

Autocodificadores variacionales

Verónica E. Arriola-Rios

Facultad de Ciencias, UNAM

25 de mayo de 2023



Antecedentes

1 Antecedentes

2 Autocodificadores variacionales (VAE)

3 Entrenamiento

4 VAE según la teoría de la información

5 VAE condicional

Temas

1 Antecedentes

- Modelos generativos
- Antecesoras
- Conceptos

Modelos discriminativos vs generativos

En **aprendizaje de máquina** se distinguen dos tipos de modelos de conceptos:

Discriminativos Se conoce como modelos *discriminativos* a aquellos capaces de realizar tareas de **clasificación**, *reconociendo* a los miembros de distintas clases.

Generativos Un modelo *generativo* además es capaz de **producir** nuevos ejemplares que pudieran pertenecer a la clase, sin haberlos visto durante su entrenamiento.

Modelado generativo

Un modelo generativo **ideal** de imágenes de robots podría crear una imagen como la siguiente de la nada.

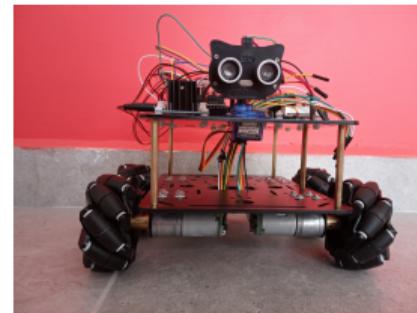


Figura: Imagen RGB 4160×3120 de un robot real.

Espacio \mathcal{X} Matrices $4160 \times 3120 \times 3$ con $x_{ijc} \in [0, 255]$.

Conjunto de datos \mathcal{X} Fotos de robots

Distribución $P(\mathcal{X})$ Distribución de probabilidad que asigna valores más altos a las matrices con imágenes de robots.

Temas

1 Antecedentes

- Modelos generativos
- **Antecesoras**
- Conceptos

Antecesoras

- Máquinas de Helmholtz
- Algoritmo vigilia-sueño (*Wake - sleep*)
- Máquinas de Boltzmann
- Redes de creencias profundas (*Deep Belief Nets DBN*)

Doersch 2021

Antecedentes

Dos teorías nutren el funcionamiento de los autocodificadores variacionales:

- Métodos variacionales Bayesianos.
- Teoría de la información.

Estado del arte: Midjourney



Figura: Chica bardo con gaita dibujada por un el modelo generativo Midjourney. ¿Pueden descubrir qué está mal?

El modelo debe capturar las dependencias entre pixeles: organización en objetos, texturas, iluminación, etc. para reducir el espacio a sólo imágenes que correspondan al concepto deseado.

$$\#\text{matrices posibles} \approx 255^{38937600} >>> \#\text{átomos en el universo} \approx 10^{82}$$

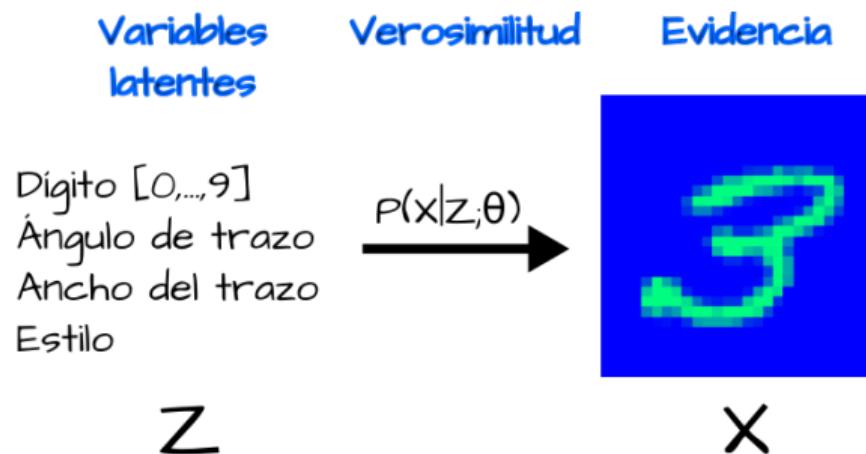
Temas

1 Antecedentes

- Modelos generativos
- Antecesoras
- Conceptos

Variables latentes

Para ilustrar el concepto, Doersch 2021 utiliza el ejemplo siguiente:



- Digamos que el sistema elige primero qué dígito dibujar, con qué estilo y a partir de ahí asigna su valor a cada pixel.
- A esta elección se le denomina *variable latente*, pues dada una imagen desconocemos el estado de las variables Z a partir de las cuales fue generada.

