Representation Learning

Luis Norberto Zúñiga Morales

10 de mayo de 2022

Contenido

Representaciones Eficientes de la Información

¿Cuál de las siguientes secuencias es más fácil de memorizar?

1. 40,27,25,36,81,57,10,73,19,68
2. 50,48,46,44,42,40,38,36,34,32,30,28,26,24,22,20,18,16,14,12,10

Representaciones Eficientes de la Información

Chase y Simon [1] en los 70s observaron que los jugadores expertos de ajedrez son capaces de memorizar las posiciones de todas las piezas en un juego únicamente viendo el tablero por 5 segundos.

Representaciones Eficientes de la Información

Chase y Simon [1] en los 70s observaron que los jugadores expertos de ajedrez son capaces de memorizar las posiciones de todas las piezas en un juego únicamente viendo el tablero por 5 segundos.

El detalle es que sólo pueden memorizarlas si se encontraban en posiciones realistas, no al azar.

Representaciones Eficientes de la Información

Chase y Simon [1] en los 70s observaron que los jugadores expertos de ajedrez son capaces de memorizar las posiciones de todas las piezas en un juego únicamente viendo el tablero por 5 segundos.

El detalle es que sólo pueden memorizarlas si se encontraban en posiciones realistas, no al azar.

Los ajedrecistas expertos son buenos observadores de patrones en ajedrez.

Autoencoder

- Un *autoencoder* es un red neuronal que se entrena para copiar la entrada en la salida .

Autoencoder

- Un *autoencoder* es un red neuronal que se entrena para copiar la entrada en la salida .
- La idea básica es que tiene una capa oculta h que determina una codificación que sirve para representar la entrada.

Autoencoder

La red consiste de dos partes principales:

- Una función codificadora (o red de reconocimiento) $h = f(x)$
- Una función de decodificación (o red generativa) $r = g(h)$

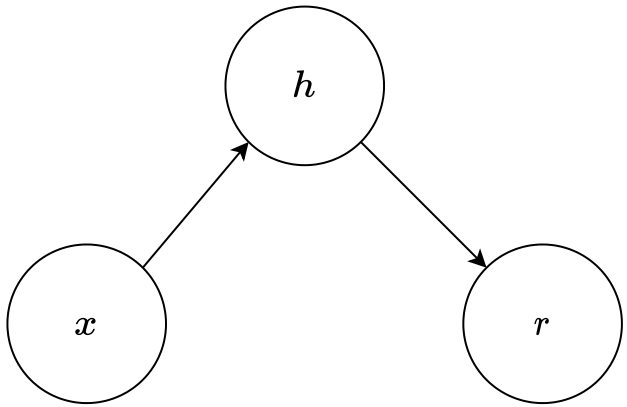


Figura: Estructura general de un autoencoder. Mapea un vector de entrada $\boldsymbol{x}$ en uno de salida $\boldsymbol{r}$, proceso que se llama reconstrucción, por medio de una representación interna de h.

Autoencoder

- Si la red neuronal del autoencoder sólo aprende

$$g(f(\boldsymbol{x})) = \boldsymbol{x}$$

en cualquier punto, no es particularmente útil.

Autoencoder

- Si la red neuronal del autoencoder sólo aprende

$$g(f(\boldsymbol{x})) = \boldsymbol{x}$$

en cualquier punto, no es particularmente útil.

- Los autoencoders se diseñan para no ser capaces de copiar perfectamente las entradas.

Autoencoder

- Si la red neuronal del autoencoder sólo aprende

$$g(f(\boldsymbol{x})) = \boldsymbol{x}$$

en cualquier punto, no es particularmente útil.

- Los autoencoders se diseñan para no ser capaces de copiar perfectamente las entradas.
- Como se obliga al modelo a copiar los aspectos más importantes de la entrada, aprende las propiedades más importantes de ella.

Undercomplete Autoencoders

- Copiar la entrada en la salida, suena inútil.

Undercomplete Autoencoders

- Copiar la entrada en la salida, suena inútil.
- El punto clave aquí es que la capa h, al entrenarse, puede adquirir propiedades útiles de los datos.

