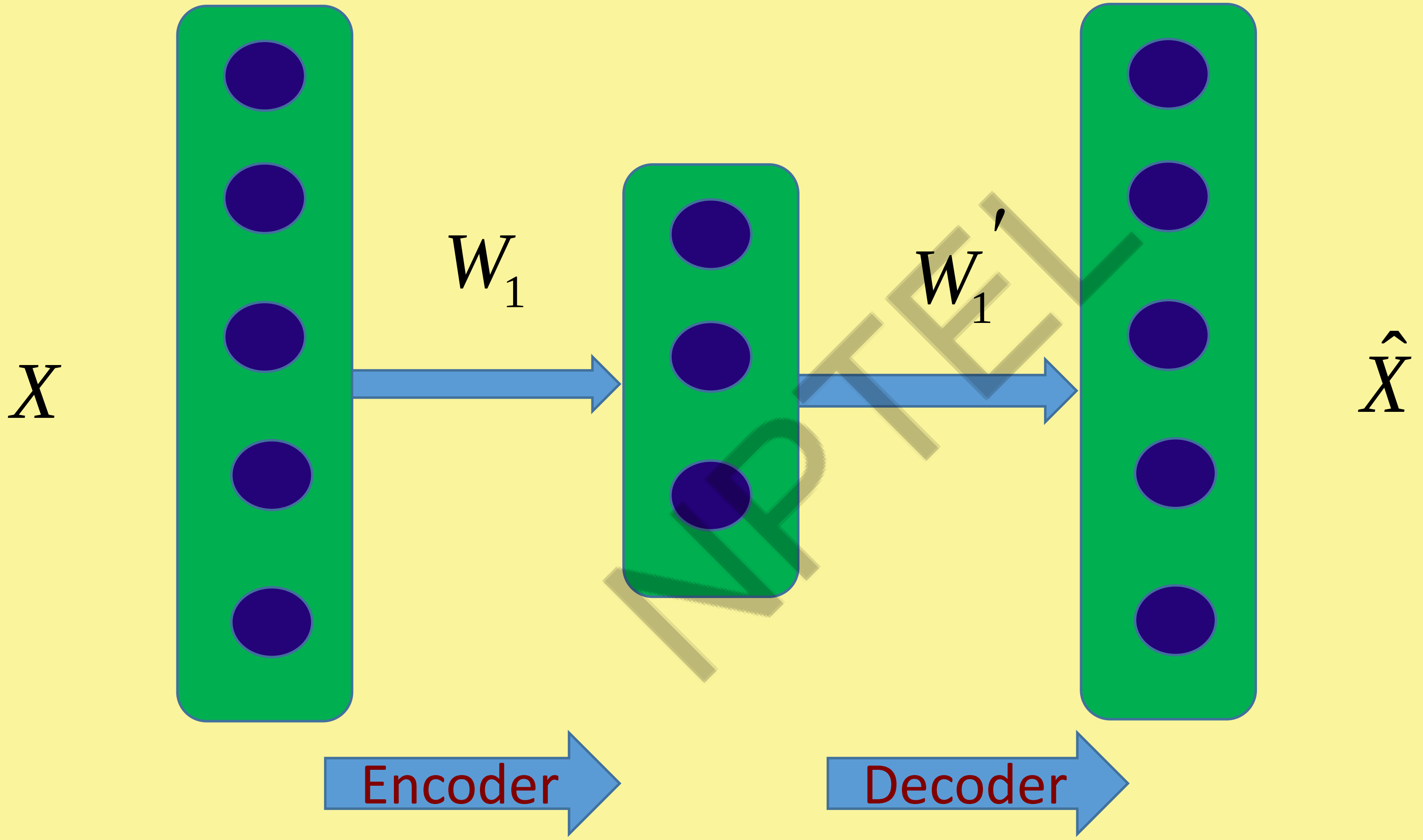
Assumption:

- High degree of correlation/structure exists in the data.
- For uncorrelated data (input features are independent), then compression and subsequent reconstruction would be difficult.

Autoencoder



Autoencoder

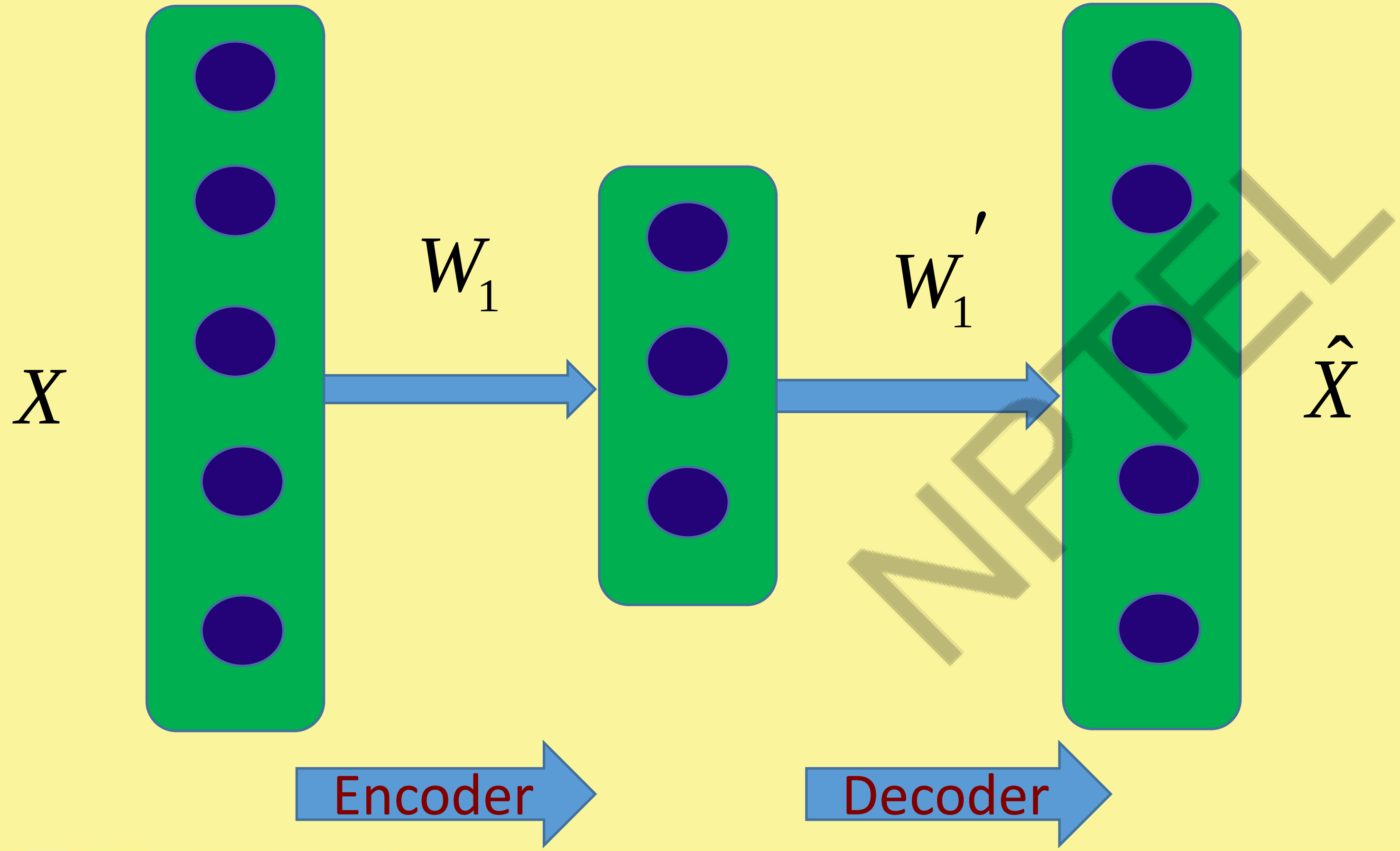


Expectation

- Sensitive enough to input for accurate reconstruction
- Insensitive enough that it does not memorize or overfit the training data