Proceso generativo

- ① Dada una función de densidad de probabilidad (PDF) $P(Z)$ definida sobre un espacio de alta dimensionalidad Z obtener por muestreo un vector Z .
- ② Sea $f(z; \theta) \rightarrow x$ una familia de funciones deterministas, parametrizadas por θ .
- ③ Encontrar los parámetros $\theta \in \Theta$ que hagan que la salida x sea semejante a los X en el conjunto de datos de entrenamiento.

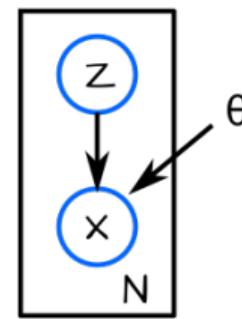


Figura: Muestreando vectores de variables latentes, es posible generar N muestras de X utilizando los mismos parámetros θ del modelo.

Entrenamiento por máxima verosimilitud

Se espera que hayan uno o más vectores z que lleven a que el modelo genere algo similar a X , por lo que la probabilidad de generar a cada X en el conjunto de entrenamiento se calcula sumando las contribuciones de todos estos posibles valores^[1]:

$$P(X) = \int P(X|z; \theta)P(z)dz$$

Se conoce como *máxima verosimilitud* a la técnica que se utilizará para maximizar $P(X|Z)$.

^[1]A esta suma se le llama *marginalización*.

Autocodificadores variacionales (VAE)

- 1 Antecedentes
- 2 Autocodificadores variacionales (VAE)
- 3 Entrenamiento
- 4 VAE según la teoría de la información
- 5 VAE condicional

Temas

② Autocodificadores variacionales (VAE)

- Fundamentos probabilistas
- Divergencia de Kullback-Leibler

¿Qué es un VAE?

“A diferencia de un codificador automático tradicional, que asigna la entrada a un vector latente, un Autocodificador variacional VAE asigna los datos de entrada a los parámetros de una distribución de probabilidad, como la media y la varianza de una Gaussiana. Este enfoque produce un espacio latente estructurado continuo, que es útil para la generación de imágenes.”

<https://www.tensorflow.org/tutorials/generative/cvae?hl=es-419>

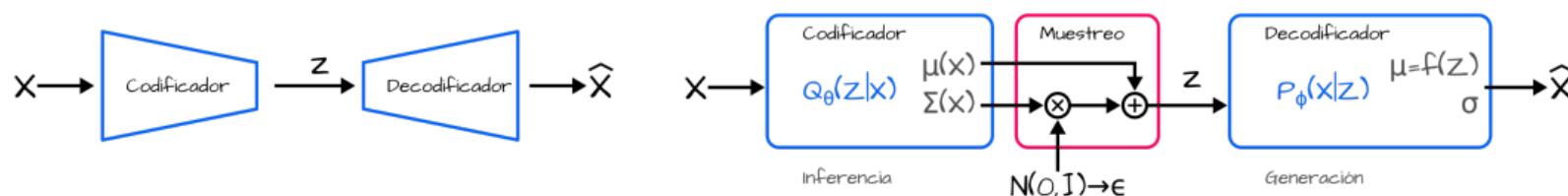


Figura: Izquierda: Autodificador tradicional. Derecha: Autocodificador variacional.

Hipótesis de VAE

Evita:

- Decidir manualmente qué información se codifica en cada dimensión de Z .
- Describir explícitamente las dependencias (estructura latente) entre las dimensiones de Z .

