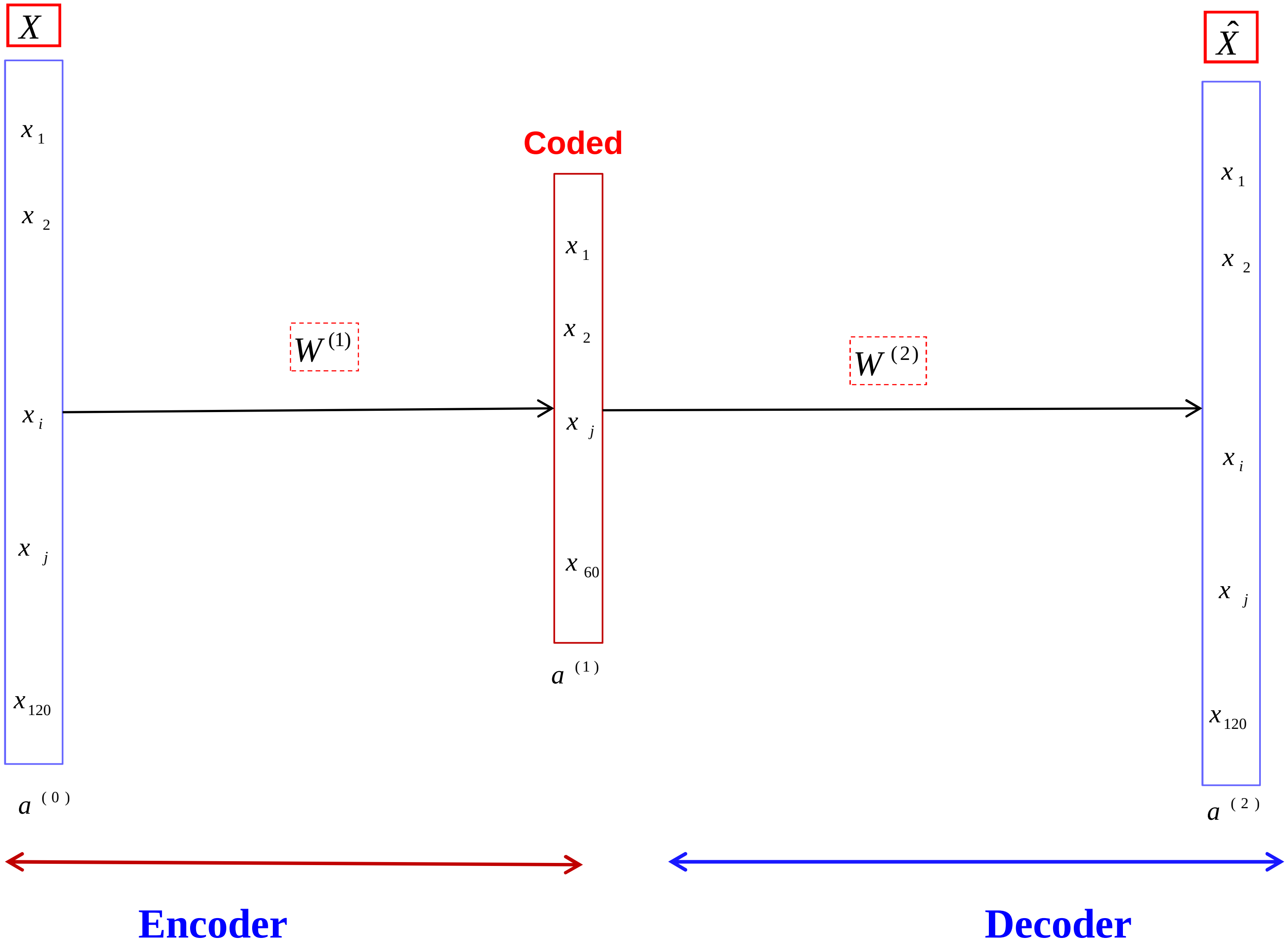
DEEP LEARNING

Deep AutoEncoder (DAE)

