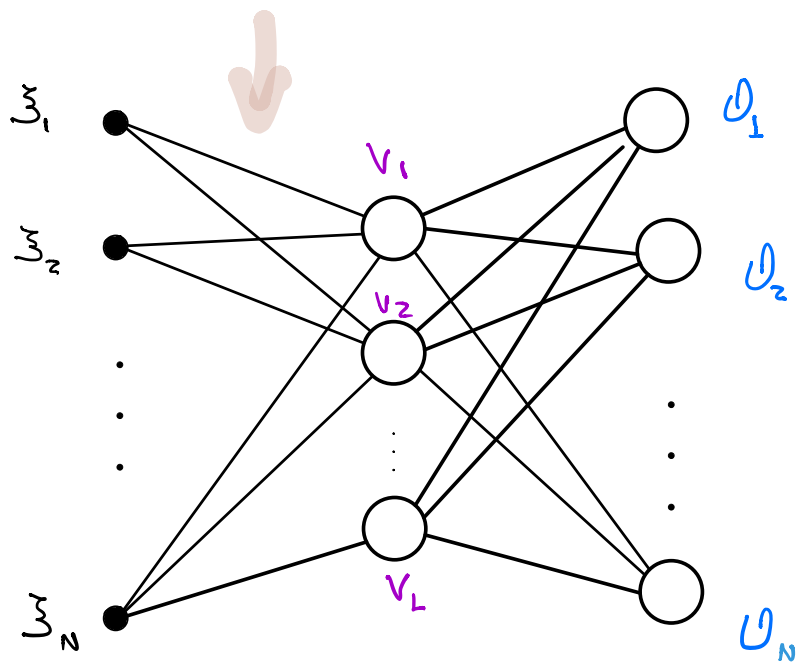


Un plan de mejoras en 5 etapas

De las redes feed forward panditas a las profundas

- ¿Cómo solucionar la elección inicial de los valores de la matriz w ?

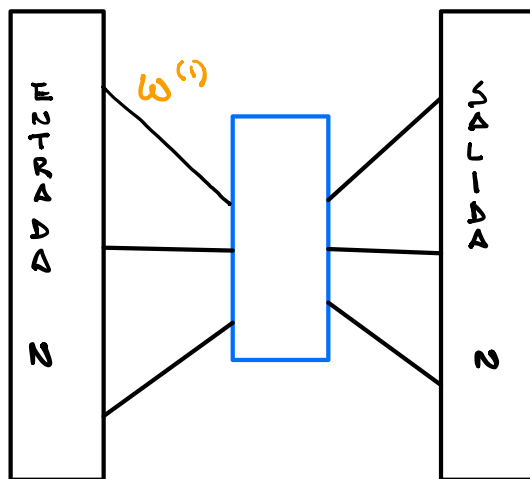
☑ **Autoencoder**³: Es un algoritmo de aprendizaje que aplica back propagation y permite el preprocesamiento de los datos con el objetivo que la salida sea igual a la entrada ($y = x$), es decir la red intenta aprender la función identidad donde $O_i^\mu \approx \xi_k^\mu$ para que la salida sea similar a la entrada. La peculiaridad de este algoritmo consiste en fijar un número de neuronas en la capa intermedia que sea inferior a la cantidad de neuronas en las capas de entrada y salida (que contienen la misma cantidad de neuronas dado que la salida debe ser la entrada). Este algoritmo no parece particularmente difícil o innovador, sin embargo cuando se imponen restricciones en la red (como limitar el número de neuronas en la capa oculta), la red es obligada a encontrar patrones o correlaciones entre los elementos, de forma que es posible aprender una representación comprimida de la entrada permitiendo reducir la dimensionalidad inicial de los datos. Entonces el autoencoder se aplica para entrenar las capas a partir de la función identidad al reducir el número de neuronas permitiendo la codificación de la información. En este sentido, se encontró que cuando se aplica un preprocesamiento de los acoplamientos y se parte de redes que han sido autoaprendidas con el autoencoder, el método de back propagation resulta mucho más eficiente que cuando se tabulan estos valores al azar.



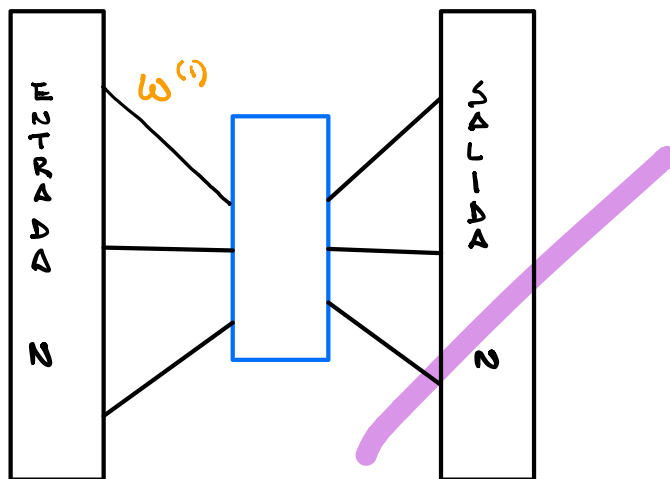
La red tiene todas neuronas en la entrada como en la salida y L neuronas en la capa oculta. Es importante que $L < N$. Debe aprender la identidad