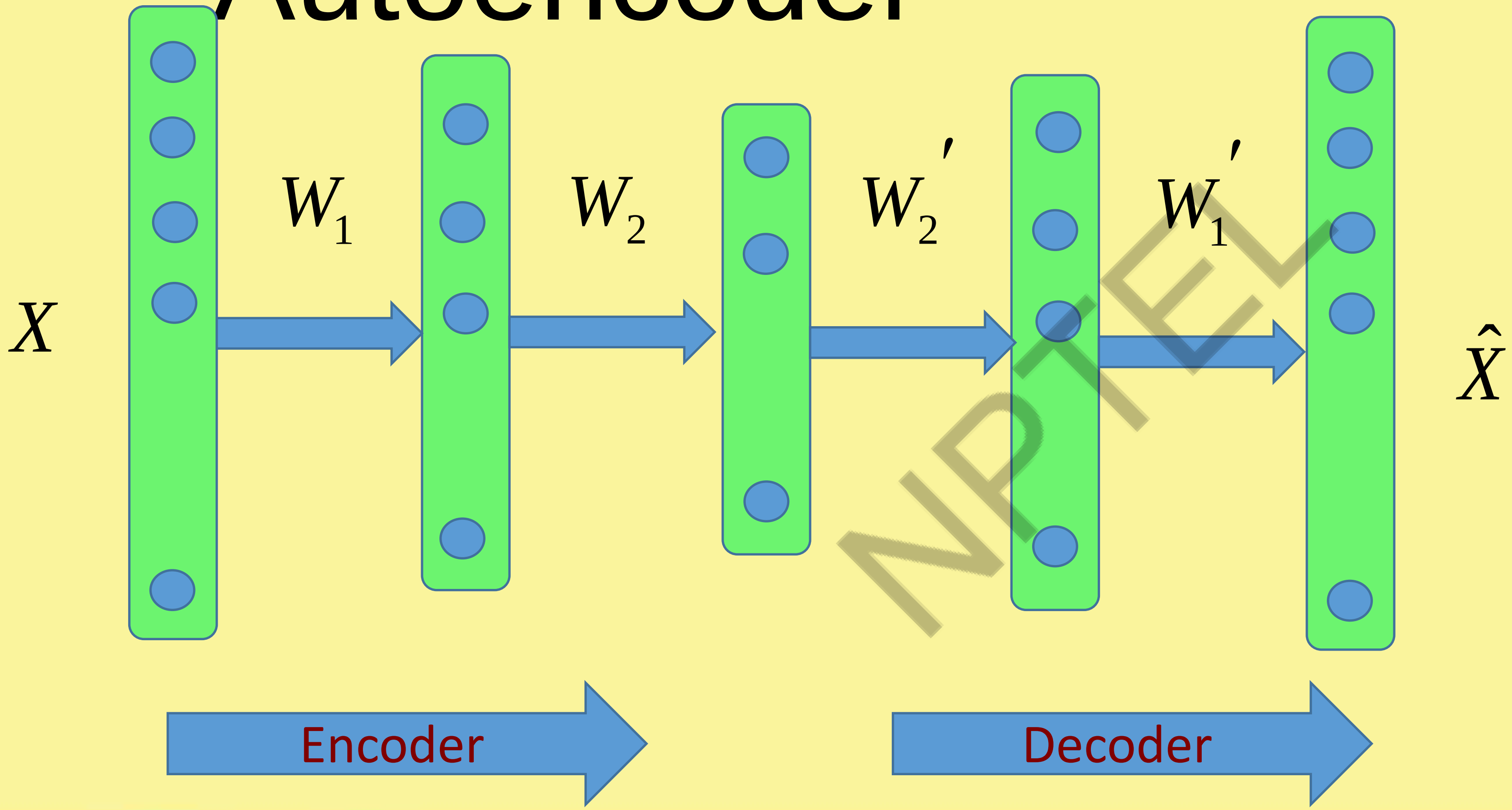
$$\text{Loss Function} \Rightarrow L(X, \hat{X}) + \text{Regularizer}$$

Undercomplete Autoencoder



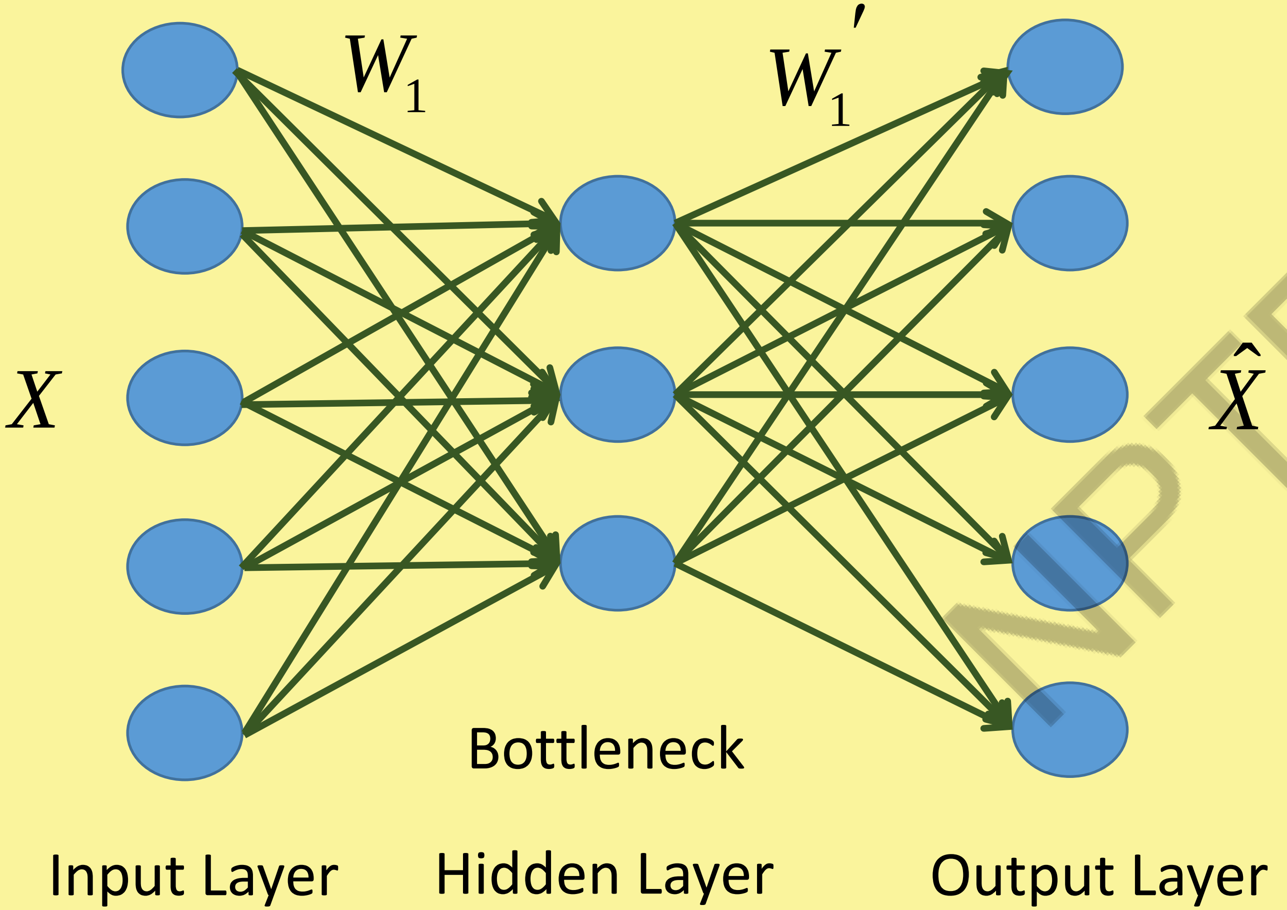
$$L(X,\hat{X}) = \frac{1}{2}\sum_{N}\left\|X-\hat{X}\right\|^2$$

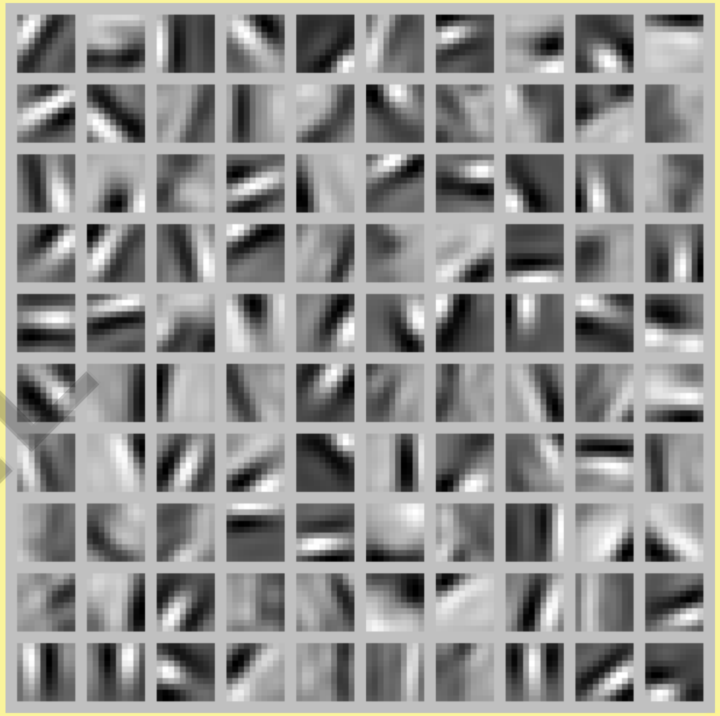
Stacked Autoencoder



$$L(X,\hat{X}) = \frac{1}{2}\sum_{N}\left\|X - \hat{X}\right\|^2$$

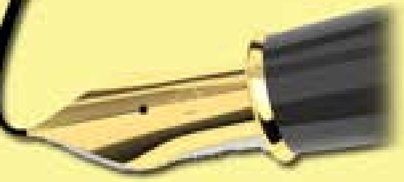
Autoencoder





NPTEL ONLINE CERTIFICATION COURSES

Thank you



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur

Topic

Lecture 29: Autoencoder vs. PCA

CONCEPTS COVERED

Concepts Covered:

- Autoencoder
  - Undercomplete Autoencoder
  - Autoencoder vs. PCA
  - Deep Autoencoder Training
  - Sparse Autoencoder
  - Denoising Autoencoder
  - Contractive Autoencoder
  - Convolution Autoencoder

Autoencoder

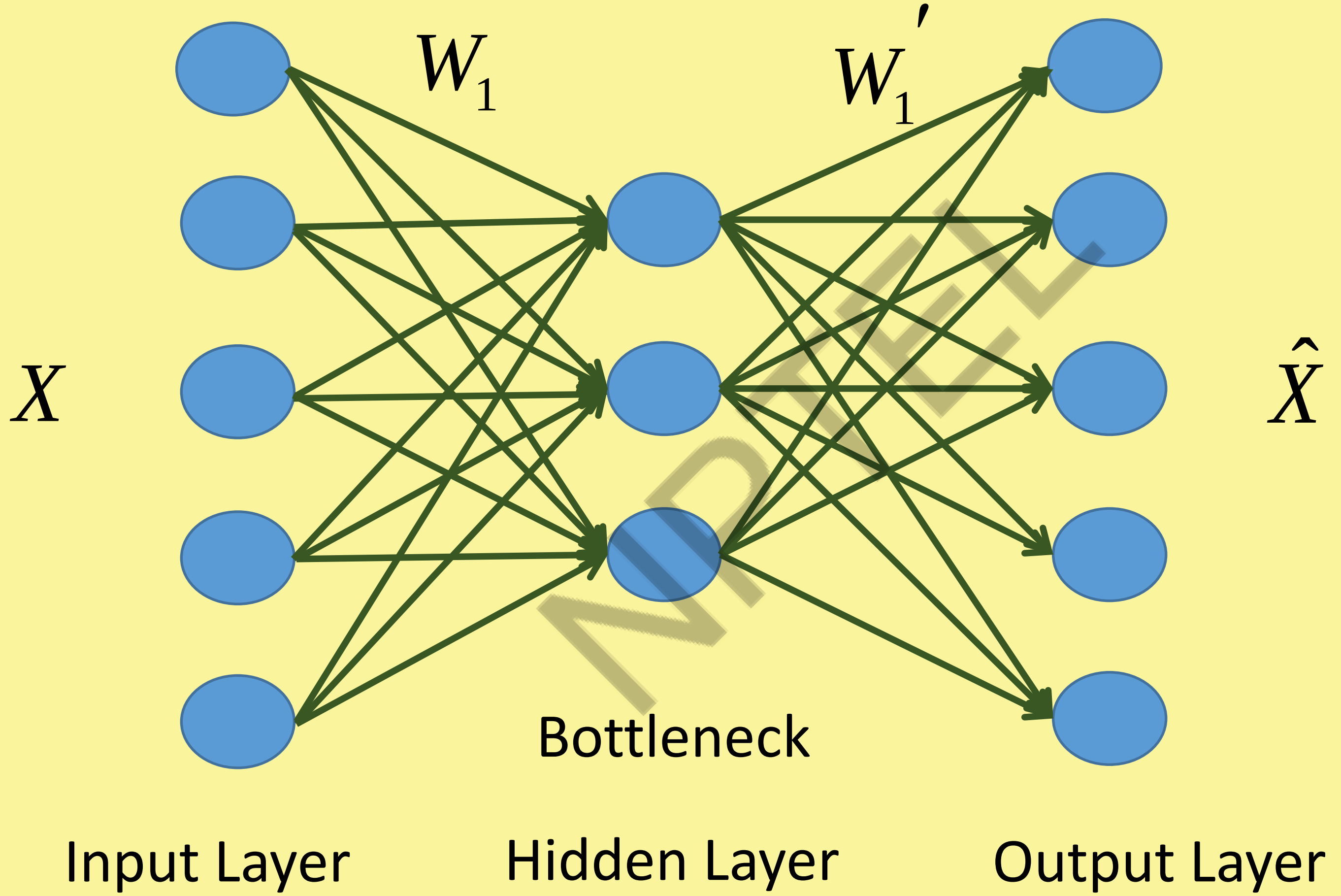
- Unsupervised Learning.
- Representation learning.
- Impose a bottleneck in the network.
- The bottleneck forces a compressed knowledge representation of the input.

Autoencoder

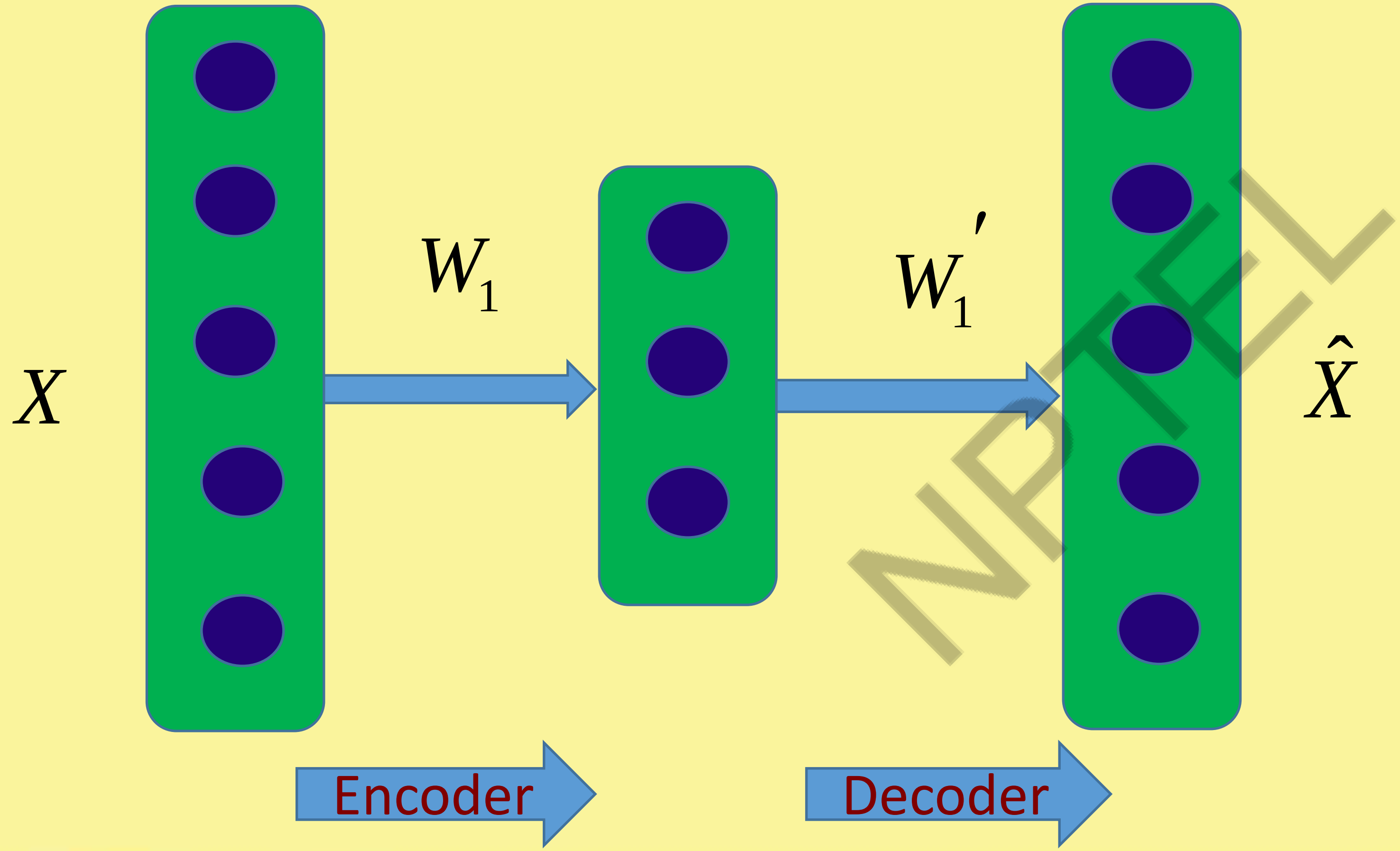
Assumption:

- High degree of correlation/structure exists in the data.
- For uncorrelated data (input features are independent), then compression and subsequent reconstruction would be difficult.

