

Codificación de soluciones del VRP a un espacio continuo

Dalianys Pérez Perera

Capítulo 1

Preliminares

Cada capítulo debería empezar con un breve resumito de qué trata, para ubicar al lector en lo que va a leer en ese momento. Sería bueno poner uno aquí :-).

Los avances (recientes¹) en el campo de *machine learning* han permitido a las (redes neuronales²) realizar una amplia variedad de tareas, entre ellas resolver problemas de optimización combinatoria. La mayoría de estos problemas, incluyendo el VRP, son difíciles de solucionar, desde un punto de vista teórico por ser NP-Duro, y desde un punto de vista práctico por lo costoso que resultan las técnicas computacionales aplicadas. (Se han examinado³) dos líneas principales con el uso de ML, una de ellas con el fin de reemplazar el excesivo cómputo por una rápida aproximación; mientras que la otra se basa en explorar el espacio de búsqueda y (aprender nuevas políticas⁴). En efecto, el empleo de (*deep reinforcement learning*⁵) ha devenido en un gran interés para aprender heurísticas automáticamente en problemas de tal complejidad.

Los problemas de enrutamiento como el TSP y el VRP, se sitúan entre los problemas de optimización más estudiados y con mayores intentos de solución en la práctica [REF (cualquiera de las que ya tienes)]. La aplicación de redes neuronales para optimizar decisiones en problemas de optimización combinatoria se remonta a ¿1982? Hopfield et al. [17]; llegando a resolver pequeñas instancias del TSP de hasta 30 ciudades. El desarrollo de (arquitecturas de redes neuronales⁶) ha allanado el camino a enfoques competitivos basados en ML no solo para dar solución a problemas de enrutamiento, sino también para inspeccionar (el espacio de búsqueda continuo que conforman un conjunto de soluciones.⁷)

En este capítulo se presenta una introducción a las técnicas (más novedosas⁸) en el campo de la Inteligencia Artificial para comprender la solución propuesta y las existentes en el estado del arte ¿del?(al) problema en cuestión. Principalmente, se enfatiza en los modelos del tipo *encoder-decoder* cuya idea constituye la esencia del objetivo de la tesis. Primero se comienza con una visión general del problema VRP y algunas de sus variantes, para luego dar paso a la teoría de *Autoencoders* y sus aplicaciones en problemas de enrutamiento.

Ups. Acabo de darme cuenta de que esto que acabo de leer es la introducción al capítulo-

1: Esto es "casi" un adjetivo. ¿Reciente de cuándo?

2: En algún lugar sería bueno poner qué son las redes neuronales, pero muy sencillito: una función de R^n en R^m que depende de varios parámetros y que con los parámetros apropiados pueden aproximar cualquier función. Esto les ha permitido realizar una amplia variedad de tareas...

3: ¿Dónde? ¿cuándo? ¿por quién? ¿para qué?

4: ¿Qué significa aprender políticas en el contexto de la optimización? :-)

5: ¿qué es esto? :-). Quizás sea bueno poner una sección en la que presentes y describas brevemente esos elementos para después poderlos mencionar sin problemas.

6: Yo sé qué es y qué significa, pero "arquitectura de redes neuronales" es algo que incorporaría a la sección en la que explicas brevemente cada uno de los elementos.

7: En el departamento hay varias profesoras que si caen en el tribunal te van a preguntar qué significa el espacio de búsqueda continuo asociado a un problema discreto :-/.

8: ¿adjetivos? :-/ Además de que tienen el inconveniente extra de que en unos años ya no será cierto :-/.

lo que dije allá arriba que pusieras :-/. Quizás sea mejor decir más rápido que esa es la introducción al capítulo. O sea, ¿empezar con parecido al último párrafo?

1.1. Problema de Enrutamiento de Vehículos

Una solución del VRP requiere la determinación de un conjunto de rutas donde cada una se realiza por un solo vehículo. Las rutas comienzan y terminan en el depósito y están conformadas por una secuencia de clientes que deben ser visitados ~~en ese?~~(~~respetando el~~) orden. La disposición de los clientes dentro del conjunto de rutas debe cumplir con las (disposiciones operativas⁹), las demandas de los mismos y además, minimizar el costo de transportación u otro criterio [REF].

En un gran número de problemas derivados del VRP se asume que la mercancía a distribuir y la flota de vehículos se encuentran inicialmente en los depósitos. La cantidad de depósitos así como su capacidad para atender a todos o solo a un subconjunto de los clientes determinan las estrategias a seguir en la planificación de las rutas.

Los clientes contienen toda la información relativa al servicio que solicitan. Pueden estar caracterizados por la cantidad y tipo de mercancía que requieren, horarios de atención y tipos de servicios. Se denota por $I = \{1, 2, \dots, n\}$ el conjunto de clientes, ($I^+ = \{0\}$ ¹⁰) el conjunto formado por el depósito. Luego para cada par de elementos ($i, j \in I^+$ ¹¹) se define el arco (i, j) con costo asociado c_{ij} . Las características de cada problema determinan el concepto de costo y factibilidad de las rutas. Para el (VRP¹²), el costo de una ruta se calcula a partir de la suma de los costos de los arcos involucrados en la misma [REF].

En las formulaciones más simples se asume que la flota la componen vehículos idénticos de capacidad Q y en algunos casos se supone que su número es ilimitado. En problemas más complejos, la flota puede estar compuesta por vehículos diferentes en cuanto a su capacidad, costo fijo y/o variable, longitud máxima de los recorridos que pueden realizar, disponibilidad, etc...

