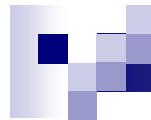
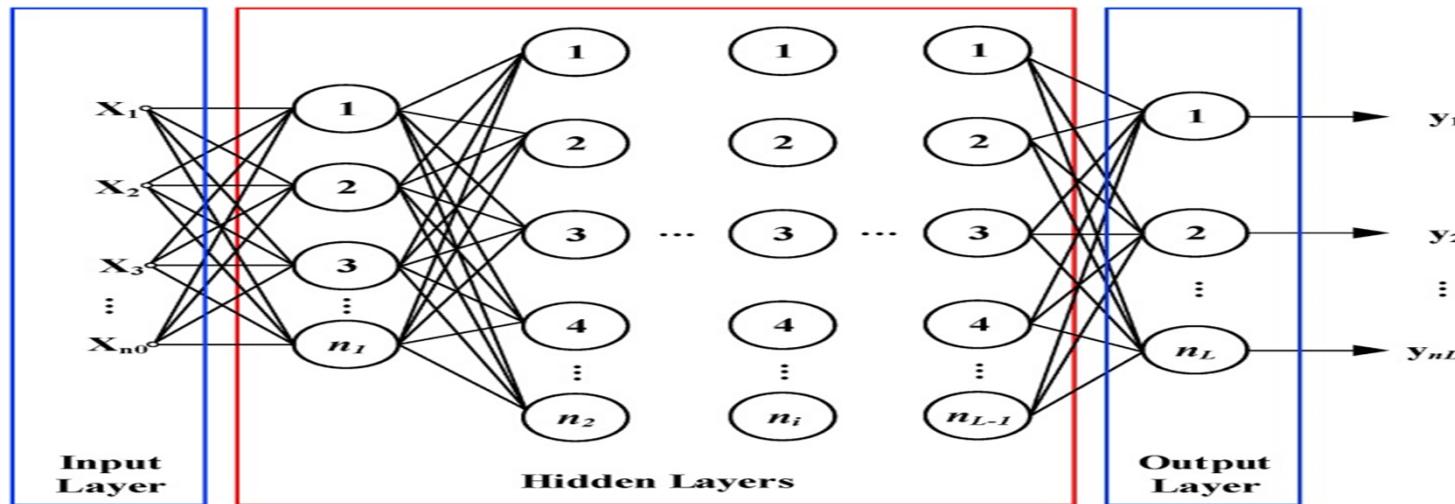


DEEP LEARNING (DL) (APRENDIZAJE PROFUNDO)



Topológica de Deep Learning



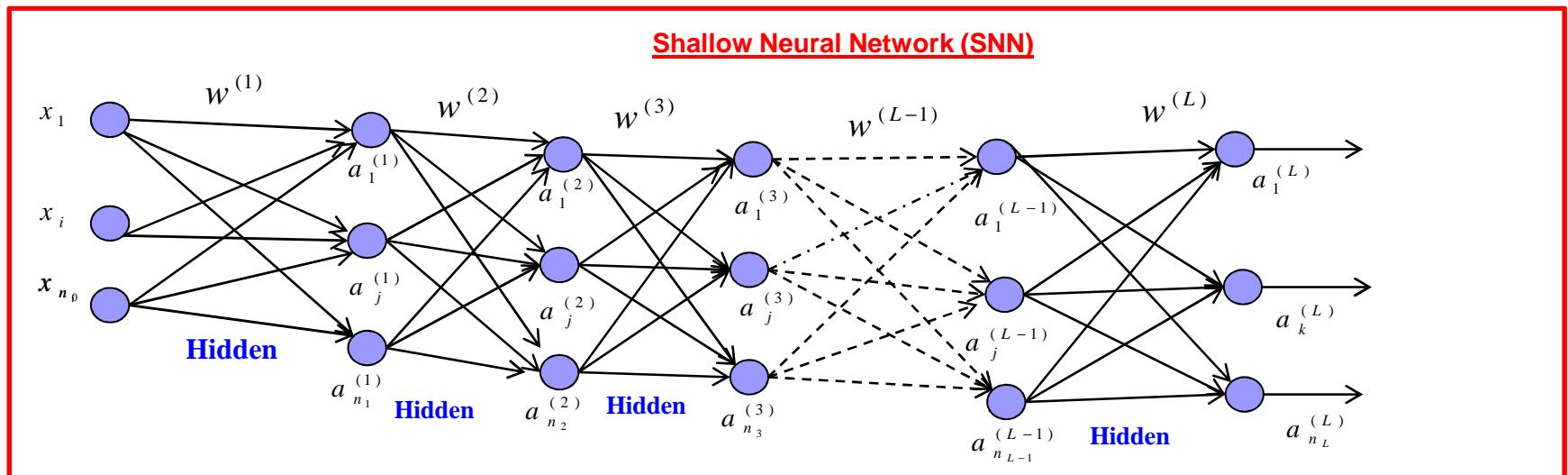
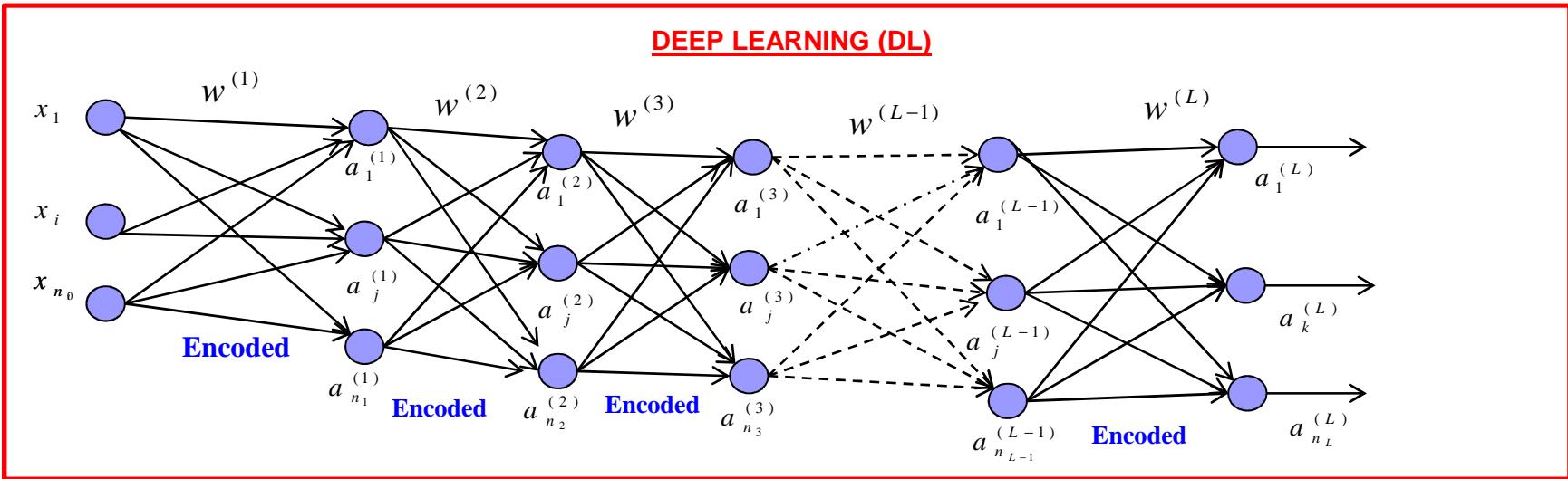
n_0 = Número de nodo de la Capa Entrada

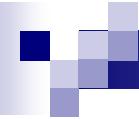
n_i = Número de nodo de la Capa Oculta i -ésima

n_L = Número de nodo de la Capa Salida

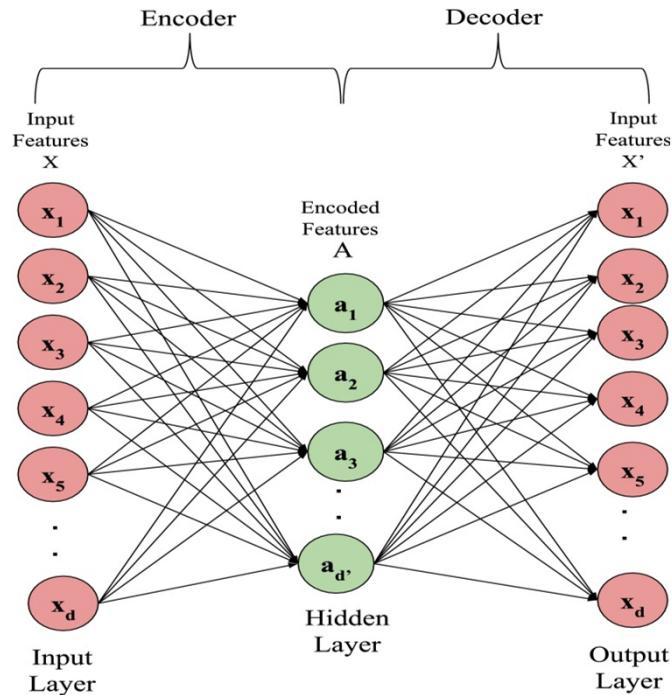
- Es un «nuevo» Multi-Layer Perceptron
- Cuál es la diferencia con Antiguo MLP ?

