Autoencoders

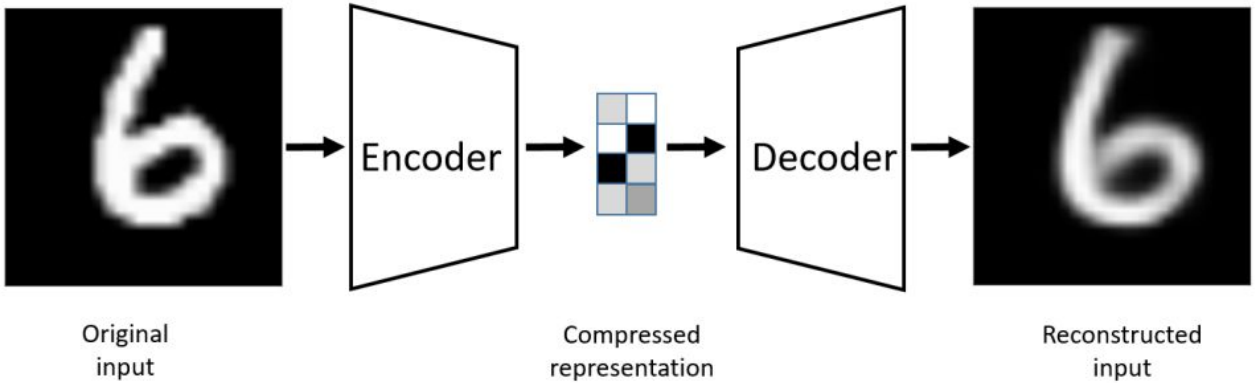
Orlando Ramos Flores

Contenido

- Autoencoder
- Definición
- Regularización
- Feed Forwards Autoencoders
  - Funciones de activación para la capa de salida
  - Error de reconstrucción
- Aplicaciones

Autoencoder

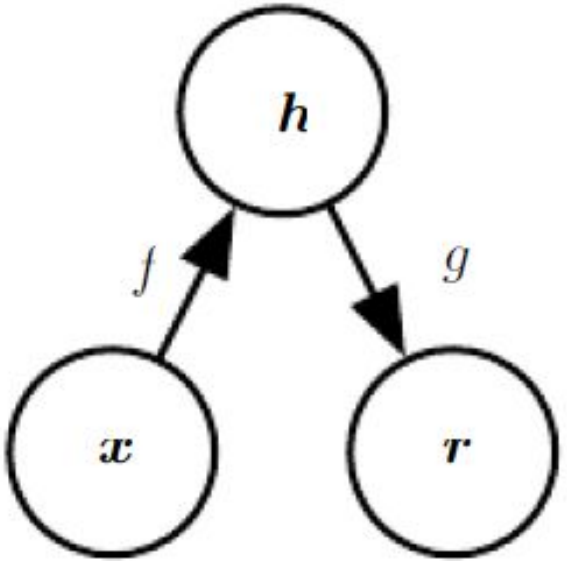
Un autoencoder es un tipo específico de red neuronal, que está diseñado principalmente para codificar la entrada en una representación comprimida y significativa, y luego decodificarla de manera que la entrada reconstruida sea lo más similar posible a la original.



Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. *arXiv preprint arXiv:2003.05991*.

Definición 1

- Un autoencoder es una red neuronal que está entrenada para intentar copiar su entrada en su salida.
- Internamente, tiene una capa oculta **h** que describe un código utilizado para representar la entrada.
- Se puede considerar que la red consta de dos partes:
  - una **función de encoder h = f (x)** y
  - un **decoder** que produce una reconstrucción **r = g (h)**.



Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Definición 1

- Si un autoencoder logra simplemente aprender a establecer **g(f (x)) = x** en todas partes, entonces no es especialmente útil.
- En cambio, los autoencoders están diseñados para no poder aprender a copiar perfectamente.
- Por lo general, están restringidos de manera que solo se les permite copiar de forma aproximada, y copiar solo la entrada que se asemeja a los datos de entrenamiento.
- Debido a que el modelo se ve obligado a priorizar qué aspectos de la entrada deben copiarse, a menudo aprende propiedades útiles de los datos.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Autoencoder: Definición 1