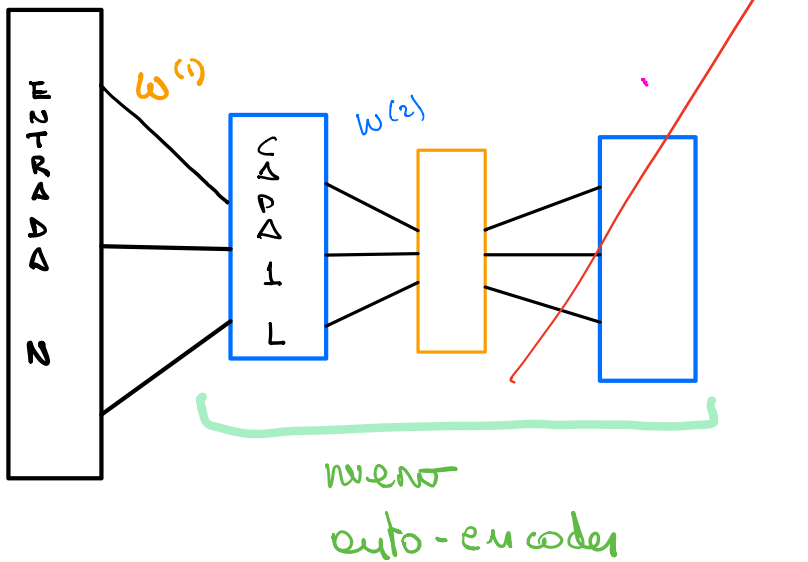
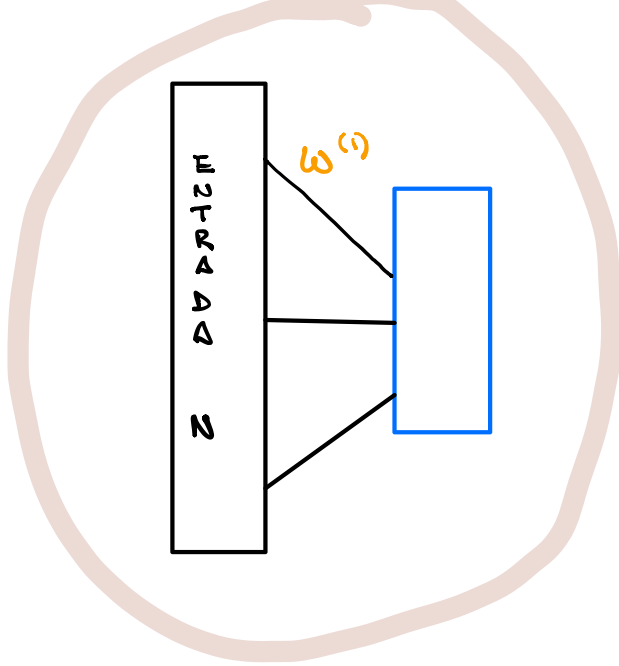
$$\vec{y}^M = \vec{x}^M$$

Entonces se pueden usar para entrenar hacia adelante

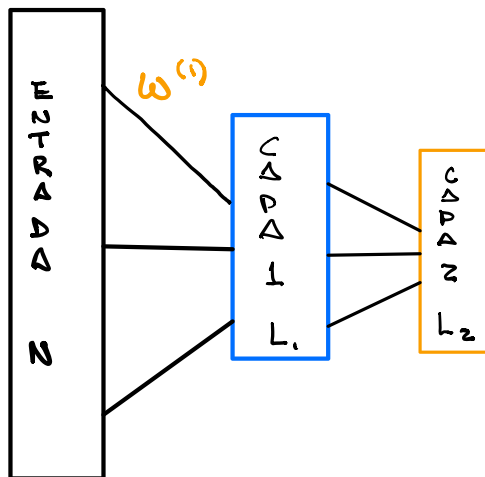


Entonces la red hasta que aprende la identidad. Ahora tiro la segunda parte





Cuando el 2º auto-encoder aprendió la identidad
fue la última parte



Repito esto hasta tener tantas capas como quiero y
comienzo recién ahora a aplicar back
propagation. Lo mismo se puede hacer con una
MÁQUINA DE BOLTZMANN.

Esto mejoró muchísimo el aprendizaje

TRANSFER LEARNING



$$724 \rightarrow 200 \rightarrow 724$$

$$724 \rightarrow 200 - 100 \rightarrow 200$$

$$724 \rightarrow 200 - 100 - 10$$

2.- Las funciones exponenciales ralentizan el aprendizaje