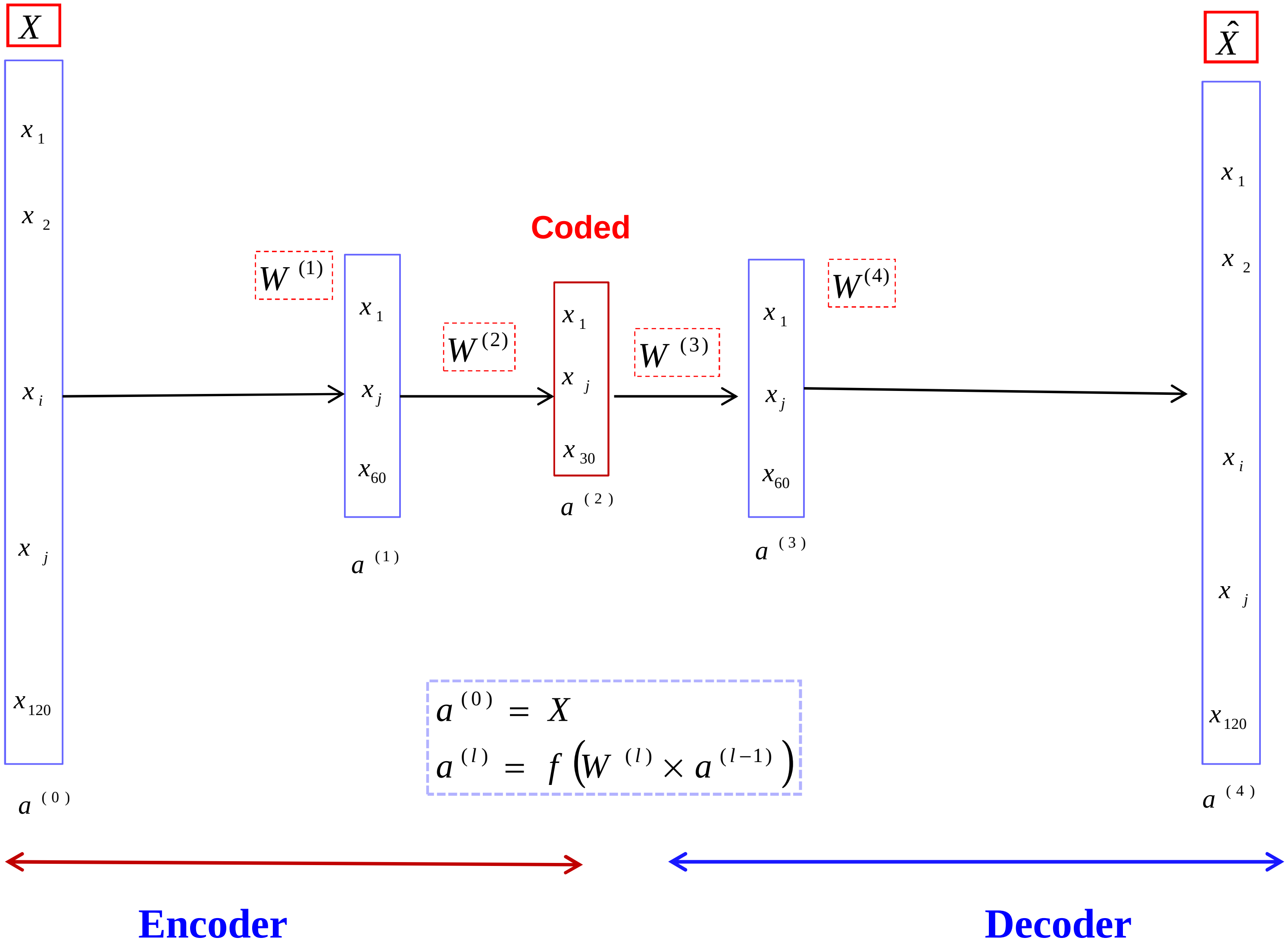
+

Algoritmo de Aprendizaje MiniBatch