Undercomplete Autoencoders

- Copiar la entrada en la salida, suena inútil.
- El punto clave aquí es que la capa h, al entrenarse, puede adquirir propiedades útiles de los datos.
- Una forma de obtener características útiles es restringir h a que tenga una dimensión menor que $\boldsymbol{x}$.

Undercomplete Autoencoders

- Copiar la entrada en la salida, suena inútil.
- El punto clave aquí es que la capa h, al entrenarse, puede adquirir propiedades útiles de los datos.
- Una forma de obtener características útiles es restringir h a que tenga una dimensión menor que $\boldsymbol{x}$.
- Un autoencoder con esa característica se le conoce como incompleto (*undercomplete*).

Undercomplete Autoencoders

- Copiar la entrada en la salida, suena inútil.
- El punto clave aquí es que la capa h, al entrenarse, puede adquirir propiedades útiles de los datos.
- Una forma de obtener características útiles es restringir h a que tenga una dimensión menor que $\boldsymbol{x}$.
- Un autoencoder con esa característica se le conoce como incompleto (*undercomplete*).
- Al ser una representación incompleta, el autoencoder debe capturar las características más importantes de los datos.

Undercomplete Autoencoders

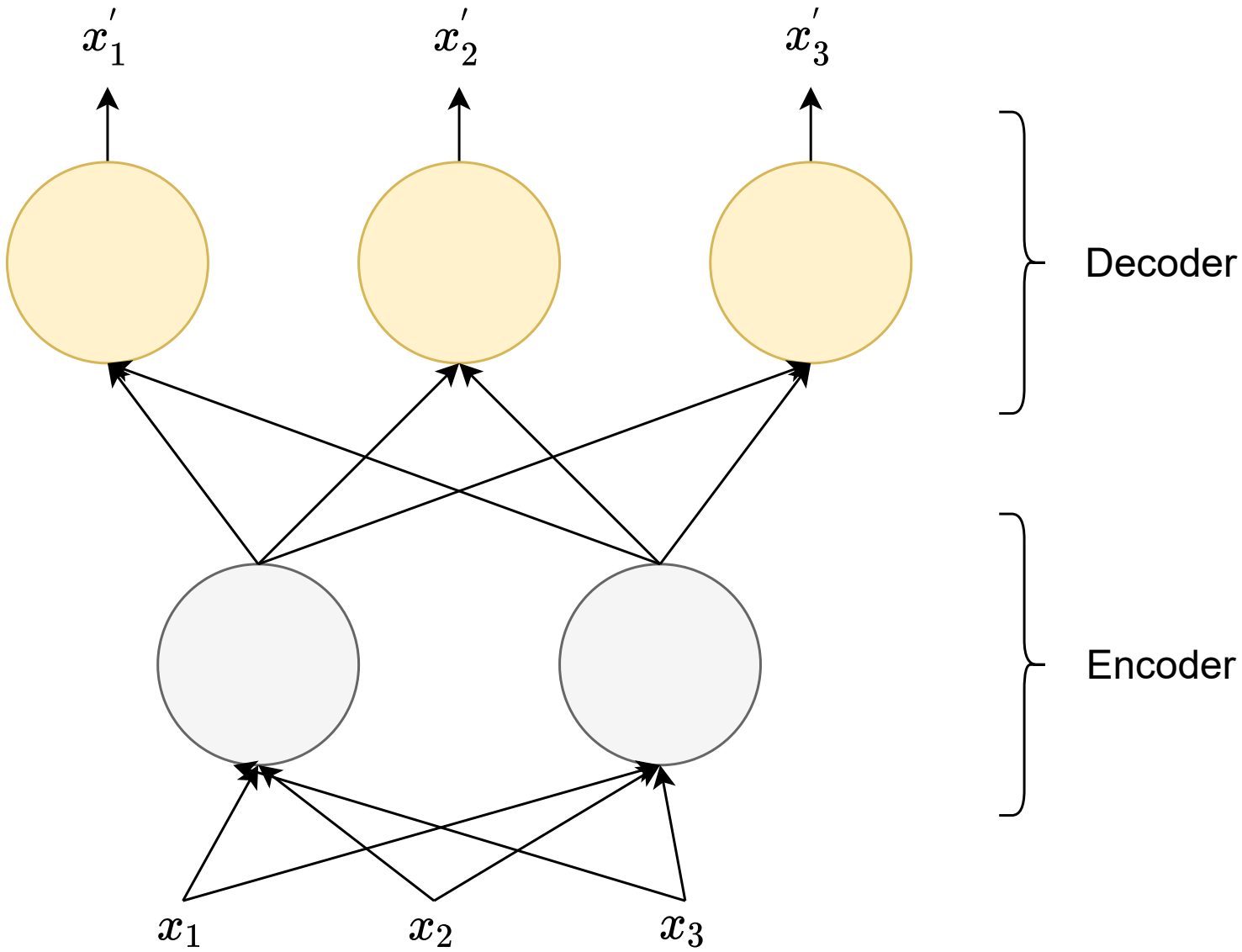


Figura: Noten que la arquitectura es la misma que la de un perceptrón.

Undercomplete Autoencoders

- Si el autoencoder utiliza únicamente activaciones lineales y la función de costo es el Error Cuadrático Medio...