Multi-layer Perceptron:



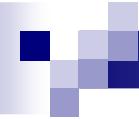


Unidad Básica : AUTO-ENCODER (AE)



Nodos Ocultos

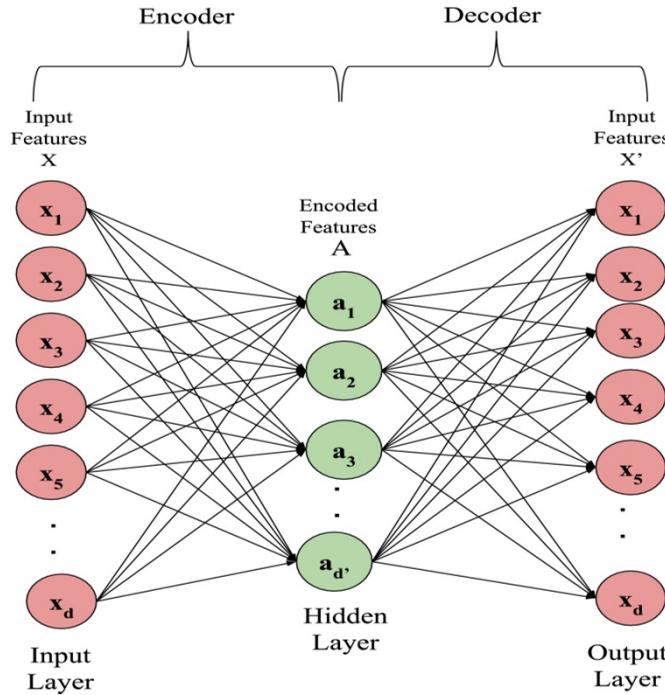
- (1) $n_h < n_i$
- (2) $n_h = n_i$
- (3) $n_h > n_i$



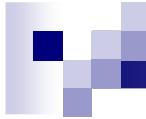
Proceso Deep Learning:

- **FASE 1: PRE-TRANING**
 - Aprendizaje No-supervisado (*auto-supervisado*)
 - Optimización de pesos de Auto-encoder.
- **FASE 2: FINE-TUNING**
 - Aprendizaje Supervisado
 - Optimización de todos los pesos del DL

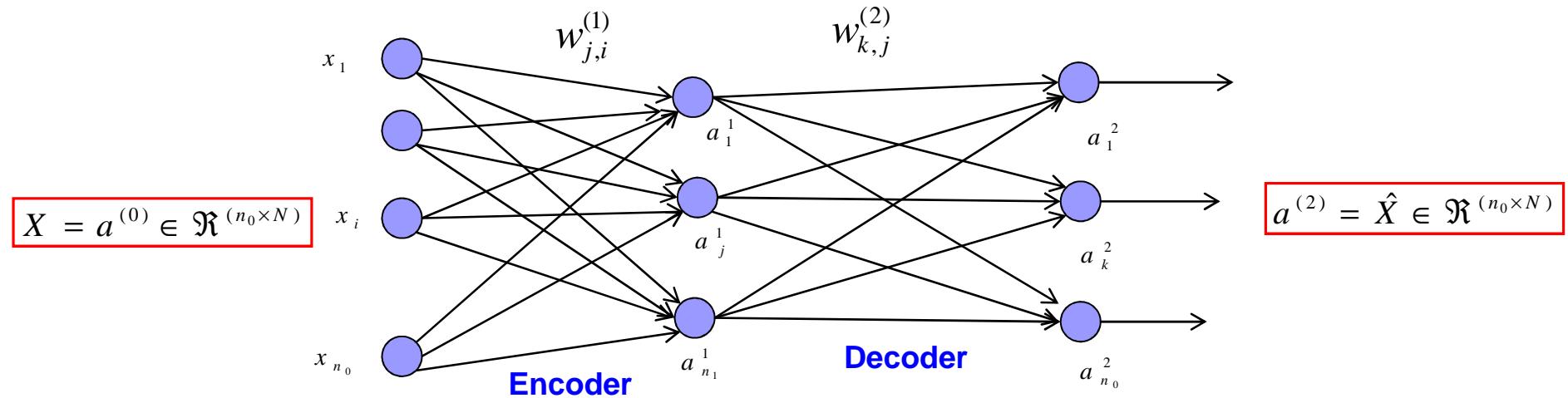
FASE 1: Entrenamiento Auto-Encoder (AE).



- **STEP 1: Pesos Decoder:**
 - Algoritmo Pseudo-inversa.
 - Algoritmo Back-Propagation.
- **STEP 2: Pesos Encoder :**
 - Algoritmo Back-Propagation.



Estructura de AE:



$$\begin{aligned} z^{(1)} &= w^{(1)} \times a^{(0)} \\ a^{(1)} &= f(z^{(1)}) \\ a^{(1)} &= \frac{1}{1 + \exp(-z^{(1)})} \end{aligned}$$

Activación
Oculta
No-Lineal

$$\begin{aligned} z^{(2)} &= w^{(2)} \times a^{(1)} \\ a^{(2)} &= f(z^{(2)}) \\ a^{(2)} &= z^{(2)} \end{aligned}$$

Activación
Salida
Lineal

Entrenamiento Pesos Decoder:

- Pesos Encoder: inicializados con valores aleatorios

$$r = \sqrt{\frac{6}{n_1 + n_0}}$$
$$w^{(1)} = rand(n_1, n_0) \times 2 \times r - r$$

Salida
Encoder

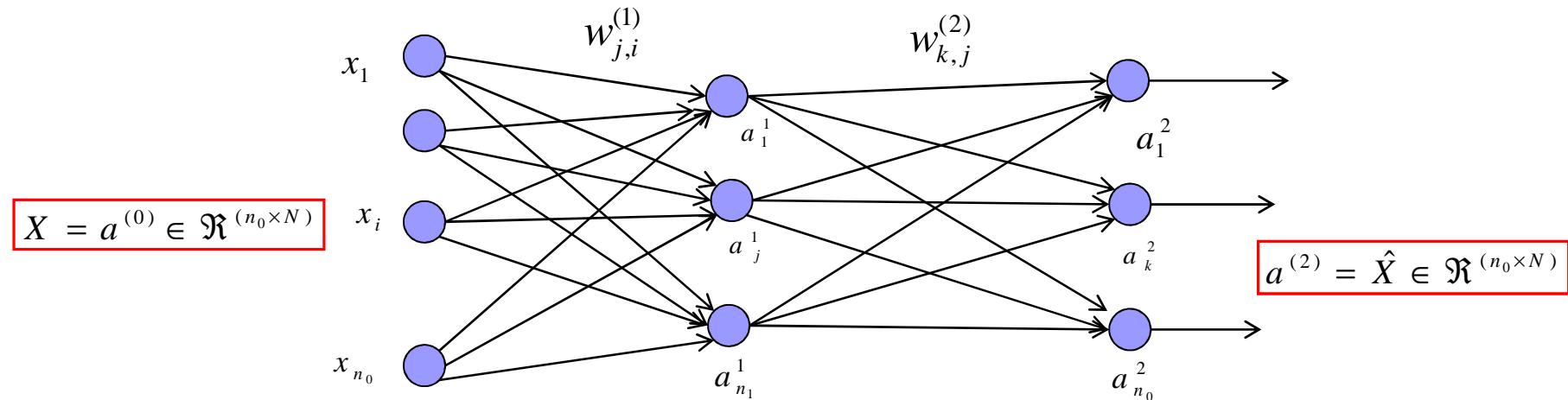
$$z^{(1)} = w^{(1)} \times a^{(0)}$$
$$H = \frac{1}{1 + \exp(-z^{(1)})}$$

