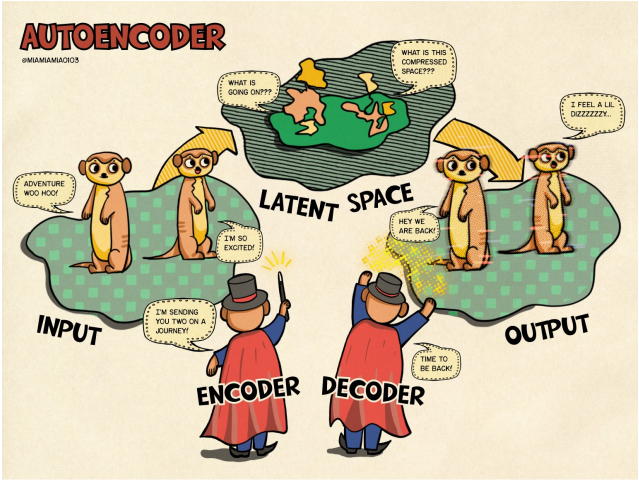
Modelos Generativos

Nahuel Costa

Grado en Ciencia e Ingeniería de Datos

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Autoencoders



Autoencoders

Los autoencoders son modelos que comprimen y reconstruyen datos. Este proceso puede derivar en diversas aplicaciones como detección de anomalías, manifold learning, compresión o generación de datos.

La idea se originó en la década de los 80 y posteriormente fue promovida por Geoffrey Hinton y compañía [1] en 2006.

Autoencoders

Estos modelos generalmente constan de dos redes:

1. Un **Encoder**, $g(\cdot)$ que comprime los datos de entrada $\mathbf{x}$ a un espacio de menores dimensiones, conocido como **espacio latente**, $\mathbf{z} = g_\phi(\mathbf{x})$.
2. Un **Decoder**, $f(\cdot)$ que a partir del espacio latente reconstruye los datos, $\mathbf{x}' = f_\theta(g_\phi(\mathbf{x}))$.

Los parámetros de cada red (ϕ, θ) se optimizan a la vez para que las reconstrucciones sean lo más parecidas a los datos originales, $\mathbf{x} \approx f_\theta(g_\phi(\mathbf{x}))$.

Autoencoders

Existen muchas variantes de autoencoders como Denoising Autoencoders [2], Sparse Autoencoders [3] o Contractive Autoencoders [4]. De aquí en adelante nos centraremos en los Variational AutoEncoders (VAE) [5] por sus propiedades generativas.

En esta librería tenéis varios modelos de autoencoders programados en Keras 3.

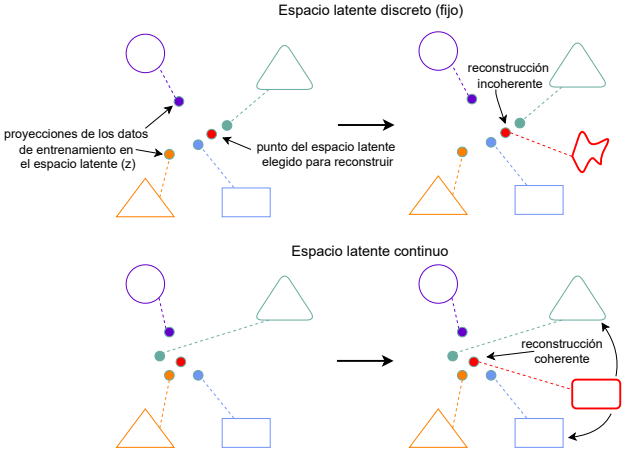
Se agradece una estrella :)

VAE

La novedad que introducen los VAE respecto a otros autoencoders es que en lugar de mapear los datos a un vector fijo se mapean a una distribución.

Esto tiene la ventaja de que se puede elegir un punto del espacio latente z que no pertenezca a la proyección de ningún dato de entrenamiento, pasarlo por el decoder y obtener una reconstrucción coherente x. Es decir, generar datos en la forma $z \rightarrow x$.