Autoencoder

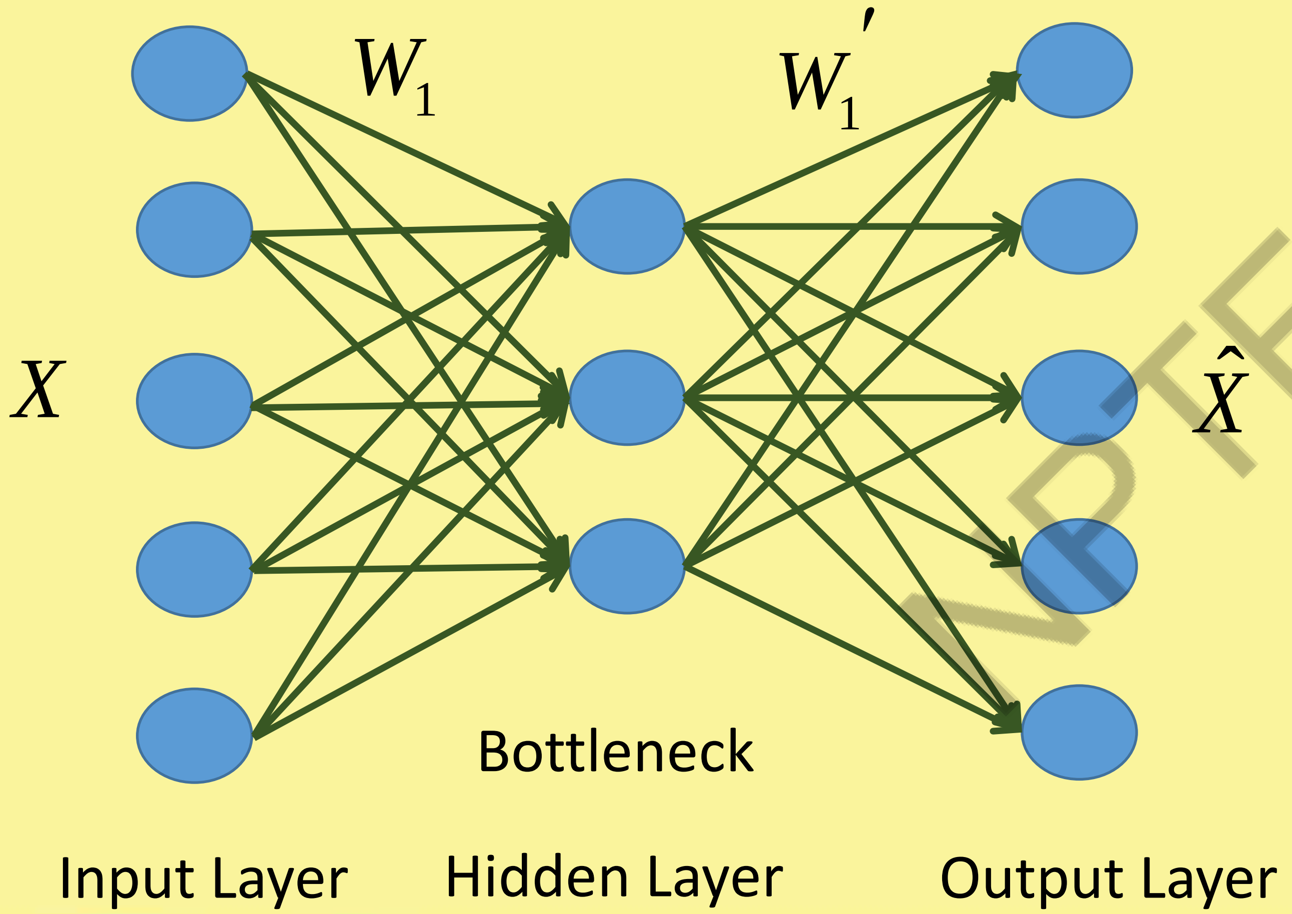


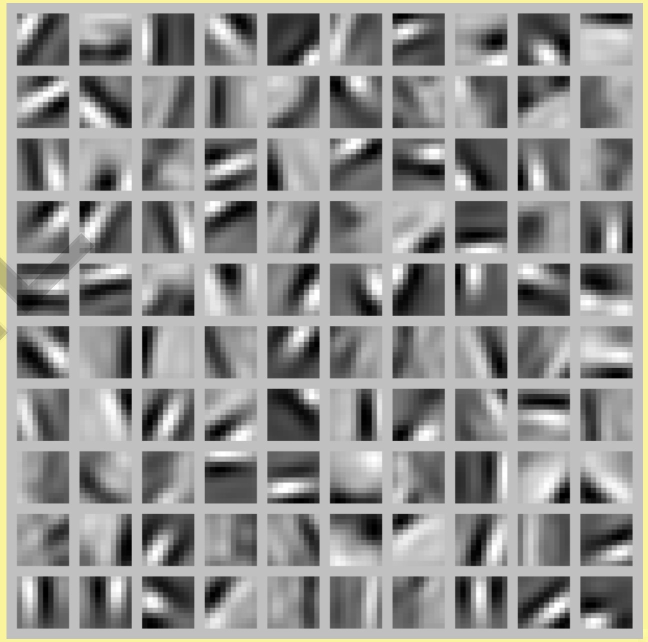
Undercomplete Autoencoder



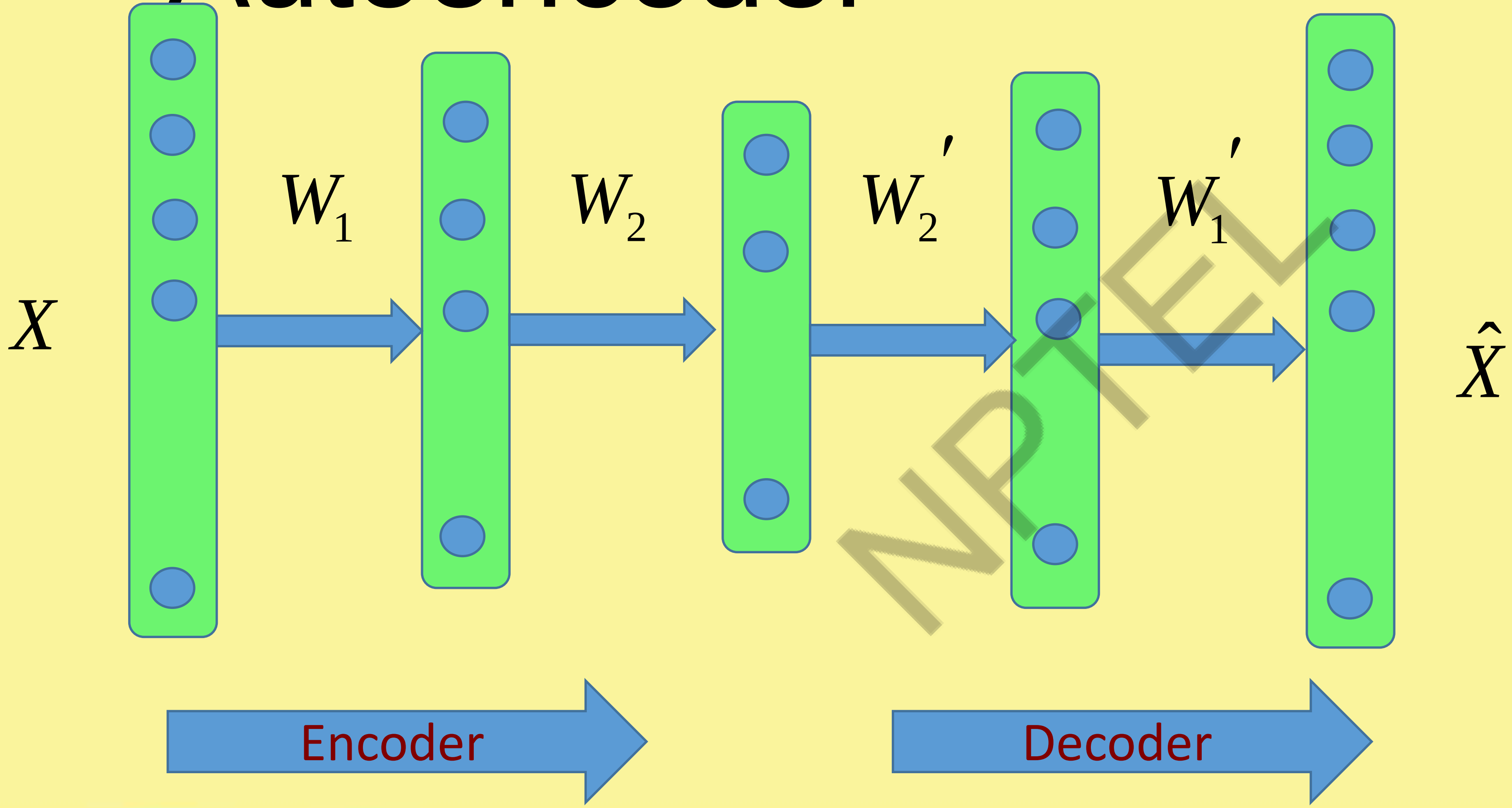
$$L(X,\hat{X}) = \frac{1}{2}\sum_N \left\| X - \hat{X} \right\|^2$$