Atendiendo a las restricciones que se consideran, los problemas de enrutamiento de vehículos se agrupan en:

VRP con restricciones de capacidad (CVRP) es la variante más estudiada hasta la actualidad según Toth y Vigo en [29]. Todos los vehículos de la flota son idénticos y con capacidad limitada, por lo que durante el recorrido no se debe exceder la capacidad de ninguno de ellos.

VRP con flota heterogénea (HVRP) la flota de vehículos no es homogénea, pues pueden presentar diferentes características.

VRP con múltiples depósitos (MDVRP) existen varios depósitos desde los cuales pueden comenzar y terminar las rutas. La determinación de los depósitos de partida y fin del recorrido son también parte de la solución del problema.

9: ¿qué es esto? :-/. Cada vez que encuentre algo que no recuerde haber leído qué significa te voy a poner un comentario de ¿qué es esto? Es el equivalente a "variable no definida" en los lenguajes de programación :-P. En este caso ¿no se pudiera poner que deben cumplir con las restricciones del problema?

10: Usualmente I^+ es conjunto de los clientes más el depósito

11: Con la definición de I^+ que diste esto no tiene mucho sentido. Creo que faltó $\cup I$ en la definición de I^+ .

12: Por lo general VRP se usa para el conjunto de todos los problemas de enrutamiento de vehículos. Creo que aquí debería ser más específica. Por ejemplo el CVRP (VRP con restricciones de capacidad)... aunque no es el único. En cualquier caso, creo que no debería ser el VRP, así general. También puedes poner, como haces más abajo: en las formulaciones más simples :-).

VRP con ventanas de tiempo (VRPTW) es una extensión del CVRP [29] donde además de la restricción de capacidad, los clientes establecen un horario conocido como ventana de tiempo en el cual se realizará el servicio.

VRP con recogida y entrega (VRPPD) los clientes solicitan el servicio de recogida y/o entrega de mercancía.

El **objetivo de un VRP** es minimizar o maximizar determinada función. Según las características del problema las funciones de optimización más frecuentes son:

- Minimizar el costo de transportación a partir de la distancia recorrida.
- Minimizar la cantidad de vehículos.
- Minimizar los costos totales, incluidos los costos fijos y los costos variables.
- Optimización jerárquica de la función objetivo: el número de vehículos es el objetivo principal de optimización, y luego se optimizará la distancia de viaje del vehículo correspondiente.

(La optimización de múltiples objetivos se refiere principalmente al VRP que tiene más de un objetivo que debe optimizarse al mismo tiempo. En la práctica, muchos problemas de enrutamiento de vehículos son problemas de optimización de toma de decisiones con objetivos múltiples.¹³)

Aquí (y entre cada par de secciones) sería bueno poner un párrafo que le recuerde al lector tres cosas: qué acaba de leer, qué tiene que ver eso con el objetivo del trabajo, y qué tiene que ver lo que acaba de leer con lo que sigue. Algo como eso debería estar entre cada par de secciones. Porque si no, es muy brusca la transición de una sección a otra... Y sí, yo sé :-), hay una pila de artículos que son así (bruscos), pero eso no significa que nosotros debamos serlo también ;-). Además, por lo general ese parrafito ayuda a la lectura.

13: Esto es verdad, pero como no tiene mucho que ver con tu trabajo, yo te propondría quitarlo, porque si lo dejas, provocas más preguntas de las que respondes, y realmente si se quita no cambia mucho el resto del trabajo, y cada vez que pase eso (que si algo se quita no cambia mucho el resto del trabajo) se debería quitar :-).

1.2. Autoencoders

Muchos (algoritmos de aprendizaje profundo no supervisado¹⁴) se basan en la idea de un (autoencoders¹⁵), así refieren Russell y Norvig en [27]. Estos modelos intentan obtener una representación o código de los datos de entrada de forma tal que la transformación sea provechosa en determinada tarea. Los autoencoders pueden aprender propiedades de los datos sin la necesidad de (labels¹⁶) o etiquetas anotadas por humanos, (por esta razón es que son clasificados como algoritmos de aprendizaje no supervisado.¹⁷)

14: Algo más que mandaría a la sección de definiciones y/o explicaciones elementales. El problema fundamental es que no todos los lectores de tu tesis tienen que ser especialistas en ML :-). De hecho, es posible que algunos de ellos se enteren de esas cosas con tu tesis :-o.

15: Para la sección de explicaciones elementales

16: Esta label yo la quietaría. Creo que etiqueta no es tan terrible... ¿o sí? :-/.

17: Mmmm... cositas como estas es las que yo pondría en la sección dedicada a la introducción al ML, que "en espíritu" pudiera llamarse: introducción al machine learning para dummies... pero solo "en espíritu". No creo que nos dejen ponerle ese título a una sección de la tesis :-/... ¡Cuánta incompreensión! :-P

La idea de los *autoencoders* ha sido parte del panorama histórico de las redes neuronales ((durante decadas)¹⁸) desde 1980 [22, 16]. Tradicionalmente han sido usados para (reduccion de dimension o extraccion de características)¹⁹. (Recientemente²⁰), las conexiones teóricas entre los modelos de (variable latente²¹) y *autoencoders* han puesto a estos últimos a la vanguardia de los (modelos generativos²²). Pueden ser considerados como un caso especial de (redes feedforward²³) y ¿por lo tanto?(ademas), ser entrenados con las mismas técnicas de descenso por(del) gradiente y *back-propagation*.

18: Te propongo quitarlo porque se puede inferir de la frase "desde 1980" :-).

19: Para la sección

20: ¿cuán recientemente?

21: ¿qué esto? :-o

22: Para la sección

23: Je, je... con esto empieza la sección "ML para dummies" :-)

Un *autoencoders* es un modelo que contiene dos partes:

Encoder: función f que transforma la entrada x en un vector latente \hat{z} de menor dimensión, $\hat{z} = f(x)$. Como el vector latente es de baja dimensión, el *encoder* es forzado a aprender solo las características más importantes de los datos de entrada.