VAE


Espacio latente discreto (fijo)
proyecciones de los datos de entrenamiento en el espacio latente (z)
punto del espacio latente elegido para reconstruir
reconstrucción incoherente
Espacio latente continuo
reconstrucción coherente

VAE

La relación entre los datos de entrada $\mathbf{x}$ y el vector latente $\mathbf{z}$ viene dada por:

- Prior $p_\theta(\mathbf{z})$
- Likelihood $p_\theta(\mathbf{x}|\mathbf{z})$
- Posterior $p_\theta(\mathbf{z}|\mathbf{x})$

La distribución de los datos entonces se puede modelar como:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z}$$

VAE

Como ya vimos antes, no es fácil calcular $p_\theta(\mathbf{x})$ de esta manera, ya que es muy costoso comprobar y sumar todos los valores posibles de $\mathbf{z}$. Para solucionar este problema, los VAE recurren a la inferencia variacional (de ahí su nombre). La inferencia variacional es uno de los métodos más recurridos de la inferencia bayesiana y se utiliza para aproximar integrales intratables. En otras palabras, es una técnica utilizada para aproximar distribuciones complejas.

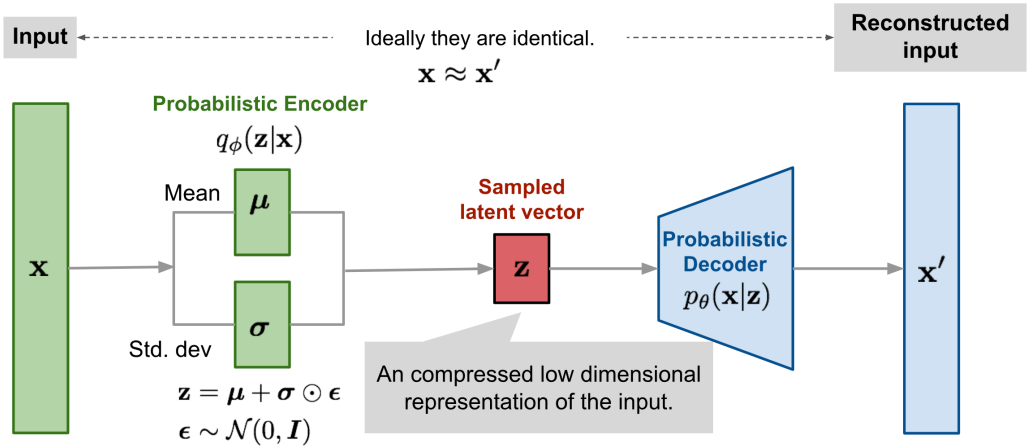
VAE

La distribución compleja que se quiere aproximar es $p_\theta(\mathbf{z}|\mathbf{x})$ (posterior). Para aproximarla mediante inferencia variacional la idea es presuponer una familia de distribuciones simples y plantear la aproximación como un problema de optimización: buscar dentro de la familia de distribuciones q_ϕ la distribución que más se aproxime a la distribución objetivo. El parámetro variacional ϕ denota los parámetros de la familia de distribuciones elegida. Por ejemplo, si asumimos que va a ser una gaussiana, ϕ serían la media y la varianza de las variables latentes.

VAE

De esta manera se introduce una función de aproximación, $q(z|x)$, modelada mediante un encoder para aprender una distribución sobre variables latentes dada una entrada x y un decoder que se encarga de modelar la probabilidad condicional $p(x|z)$.

VAE


Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
$\boldsymbol{\sigma}$
Std. dev
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
$\mathbf{x}$
Sampled latent vector
$\mathbf{z}$
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$
An compressed low dimensional representation of the input.

Función de error: ELBO

La distribución estimada por el encoder, $q_\phi(\mathbf{z}|\mathbf{x})$, debe ser próxima a la real $p_\theta(\mathbf{z}|\mathbf{x})$. Para cuantificar la distancia entre estas dos distribuciones se usa la divergencia de Kullback-Leibler, $D_{\mathrm{KL}}(Y|X)$, que se encarga de medir cuánta información se pierde si se usa la distribución Y para representar X.

En este caso se quiere minimizar $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))$ con respecto a ϕ.

Función de error: ELBO

Usando el teorema de Bayes, se puede reescribir como:

$$
\begin{aligned}
D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) = & \log p(\mathbf{x}) \\
& - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \\
& + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))
\end{aligned}
$$

Función de error: ELBO

El primer término es el log-likelihood de los datos, que queremos maximizar.