AutoEncoder Mono-Capa

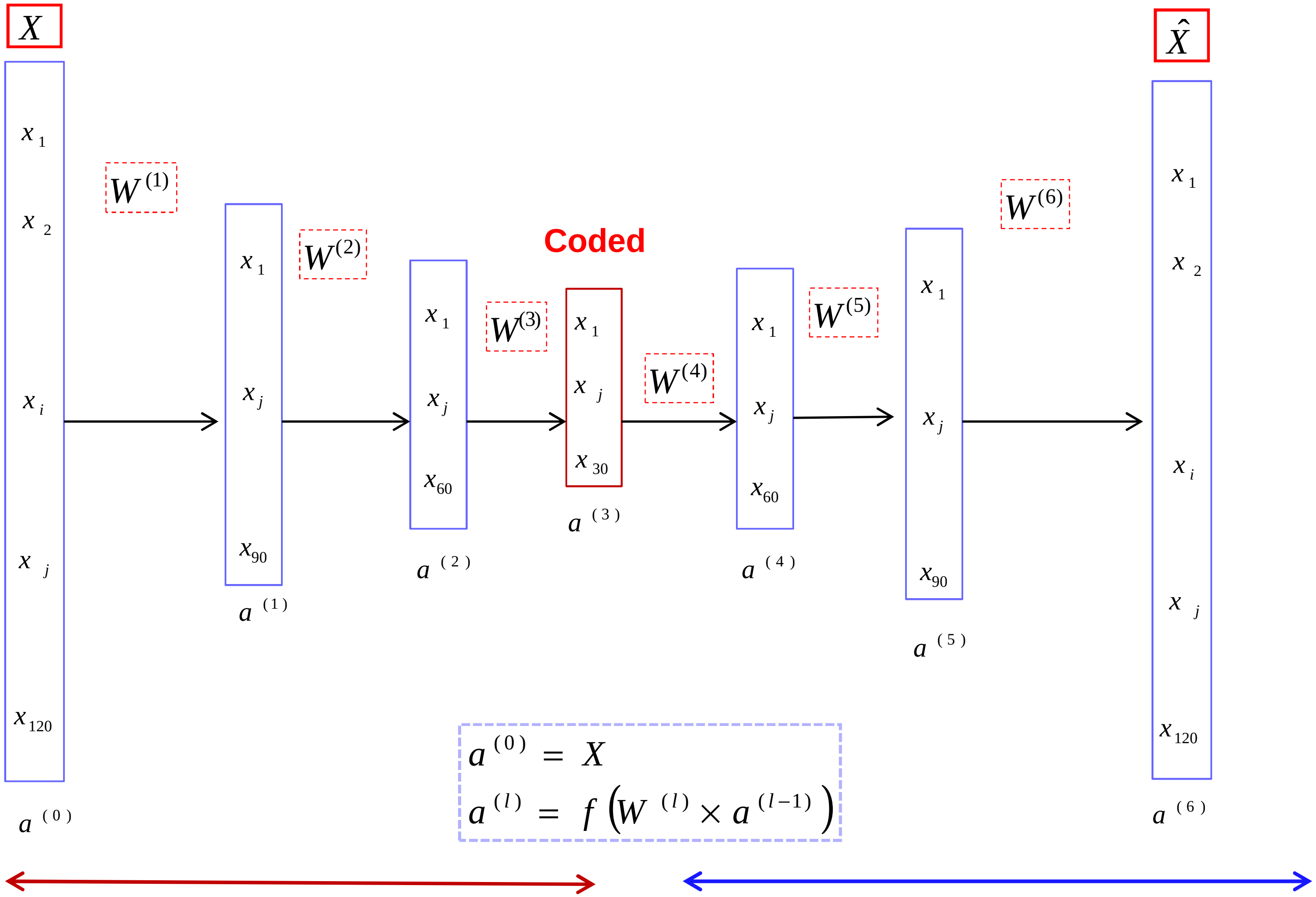


Deep AutoEncoder (DAE)