Decoder: Trata de reproducir la entrada \tilde{x} a partir del vector latente \hat{z} , $\tilde{x} = g(\hat{z})$. Aunque \hat{z} tiene una menor dimensión, posee el tamaño suficiente para permitir al *decoder* recuperar la entrada.

Hay algo que algunas personas llaman "voz pasiva", y es cuando se escribe usando los participios. En los artículos en inglés se usa mucho, y de hecho, hay revistas que te obligan a escribir así :-o, pero por lo general no se considera buen estilo de escritura, sobre todo porque nadie habla así :-/. Nadie dice: La habitación fue entrada por mí, después de que la puerta fue abierta, también por mí. Digo :-o, yo espero que tú no hables así :-/. Por lo general tú dices: abrí y la puerta y entré. Uno los problemas que tiene la voz pasiva es que obliga a que la lectura sea más lenta. En fin, que "tremenda muela bizca" :-), pero en resumen, siempre que se pueda, es preferible evitar la voz pasiva. Por eso, te voy a señalar algunos fragmentos que estén con voz pasiva y que se pueden cambiar a "la forma normal como la gente habla" sin mucho problema.

El modelo (es entrenado²⁴) de forma tal que $x \approx g(\hat{z})$, o sea, el proceso de codificación sea revertido por el decodificador lo mejor posible. Para los (modelos generativos²⁵) no resulta especialmente útil conseguir que $g(f(x)) = x$ en todo momento (¿por qué?²⁶). En cambio, se espera que al entrenar el modelo para que genere la misma entrada, se obtenga una representación capaz de agrupar aspectos y propiedades significativas de los datos [13].

24: Voz pasiva

25: Para la sección

26: Cuando veas una pregunta como esta así, aparentemente salida de la nada, es que me surgió cuando estaba leyendo. Bueno, no solo a mí, sino a un posible lector genérico. Y no es buena idea que el lector vaya por ahí haciéndose preguntas que no son las que a ti te interesan. Quizás en este caso sea bueno explicarlo brevemente.

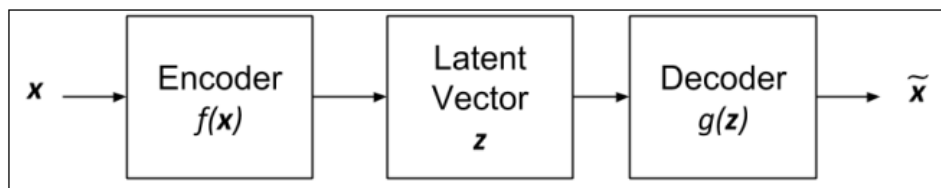


Figura 1.1: Diagrama general de un *autoencoders*

Un ejemplo de *autoencoders* muy simple es el *autoencoders* lineal, donde f y g son lineales

y comparten la matriz de pesos W :

$$\begin{aligned}\hat{z} &= f(x) = Wx \\ \tilde{x} &= g(\hat{z}) = W^\top \hat{z}\end{aligned}$$

El proceso de aprendizaje se describe como minimizar una función

$$\mathcal{L}(x, g(f(x))), \quad (1.1)$$

donde \mathcal{L} es una función de pérdida que penaliza a $g(f(x))$ por ser diferente de x . Un ejemplo sería calculando el error cuadrático medio $\sum_j \|x_j - g(f(x_j))\|^2$.

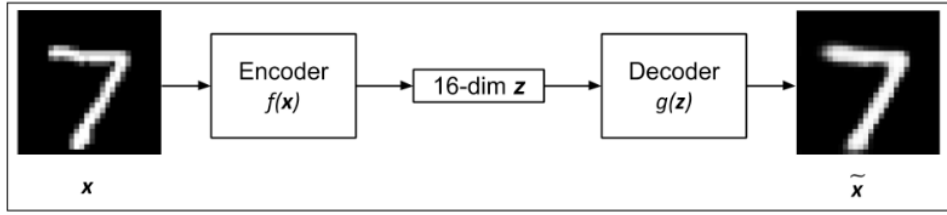


Figura 1.2: Un *autoencoders* con dígitos de MNIST como entrada y salida. El espacio latente es de dimensión 16.

Si la entrada (es tratada²⁷) como una distribución, se puede interpretar el *encoder* como un codificador de la distribución $p(z|x)$, y el *decoder*, como un decodificador de la distribución $p(x|z)$. La función de pérdida del *autoencoders* (es expresada²⁸) como:

$$\mathcal{L} = -\log p(x|z) \quad (1.2)$$

Luego, si se asume que la distribución de la salida del *decoder* es Normal, entonces la función de pérdida se traduce en:

$$\mathcal{L} = -\log p(x|z) = -\log \prod_{i=1}^m \mathcal{N}(x_i, \tilde{x}_i, \sigma^2) = -\sum_{i=1}^m \log \mathcal{N}(x_i, \tilde{x}_i, \sigma^2) \propto \sum_{i=1}^m (x_i - \tilde{x}_i)^2 \quad (1.3)$$

En la ecuación (anterior) 1.3, $\mathcal{N}(x_i, \tilde{x}_i, \sigma^2)$ representa una distribución Normal con media \tilde{x}_i y varianza σ^2 , siendo esta última constante. La salida del *decoder* \tilde{x}_i se asume independiente y de dimensión m .