Undercomplete Autoencoders

- Si el autoencoder utiliza únicamente activaciones lineales y la función de costo es el Error Cuadrático Medio...
- ¡Un autoencoder incompleto termina por hacer PCA!

Undercomplete Autoencoders

Vámonos a Google Colab...

Undercomplete Autoencoders

Vámonos a Google Colab...

Ejercicio

Aplicar PCA en el mismo conjunto de datos X y graficar, en el mismo canvas, la reducción de dimensionalidad que hace PCA y la realizada por el autoencoder programado.

Undercomplete Autoencoders

Nota

Es buena idea ver a los autoencoders como una forma de aprendizaje auto supervisado. Es decir, mediante una técnica de aprendizaje supervisado, generan automáticamente las etiquetas de clase, solo que en este caso son igual que las entradas.

Stacked Autoencoders

Un autoencoder no es más especial que otras redes neuronales que hayan visto antes:

- Pueden tener múltiples capas ocultas.
- En este caso se conocen como *deep* o *stacked autoencoders*.

Stacked Autoencoders

Un autoencoder no es más especial que otras redes neuronales que hayan visto antes:

- Pueden tener múltiples capas ocultas.
- En este caso se conocen como *deep* o *stacked autoencoders*.

Pregunta

¿En qué creen que ayude tener más capas ocultas al autoencoder?

Stacked Autoencoders

Un autoencoder no es más especial que otras redes neuronales que hayan visto antes:

- Pueden tener múltiples capas ocultas.
- En este caso se conocen como *deep* o *stacked autoencoders*.