$$a^{(0)} = X$$
$$a^{(l)} = f\left(W^{(l)} \times a^{(l-1)}\right)$$

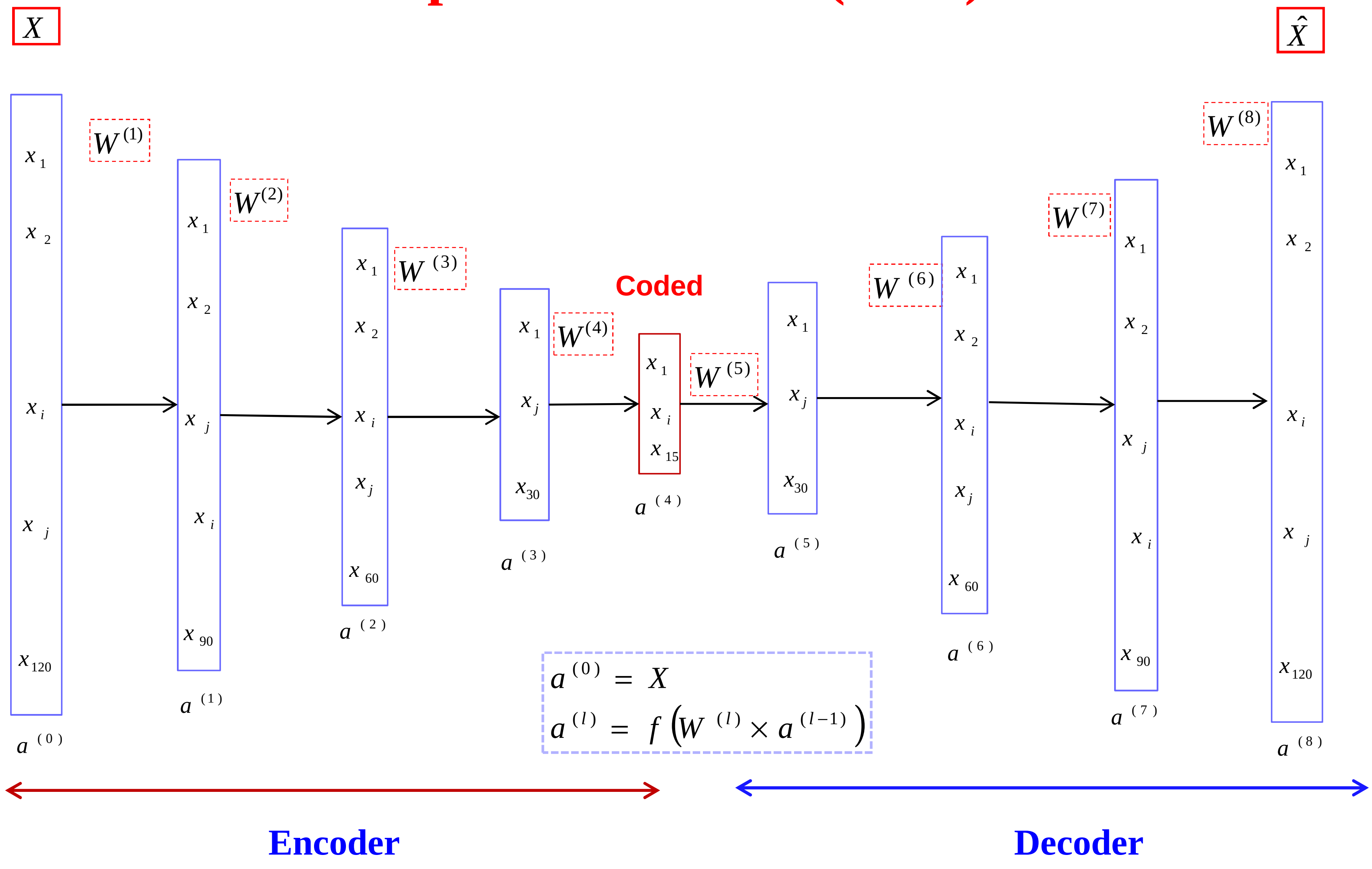
Deep AutoEncoder (DAE)