Autoencoder





Deep Autoencoder



$$L(X,\hat{X}) = \frac{1}{2}\sum_{N}\left\|X - \hat{X}\right\|^2$$

Autoencoder
vs.
PCA

What is PCA?

NPTEL ONLINE CERTIFICATION COURSES

Thank you

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas
Department : E & ECE, IIT Kharagpur

Topic

Lecture 30: Autoencoder vs. PCA

CONCEPTS COVERED

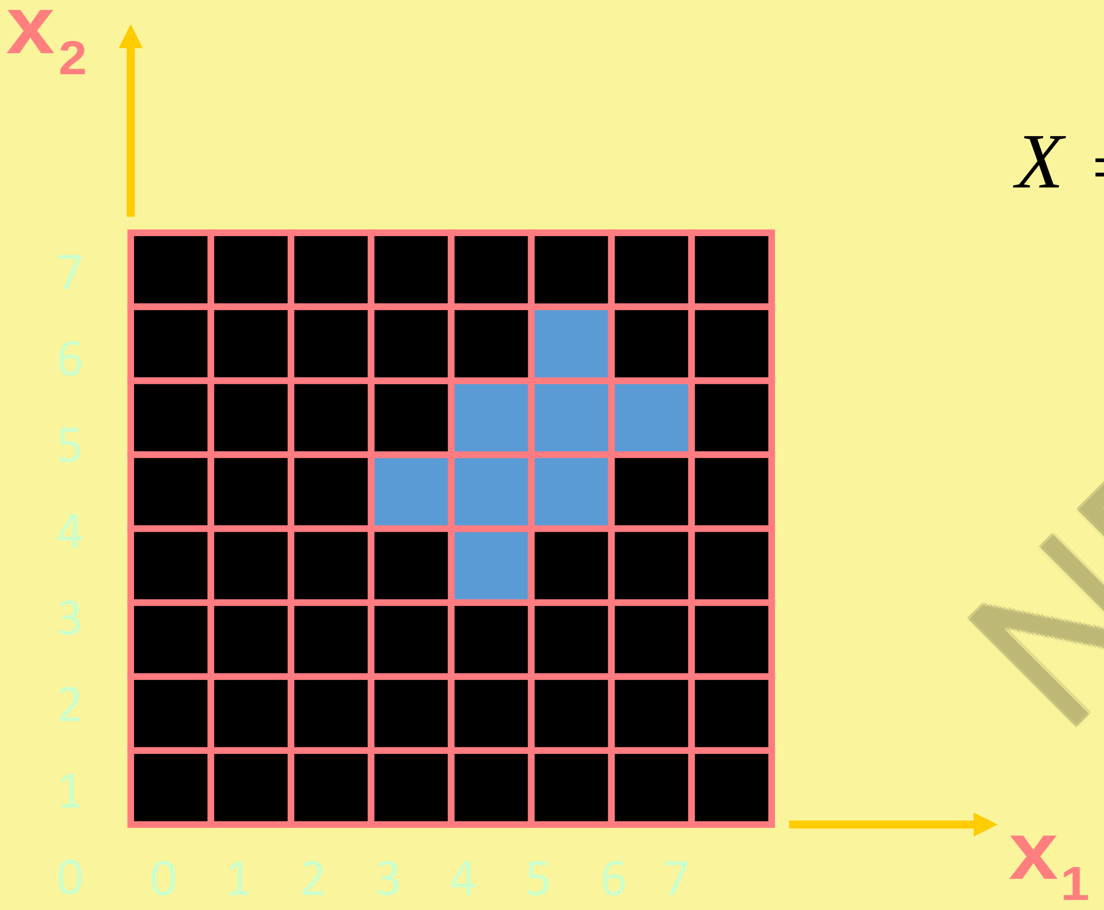
Concepts Covered:

- Autoencoder
  - Undercomplete Autoencoder
  - Autoencoder vs. PCA
  - Deep Autoencoder Training
  - Sparse Autoencoder
  - Denoising Autoencoder
  - Contractive Autoencoder
  - Convolution Autoencoder

Autoencoder
vs.
PCA

What is PCA?

PCA



$$X = \left\{ \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ 4 \end{bmatrix}, \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ 6 \end{bmatrix}, \begin{bmatrix} 6 \\ 5 \end{bmatrix} \right\}$$

$$\mu_X = \begin{bmatrix} 4.5 \\ 4.5 \end{bmatrix}$$

PCA

$$X = \left\{ \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ 4 \end{bmatrix}, \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ 6 \end{bmatrix}, \begin{bmatrix} 6 \\ 5 \end{bmatrix} \right\} \qquad \mu_X = \begin{bmatrix} 4.5 \\ 4.5 \end{bmatrix}$$

$$(X_1 - \mu_X)(X_1 - \mu_X)^t = \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix} \begin{bmatrix} -1.5 & -0.5 \end{bmatrix} = \begin{bmatrix} 2.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix}$$

$$(X_2 - \mu_X)(X_2 - \mu_X)^t = \begin{bmatrix} 0.25 & 0.75 \\ 0.75 & 2.25 \end{bmatrix}$$