Para ello asume que:

- No existe una interpretación simple de las dimensiones de Z .
- Afirma que se pueden obtener muestras de Z a partir de la distribución normal

$$\mathcal{N}(0, I)$$

Objetivo

Optimizar:

$$P(X) = \int P(X|z; \theta)P(z)dz$$

$P(Z)$ se estima como

$$P(Z) = \mathcal{N}(0, I)$$

$P(X|Z)$ puede ser cualquier distribución de probabilidad con tal que sea computable y continua en sus parámetros θ .

En los VAE se suele utilizar una distribución Gaussiana:

$$P(X|Z; \theta) = \mathcal{N}(X|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)}$$

$$\mu = f(Z; \theta)$$

$$\Sigma = \sigma^2 I$$

Si los datos de entrada son binarios, se puede usar Bernoulli.

¿Por qué normales?

- Se quieren generar ejemplares semejantes a X , no idénticos.
- Para utilizar retropropagación, se requieren funciones continuas.

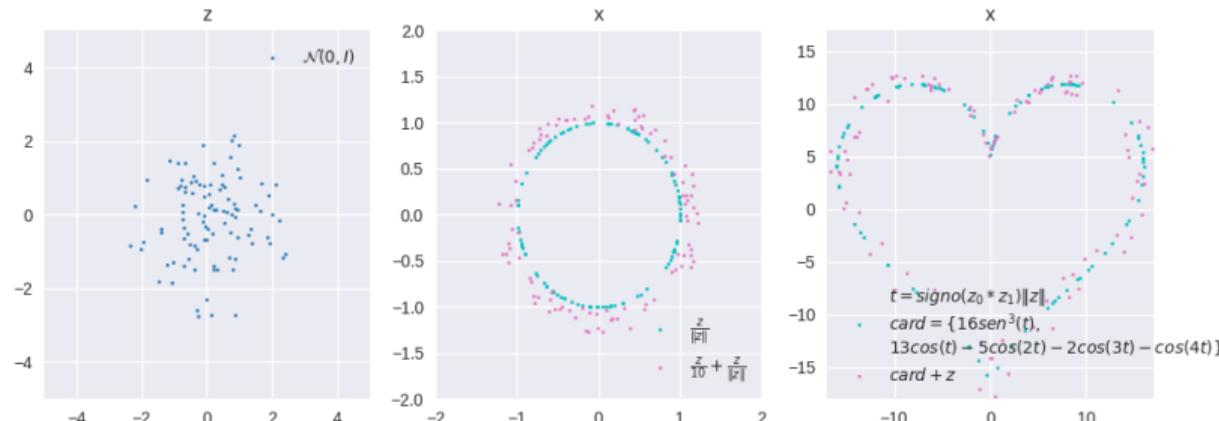


Figura: Mapeo de las muestras de la función normal a otros espacios.

Para un argumento más formal, revisar el muestreo de la transformación inversa, como sugiere Doersch 2021.

¿Verosimilitud ponderada?

Optimizar:

$$P(X) = \int P(X|z; \theta)P(z)dz$$

- Para evaluar la función objetivo $P(X)$ es necesario estimar la integral sobre todos los Z .
- Hay vectores Z que es más probable que contribuyan a X que otros.
- Es más eficiente estimar la integral si se usan sólo Z con contribuciones altas.
- Para ello serviría utilizar la función $Q(Z|X)$, que será una aproximación a $P(Z|X)$.

- Sin embargo, es necesario resolver **dos problemas**:
 - ① Para obtener X necesitamos muestrear a Z de $P(Z)$, no de $P(Z|X)$.
 - ② Muestrear requiere demasiadas operaciones.
- Soluciones:
 - ① No se optimiza exactamente $P(X)$, si no algo semejante obtenido usando teoría de la información.
 - ② Se aproxima el proceso de muestreo con las características del descenso por el gradiente estocástico aplicado a la función que se va a optimizar.

Inferencia variacional

- La *inferencia variacional* aproxima a la función $P(Z|X)$ mediante una familia de distribuciones $Q_\lambda(Z|X)$.
- El *parámetro variacional* λ indexa a la familia de distribuciones.