El segundo término, es el log-likelihood esperado de los datos bajo la posterior aproximada (encoder).

El tercer término es la divergencia KL entre la posterior aproximada y el prior.

Función de error: ELBO

Combinando estos términos, se puede definir la función de pérdida de un VAE como:

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}q\phi(\mathbf{z}|\mathbf{x})[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathsf{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))$$

También conocida como Evidence Lower bound (ELBO).

Función de error: ELBO

El primer término es la reconstrucción de x que tiende a hacer el esquema de codificación-decodificación lo más eficiente posible maximizando el log-likelihood $\log p_\theta(\mathbf{x}|\mathbf{z})$ con muestreo de $q_\phi(\mathbf{z}|\mathbf{x})$ (encoder).

El segundo término regulariza las variables latentes (representadas por z) minimizando la divergencia KL entre la aproximación variacional (encoder) y la distribución a priori de z. Esto fuerza a obtener un espacio latente en el que las distribuciones devueltas por el encoder se aproximen a una normal estándar.

Aplicaciones

Las aplicaciones de los VAE son innumerables. Algunos ejemplos más allá de la generación de imágenes son su aplicación a modelado de texto y su semántica [6], detección de anomalías [7], estimación de la vida útil de sistemas industriales [8], detección de arritmias [9] o diseño molecular [10].

Posterior collapse

Los VAE son conocidos por ser susceptibles al **posterior collapse**. Esto es un problema que se refiere a la situación en la que la posterior aproximada colapsa con la prior debido a la desaparición (vanishing) del término KL. Esto fuerza a la posterior aproximada a ser independiente de los datos, es decir, $q(\mathbf{z}|\mathbf{x}) = q(\mathbf{z}) = p(\mathbf{z})$. En este caso, la representación latente de la entrada es ignorada por el decoder y por tanto no aprende ninguna representación útil de los datos.

Trucos de entrenamiento

Hay varias aproximaciones para solventar este problema, entre ellas:

- **KL Divergence Annealing**: Incrementa gradualmente el peso del término de divergencia KL en la función de pérdida durante el entrenamiento para permitir que el modelo aprenda primero a reconstruir los datos antes de imponer una regularización fuerte en el espacio latente.
- **Free Bits**: Establece un umbral mínimo para la divergencia KL en cada dimensión de la variable latente, asegurando que cada dimensión contribuya con al menos un "mínimo de información" y evitando que colapsen a valores triviales.

Trucos de entrenamiento

- **Minimum Desired Rate**: Similar a *Free Bits*, pero aplica una penalización para mantener un mínimo de información en el espacio latente, forzando que la compresión no pierda relevancia.
- **Dropout**: Desconecta aleatoriamente conexiones de la red durante el entrenamiento para evitar que el modelo dependa demasiado de ciertas neuronas, manteniendo así la relevancia de las variables latentes.
- **Independent Hidden States**: Obliga a que las representaciones ocultas del encoder sean independientes entre sí, asegurando que las variables latentes no se vuelvan redundantes o que algunas dimensiones se ignoren.

Variantes del VAE

La literatura de los VAE es muy extensa y se han propuesto múltiples variantes. Veremos algunas de ellas.

Uno de los problemas de los VAE es que generan imágenes borrosas. Sin embargo, esto no pasa con modelos que optimizan explícitamente la probabilidad de los datos, como los normalizing flows.

Variantes del VAE

¿Por qué pasa esto?

Normalmente para cuantificar la calidad de las reconstrucciones se utilizan métricas como RMSE o BCE, por lo tanto, la red tiene como objetivo minimizar estas métricas. Durante el entrenamiento el modelo puede acertar una reconstrucción en concreto pero fallar en el resto, por lo que el error será alto. Una forma de minimizar el error es no acertar reconstrucciones concretas si no producir generaciones borrosas pero cercanas a los datos reales de forma que cuando se hace la media de las reconstrucciones el error obtenido sea bajo.