Algunos ejemplos de estos modelos son los *regularized autoencoders* que son efectivos en el aprendizaje de representaciones para tareas de clasificación y los *variational autoencoders*, con aplicaciones como modelos generativos. **NUEVO PÁRRAFO.**

Los *autoencoders* se aplican a muchos problemas, desde el reconocimiento facial, detección de características, detección de anomalías, traducción automática [4, 28] hasta generar aleatoriamente nuevos datos que son similares a los datos de entrada o de entrenamiento.

27: Voz pasiva

28: Voz pasiva

TODO: citar ejemplos de los usos anteriores

Aquí hace falta una transición a la sección siguiente. Recuerda: ¿qué acabo de leer?, ¿por qué es relevante? ¿qué tiene que ver con lo que sigue? Pude ser algo muy sencillo como: uno de los autoencoders que “no sé qué” son los Regularized Autoencoders que se presentan a continuación. O dentro de los autoencoders, los Regularized Autoencoders “son relevantes por algo” y se presentan en la siguiente sección.

1.2.1. Regularized Autoencoders

(Idealmente, se pudiera entrenar satisfactoriamente cualquier arquitectura *autoencoders* seleccionando la dimensión del código y las capacidades del *encoder* y el *decoder* basadas en la complejidad de la distribución que se va a modelar.²⁹) No obstante, dado que se desea que el modelo descubra atributos latentes dentro de los datos, es importante asegurar que el *autoencoders* no esté aprendiendo simplemente una forma eficiente de memorizar los datos de entrenamiento. Similar a los problemas de aprendizaje **supervizado**, existen varias técnicas de regularización en la red neuronal con el objetivo de garantizar buenas propiedades de generalización.

Los *regularized autoencoders* proporcionan (esta capacidad³⁰), mediante una función de pérdida que le otorga al modelo otras propiedades además del sentido de recuperar la misma entrada en la salida. Entre estas propiedades se puede mencionar (la dispersión en la capa de representación³¹), (valores muy pequeños de la derivada del vector latente y manejo del ruido o falta de información en las entradas³²).

Un *sparse autoencoder* (SAE) es (simplemente) un (*autoencoders*³³) cuyo criterio de aprendizaje envuelve una función de penalidad $\Omega(z)$ sobre la dispersión de la capa de código o codificación \hat{z} en conjunto con el error de reconstrucción.

$$\mathcal{L}(x, \tilde{x}) + \Omega(\hat{z}) \quad (1.4)$$

En la ecuación anterior falta algo... un operador... Eso es una definición de algo (sospecho que puede ser la función de pérdida de los SAE), pero bueno, la idea es que ahí falta algo. Es como si hubieras puesto: $8 + 2$:-/.

Estos modelos (son típicamente usados³⁴) para aprender particularidades en otro tipo de tareas como la clasificación. La idea intuitiva detrás de los mismos es forzarlo a activar algunas regiones de la red selectivamente en dependencia de los datos de entrada, de manera que (sean extraídas³⁵) la mayor cantidad de peculiaridades o al menos las más importantes.

Existen dos formas de imponer la restricción de dispersión; ambas implican medir las activaciones de la capa oculta para cada (*batch*³⁶) de entrenamiento y añadir algún término a la función de pérdida con el fin de penalizar excesivas activaciones. Estos términos son:

- **Regularización L1:** penaliza el valor absoluto del vector de activaciones a en la capa (\hat{z} ³⁷) para la observación i , escalado por un parámetro de ajuste λ .

29: Esta oración me costó trabajo leerla y entenderla :-/. ¿Qué quieres decir con ella? ¿Puedes describirla con otras palabras?

30: ¿cuál? ¿propiedades de generalización?

31: ¿qué es esto? :-/ (Fíjate que aquí yo no sé, y no me queda claro si debería ir a la sección para Dummies o es que se debería explicar mejor aquí :-/).

32: ¿por qué esto es relevante o importante?

33: ¿con s o sin s?

34: Voz pasiva

35: Voz pasiva, ¿que se puedan extraer?

36: Para la sección.

37: ¿ \hat{z} no era el resultado del encoder? ¿O aquí se usa con otro significado?

$$\mathcal{L}(x, \tilde{x}) + \lambda \sum_i |a_i^{(\hat{z})}| \quad (1.5)$$

En la ecuación anterior vuelve a faltar algo :-(. Es otra vez como 2 + 5 :-(.

- **Divergencia de Kullback-Leibler (KL):** (~~en-esencia~~) es una medida no simétrica de similitud o diferencia entre dos funciones de distribución de probabilidad. Se define a ρ como un parámetro de dispersión, el cual denota el promedio de activación de una neurona sobre una colección de observaciones. Su valor puede ser aproximado mediante la fórmula ($\hat{\rho}_j = \frac{1}{m} \sum_i [a_i^{(\hat{z})}(x)]$ ³⁸) donde el subíndice j denota una neurona específica en la capa (\hat{z} ³⁹), y m la cantidad de observaciones x . Finalmente el término agregado resulta en:

$$\mathcal{L}(x, \tilde{x}) + \sum_j KL(\rho || \hat{\rho}_j) \quad (1.6)$$

38: ¿qué es $z_j^{(\hat{z})}$? :-(.

39: Mmm... dos veces ya no debe ser coincidencia... Entonces :-(. me perdí la parte en que \hat{z} se convirtió en capa de la red :-(.

Falta algo en la ecuación, probablemente el miembro izquierdo. De hecho, ahora mismo ni siquiera es una ecuación :-/, pero debería serlo.

siendo ρ la distribución (idealmente presente en las observaciones de los nodos de la capa oculta⁴⁰).

40: ¿qué significa esto?