- Tradicionalmente, los autoencoders se usaban para la reducción de la dimensionalidad o el aprendizaje de características.
- Los autoencoders pueden considerarse un caso especial de las redes feedforward (FFN) y pueden entrenarse con todas los mismos técnicas, típicamente descenso de gradiente de mini-batch siguiendo gradientes calculados por retropropagación.
- A diferencia de las FFN generales, los autoencoders también pueden entrenarse mediante recirculación, un algoritmo de aprendizaje basado en la comparación de las activaciones de la red en la entrada original con las activaciones en la entrada reconstruida.
- La recirculación se considera biológicamente más plausible que la retropropagación, pero rara vez se usa para aplicaciones de aprendizaje automático.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Definición 2

- En la mayoría de las arquitecturas típicas, el **encoder** y el **decoder** son redes neuronales ya que se pueden entrenar fácilmente con bibliotecas de software existentes como TensorFlow o PyTorch con retropropagación.
- En general, el **encoder** se puede escribir como una **función g** que dependerá de algunos parámetros: $h_i = g(x_i)$
- Donde, $h_i \epsilon \mathbb{R}^q$ (la representación de características latentes) es la salida del bloque **decoder** cuando se evalúa en la entrada x_i .
- Hay que tener en cuenta que se tendrá $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$

Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

Definición 2

- El **decoder** (y la salida de la red se indicará con $\tilde{x}_i$) se puede escribir entonces como una segunda función genérica f de las características latentes: $\tilde{x}_i = f(h_i) = f(g(x_i))$ donde $\tilde{x}_i \epsilon \mathbb{R}^n$
- Entrenar un autoencoder simplemente significa encontrar las funciones $g(\cdot)$ y $f(\cdot)$ que satisfagan $arg\,min_{f,g} < [\Delta(x_i, f(g(x_i)))] >$
- Donde Δ indica una medida de cómo difieren la entrada y la salida del autoencoder (básicamente, la función de pérdida penalizará la diferencia entre la entrada y la salida) y $< \cdot >$ indica el promedio de todas las observaciones.

Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

Definición 2

- Dependiendo de cómo se diseñe el autoencoder, puede ser posible encontrar **f** y **g** para que el autoencoder aprenda a reconstruir la salida perfectamente, aprendiendo así la función de identidad.
- Esto no es muy útil, como se discutió al principio, y para evitar esta posibilidad, se pueden usar dos estrategias principales:
  - crear un cuello de botella y
  - agregar regularización de alguna forma
- Nota: se quiere que el autoencoder reconstruya la entrada lo suficientemente bien. Aún así, al mismo tiempo, debería crear una representación latente (la salida del codificador) que sea útil y significativa.

Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

Regularización

- Intuitivamente, significa hacer cumplir la escasez en la salida de características latentes.
- La forma más sencilla de lograr esto es agregar un término de regularización L1 o L2 a la función de pérdida.
- Eso se verá así para el término de regularización L2:

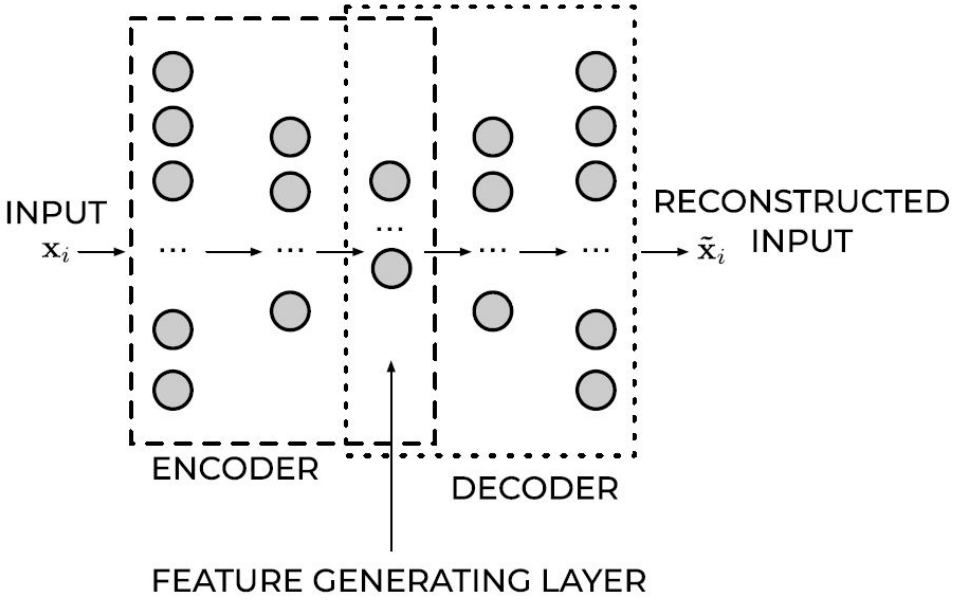
$$arg\,min_{f,g}(\mathbb{E}[\Delta(x_i, g(f(x_i)))] + \lambda \sum_i \theta_i^2)$$

- En la fórmula θ_i son los parámetros (pesos) en las funciones f y g.

Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

Feed Forward Autoencoders

- Un Feed-Forward Autoencoder (FFA) es una red neuronal compuesta de capas densas.
- La cantidad de neuronas en las capas al principio disminuye a medida que se avanza por la red hasta que llega a la mitad y luego comienza a crecer nuevamente hasta que la última capa tiene la misma cantidad de neuronas que las dimensiones de entrada.



Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

Feed Forward Autoencoders

- Una arquitectura FFA típica (aunque no es un requisito obligatorio) tiene un número impar de capas y es simétrica con respecto a la capa intermedia.
- Normalmente, la primera capa tiene un número de neuronas $\mathbf{n_1=n}$ (el tamaño de la observación de entrada $\mathbf{x_i}$).
- A medida que se avanza hacia el centro de la red, el número de neuronas en cada capa disminuye en alguna medida.
- La capa intermedia (recordar que se tiene un número impar de capas) suele tener el menor número de neuronas. El hecho de que el número de neuronas en esta capa sea más pequeño que el tamaño de la entrada es el **cuello de botella** que se mencionó anteriormente.

Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

Feed Forward Autoencoders

- En casi todas las aplicaciones prácticas, las capas posteriores a la intermedia son una versión reflejada de las capas anteriores a la intermedia.
- Por ejemplo, un autoencoder con 3 capas podría tener el siguiente número de neuronas:
  - n1=10, n2=5 y luego n3= n1=10 (suponiendo que se está trabajando en un problema donde la dimensión de entrada es n=10).
  - Todas las capas hasta incluso la intermedia forman lo que se llama el **encoder**, y todas las capas desde e incluso la intermedia hasta la salida forman lo que se llama el **decoder**.
- Si el entrenamiento FFA tiene éxito, el resultado será una buena aproximación de la entrada, en otras palabras, $\tilde{x}_i \approx x_i$.

Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

Feed Forward Autoencoders

- El **encoder** puede reducir el número de dimensiones de la observación de entrada (**n**) y crear una **representación aprendida** ($\mathbf{h_i}$) de la entrada que tiene una dimensión más pequeña **q < n**.
- Esta **representación aprendida** es suficiente para que el **decoder** pueda reconstruir la entrada utilizando solo un número mucho más pequeño (**q**) de características que las observaciones de entrada que tienen inicialmente (**n**).

Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

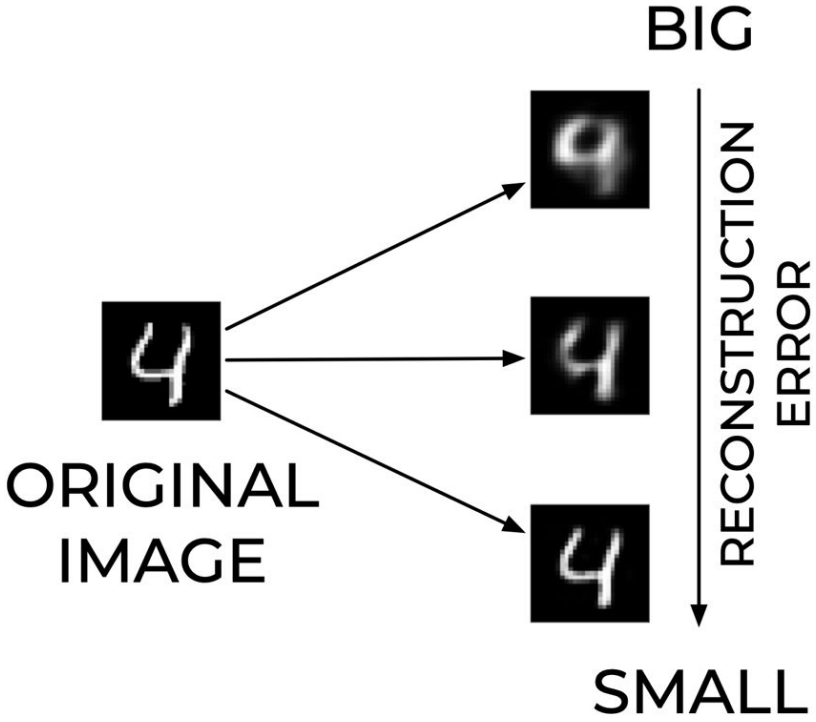
Funciones de activación para la capa de salida