Pesos
Decoder

$$w^{(2)} = X \times H^T \times \left(H \times H^T + \frac{I}{C} \right)^{-1}$$

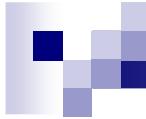
Entrenamiento Pesos Encoder:

- Algoritmo Back-propagation : Descenso del Gradiente



$$E = \frac{1}{2} \sum_{n=1}^N C_n = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^{n_0} (a_{k,n}^{(2)} - x_{k,n})^2$$

Función de Costo



Gradiente Encoder:

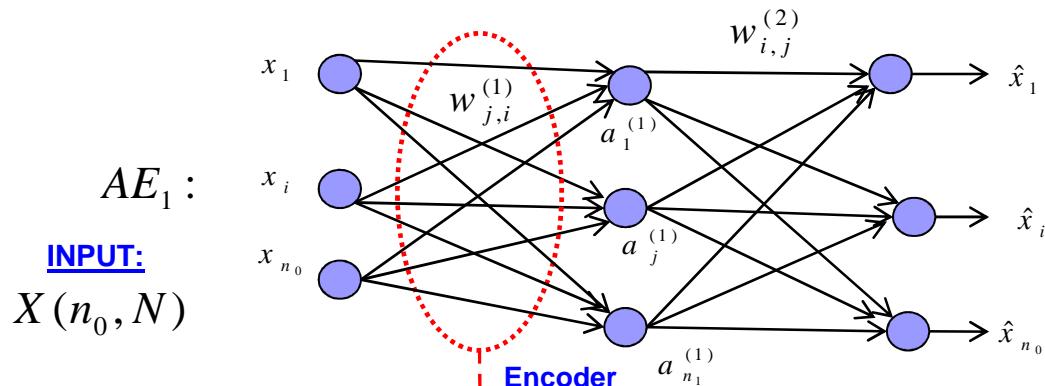
**Notación Matricial:
Gradiente**

$$\frac{\partial C}{\partial w^{(1)}} = \left\{ \left(w^{(2)} \right)^T \times e^{(2)} \right\} * f' \left(z^{(1)} \right) \times \left(a^{(0)} \right)^T$$

**Notación Matricial:
Actualización
Pesos Encoder**

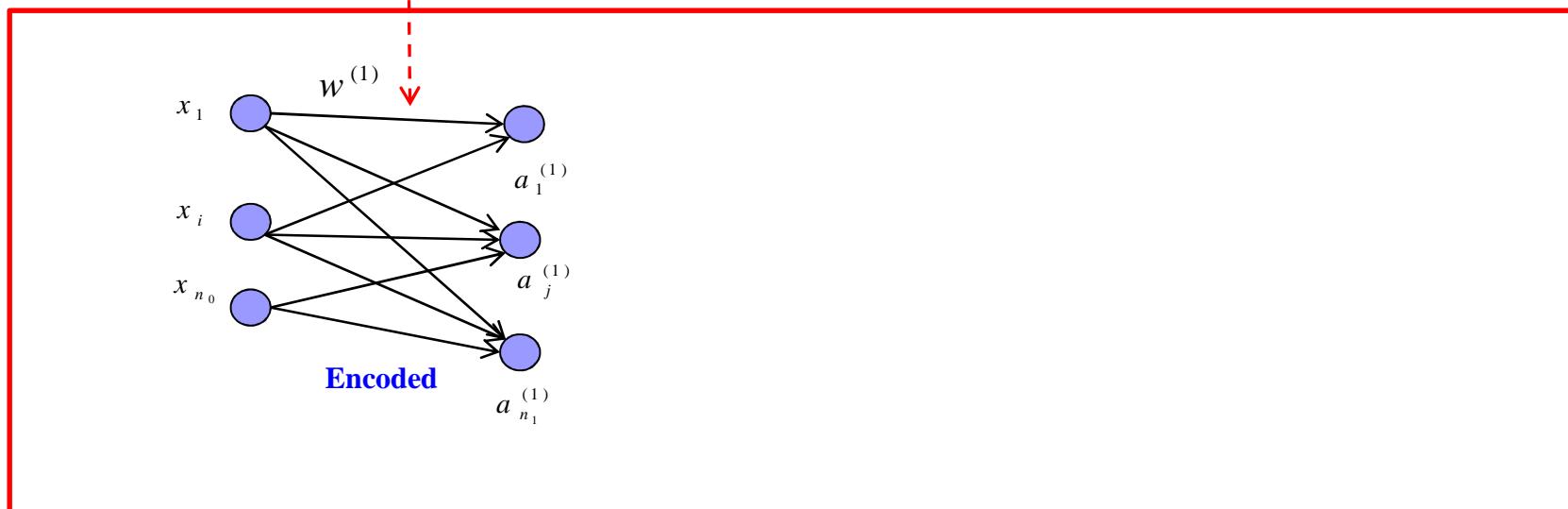
$$w^{(1)}(t) = w^{(1)}(t-1) - \mu \times \frac{\partial C}{\partial w^{(1)}}, \quad t = 1, \dots, MaxIter$$

Construcción de DL usando AE Apilados (AEA)



Steps:

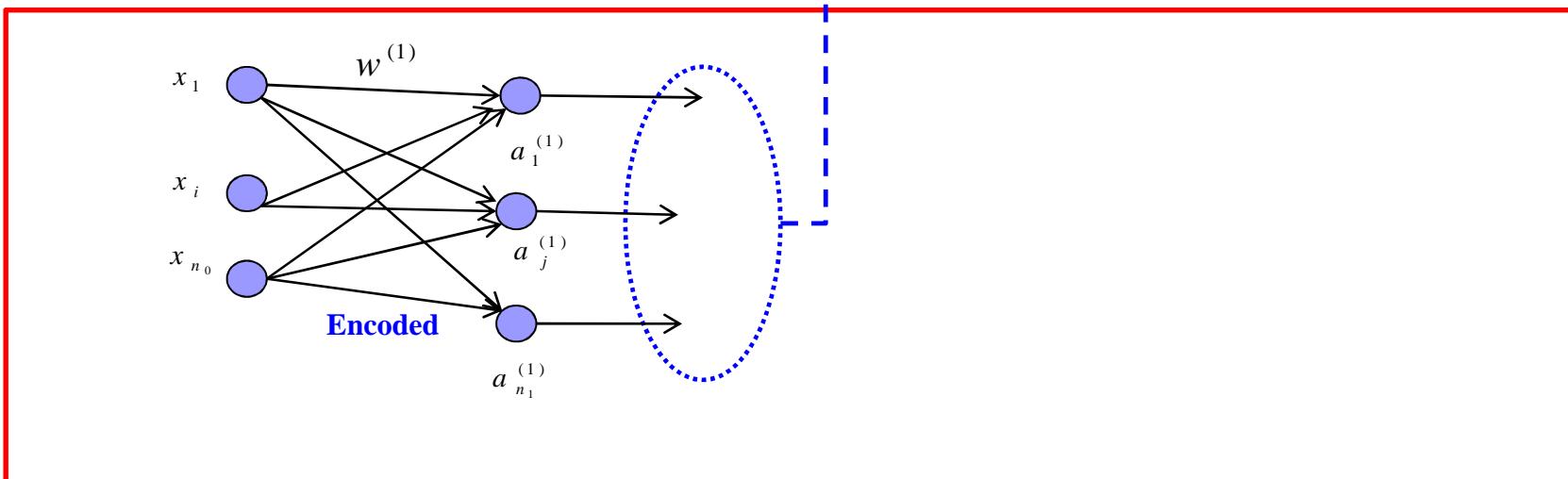
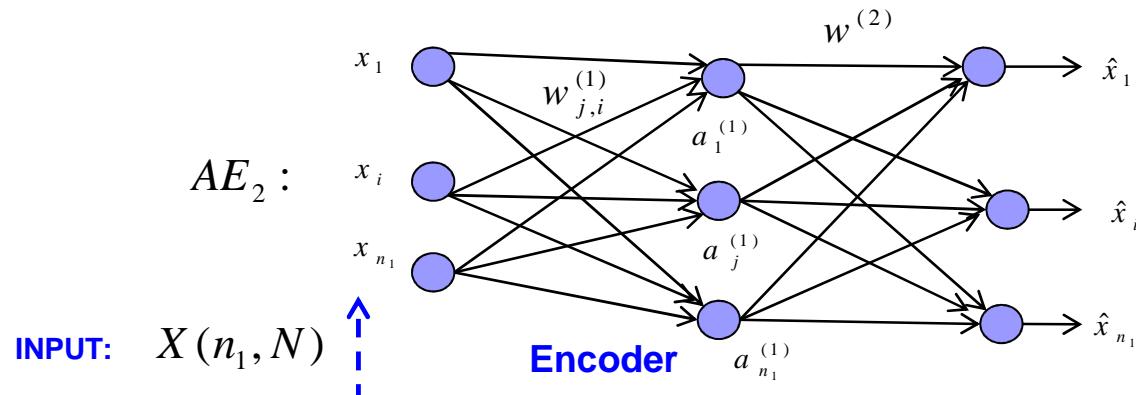
1. Entrenar el AE
2. Retornar los pesos del Encoder para diseñar la primera capa del DL