Otro enfoque para desarrollar un modelo generalizable es corromper ligeramente los datos de entrada, o sea, añadirle ruido a la información, manteniendo los datos originales como objetivo. Conocidos como *denoising autoencoders* (DAE) se orientan a minimizar la función $\mathcal{L}(x, g(f(\tilde{x})))$ donde \tilde{x} es una copia de x alterada por alguna variante de ruido. Esta técnica garantiza que el modelo no pueda simplemente desarrollar un mapeo que memorice los datos de entrenamiento porque en esta versión no son iguales la entrada y la salida.

Se esperaría que para entradas muy similares, la codificación aprendida también fuera muy similar. El modelo podría entrenarse explícitamente para que este sea el caso, requiriendo que la derivada de las activaciones de la capa oculta sea pequeña con respecto a la entrada. En otras palabras, que para pequeños cambios en la entrada, se mantenga un estado codificado muy similar. Esta otra estrategia de regularización da lugar a los *contractive autoencoders*. Su idea es bastante parecida a un *denoising autoencoders* en el sentido de que esas pequeñas perturbaciones en la entrada pueden considerarse ruido y se desea que el modelo sea robusto ante las mismas.

Los *contractive autoencoders* penalizan las instancias donde un pequeño cambio en la entrada conduce a un gran cambio en el espacio de codificación. El término de penalización que se suma a la función de pérdida es aplicado durante el entrenamiento y se plantea de la siguiente forma:

$$\mathcal{L}(x, g(f(x))) + \Omega(\hat{z}, x) \quad (1.7)$$

$$\Omega(\hat{z}, x) = \lambda \sum_i \|\nabla z_i\|^2 \quad (1.8)$$

A la primera ecuación creo que le falta la parte izquierda

De esta forma obliga al modelo a aprender una función que no cambia mucho cuando x varía ligeramente. Como la penalidad (es aplicada⁴¹) sobre los datos de entrenamiento, está forzado a capturar información sobre la distribución del conjunto de entrenamiento.

Además de los métodos descritos anteriormente, que se interpretan como *regularized autoencoders*, casi cualquier modelo generativo con variables latentes y dotado de un procedimiento de inferencia ESPACIO((⁴²) para (computar⁴³) representaciones latentes dada una entrada) puede verse como un caso particular de *autoencoders* [13]. Dos enfoques de modelos generativos que enfatizan esta conexión con los *autoencoders* son los descendientes de la máquina de Helmholtz¹ (Hinton et al.[14]) como los *variational autoencoders* y las *generative stochastic networks*²(GSNs).

Acabo de leer las noticas al pie. Te propongo que en vez de notas al pie lo pongas aquí como parte del cuerpo. Aunque para eso tendrías que cambiar un poco la redacción. Sería algo como: son los descendientes de la maquina de Helmholtz, las generative stochastic networks y los variational autoencoders.

Las maáquinas de helmholtz bla bla bla. Las stochastic ... bla bla bla. y las variational autoencoders (también dar una explicación en una oración. Ahí seguirías con el párrafo que le sigue a este...

Estos modelos aprenden naturalmente codificaciones de la entrada (de alta calidad⁴⁴). Esta capacidad (es adquirida⁴⁵) debido a que (son entrenados⁴⁶) (para maximizar la probabilidad de los mismos datos⁴⁷), más allá de copiar la entrada en la salida. En la siguiente sección se profundizará sobre el modelo *variational autoencoders*, su función, utilidad y aplicaciones.

¿por qué los variational autoencoders tienen una sección los otros no? :-/ No quiero que tengan una sección si no hace falta, pero sería bueno dejar claro por qué ese "es especial", y esa explicación, además, sería la transición a la siguiente sección :-D.

41: Voz pasiva

42: Este espacio no va

43: ¿calcular? ¿obtener? Es que por lo general, computar suena feo... al menos a mí :-), y prefiero evitarlo tanto como se pueda.

44: ¿qué es de alta calidad? ¿la entrada o las codificaciones? Creo que se debería cambiar el orden de la oración.

45: Voz pasiva

46: Voz pasiva. ¿se entrenan?

47: ¿qué significa maximizar la probabilidad de los mismos datos?

1.2.2. Variational Autoencoders

Los *variational autoencoders* (VAE) fueron descubiertos simultáneamente por Kingma y Welling en diciembre de 2013 y por Rezende, Mohamed y Wierstra en enero de 2014 [10] y pertenecen a la familia de modelos generativos [13, 3, 10]. Son un tipo de *autoencoders* más avanzado que mezcla ideas de aprendizaje profundo con inferencia bayesiana [REF].

Como se describió en la sección anterior, un *autoencoders* clásico transforma su entrada a un vector de un espacio a través del encoder, y luego lo decodifica en una salida con las

¹La máquina de Helmholtz (llamada así por Hermann von Helmholtz y su concepto de energía libre de Helmholtz) es un tipo de red neuronal artificial que puede describir la estructura oculta de un conjunto de datos al ser entrenada para crear un modelo generativo de dicho conjunto [2].

²Son una generalización del *denoising autoencoders* que incluye variables latentes en el proceso generativo mediante cadenas de Markov, además de las variables conocidas de entrada [8].

mismas dimensiones que la entrada original. Imponiendo varias restricciones al vector codificado o vector de contexto, como también se le nombra, el modelo podrá aprender mayores o menores características latentes interesantes de la representación de los datos. Por lo general su función se basa en comprimir los datos de entrada en una representación con menos información. Chollet *et al.* [10] sostiene que los VAEs, superan a los *autoencoders* clásicos con un poco de "magia estadística" que los obliga a aprender ese espacio deseado.

La frase me encanta, pero si dices magia estadística deberías al menos mencionar ligeramente en qué consiste :-).