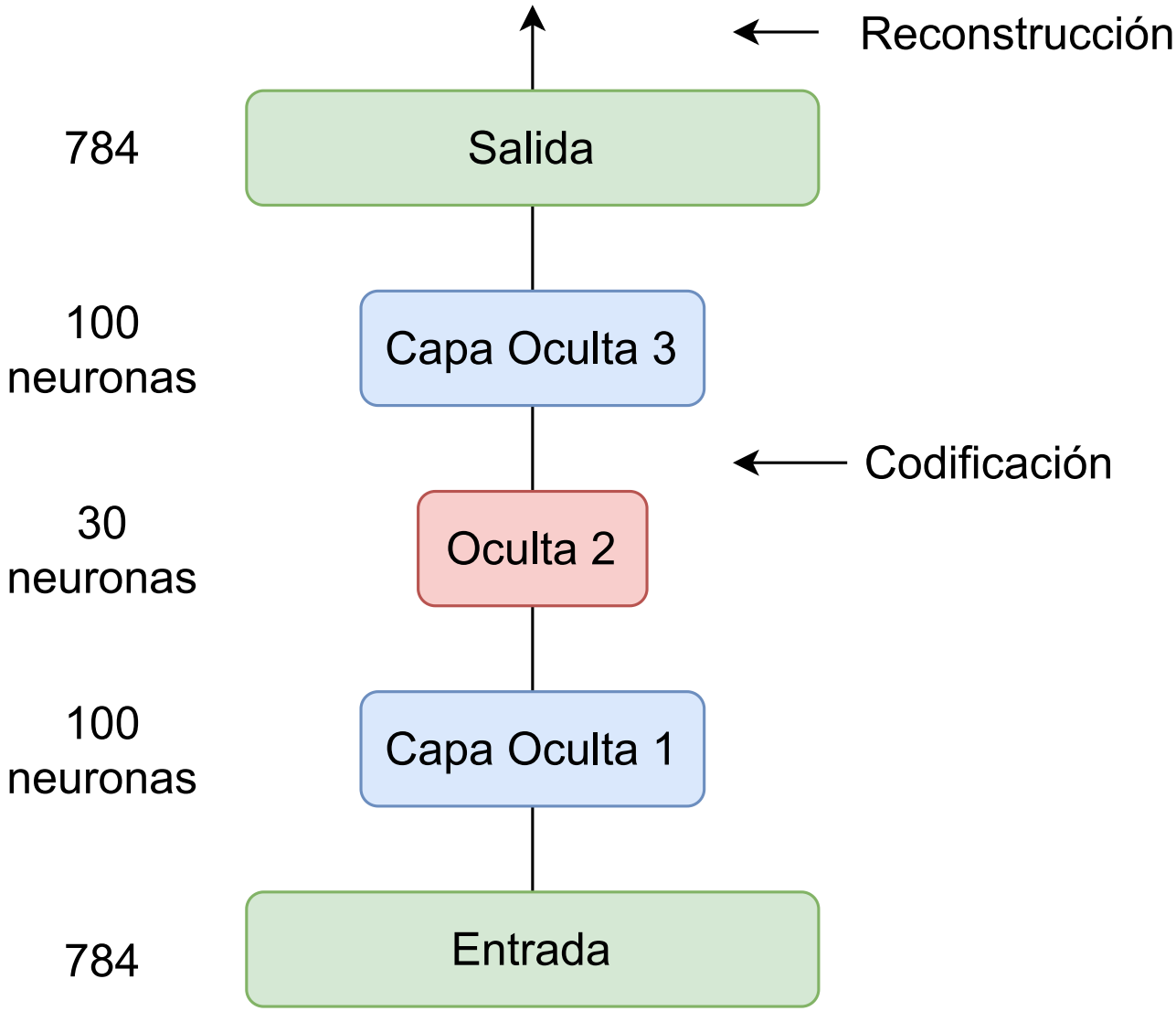
Pregunta

¿En qué creen que ayude tener más capas ocultas al autoencoder?

Respuesta

Más capas permiten al autoencoder aprender codificaciones más complejas.

Stacked Autoencoders



Stacked Autoencoders

Vámonos a Google Colab...

Stacked Autoencoders

Vámonos a Google Colab...

Ejercicio

¿Qué pasa si aumentan el número de epochs durante el entrenamiento? Repitan el entrenamiento con 20, 30, 40 y 50 epochs y comparen las imágenes reconstruidas.

Stacked Autoencoders

Vámonos a Google Colab...

Ejercicio

¿Qué pasa si aumentan el número de epochs durante el entrenamiento? Repitan el entrenamiento con 20, 30, 40 y 50 epochs y comparen las imágenes reconstruidas.

Ejercicio

Hagan lo mismo pero con el conjunto de datos Fashion MNIST.

Stacked Autoencoders

Vámonos a Google Colab...

Ejercicio

¿Qué pasa si aumentan el número de epochs durante el entrenamiento? Repitan el entrenamiento con 20, 30, 40 y 50 epochs y comparen las imágenes reconstruidas.

Ejercicio

Hagan lo mismo pero con el conjunto de datos Fashion MNIST.

Pregunta

¿Qué aplicación se les ocurre puede tener esta funcionalidad de los autoencoders?

Unsupervised Pretraining

Pregunta

¿Para qué sirve el *transfer learning*?