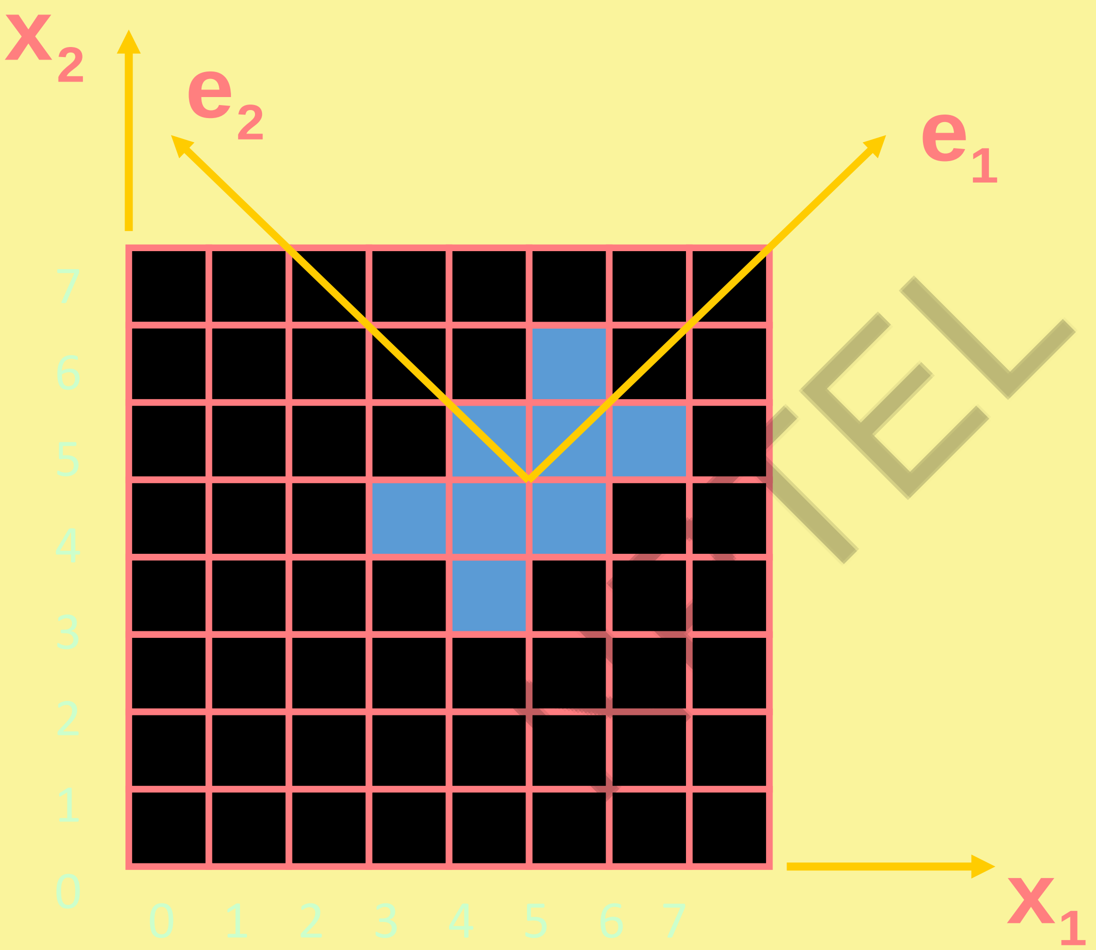
PCA

$$C_X = \begin{bmatrix} 0.75 & 0.375 \\ 0.375 & 0.75 \end{bmatrix}$$

$$\begin{vmatrix} 0.75-\lambda & 0.375 \\ 0.375 & 0.75-\lambda \end{vmatrix} = 0 \Rightarrow \lambda_1 = 1.125 \;\&\; \lambda_2 = 0.375$$

$$\lambda_1 \Rightarrow e_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \lambda_2 \Rightarrow e_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

PCA



PCA

ORIGINAL

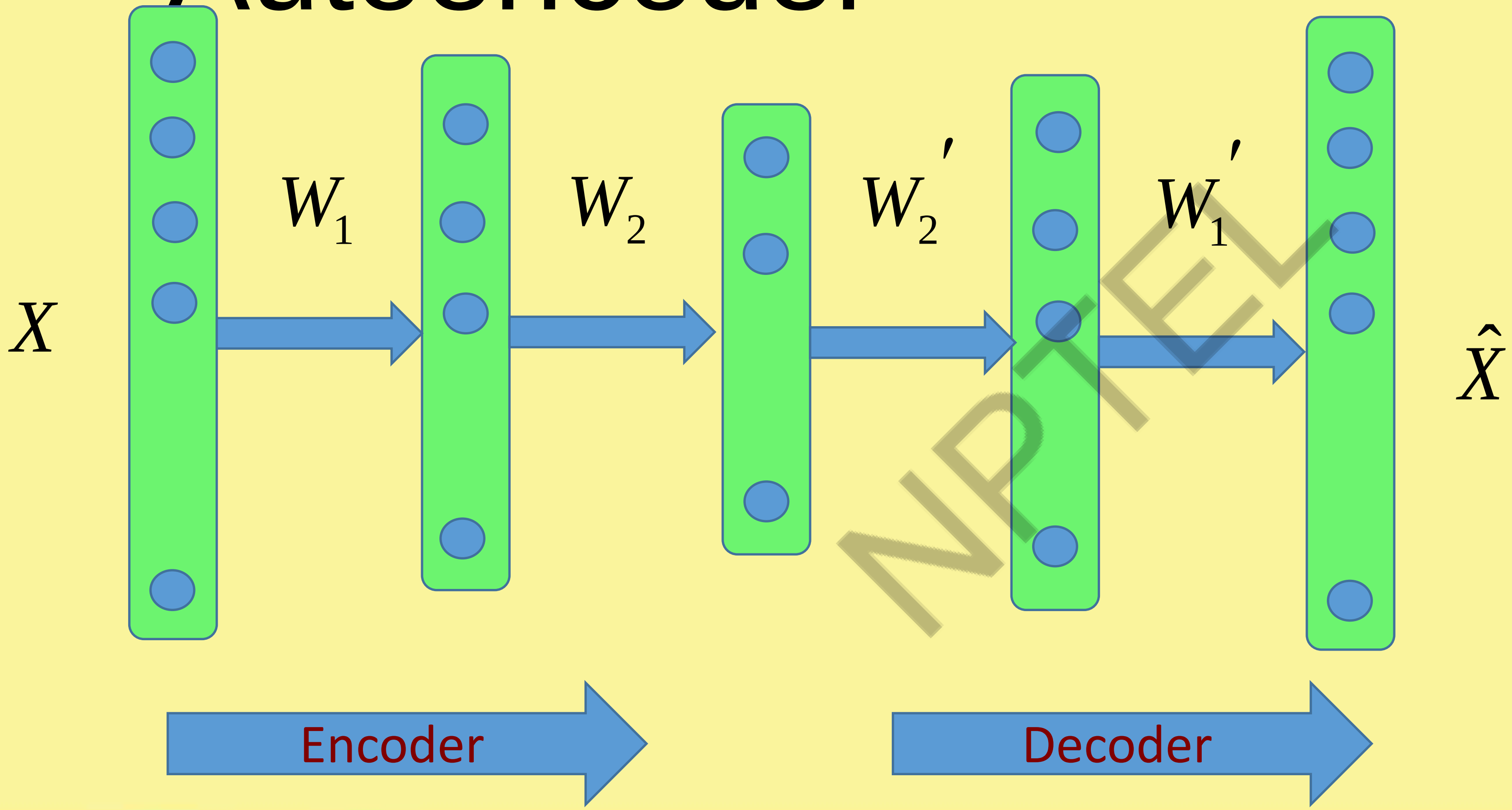
1

5

25

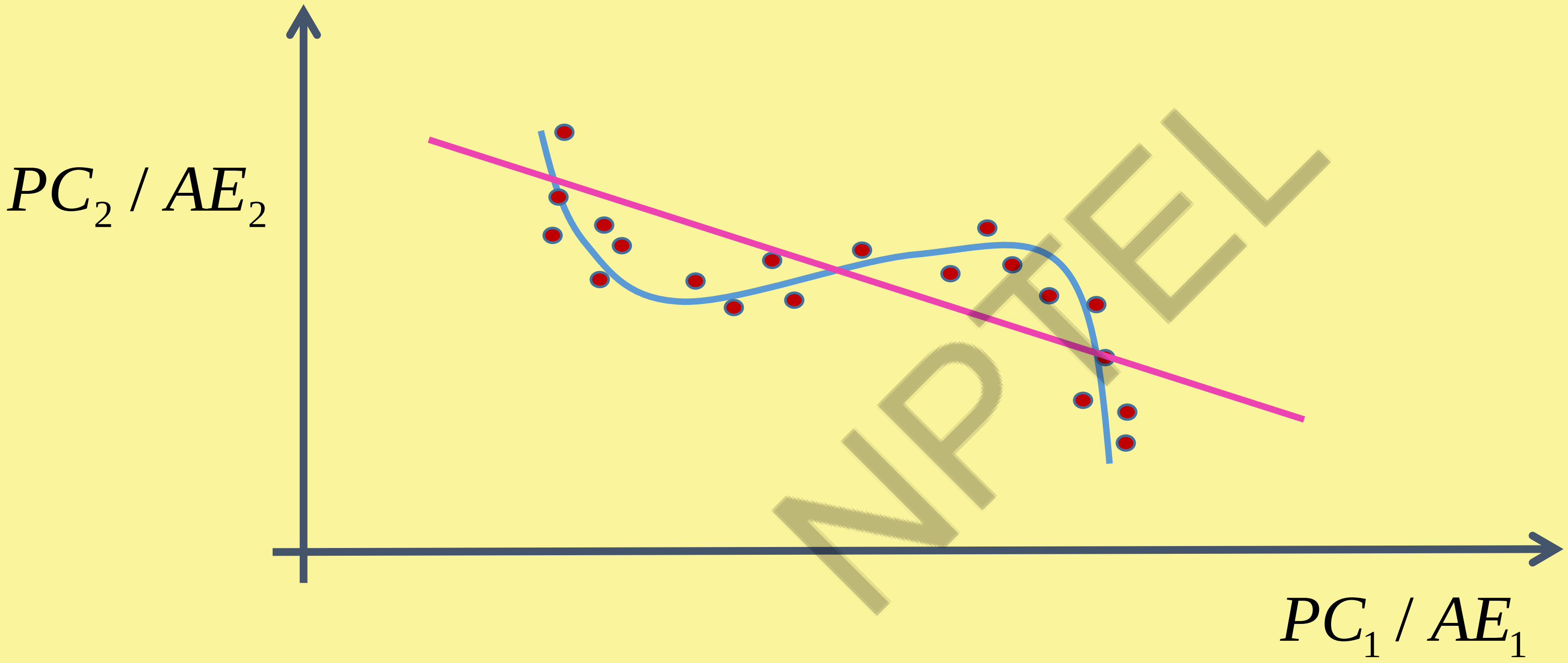
What does PCA do?