Un VAE, en lugar de comprimir la entrada en un código fijo del espacio latente, la convierte en dos parámetros de una distribución estadística: la media y la varianza. Luego, utiliza ambos parámetros para generar aleatoriamente un elemento de esa distribución, y decodificarlo por medio del *decoder*. En esencia, es un tipo de *autoencoders* cuya distribución de codificaciones (es regularizada⁴⁸) durante el entrenamiento para asegurar que su espacio latente (tenga buenas propiedades⁴⁹) que permitan generar nuevos datos. El término *variational* proviene de la estrecha relación que existe entre la regularización y el método de inferencia variacional en estadística. La (estocasticidad⁵⁰) de este proceso mejora la robustez y obliga al espacio latente a codificar representaciones significativas.

Revisé hasta aquí :-(. Mañana sigo :-D.

48: Voz pasiva

49: ¿qué significa que un espacio latente tenga buenas propiedades? Quizás eso sea bueno ponerlo en la sección "para dummies"

50: ¿aleatoriedad?

1.2.3. Principios del VAE

Los modelos generativos (prestan gran interés en aproximar⁵¹) la verdadera distribución de su entrada usando redes neuronales:

$$x \sim P_{\theta}(x) \quad (1.9)$$

En la ecuación anterior, θ hace referencia a los parámetros determinados durante el entrenamiento. Por ejemplo, en el contexto de la base de datos de rostros de celebridades **ESPACIO**(CelebA (⁵²)), sería equivalente a encontrar una distribución que pueda dibujar rostros humanos. Similarmente, en el conjunto de datos MNIST que contiene dígitos del 0 al 9, la distribución puede generar (distinguibles dígitos⁵³) escritos a mano.

(TODO: poner footnote referente a ambos datasets(CelebA, MNIST) con la url de acceso a los mismos⁵⁴)

En *machine learning*, para lograr cierto nivel de inferencia, interesa encontrar (la $P_{\theta}(x, z)$ ⁵⁵), traducida como la probabilidad conjunta entre la entrada x , y la variable latente z [3]. Las variables latentes no son parte del (dataset⁵⁶) pero (recopilan ciertas propiedades observables en los datos⁵⁷). Luego $P_{\theta}(x)$ puede ser **calculada** como la distribución marginal:

$$P_{\theta}(x) = \int P_{\theta}(x, z) dz \quad (1.10)$$

El problema (con la ecuación⁵⁸) 1.10 es intratable, pues la ecuación no tiene una forma

51: No estoy seguro de que esta frase tenga sentido en español :-(. ¿Qué quieres decir con eso? :-)

52: Este espacio no debería estar aquí

53: Creo que aquí es mejor escribirlo en el formato sustantivo y después adjetivo.

54: Quizás esto pueda estar en la sección de introducción al ML.

55: Quita esto de aquí y ponlo después que lo digas con palabras: interesa encontrar la probabilidad de TAL COSA, que se representa por AQUÍ PONES LA ECUACIÓN. Casi siempre es más fácil entender primero la idea con palabras y después, la ecuación. Aquí lo tienes al revés.

56: ¿conjunto de entrenamiento?

57: Esto sería bueno explicarlo al menos una vez. Quizás en la sección de introducción.

58: ¿con la ecuación o "al tratar de resolver la ecuación"?

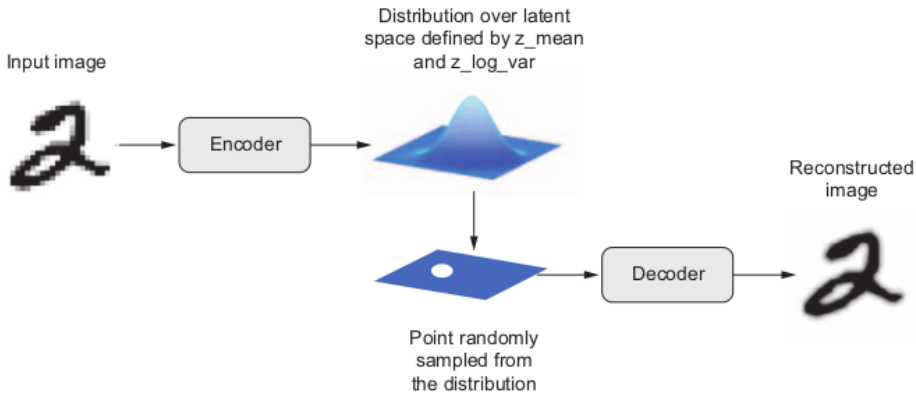


Figura 1.3: Ejemplo de VAE transformando una imagen a 2 vectores, z_μ y z_σ , los cuales definen una distribución de probabilidad sobre el espacio latente y son usados para generar un punto que será decodificado. **En algún momento del texto deberías referenciar esta imagen y explicar qué hace. Eso además del caption, que está bien. Es que se considera “mal estilo” tener imágenes o tablas en el documento que no estén referenciadas en el texto. De hecho, es uno de los puntos que se evalúan para aceptar o rechazar trabajos en eventos y revistas.**

analítica o un estimador eficiente. (No puede ser diferenciable con respecto a sus parámetros⁵⁹) . En efecto, la optimización o proceso de *backpropagation* en la red neuronal no sería factible [10].

Usando el teorema de Bayes se tiene una alternativa para la expresión en 1.10:

$$P_\theta(x) = \int P_\theta(x|z)P(z)dz \quad (1.11)$$

$P(z)$ es una distribución de probabilidad sobre z . No está condicionada a ninguna de las observaciones. Si z es discreto y $P_{\theta(x|z)}$ es una distribución Normal, entonces $P_\theta(x)$ (es una mezcla⁶⁰) de funciones Normales. Si z es continuo, $P_\theta(x)$ es (una mezcla infinita⁶¹) de distribuciones Normales.