Por ejemplo, si Q es una Gaussiana, $\lambda_{x_i} = (\mu_{x_i}, \sigma_{x_i}^2)$

(Altosaar 2016)

Temas

② Autocodificadores variacionales (VAE)

- Fundamentos probabilistas
- Divergencia de Kullback-Leibler

Divergencia de Kullback-Leibler

- “La *divergencia de Kullback-Leibler* (KL), también conocida como **divergencia de la información, ganancia de la información o entropía relativa**, mide el número esperado de extra bits requeridos en muestras de código de P cuando se usa un código basado en Q, en lugar de un código basado en P.”^[2]

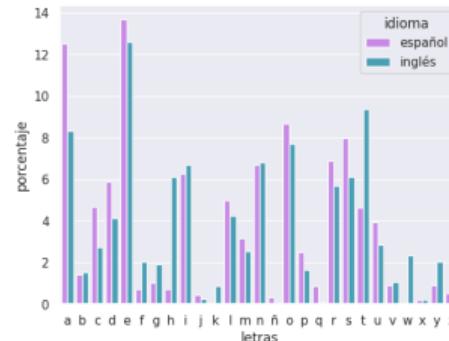


Figura: Una codificación óptima asigna códigos más cortos a los ejemplares más frecuentes. Cuando se codifica con respecto a la distribución equivocada se requieren más bits.

[2] https://es.wikipedia.org/wiki/Divergencia_de_Kullback-Leibler

- “Generalmente P representa la *verdadera* distribución de los datos, observaciones, o cualquier distribución teórica. La medida Q generalmente representa una teoría, modelo, descripción o aproximación de P.”^[2]
- Hay una descripción muy curiosa de este concepto con gusanos extraterrestres en:

<https://datascience.eu/es/aprendizaje-automatico/explicacion-de-la-divergencia-kullback-leibler/>

Definición (Divergencia de Kullback-Leibler)

Sean P y Q las distribuciones de dos variables aleatorias continuas, con p y q sus densidades, entonces la *Divergencia de Kullback-Leibler (KL) \mathcal{D}* se define como:

$$\begin{aligned}\mathcal{D}(P\|Q) &= \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx \\ &= \int_{-\infty}^{\infty} p(x)[\ln p(x) - \ln q(x)] dx\end{aligned}$$

- Las unidades son bits si el logaritmo es base 2,
- se llaman *nats* si la base es e .

Entrenamiento

- 1 Antecedentes
- 2 Autocodificadores variacionales (VAE)
- 3 Entrenamiento
- 4 VAE según la teoría de la información
- 5 VAE condicional

Temas

3 Entrenamiento

- ELBO
- Estimación del gradiente

Divergencia KL para el VAE

Optimizar:

$$P(X) = \int P(X|z; \theta)P(z)dz$$

- Considerando que queremos tomar z de $Q(Z|X)$ se define la divergencia KL entre $P(Z|X)$ y una función arbitraria $QA(Z)$ como:

$$\mathcal{D}[QA(Z)\|P(Z|X)] = E_{z \sim QA} [\log QA(Z) - \log P(Z|X)]$$

recordando que $E_{z \sim QA}$ en el lado derecho es^[3]:

$$E_{z \sim QA} = \int_{-\infty}^{\infty} QA(z)[\log QA(Z) - \log P(Z|X)]dz$$

^[3]Según la definición de valor esperado y la ley del estadístico inconsciente.

<https://blog.nekomath.com/proba1-valor-esperado-de-una-variable-aleatoria/>

<http://blog.espol.edu.ec/estg1003/valor-esperado-de-una-funcion/>

Función de pérdida: -ELBO

- Eligiendo $Q_A(Z)$ como $Q(Z|X)$, utilizando Bayes y despejando, se llega a:
(Doersch 2021)

$$\log P(X) - \mathcal{D}[Q(Z|X)\|P(Z|X)] = E_{Z \sim Q}[\log P(X|Z)] - \mathcal{D}[Q(Z|X)\|P(Z)] \quad (1)$$

- Se conoce como *Cota inferior a la evidencia (Evidence Lower BOund) ELBO* al lado derecho de esta ecuación:

$$ELBO = E_{Z \sim Q}[\log P(X|Z)] - \mathcal{D}[Q(Z|X)\|P(Z)] \quad (2)$$

- La función de pérdida entonces es $-\text{ELBO}$: (Altosaar 2016)

$$L = \sum_{i=1}^N l_i$$

$$l_i(\theta, \phi) = -E_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] + \mathcal{D}[q_\theta(z|x_i)\|p(z)]$$

donde N es el número total de datos en X .