$$a^{(0)} = X$$
$$a^{(l)} = f\left(W^{(l)} \times a^{(l-1)}\right)$$

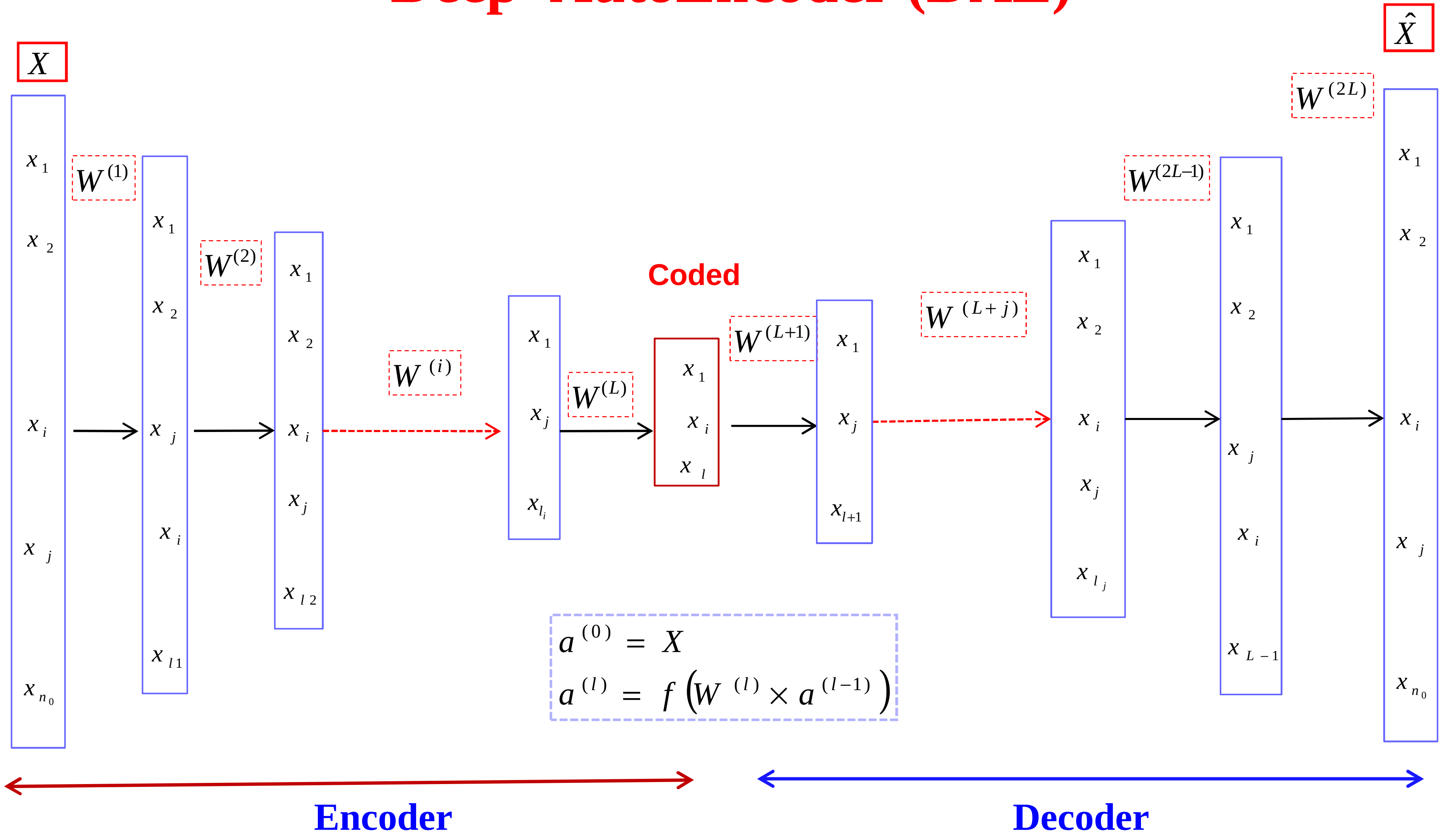
Encoder **Decoder**

Deep AutoEncoder (DAE)



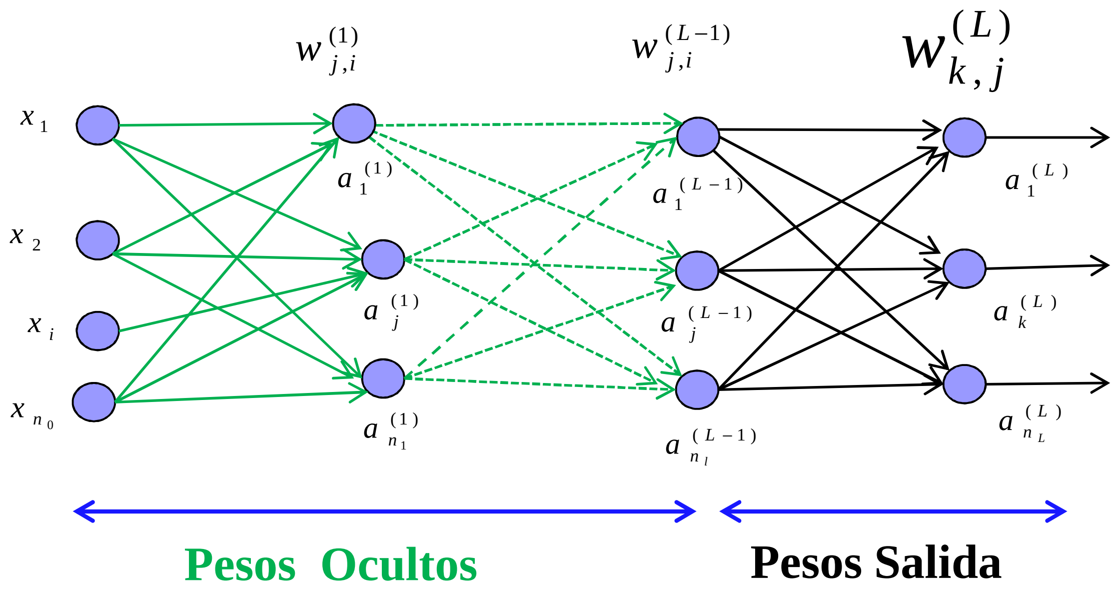
$$a^{(0)} = X$$
$$a^{(l)} = f\left(W^{(l)} \times a^{(l-1)}\right)$$