Deep Autoencoder



$$L(X,\hat{X}) = \frac{1}{2}\sum_{N}\left\|X - \hat{X}\right\|^2$$

Autoencoder vs. PCA



Experimental Setup for Dimensionality Reduction

- Dataset used: MNIST (a large database of handwritten digits)
  - Total train Images: 60,000
  - Total test Images: 10,000
  - Image dimension: 28×28 (784)
  - Dimensionality reduction: $784 \rightarrow 2$
  - Reconstruction: $784 \rightarrow 30$
- Optimizer used: Adam (Learning rate-10^{-4})
- Loss Function: Mean Squared Error
- Trained for 100 iterations

Source: G. E. Hilton and R. R. Salakhutdinov: "Reducing the Dimensionality of Data with Neural Networks", Science, Vol 313, July 2006, pp. 504-507.

MNIST Data set: Example

Autoencoder converges to PCA

PCA

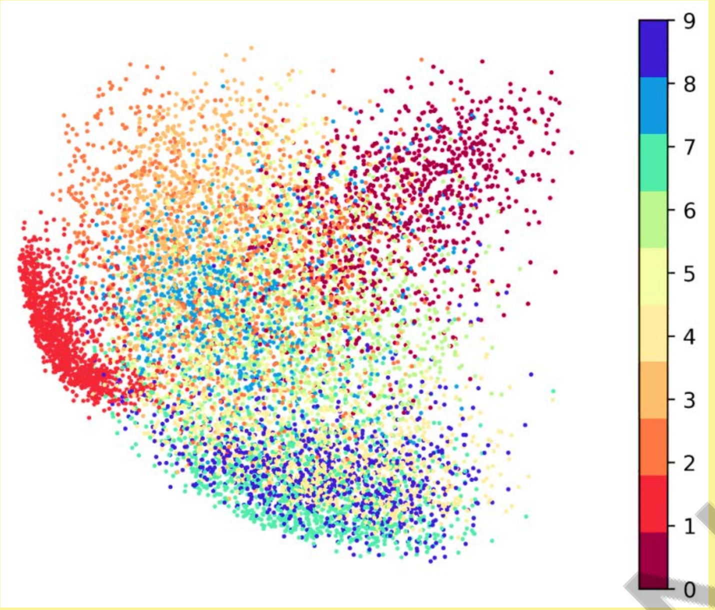
2-Layer Autoencoder

$$784 \downarrow 2 \downarrow 784$$

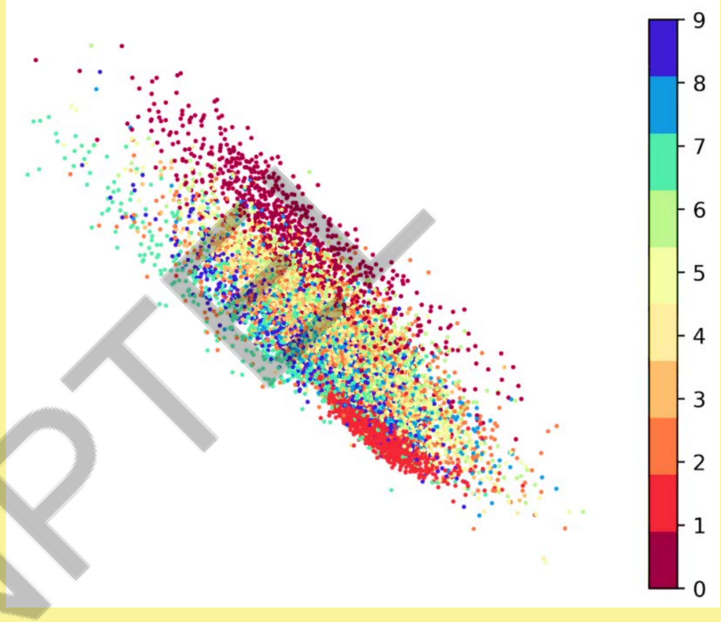
Source: G. E. Hilton and R. R. Salakhutdinov: “Reducing the Dimensionality of Data with Neural Networks”, Science, Vol 313, July 2006, pp. 504-507.

Deep vs. Shallow autoencoder

784
↓
2 (Sigmoid)
↓
784



2-Layered AE with
Sigmoid Non-Linearity

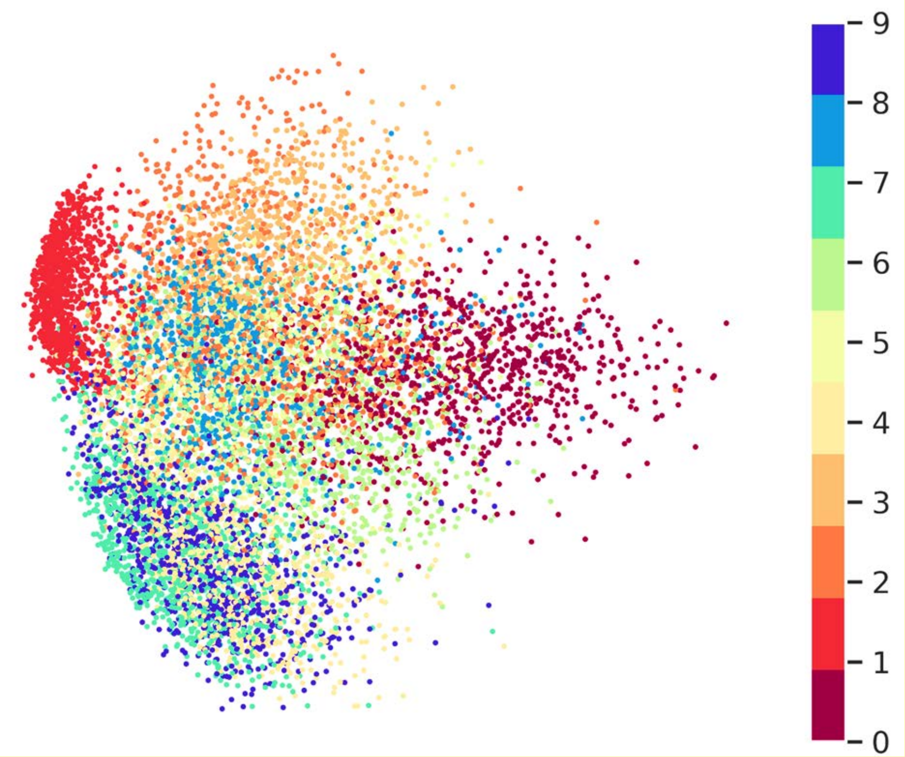


Deep AE without any
Non-Linear Activations

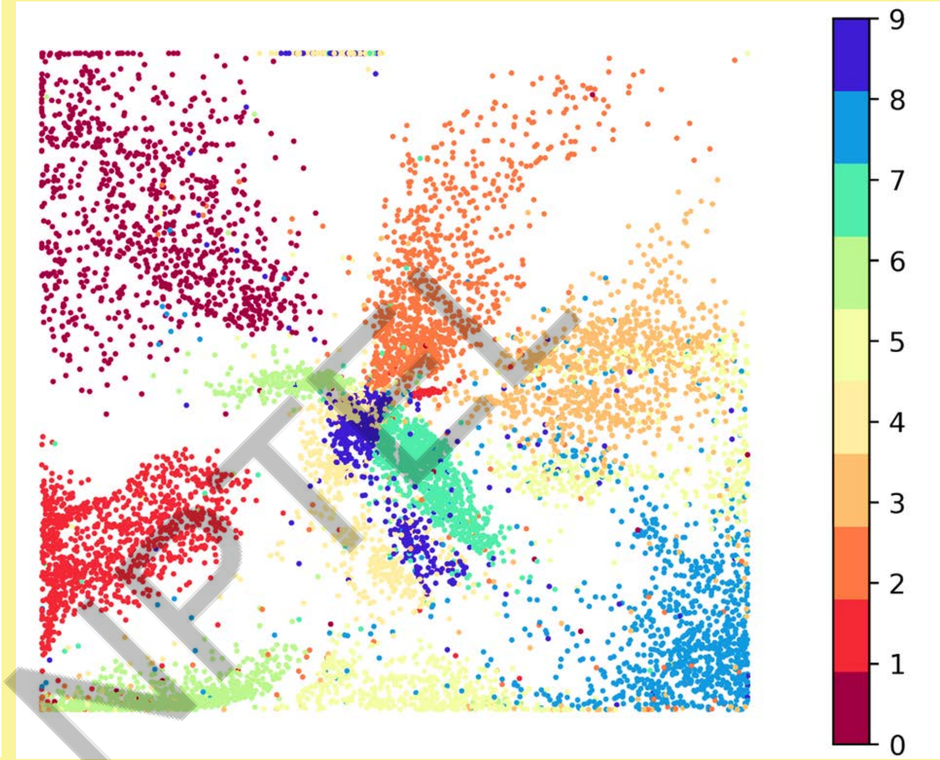
784
↓
1000
↓
500
↓
250
↓
2
↓
250
↓
500
↓
1000
↓
784

Source: G. E. Hilton and R. R. Salakhutdinov: “Reducing the Dimensionality of Data with Neural Networks”, Science, Vol 313, July 2006, pp. 504-507.