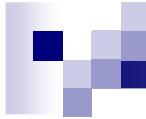
DEEP LEARNING



DL usando AE

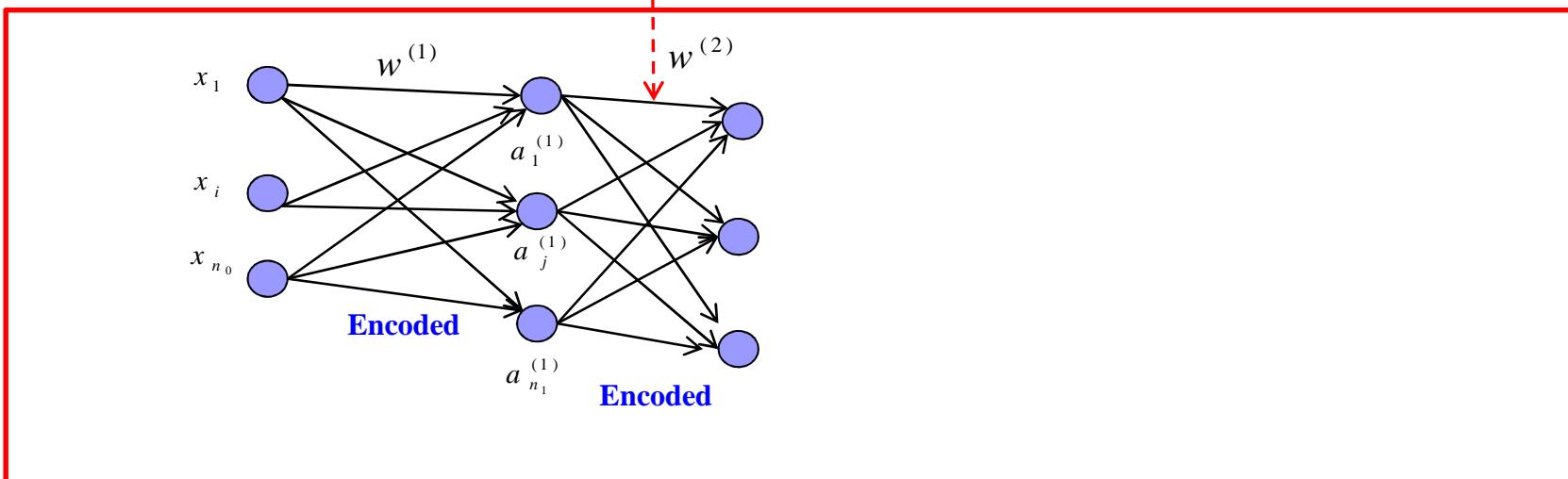
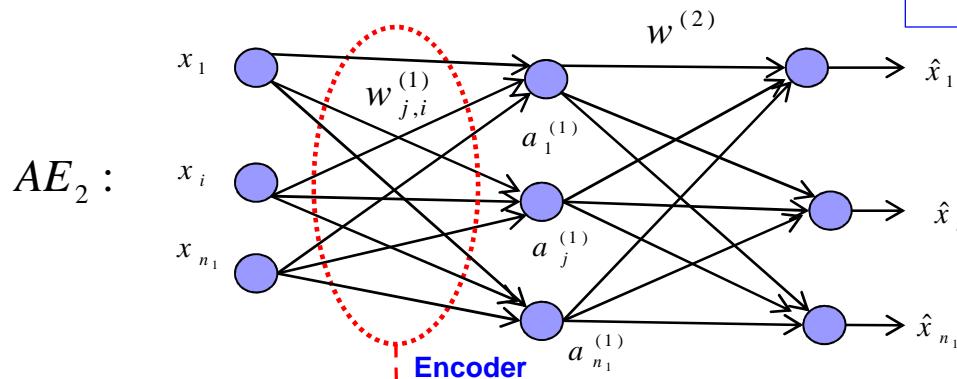


DEEP LEARNING



DL usando AE

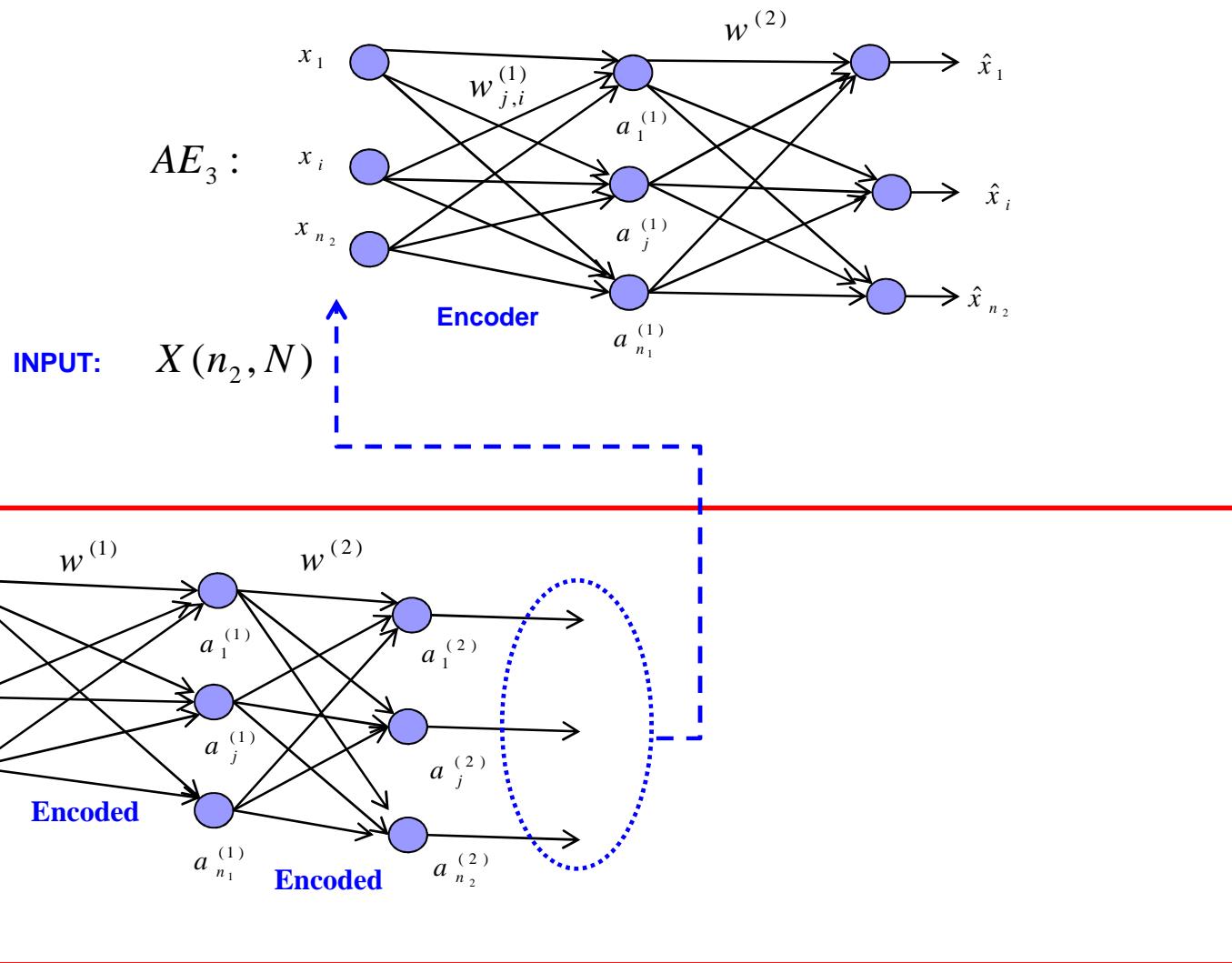
1. Entrenar el AE
2. Retornar los pesos del Encoder para diseñar la segunda capa del DL