β-VAE

Se puede resolver este problema modificando el encoder para evitar fusionar entradas distintas en el mismo espacio latente o del decoder añadiendo información que falta en el espacio latente. Sin embargo, una solución aún más sencilla es reducir la penalización sobre el término KL, haciendo que el modelo se acerque más a un autoencoder determinista:

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}q\phi(\mathbf{z}|\mathbf{x})[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))$$

En esto consiste el β-VAE [11] (2017).

β-VAE

Si se fija $\beta = 1$, la función objetivo es la misma utilizada en los VAE estándar; si se fija a 0, entonces es la misma utilizada por los autoencoders estándar. Si se usa $\beta < 1$, se almacenan más bits sobre cada entrada y, por tanto, se pueden reconstruir las imágenes de forma menos borrosa. Si se utiliza $\beta > 1$, se obtiene una representación más comprimida.

β-VAE

Una ventaja de utilizar $\beta > 1$ es que fomenta el aprendizaje de una representación latente "desenredada". Intuitivamente, esto significa que cada dimensión latente representa un factor/característica diferente de los datos de entrada. Por ejemplo, un modelo entrenado con fotos de rostros humanos podría captar la suavidad, el color de la piel, el color o la longitud del pelo, la emoción, si se llevan gafas y muchos otros factores relativamente independientes en dimensiones separadas.

InfoVAE

Otro problema de los VAE estándar es que a veces al optimizar el ELBO se pone demasiado énfasis en el segundo término (la regularización), lo que puede llevar a representaciones latentes que no capturen bien la estructura de los datos y a muestras generadas que no son diversas.

El InfoVAE [12] (2017) modifica la función de pérdida añadiendo un término de **maximización de la información mutua** entre las variables latentes z y los datos observados x.

InfoVAE

La función de pérdida de InfoVAE se puede escribir como:
$\mathcal{L}_{\text{InfoVAE}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \alpha D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})) + \lambda D_{\text{KL}}(q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}))$

Donde el primer término es el error de reconstrucción, el segundo es la divergencia KL entre la distribución posterior aproximada y la prior y el tercero es la divergencia KL entre la distribución marginal de las variables latentes y la distribución prior.

α y λ son hiperparámetros que controlan la importancia relativa de cada término. De esta manera, al maximizar la información mutua entre z y x, se asegura que las representaciones latentes z contengan más información relevante sobre los datos x.

Multimodal VAEs

Los VAE multimodales [13], [14] son una extensión del modelo VAE estándar para manejar y aprender representaciones conjuntas de datos provenientes de diferentes modalidades. Estas modalidades pueden incluir texto, imágenes, audio, video, etc. La idea principal es capturar las dependencias y correlaciones entre diferentes tipos de datos, aprendiendo una representación latente compartida que pueda ser utilizada para tareas como la generación de datos o la imputación de información perdida.
Hasta ahora se han propuesto varios enfoques para el aprendizaje multimodal de VAEs, en la siguiente review se recopilan algunos de los trabajos más influyentes [15].

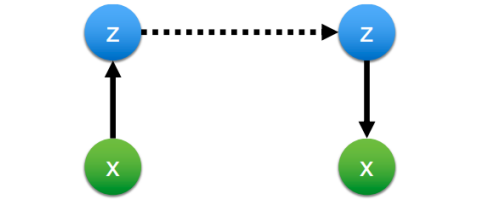
Hierarchical VAEs

Un problema de los VAEs es que sufren de priors simples.
Una forma de suavizar esto es con jerarquías de variables latentes. Jerarquizando tanto el modelo de inferencia (encoder) como el modelo generativo (decoder) se consiguen mejores probabilidades:

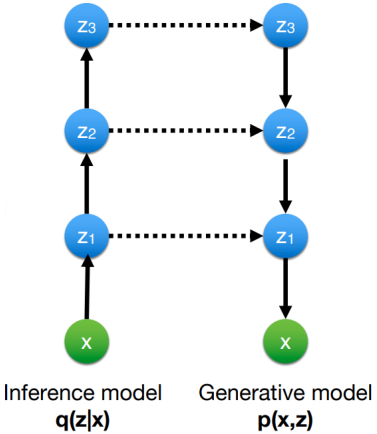
$$p(x) = \int p(x|z_1)p(z_1|z_2)...p(z_k)dz_{1:k} \tag{1}$$

$$\text{ELBO} = \mathbb{E}_{q(\cdot|x)}[\log p(x|z_{1:k})] - \text{KL}(q(z_{1:k}|x)||p(z)) \tag{2}$$

Hierarchical VAEs



"Flat" VAE



"Hierarchical" VAE

Hierarchical VAEs

Se han publicado muchos trabajos que exploran diferentes tipos de modelos HVAE, entre los que destacan VD-VAE (Very Deep VAE) [16] para generación de imágenes o Bit-swap [17] para compresión de datos.