La siguiente pregunta es curiosidad mía, y no necesariamente para incluirlo en el documento: pero ¿se tienen las expresiones o se conocen las formas de esas mezclas que mencionas? :-/

En la práctica, al intentar construir una red neuronal para aproximar $P_{\theta(x|z)}$ sin una función de pérdida adecuada, simplemente ignorará z y llegará a una solución trivial [3]:

$$P_{\theta(x|z)} = P_\theta(x)$$

Por lo tanto la ecuación 1.11 no ((nos)⁶²) proporciona una buena estimación de $P_\theta(x)$. (En efecto⁶³) , la ecuación 1.11 puede ser expresada también de la forma:

59: ¿Por qué? Estas preguntas de por qué se deberían responder con una oración o con una referencia.

60: ¿qué tipo de mezcla?

61: ¿qué tipo de mezcla?

62: Aquí te propongo quitar el “nos” por dos motivos. El primero es que se puede quitar y no afecta el resto del documento, y la segunda es que pone la oración en primera persona y siempre que podamos, debemos evitar eso... :-o. Quiero decir: siempre que se pueda, se debe evitar eso ;-).

63: Esto es completamente secundario y lo puedes ignorar si quieres, pero si nos da tiempo, revisa que escribiste varias veces “En efecto”. Son solo 3 veces en este capítulo, pero esta y la anterior están “cerquita”, y pudiera empezar a dar la sensación de “muletilla”. Eso no es problema, todos tenemos muuuuchas de esas (orales y literarias), pero es bueno tomar conciencia de ellas y sustituirla por expresiones equivalentes.

$$P_{\theta}(x) = \int P_{\theta}(z|x)P(x)dz \quad (1.12)$$

No obstante, sigue siendo (intratable $P_{\theta}(z|x)$ ⁶⁴). El objetivo de un VAEs es encontrar una distribución que estime ((los mas cerca posible la)⁶⁵) $P_{\theta}(z|x)$ ¿de la mejor manera posible?.

Sería bueno poner alguna transición a la sección que sigue. Aunque sea una oración, que incluso pudiera estar en el párrafo anterior.

1.2.4. Inferencia en el VAE

Con el propósito de obtener $P_{\theta}(z|x)$, el VAE introduce un modelo de inferencia variacional (ESPACIO(encoder)⁶⁶):

$$Q_{\phi}(z|x) \approx P_{\theta}(z|x) \quad (1.13)$$

donde $Q_{\phi}(z|x)$ proporciona un buen estimador de $P_{\theta}(z|x)$; (siendo además de parametrizable,⁶⁷ (finalmente) (manejable⁶⁸). $Q_{\phi}(z|x)$ (puede ser aproximado⁶⁹) (por una red neuronal profunda mediante la optimización de los parámetros ϕ .⁷⁰)

A partir del esquema de la figura 1.1, se considera a Q_{ϕ} para inferir las posibles (variables ocultas⁷¹) que (son usadas⁷²) (en la generacion de⁷³) una observación. Tal modelo puede ser construido mediante una arquitectura de redes neuronales donde el modelo *encoder* aprende una función de transformación de x a z y de igual forma el *decoder* de z a x como se visualiza en la figura 1.4.

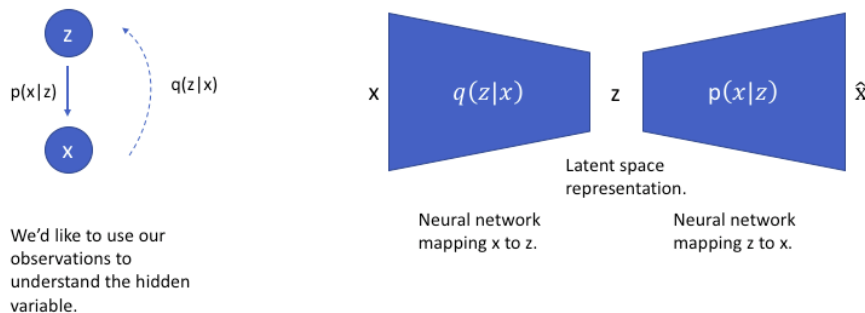


Figura 1.4: Representación del VAE mediante modelos neuronales. Me parece muy buena la imagen, lo único es que no me gusta que estén las explicaciones en inglés :-(. Lo terrible es que no estoy seguro de por qué. ahora mismo la mayor "preocupación" es que sospecho que es algo que señalarían los miembros del tribunal cuando la revisen :-(. En cualquier caso, este es un problema muuuuuuy pequeño, y se puede ignorar por ahora :-).

(Tradicionalmente⁷⁴) $Q_{\phi}(z|x)$ se distingue por ser una distribución gaussiana (multivariante⁷⁵)

3.

64: Aquí falta un verbo. ¿Qué es intratable? Resolverlo, explicarlo, dibujarlo... me imagino que sea resolverlo, pero sería bueno dejarlo de manera explícita

65: Es que no estoy seguro de que "estimar lo más cerca posible" le vaya a gustar a la gente de estadística :-/.

66: Usualmente antes de abrir un paréntesis se deja un espacio. Yo sé, en un lenguaje de programación ser vería terriblemente feo, pero en los documentos escritos, no tanto :-)

67: Te propongo cambiar esta parte por: que además de ser parametrizable, es manejable. Y bueno (este bueno es una muletilla mía oral y escrita. Es para que veas que todos las tenemos :-P), sigue estando la pregunta de qué significa que se sea manejable

68: ¿Qué significa esto?

69: Voz pasiva

70: El hecho de que Q_{ϕ} se pueda aproximar por una red neuronal profunda yo lo pondría al comienzo del párrafo, porque esa es la primera pregunta que un lector se haría: ¿qué forma tiene esa Q ? Así tú se lo dices rápido, y ya desaparece la pregunta. Después das el resto de los argumentos y ventajas.