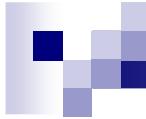


DEEP LEARNING



DL usando AE

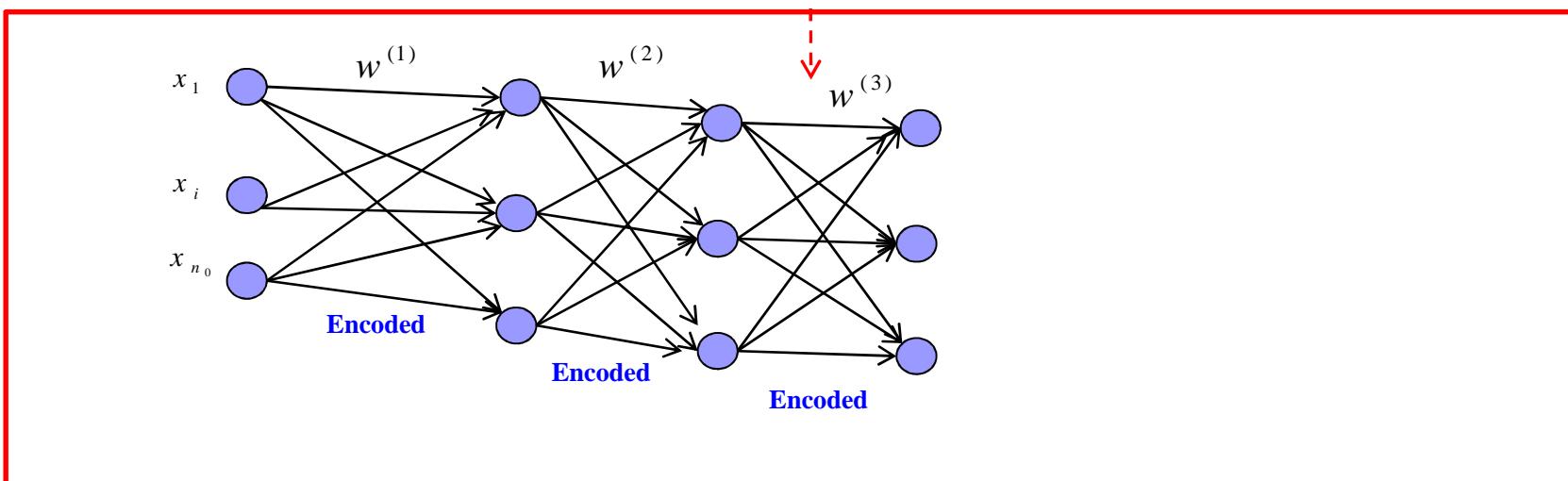
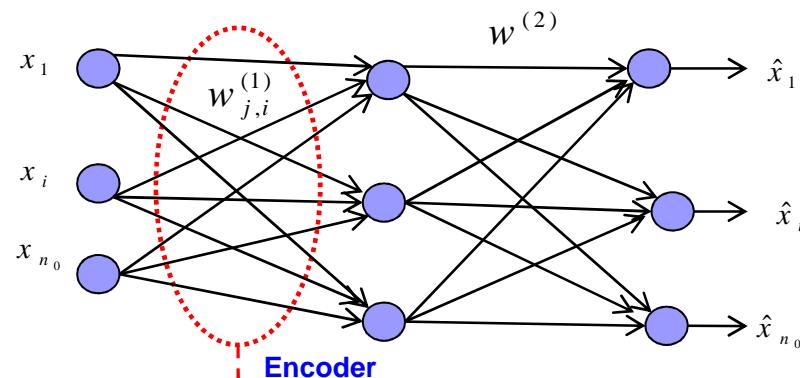




DL usando AE

- Entrenar el AE
- Retornar los pesos del Encoder para diseñar la tercera capa del DL

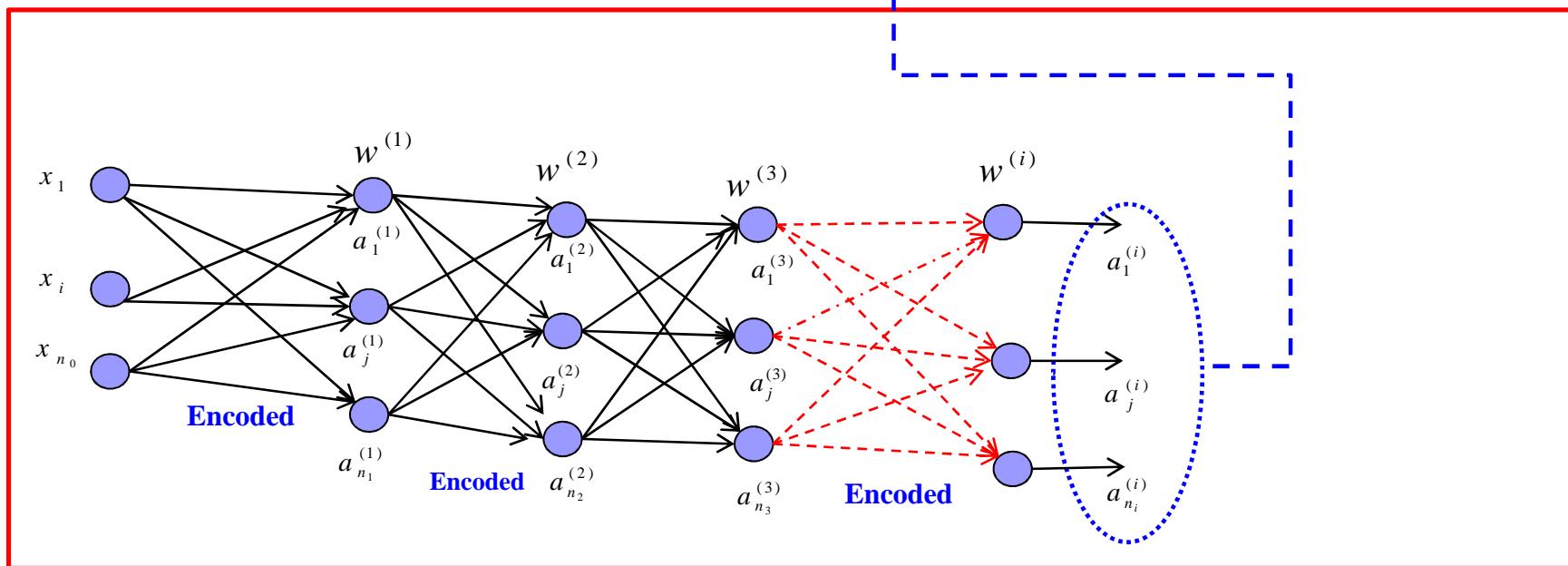
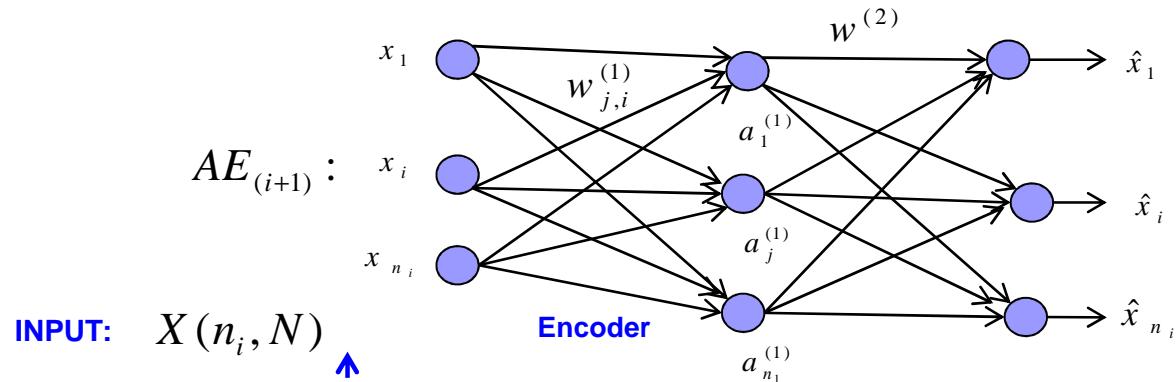
$AE_3 :$