Optimización

- Se busca minimizar la pérdida con descenso por el gradiente estocástico, considerando los ejemplares de entrenamiento X :

$$\mathbb{E}_{X \sim D} [\log P(X) - \mathcal{D}[Q(Z|X)\|P(Z|X)]] = \mathbb{E}_{X \sim D} [\mathbb{E}_{Z \sim Q} [\log P(X|Z)] - \mathcal{D}[Q(Z|X)\|P(Z)]]$$

- Dado que cada x_i en X tiene sus propios parámetros, el gradiente puede entrar a las integrales y calcularse para:

$$\log P(X|Z) - \mathcal{D}[Q(Z|X)\|P(Z)] \tag{3}$$

- Así ha aparecido al estructura de un **autocodificador**: Q codifica a X en Z y P decodifica Z para reconstruir X

Temas

3 Entrenamiento

- ELBO
- Estimación del gradiente

Divergencia

$$\nabla_{\theta, \phi} [\log P_\phi(X|Z) - D[Q_\theta(Z|X) \| P(Z)]] \quad (4)$$

- La divergencia para dos Gaussianas se puede calcular (Doersch 2021):

$$D[\mathcal{N}(\mu(X), \sigma(X)) \| \mathcal{N}(0, I)] = \frac{1}{2} (\text{tr}(\Sigma(X)) + \mu(X)^T \mu(X) - k - \log |\Sigma(X)|)$$

donde k es la dimensionalidad de la distribución y $|\Sigma|$ indica determinante.

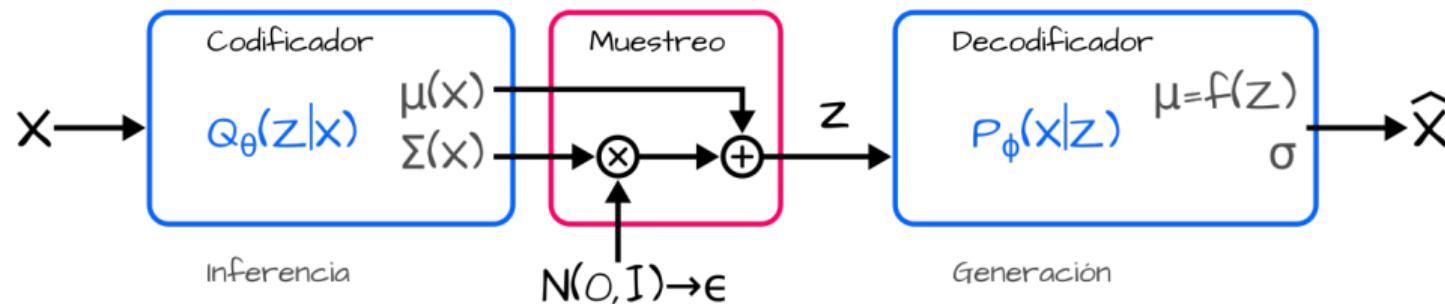
Verosimilitud

- **Reparametrización:** Para calcular $\nabla_{Z \sim Q} [\log P(X|Z)]$ es necesario muestrear Z a partir de $Q(Z|X)$. Dado que es posible calcular el gradiente para entradas aleatorias, pero no de unidades estocásticas dentro de la red, se reparametriza el cálculo de $Q(Z|X)$ de modo que:
 - 1 la muestra sea extraída de una distribución sin parámetros entrenables y
 - 2 sobre ella se aplique alguna transformación determinista, cuyos parámetros sí se entranan.