Deep AutoEncoder (DAE)



$$a^{(0)} = X$$
$$a^{(l)} = f\left(W^{(l)} \times a^{(l-1)}\right)$$

Topología de DEA := (MLP)



$$
\begin{aligned}
a^{(0)} &= X \\
z^{(l)} &= W^{(l)} \times a^{(l-1)} \\
a^{(l)} &= f\left(z^{(l)}\right)
\end{aligned}
$$

$$
\begin{aligned}
z^{(L)} &= W^{(L)} \times a^{(L-1)} \\
a^{(L)} &= f\left(z^{(L)}\right)
\end{aligned}
$$

Aprendizaje Adaptativo MiniBatch:

Algoritmo ADAM Mejorado (mADAM)

Training DAE: MiniBatch-mADAM

- Considerar una base de datos de *N*-muestras dada como:

$$\boxed{\{X_i, Y_i\}_{i=1}^{N}, \quad X_i \in \Re^d, \; Y_i \in \Re^m}$$

- X_i : representa la data de training
- Y_i: representa la data deseada
- d: denota el número de variables de entrada
- m: denota el número de salidas

Training DAE: MiniBatch-mADAM

- **Paso 1**: Re-ordenar aleatoriamente la localización de cada muestra de la base de datos de training.

- **Paso 2**: Dividir las ***N***-muestras de la base de datos en Batchs de ***M***-muestras.

$$\boxed{\begin{array}{l} B = \dfrac{N}{M} \\ B : \text{Número de Batch} \end{array}}$$

- **Paso 3**: Entrenar el DAE usando un Número Máximo de Épocas.
  - Cada Épocas ajusta los pesos del DAE ***B***-veces vía el **mADAM**.

Ajuste Pesos de DAE: MiniBatch-mADAM

Costo (MSE):

$$E = \frac{1}{2M}\sum_{n=1}^{M} C_n = \frac{1}{2M}\sum_{n=1}^{M}\sum_{k=1}^{n_L}\left(a_{k,n}^{(L)} - a_{k,n}^{(0)}\right)^2$$

Notación Matricial:
Gradiente
Pesos de Salida

$$\frac{\partial E}{\partial w^{(L)}} = \delta^{(L)} \times \left(a^{(L-1)}\right)^T$$

$$\delta^{(L)} = e^{(L)} \otimes f'\left(z^{(L)}\right)$$

Notación Matricial:
Gradiente
Pesos Ocultos

$$\frac{\partial E}{\partial w^{(l)}} = \left\{\left(w^{(l+1)}\right)^T \times \delta^{(l+1)}\right\} \otimes f'\left(z^{(l)}\right) \times \left(a^{(l-1)}\right)^T$$

$$l = (L-1), (L-2), \ldots, 3, 2, 1$$

Training DAE: miniBatch mADAM

Actualización Pesos de Salida

$$\boxed{\begin{aligned} & w^{(L)}(t) = w^{(L)}(t-1) - \mu \times gAdam \\ & t = 1,2,\ldots, MaxIter \\ & u \in (0,1) \end{aligned}}$$

Training DAE : miniBatch mADAM

Actualización Pesos de Salida

$$gAdam = \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \times \frac{V^{(L)}(t)}{\sqrt{S^{(L)}(t)} + \varepsilon}, \qquad \varepsilon = \left(10^{-6}, 10^{-8}\right)$$

$$V^{(L)}(t) = \beta_1 \times V^{(L)}(t-1) + (1-\beta_1)\frac{\partial E}{\partial w^{(L)}(t-1)}$$

$$S^{(L)}(t) = \beta_2 \times S^{(L)}(t-1) + (1-\beta_2)\left(\frac{\partial E}{\partial w^{(L)}(t-1)}\right)^2$$

$$w^{(L)}(0) = \text{random}$$

$$V^{(L)}(0),\ S^{(L)}(0) = W^{(L)}.shape$$

$$V^{(L)}(0),\ S^{(L)}(0) = 0$$

$$\beta_1 = 0.9 \qquad \beta_2 = 0.999$$

Training DAE: miniBatch mADAM

Actualización Pesos Ocultos

$$\boxed{\begin{aligned} & w^{(l)}(t) = w^{(l)}(t-1) - \mu \times gAdam \\ & l = L-1, L-2, \ldots 2,1 \\ & t = 1,2,\ldots, MaxIter \end{aligned}}$$