DEEP LEARNING



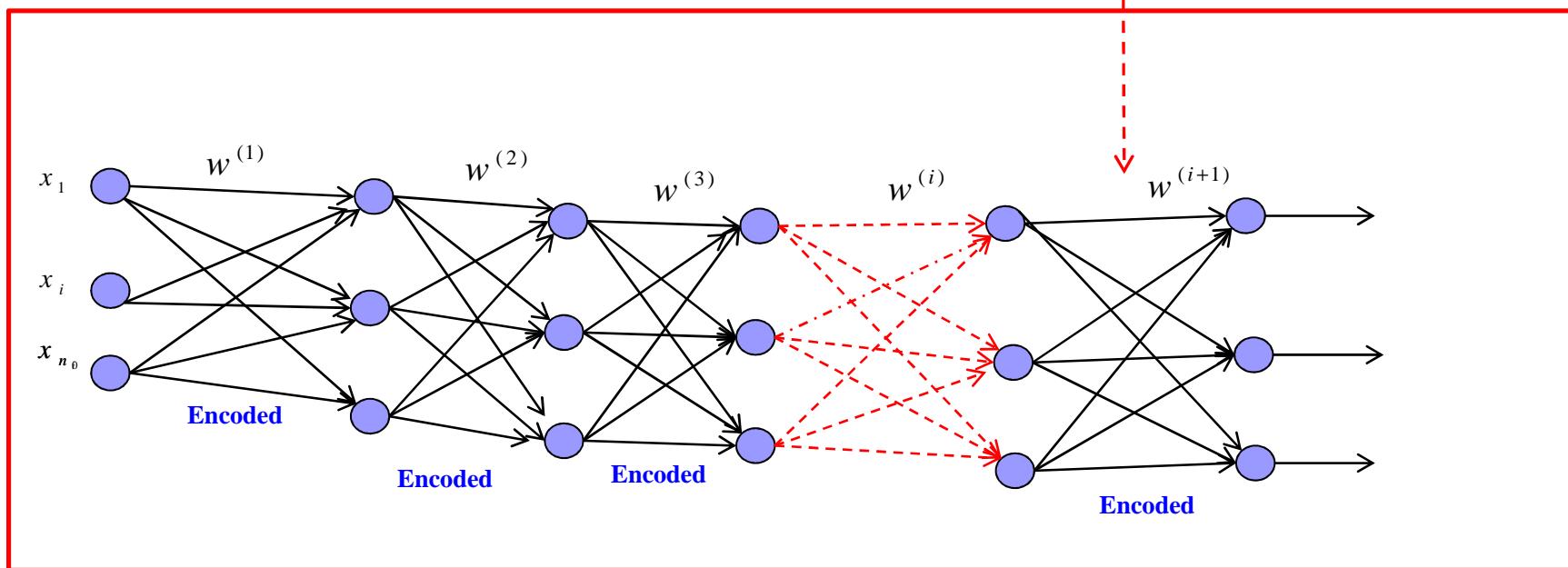
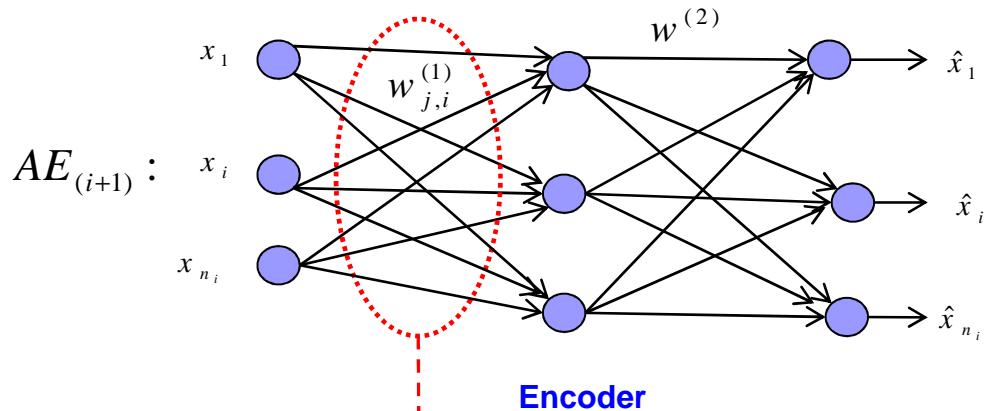
DL usando AE