Deep Autoencoder with Non-Linear Activations



Principal component analysis (PCA)



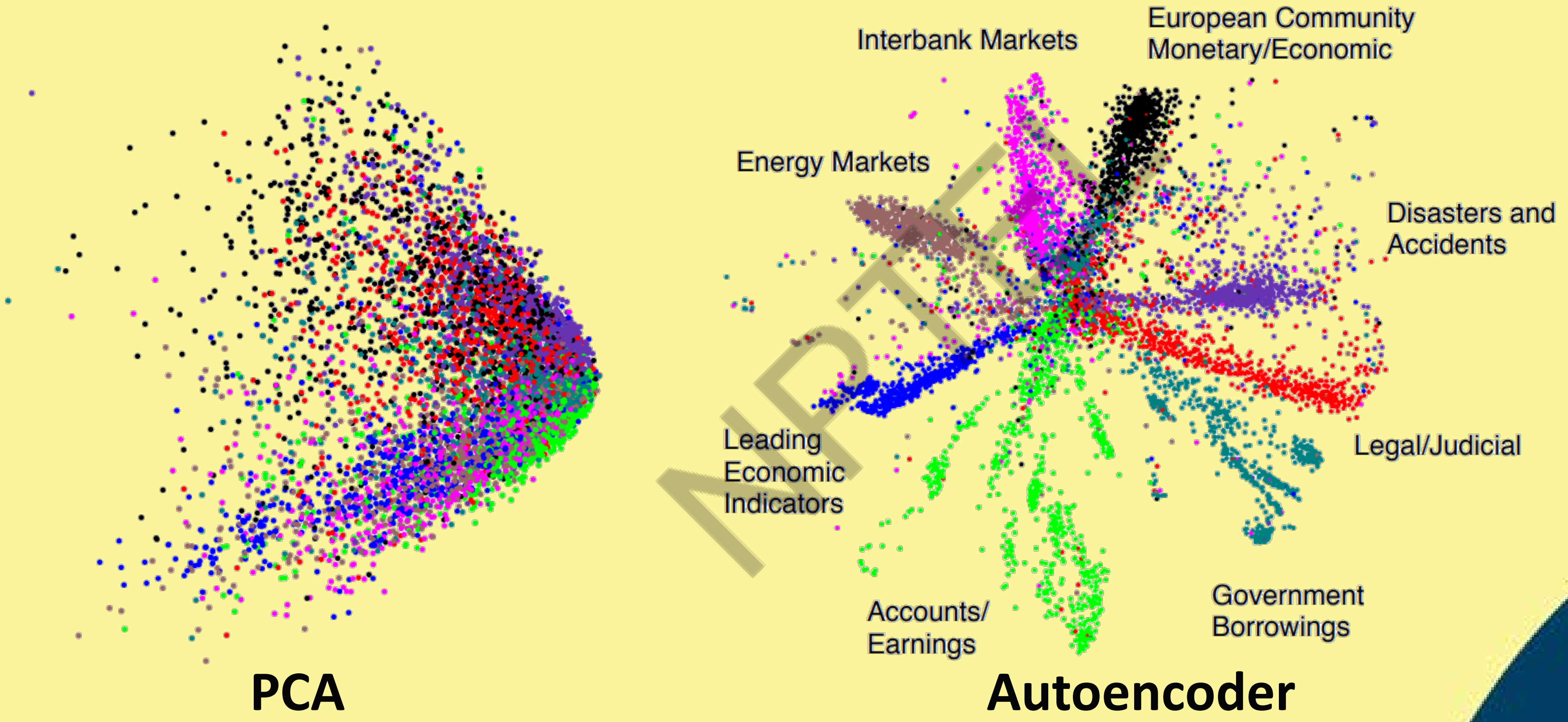
Deep Autoencoder with Non-Linear Activations

$784\ (ReLU)$
$\downarrow$
$1000\ (ReLU)$
$\downarrow$
$500\ (ReLU)$
$\downarrow$
$250\ (ReLU)$
$\downarrow$
$\mathbf{2\ (\boldsymbol{Sigmoid})}$
$\downarrow$
$250\ (ReLU)$
$\downarrow$
$500\ (ReLU)$
$\downarrow$
$1000\ (ReLU)$
$\downarrow$
784

Source: G. E. Hilton and R. R. Salakhutdinov: "Reducing the Dimensionality of Data with Neural Networks", Science, Vol 313, July 2006, pp. 504-507.

Autoencoder for Dimensionality Reduction

Articles from Reuter corpus were mapped to a 2 dimensional vector



Source: G. E. Hilton and R. R. Salakhutdinov: "Reducing the Dimensionality of Data with Neural Networks", Science, Vol 313, July 2006, pp. 504-507.

Reconstruction from Latent Space

Original

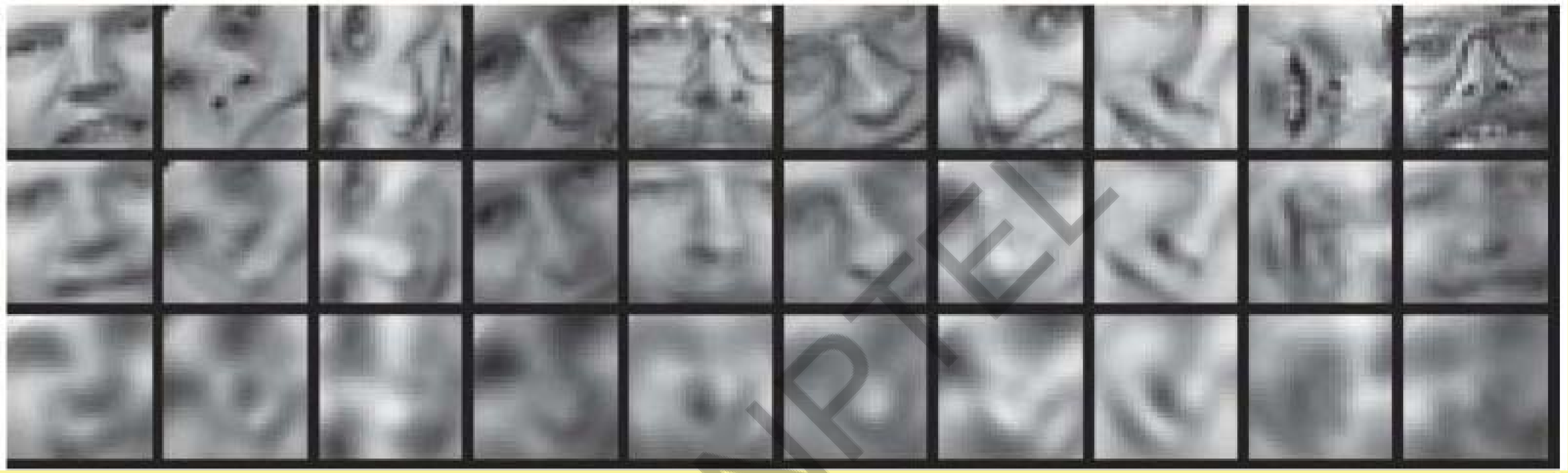
Autoencoder

PCA

$784\ (ReLU) \rightarrow 1000\ (ReLU) \rightarrow 500\ (ReLU) \rightarrow 250\ (ReLU) \rightarrow \mathbf{30\ (ReLU)} \rightarrow 250\ (ReLU) \rightarrow 500\ (ReLU) \rightarrow 1000\ (ReLU) \rightarrow 784$

Source: G. E. Hilton and R. R. Salakhutdinov: “Reducing the Dimensionality of Data with Neural Networks”, Science, Vol 313, July 2006, pp. 504-507.

Reconstruction from Latent Space



(a) Original (b) 30-D AE (c) 30-D PC

Source: G. E. Hilton and R. R. Salakhutdinov: "Reducing the Dimensionality of Data with Neural Networks", Science, Vol 313, July 2006, pp. 504-507.

NPTEL ONLINE CERTIFICATION COURSES

Thank you