Training DAE : miniBatch mADAM

Actualización Pesos Ocultos

$$gAdam = \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \times \frac{V^{(l)}(t)}{\sqrt{S^{(l)}(t)}+\varepsilon}$$

$$V^{(l)}(t) = \beta_1 V^{(l)}(t-1) + (1-\beta_1)\frac{\partial E}{\partial w^{(l)}(t-1)}$$

$$S^{(l)}(t) = \beta_2 S^{(l)}(t-1) + (1-\beta_2)\left(\frac{\partial E}{\partial w^{(l)}(t-1)}\right)^2$$

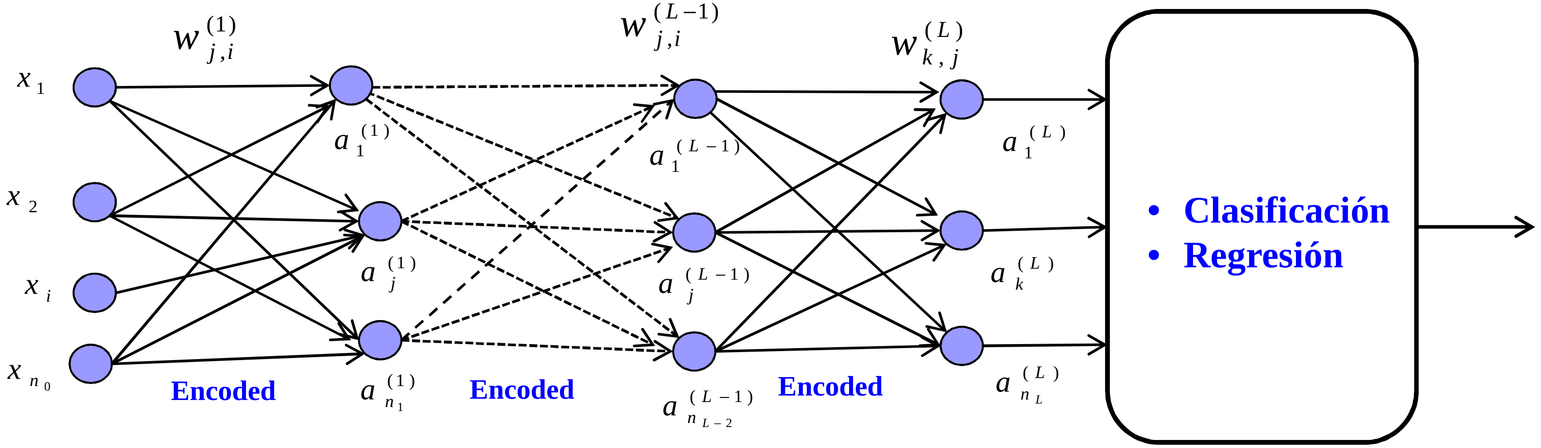
$$w^{(l)}(0) = \text{random} \quad , \; l = 1,2,\ldots,L-1$$