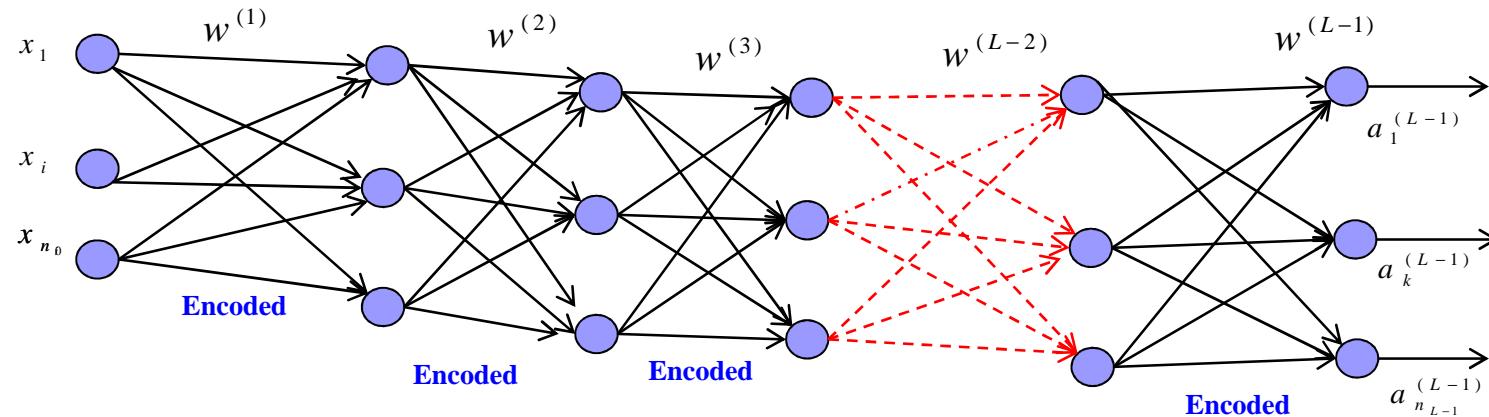


DL usando AE

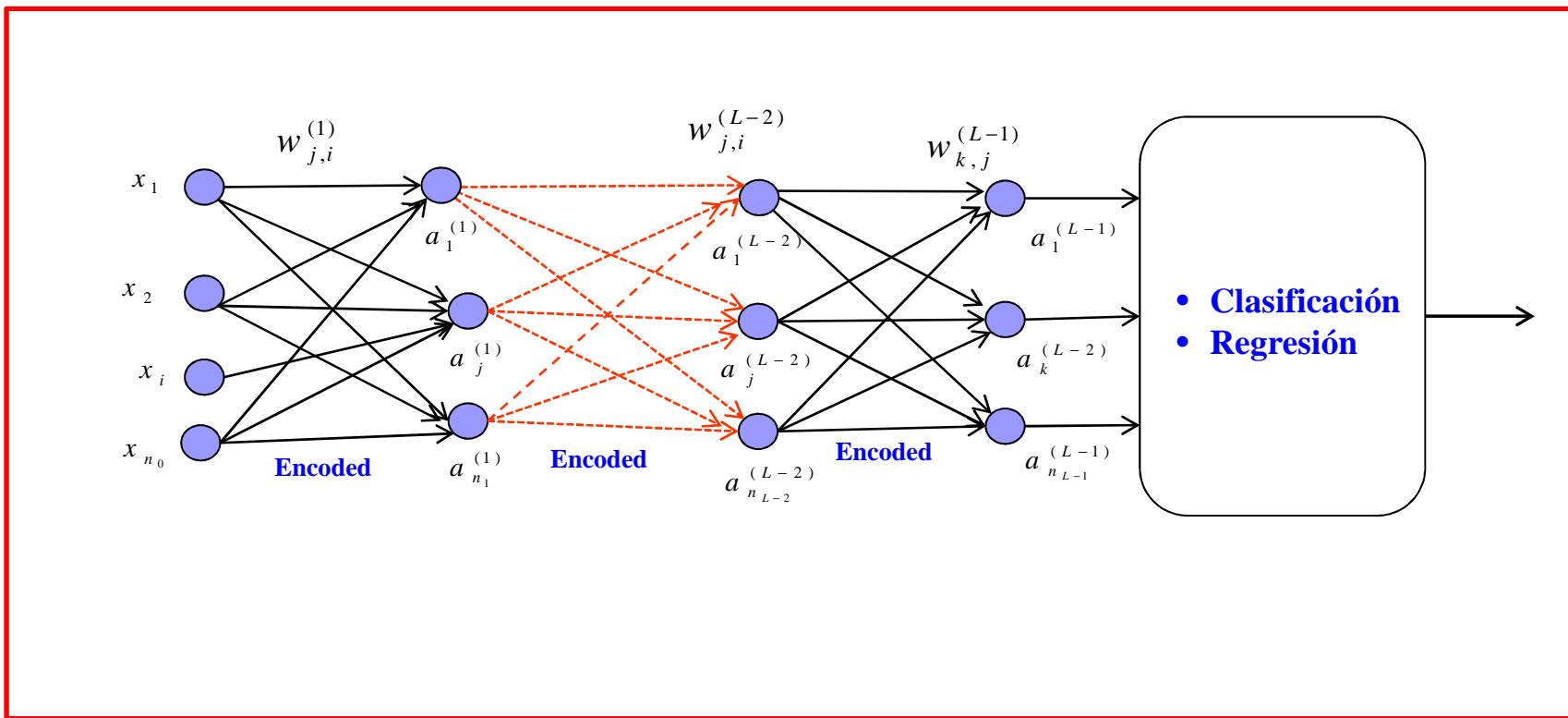
- Entrenar el AE
- Retornar los pesos del Encoder para diseñar la $(i+1)$ -ésima capa del DL



Construcción de DL usando AE Apilados

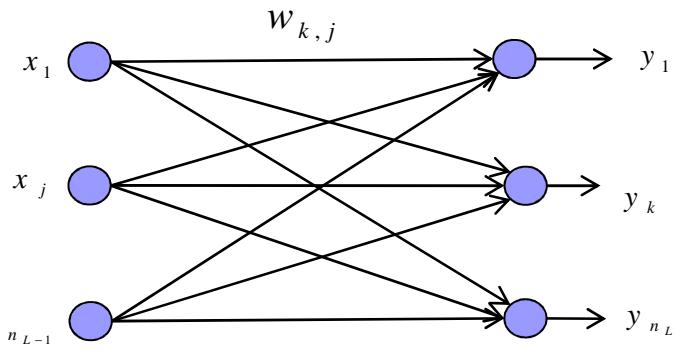


Deep Learning

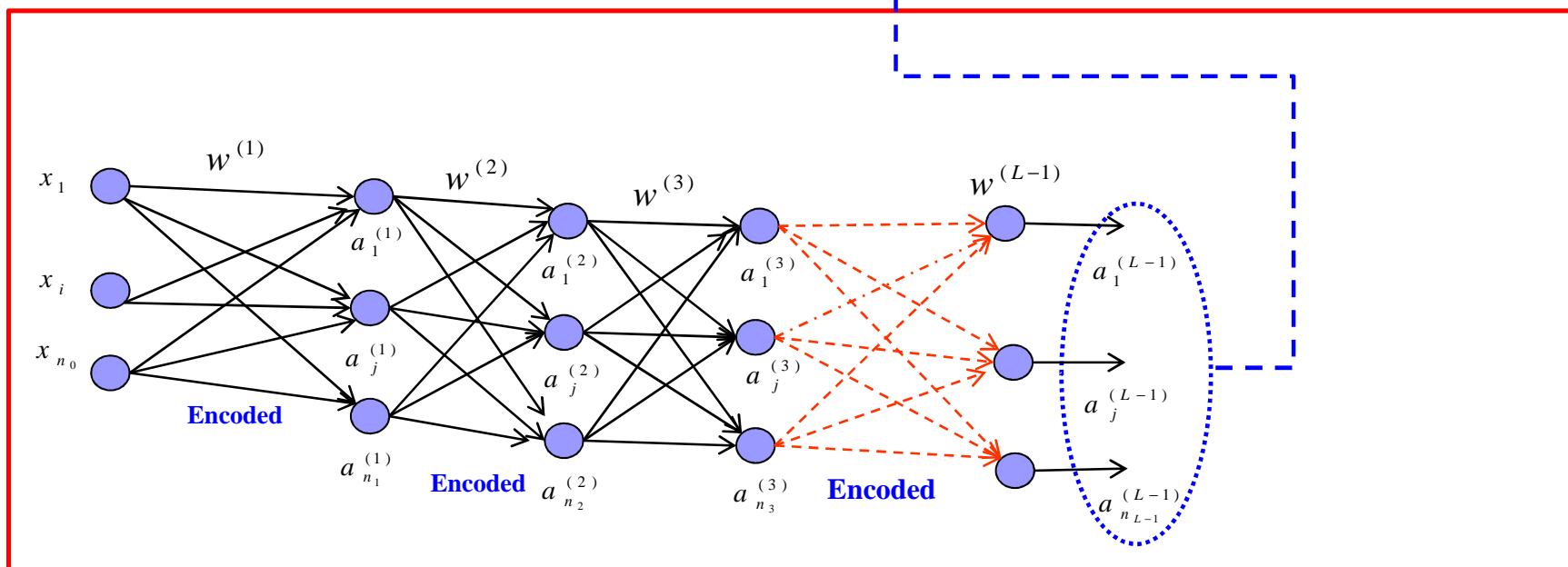


Clasificador Softmax:

Número de Clases: n_L
(nodos salidas)



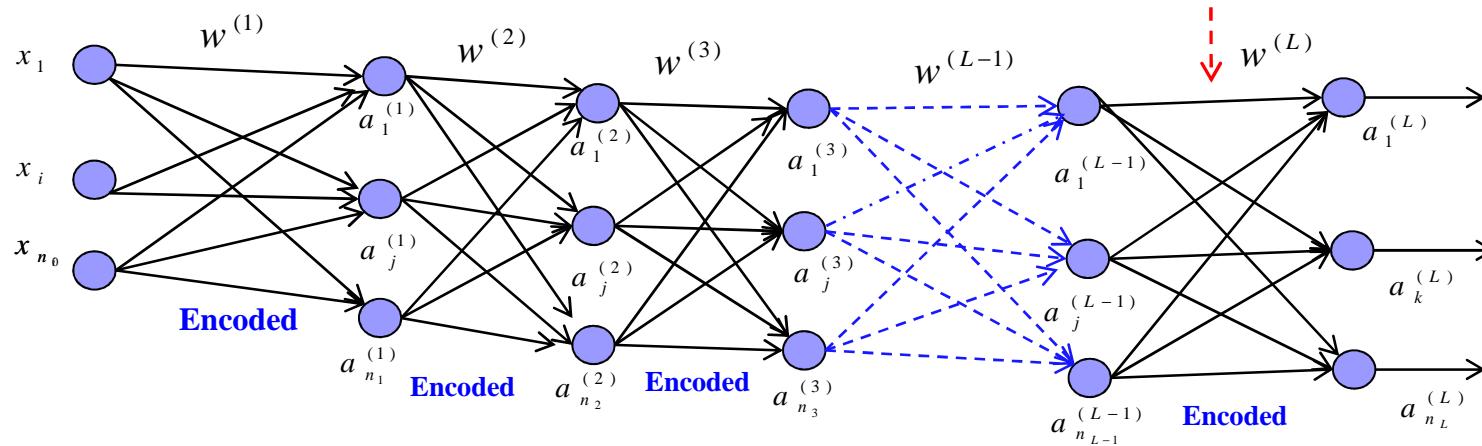
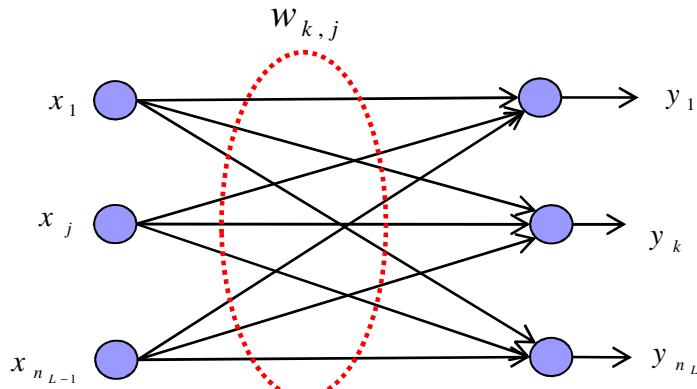
INPUT: $X(n_{L-1}, N)$