Para la función Gaussiana con Σ una matriz diagonal esto es posible pues:

$$\mathcal{N}(\mu, \Sigma) = \mu(X) + \Sigma^{1/2}(X)\mathcal{N}(0, I)$$

Codificador/Decodificador



En el ejemplo de la generación de imágenes:

- El decodificador debe dar a la salida tantos juegos de valores para los parámetros como pixeles hay en la imagen a reconstruir. (Altosaar 2016)

Generación

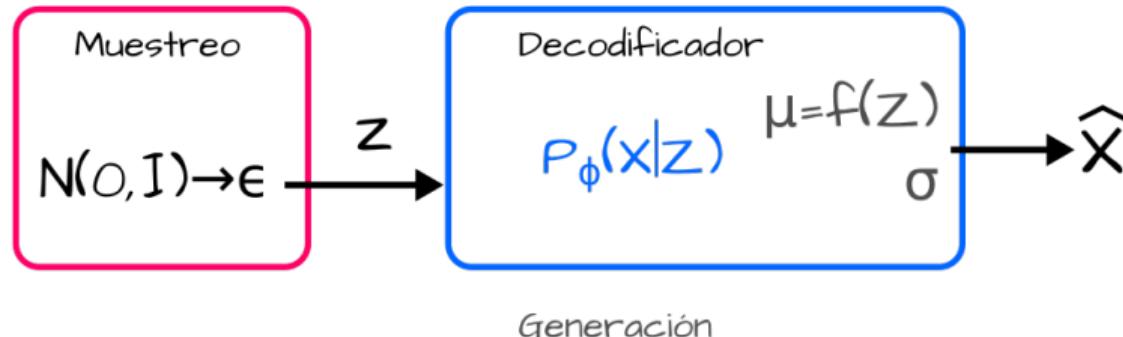


Figura: Para generar ejemplares nuevos la parte codificadora se desconecta y Z se muestrea de una Gaussiana con media cero y varianza uno.

Evaluación

- Aún no es tratable el evaluar la **probabilidad de que el modelo produzca algún ejemplar de prueba.**
- Dado que $\mathcal{D}[Q(Z|X)\|P(Z|X)] > \varepsilon$ con $\varepsilon > 0$, es posible usar el lado de derecho de la ecuación (1) para obtener una cota inferior a $\log P(X)$. De ahí el nombre ELBO.

$$\text{ELBO} = E_{Z \sim Q}[\log P(X|Z)] - \mathcal{D}[Q(Z|X)\|P(Z)] \quad (2 \text{ revisada})$$

- Esto esto es útil para detectar qué tan bien captura el modelo a algún ejemplar X.

Implementaciones

La implementación más clara y fácil de seguir se encuentra en un tutorial de Tensorflow. Utilizan redes convolucionales para estimar $Q(Z|X)$ y $P(X|Z)$:

<https://www.tensorflow.org/tutorials/generative/cvae?hl=es-419>

Otra implementación de un VAE para generar dígitos se encuentra en:

https://github.com/cdoersch/vae_tutorial

VAE según la teoría de la información

- 1 Antecedentes
- 2 Autocodificadores variacionales (VAE)
- 3 Entrenamiento
- 4 VAE según la teoría de la información
- 5 VAE condicional

Interpretación

$$\log P(X) - \mathcal{D}[Q(Z|X)\|P(Z|X)] = E_{Z \sim Q}[\log P(X|Z)] - \mathcal{D}[Q(Z|X)\|P(Z)] \quad (1 \text{ revisada})$$

Es posible interpretar cada término de la forma siguiente: (Doersch 2021)

$\log P(X)$ Número total de bits que se requieren para construir una X dada bajo nuestro modelo, utilizando una codificación ideal.

$\log P(X|Z)$ Mide la cantidad de información requerida para reconstruir X a partir de Z con una codificación ideal.

$\mathcal{D}[Q(Z|X)\|P(Z)]$ Información extra que se requerirá para obtener Z a partir de $Q(Z|X)$ en lugar de $P(Z)$.