La arquitectura de VD-VAE es un VAE convolucional simple. Para cada capa, el prior y el posterior son gaussianas diagonales. Descubrieron una técnica en el que el remuestreo del vecino más cercano (en el decoder) funcionaba mucho mejor que la convolución transpuesta y evitaba el posterior collapse. Esto permitió el entrenamiento con el ELBO original, sin necesidad de ninguno de los trucos discutidos anteriormente.

Hierarchical VAEs

En Bit-swap presentan un VAE jerárquico con el que demuestran que el esquema de compresión bits-back (BB-ANS) se puede usar con modelos de variables latentes para convertirse en esquemas de compresión eficientes.

Hierarchical VAEs y compresión

Jiang et al. en NPC (Non-Parametric learning by Compression) [18], propusieron un método basado compresión que se apoya en tres módulos reemplazables: un método de agregación, un compresor y una métrica de distancia. Utilizaron el VAE de Bit-Swap para realizar clasificación de imágenes, superando a métodos supervisados y semisupervisados.

En el blog de Jiang hay un par de posts donde explica esto y la posibilidad de utilizar modelos generativos para compresión y tareas posteriores como clasificación/regresión.

[1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[2] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

[3] A. Makhzani and B. Frey, "K-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013.

[4] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th international conference on international conference on machine learning*, pp. 833–840, 2011.

[5] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[6] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," in *International conference on machine learning*, pp. 3881–3890, PMLR, 2017.

[7] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, "Anomaly detection for time series using vae-lstm hybrid model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4322–4326, Ieee, 2020.

[8] N. Costa and L. Sánchez, "Variational encoding approach for interpretable assessment of remaining useful life estimation," *Reliability Engineering & System Safety*, vol. 222, p. 108353, 2022.

[9] N. Costa, L. Sanchez, and I. Couso, "Semi-supervised recurrent variational autoencoder approach for visual diagnosis of atrial fibrillation," *Ieee Access*, vol. 9, pp. 40227–40239, 2021.

[10] C. De Donno, S. Hediyeh-Zadeh, A. A. Moinfar, M. Wagenstetter, L. Zappia, M. Lotfollahi, and F. J. Theis, "Population-level integration of single-cell datasets enables multi-scale analysis across samples," *Nature Methods*, vol. 20, no. 11, pp. 1683–1692, 2023.

[11] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework.," *ICLR (Poster)*, vol. 3, 2017.

[12] S. Zhao, J. Song, and S. Ermon, "Infovae: Information maximizing variational autoencoders," *arXiv preprint arXiv:1706.02262*, 2017.

[13] M. Wu and N. Goodman, "Multimodal generative models for scalable weakly-supervised learning," *Advances in neural information processing systems*, vol. 31, 2018.

[14] Y. Shi, B. Paige, P. Torr, *et al.*, "Variational mixture-of-experts autoencoders for multi-modal deep generative models," *Advances in neural information processing systems*, vol. 32, 2019.

[15] G. Sejnova, M. Vavrecka, and K. Stepanova, "Benchmarking multimodal variational autoencoders: Cdsprites+ dataset and toolkit," 2023.

[16] R. Child, "Very deep vaes generalize autoregressive models and can outperform them on images," *arXiv preprint arXiv:2011.10650*, 2020.

[17] F. Kingma, P. Abbeel, and J. Ho, "Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables," in *International Conference on Machine Learning*, pp. 3408–3417, PMLR, 2019.

[18] Z. Jiang, Y. Dai, J. Xin, M. Li, and J. Lin, "Few-shot non-parametric learning with deep latent variable model," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26448–26461, 2022.