Unsupervised Pretraining

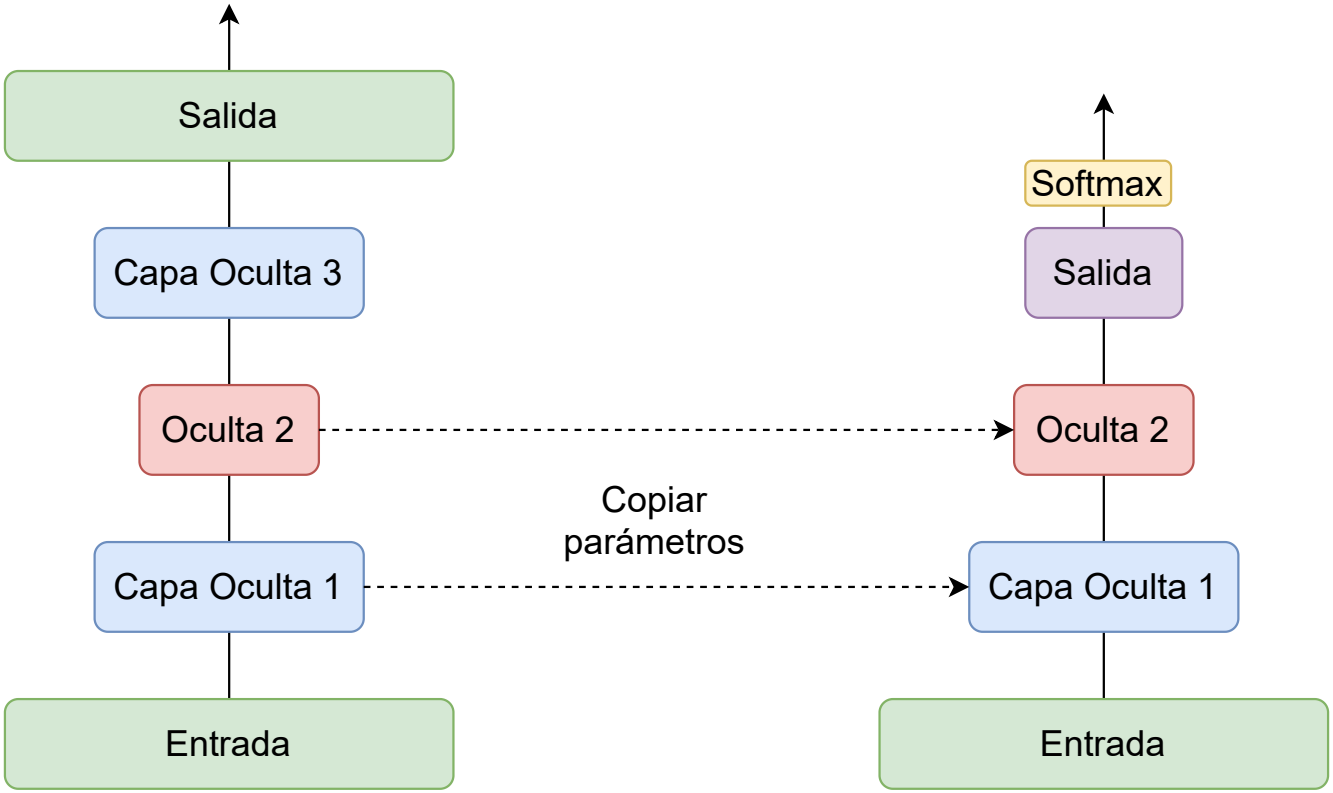
Pregunta

¿Para qué sirve el *transfer learning*?

De manera similar, si:

- Se tiene un gran conjunto de datos donde la mayoría de los datos se encuentran sin etiquetar.
- Se puede entrenar un *stacked autoencoder* usando toda la información...
- Reusar las capas bajas para crear una red neuronal para la tarea de clasificación y entrenar el modelo con la información etiquetada.

Unsupervised Pretraining



Fase 1

Entrenar el autoencoder usando toda la información

Fase 2

Entrenar el clasificador con la información anotada

Unsupervised Pretraining

Pregunta

¿Por qué creen que en la práctica esto es buena idea?

Unsupervised Pretraining

Pregunta

¿Por qué creen que en la práctica esto es buena idea?

Respuesta

En práctica, es muy sencillo obtener información en grandes volúmenes, pero es muy difícil anotarla manualmente.

Unsupervised Pretraining

Bibliografía Sugerida

[1] William G. Chase and Herbert A. Simon. Perception in chess. *Cognitive Psychology*, 4(1):55–81, 1973.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[3] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2nd edition, 2019.