$$V^{(l)}(0),\; S^{(l)}(0) = w^{(l)}.\text{shape}$$

$$V^{(l)}(0),\; S^{(l)}(0) = 0$$

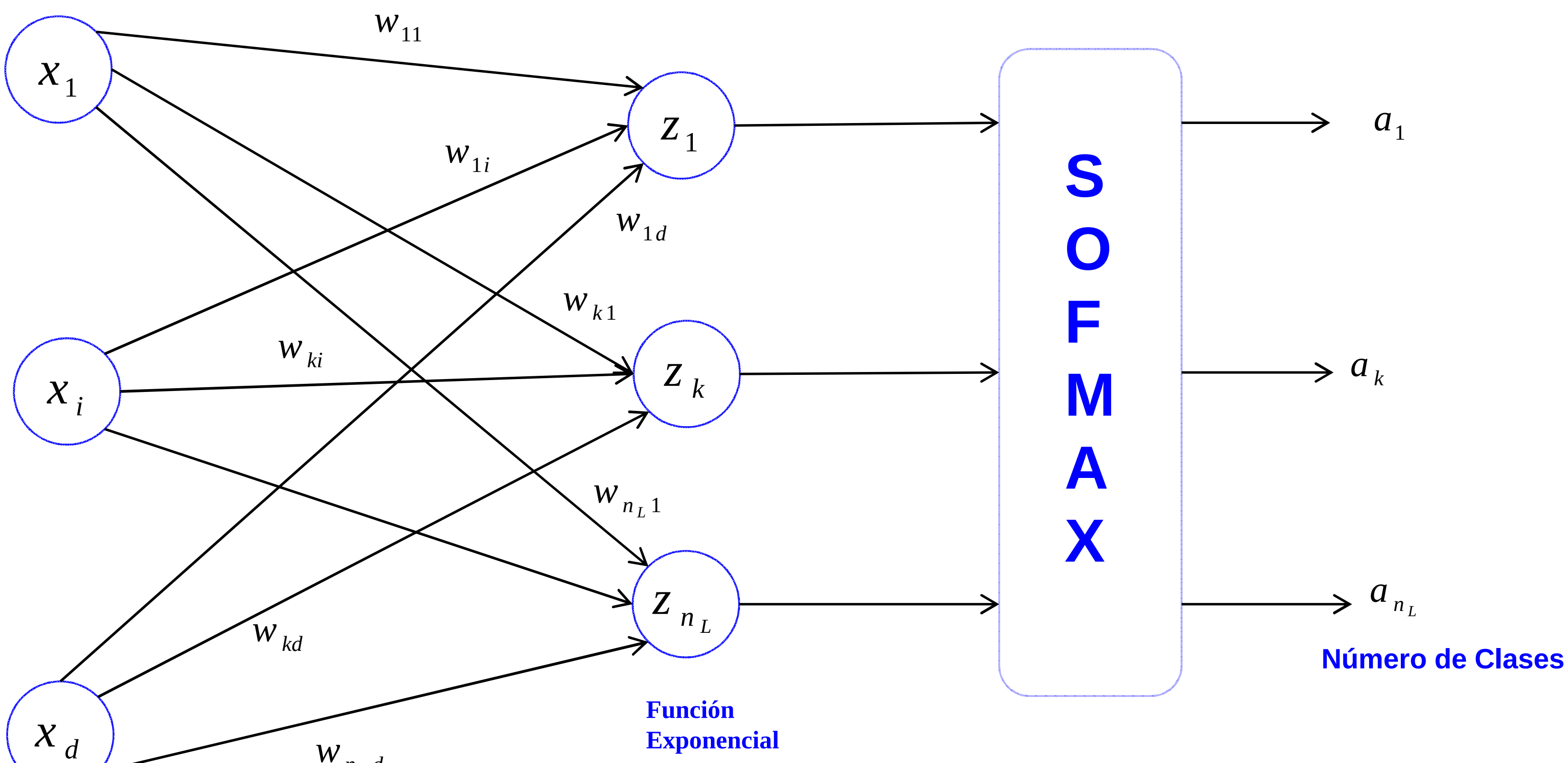
$$\beta_1 = 0.9 \qquad \beta_2 = 0.999$$

Deep AutoEncoder



Regresión Softmax usando mADAM

Softmax



Data Entrada

$$z_k = \exp\left(\sum_{i=1}^{d} w_{k,i} x_i\right)$$

$$k = 1, \ldots, n_L$$

$$a_k = \frac{z_k}{\sum_{j=1}^{n_L} z_j}$$

Training Softmax usando MiniBatch mADAM

Training Softmax: MiniBatch mADAM

Entropía Cruzada:

$$E = -\frac{1}{M}\sum_{n=1}^{M}\sum_{k=1}^{n_L} Y_{k,n}\log\left(a_{k,n}\right)$$

- **M** : número de muestras, $\mathbf{n_L}$: numero de salidas, $\mathbf{Y_{k,n}}$: valores deseados

Notación Matricial :
Gradiente Pesos

$$\frac{\partial E}{\partial W} = -\frac{1}{M}\left\{(Y - A)\times X^T\right\}$$

Notación Matricial :
Ajuste Pesos

$$w(t) = w(t-1) - \mu \times gAdam$$
$$t = 1,2,\dots, MaxIter$$
$$u \in (0,1)$$

Training Softmax : miniBatch mADAM

Actualización Pesos de Salida

$$gAdam = \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \times \frac{V(t)}{\sqrt{S(t)} + \varepsilon}, \qquad \varepsilon = \left(10^{-6}, 10^{-8}\right)$$

$$V(t) = \beta_1 \times V(t-1) + (1-\beta_1)\frac{\partial E}{\partial w\left(t-1\right)}$$

$$S(t) = \beta_2 \times S(t-1) + (1-\beta_2)\left(\frac{\partial E}{\partial w\left(t-1\right)}\right)^2$$

$$\begin{aligned} w(0) &= \text{random} \\ V(0), S(0) &= w.shape \\ V(0), S(0) &= 0 \\ \beta_1 = 0.9 \qquad \beta_2 &= 0.999 \end{aligned}$$

TAREA 3 :

Deep Learning :
DAE
+
mADAM

CONTINUARÁ....