- ReLU
- Sigmoid
- Loss Function
- Mean Square Error
- Binary Cross-Entropy

Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

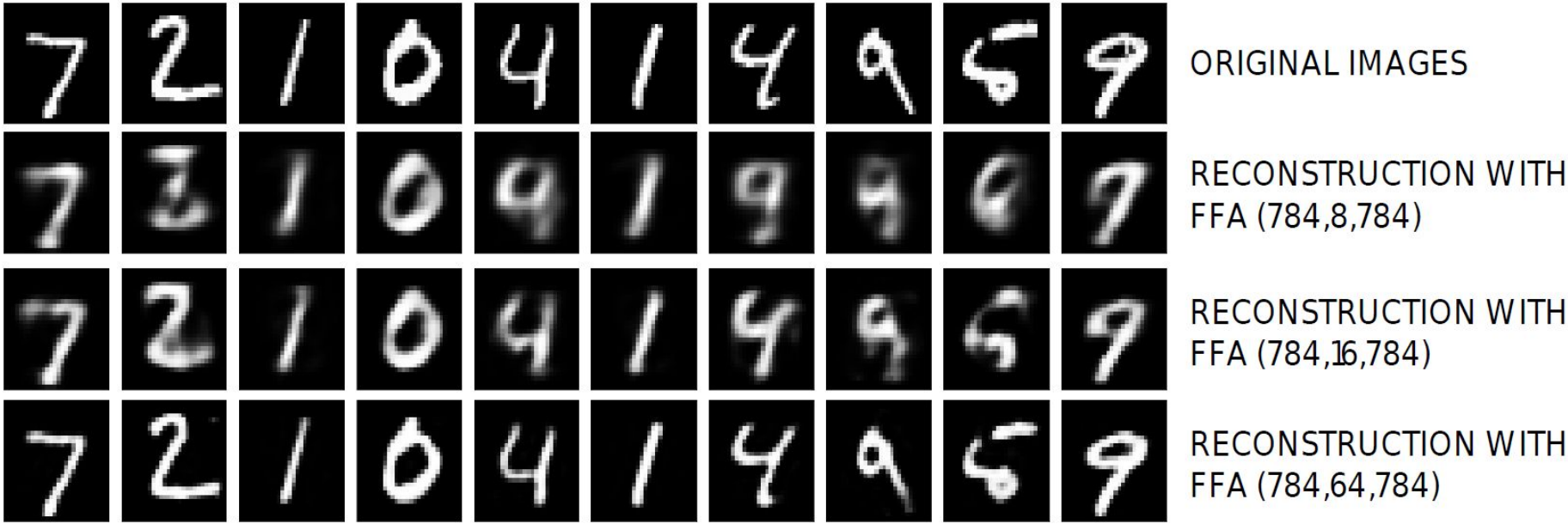
Reconstruction Error

- El error de reconstrucción (RE) es una métrica que indica qué tan bien (o mal) el autoencoder pudo reconstruir la observación de entrada x_i. El RE más típico utilizado es el MSE
- Hay una explicación fácil e intuitiva del error de reconstrucción:
  - Cuando el RE es significativo, el autoencoder no pudo reconstruir bien la entrada.
  - Mientras que cuando es pequeño, la reconstrucción fue exitosa.



Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

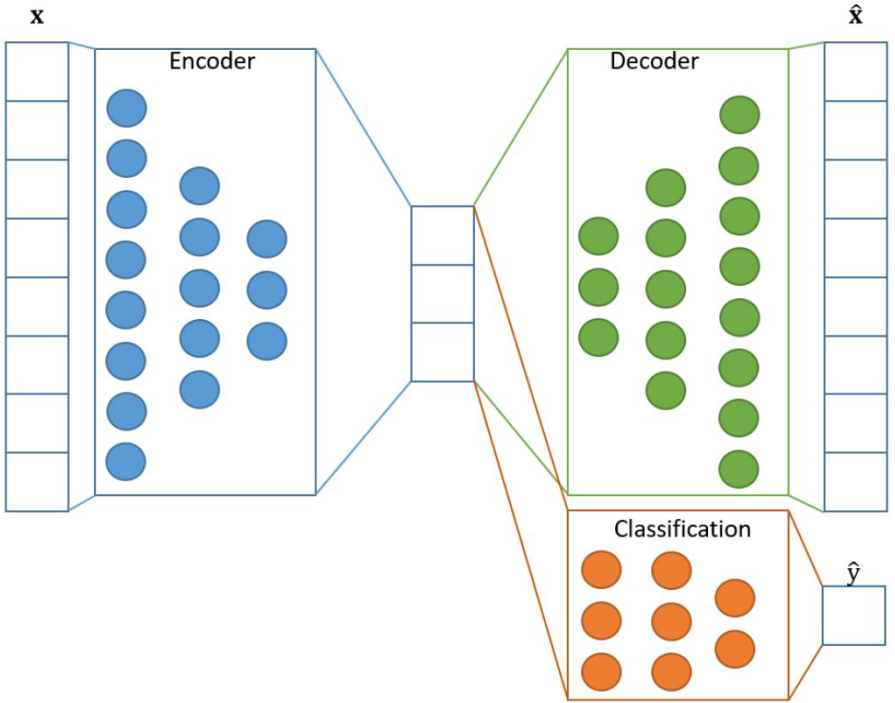
Reconstruction Error



Michelucci, U. (2022). An Introduction to Autoencoders. *arXiv preprint arXiv:2201.03898*.

Aplicaciones

- **Clasificación**.
  - El decoder se deja de lado y el encoder se usa como la primera parte de un modelo de clasificación.
  - Sus pesos pueden ajustarse o permanecer fijos durante el entrenamiento.
  - Se puede encontrar una estrategia más simple en la que se entrena una máquina de vectores de soporte (SVM) en las características de salida del encoder.
  - Otro enfoque utiliza autoencoders como técnica de regularización para una red de clasificación.



Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. *arXiv preprint arXiv:2003.05991*.

Aplicaciones

- **Agrupamiento**
  - Suponiendo que los datos tienen alguna representación latente de baja dimensión, se pueden usar autoencoders para calcular tales representaciones de los datos, que se componen de muchas menos características.
  - El decodificador se deja de lado, de manera similar al uso en la clasificación.
  - La representación latente (la salida del codificador) de cada punto de datos se mantiene y sirve como entrada para cualquier algoritmo de agrupamiento dado (por ejemplo, K-means).
  - La principal desventaja de usar autoencoders para la agrupación en clústeres es que los embeddings se entrenan únicamente para la reconstrucción y no para la aplicación de agrupación en clústeres.

Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. *arXiv preprint arXiv:2003.05991*.

Aplicaciones

- **Detección de anomalías**
  - El objetivo es aprender un perfil normal dado de los ejemplos de datos normales y luego identificar las muestras que no se ajustan al perfil normal como anomalías.
  - Esto se puede aplicar en diferentes aplicaciones, como detección de fraude, monitoreo de sistemas, etc.
  - El uso de autoencoders para estas tareas parte del supuesto de que un autoencoder entrenado aprendería el subespacio latente de las muestras normales.
  - Una vez entrenado, daría como resultado un error de reconstrucción bajo para muestras normales y un error de reconstrucción alto para las anomalías.

Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. *arXiv preprint arXiv:2003.05991*.

Aplicaciones

- **Sistemas de recomendación**
  - Un sistema de recomendación es un modelo o sistema que busca predecir las preferencias o afinidades de los usuarios con los artículos.
  - Un ejemplo básico del uso de autoencoders para sistemas de recomendación es el modelo AutoRec.
  - El modelo AutoRec tiene dos variantes: AutoRec basado en el usuario (U-AutoRec) y AutoRec basado en items (I-AutoRec).
  - En U-AutoRec, el autoencoder aprende una representación de menor dimensión de las preferencias de items para usuarios específicos.
  - Mientras que en I-AutoRec, el autoencoder aprende una representación de menor dimensión de las preferencias del usuario para items específicos.

Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. *arXiv preprint arXiv:2003.05991*.

Aplicaciones

- **Reducción de dimensionalidad**
  - Los datos del mundo real, como texto o imágenes, a menudo se representan utilizando una representación dispersa de alta dimensión.
  - El objetivo de la reducción de la dimensionalidad es aprender una variedad de dimensiones inferiores, el llamado espacio de "dimensionalidad intrínseca".
  - El uso de autoencoders para la reducción de dimensionalidad es sencillo.
  - De hecho, la reducción de dimensionalidad la realiza cada autoencoder en la capa de cuello de botella.
  - La proyección de la entrada original en la representación de cuello de botella de menor dimensión es una operación de reducción de dimensión a través del encoder y bajo el objetivo dado al decoder.

Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. *arXiv preprint arXiv:2003.05991*.