DEEP LEARNING

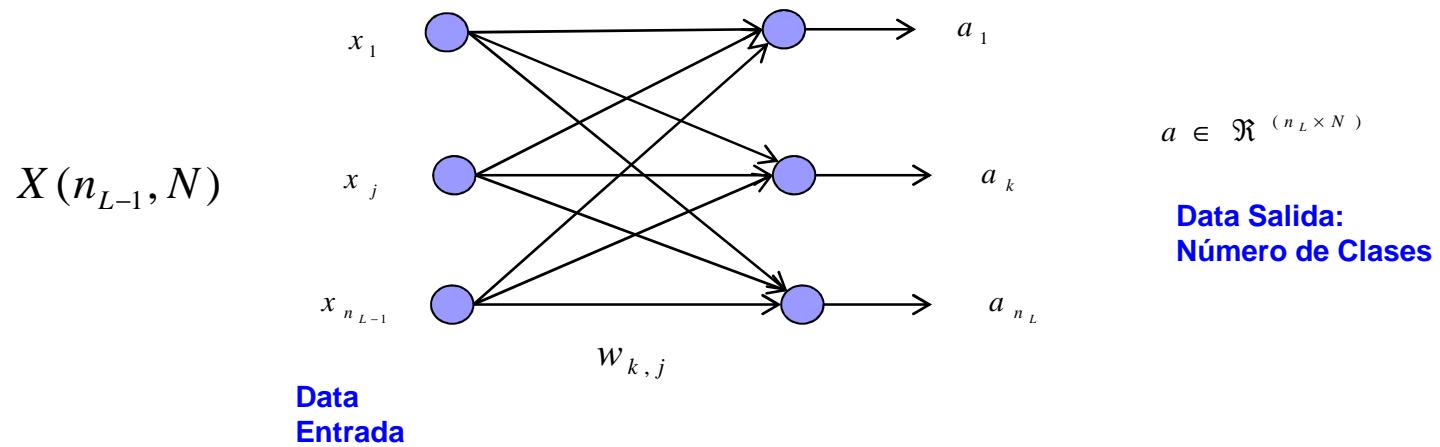
Deep Learning+ Clasificador Softmax:

- Entrenar el Softmax
- Retornar los pesos del Softmax para diseñar la **L-ésima** capa del DL



DEEP LEARNING

Clasificación Softmax: (Feed-forward)



Activación Softmax
de la n -ésima muestra

$$\begin{aligned} z &= w \times X \\ z &= \exp(z) \\ a_n &= \frac{z(:, n)}{\sum_{k=1}^{n_L} z_{k,n}}, \quad n = 1, \dots, N \end{aligned}$$

Clasificador Softmax: Entrenamiento

Entropía Cruzada:

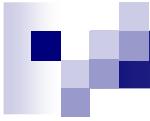
$$Cost = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^{n_L} T_{k,n} \log(a_{k,n}) + \frac{\lambda}{2} \|w\|_2^2$$

- N : número de muestras, n_L : numero de clases,
- $T_{k,n}$: valores deseados (etiquetas), λ : penalidad de pesos.

Notación Matricial :
Modificación de W

$$w(k) = w(k-1) - \mu \frac{\partial Cost}{\partial w}, \quad k = 1, \dots, MaxIter$$

$$\frac{\partial Cost}{\partial w} = -\frac{1}{N} \left((T - A) \times X^T \right) + \lambda \times w(k-1)$$



Clasificación Multi-clases: Codificación de Etiquetas Binarias

- Número de Clases: 3

$$\text{Class 1} = [1 \quad 0 \quad 0]$$

$$\text{Class 2} = [0 \quad 1 \quad 0]$$

$$\text{Class 3} = [0 \quad 0 \quad 1]$$

- Número de Clases: 4

$$\text{Class 1} = [1 \quad 0 \quad 0 \quad 0]$$

$$\text{Class 2} = [0 \quad 1 \quad 0 \quad 0]$$

$$\text{Class 3} = [0 \quad 0 \quad 1 \quad 0]$$

$$\text{Class 4} = [0 \quad 0 \quad 0 \quad 1]$$

- Número de Clases: 5

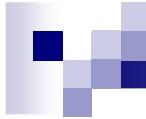
$$\text{Class 1} = [1 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\text{Class 2} = [0 \quad 1 \quad 0 \quad 0 \quad 0]$$

$$\text{Class 3} = [0 \quad 0 \quad 1 \quad 0 \quad 0]$$

$$\text{Class 4} = [0 \quad 0 \quad 0 \quad 1 \quad 0]$$

$$\text{Class 5} = [0 \quad 0 \quad 0 \quad 0 \quad 1]$$



Clasificación Multi-clases: Codificación de Etiquetas Binarias

- Número de Clases: 10

$Class \ 1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$
$Class \ 2 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$
$Class \ 3 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$
$Class \ 4 = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$
$Class \ 5 = [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$
$Class \ 6 = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$
$Class \ 7 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$
$Class \ 8 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]$
$Class \ 9 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]$
$Class \ 10 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]$

Métricas de Rendimiento: 2-clases

Matriz de Confusión (CM)

		Deseado Valor	
		P	N
P R E D I CHO	P	TP	FP
	N	FN	TN

$$R = \frac{TP}{TP + FN}$$

$$P = \frac{TP}{TP + FP}$$

$$F = 2 \times \frac{P \times R}{P + R}$$

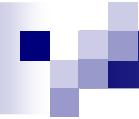
$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

Recall
(TPR))

Precision
(precisión)

F-score

Accuracy
(exactitud)



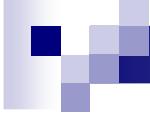
Métricas de Rendimeinto : Multi-Clases

$$\text{Re call } (j) = \frac{CM_{j,j}}{\sum_{i=1}^{n_L} CM_{i,j}}, \quad j = 1, \dots, n_L$$

$$\text{Precision } (i) = \frac{CM_{i,i}}{\sum_{j=1}^{n_L} CM_{i,j}}, \quad i = 1, \dots, n_L$$

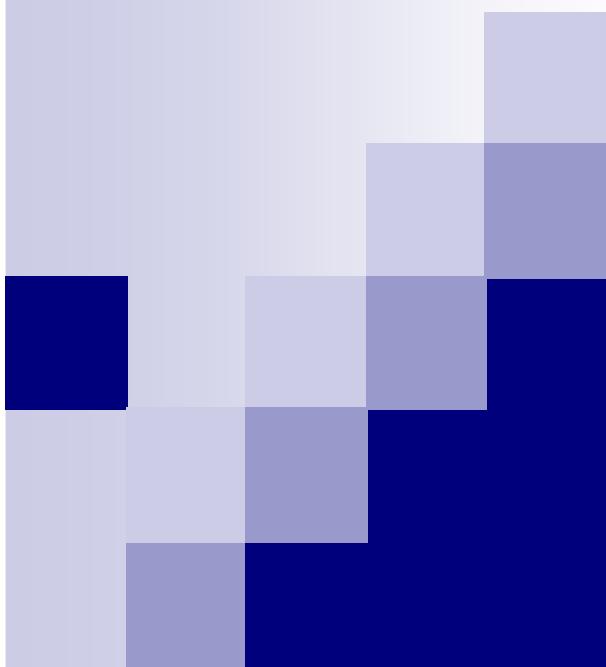
$$\text{F - score } (j) = 2 \times \frac{\text{Pr ecison } (j) \times \text{Re call } (j)}{\text{Pr ecision } (j) + \text{Re call } (j)}$$

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{n_L} CM_{i,i}$$



TRABAJO 2^a

Deep Learning: Clasificación



CONTINUARÁ....