Se puede pensar que es información extra que obtenemos acerca de X .

Dado que $P(Z) = \mathcal{N}(0, I)$, obliga a que $Q \sim \mathcal{N}$, esto tiene el efecto de mantener representaciones de objetos de la misma clase en regiones cercanas de Z .

El número total de bits $-\log P(X)$ es la suma de estos dos pasos menos la penalización debida a que Q no es una codificación óptima $\mathcal{D}[Q(Z|X)\|P(Z|X)]$.

VAE condicional

- 1 Antecedentes
- 2 Autocodificadores variacionales (VAE)
- 3 Entrenamiento
- 4 VAE según la teoría de la información
- 5 VAE condicional

VAE condicional

- Condiciona el proceso generativo completo a una entrada.
- *Ejemplo:* Dada una imagen remover una parte y llenar con contenido coherente.

Definición

Dada una entrada X y una salida Y , crear un modelo $P(Y|X)$ que maximize esta probabilidad^a

^aY es lo que solía ser nuestra X.

$$\log P(Y|X) - \mathcal{D}[Q(Z|Y, X)\|P(Z|X)] = E_{Z \sim Q(\cdot|Y, X)}[\log P(Y|Z, X)] - \mathcal{D}[Q(Z|Y, X)\|P(Z|X)] \quad (5)$$

Diagrama

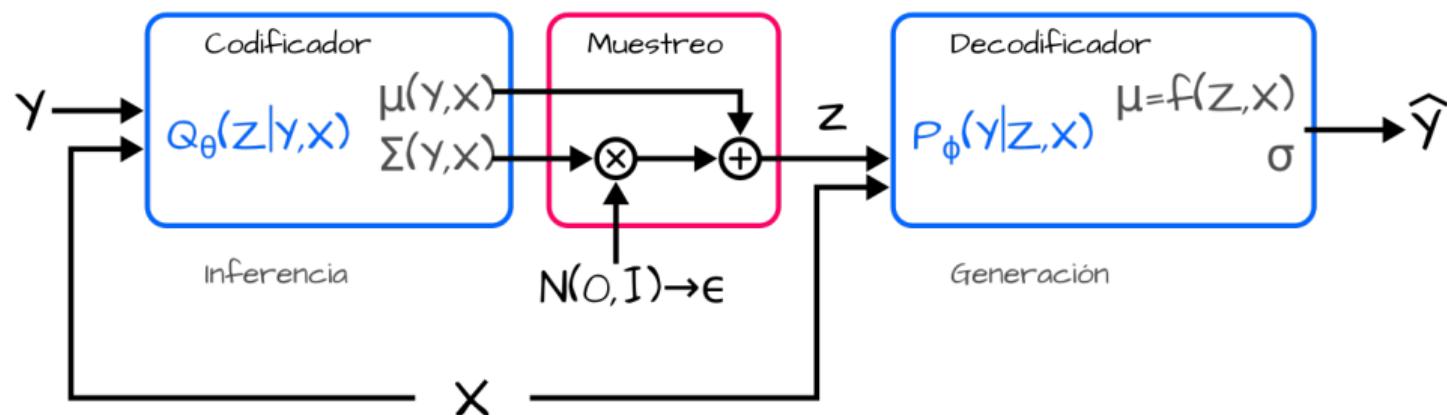


Figura: El sistema ahora depende también de una entrada fija.

Referencias I

-  Altosaar, Jaan (2016). «Tutorial - What is a Variational Autoencoder?» en. En: DOI: 10.5281/ZENODO.4462916. URL:
<https://jaan.io/what-is-variational-autoencoder-vae-tutorial/>.
-  Doersch, Carl (ene. de 2021). *Tutorial on Variational Autoencoders*. Carnegie Mellon/UC Berkeley. URL: <https://arxiv.org/pdf/1606.05908.pdf>.
-  Rivera, Mariano (2018). *Autocodificadores Variacionales*. URL:
http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_profundo/vae/vae.html.

Licencia

Creative Commons
Atribución-No Comercial-Compartir Igual