71: ¿cuáles son estas variables?

72: Voz pasiva

73: ¿para generar? ¿al generar?

74: ¿Para quién? ¿dónde? ¿en qué contexto? ¿desde cuándo? Esa es una palabra que si no se especifica bien puede provocar muuuuuchas preguntas :-/.

75: Yo siempre he oído multivariada :-/. ¿Esa palabra "multivariante" la sacaste de algún documento "serio en español"? Por si acaso le voy a preguntar a las profes de estadística y probabilidades.

$$Q_{\phi}(z|x) = \mathcal{N}(z; \mu(x), \text{diag}(\sigma(x))) \quad (1.14)$$

Tanto la media $\mu(x)$ como la desviación estandar $\sigma(x)$, (son computadas⁷⁶) por el modelo neuronal *encoder* usando los datos de entrada. Para construir un verdadero modelo gaussiano (multivariante⁷⁷), sería necesario definir una matriz de covarianza que describe la correlación entre las dimensiones. Sin embargo, dicho modelo (es simplificado⁷⁸) para que la matriz (solo tenga valores distintos de cero en la diagonal⁷⁹), permitiendo describir esta información en un vector. Que la (La) matriz diagonal implica que los elementos z son independientes.

Transición a lo que sigue. Recuerda las tres preguntas: ¿qué acabo de leer? ¿qué tiene que ver con el objetivo del trabajo? ¿y cuál es la relación con lo que sigue? Realmente hay una 4ta pregunta que debería estar presente siempre, pero que muchas veces se puede ignorar, aunque cuando se responde, los resultados son espectaculares. Idealmente en estas transiciones, lo que se le debería responder al lector es ¿y a mí qué? Cuando el lector termina la sección (inconscientemente) se hace esa pregunta: ¿y a mí qué? Las tres que te pongo, son un sustituto “bastante eficiente”, pero lo ideal sería poder responderle esa :-). Pero no es obligatorio. Solo déjalo corriendo en background :-D.

1.2.5. Aplicación de los VAEs a problemas similares ¿(similares a qué)?

Una de las aplicaciones más recientes de los VAEs (autoencoders⁸⁰) en general, que comparte ideas (con este trabajo⁸¹) (es concebida⁸²) en [26]. Rákos et al. aborda el diseño de (estrategias de control⁸³) seguras y óptimas para vehículos autónomos. Presentan un *variational autoencoder* para comprimir la información referente a las trayectorias, seleccionando (útiles características latentes⁸⁴) en los datos con valores pequeños de pérdida y error de reconstrucción. Muestran que la representación aprendida durante la compresión separa las trayectorias de acuerdo con tres clases de maniobra (mantenimiento de carril, cambio de carril derecho o cambio de carril izquierdo). La propuesta parte de la hipótesis de predecir (el vector de contexto⁸⁵) de la trayectoria en lugar de la trayectoria en sí. Sin embargo, (los resultados invierten dicha teoría⁸⁶) debido a que (las trayectorias no pueden decodificar suficiente información de los movimientos alrededor del vehículo⁸⁷).

Aquí puse un cambio de página que no hace falta, para tener espacio para los comentarios.

76: Aquí hay una voz pasiva, y yo preferiría usar otro verbo que no fuera computar: aquí incluso puedes escoger entre calculadas y estimadas.

77: pendiente de consulta con los profes de estadística

78: Voz pasiva, y además: ¿por quién es simplificado? ¿cuándo? ¿en qué contexto?

79: :-o. Eso me parece mágico. Creo que deberías explicarlo un poquito más, aquí o en la sección. El problema es que como dices, si la matriz de covarianza es diagonal entonces las variables son independientes. ¿significa esto que los datos “se cañonean” para que sean independientes y se ignoran las dependencias, ¿o es algo más ingenioso, inteligente, y más del agrado de los estadísticos? :-P?

80: ¿Las AE en VAE no significan ya AutoEncoders?

81: Por ejemplo, una de las ventajas de poner en las transiciones ¿qué tiene que ver eso con este trabajo? es que llegado este punto el lector tiene “fresco” lo que queremos hacer. Ahora mismo solo lo vio en el resumen y en la introducción. Y eso es algo que se debería tener siempre presente.

82: Voz pasiva

83: ¿Qué es una estrategia de control?

84: Esto está en inglés, que es donde se debe poner el adjetivo antes del sustantivo. Como casi todo lo que hacemos nosotros, lo leemos en inglés, llega un momento en que casi empezamos a pensar (y escribir! :-o medio ne inglés :-/. A todos nos pasa. De todas formas, sería mejor poner el útil detrás de características latentes :-D.

85: ¿qué es un vector de contexto?

86: ¿qué significa esto?

87: Aquí entiendo lo de los movimientos alrededor del vehículo, peor no entiendo qué significa que las trayectorias no lo puedan decodificar :-(.

³En probabilidad y estadística, una distribución gaussiana multivariante, es una generalización de la distribución normal unidimensional a dimensiones superiores [1].

Los problemas de enrutamiento son especialmente significativos en gran variedad de aplicaciones modernas, como (~~por ejemplo en~~) la planificación de rutas, mantenimiento de redes, desarrollo de sistemas nanoelectrónicos de alto rendimiento, entre otros; y además, están caracterizados por datos (~~esparcidos~~⁸⁸) y no balanceados. **Esto me parece muy interesante, pero más apropiado para algún tipo de introducción, quizás para la del trabajo completo... pero aquí me saca un poco del flujo de la lectura. Creo que incluso tiene la característica de que en esta posición se puede quitar y la sección queda más o menos igual.**

88: ¿dispersos? ¿ralos?

En [30] se manifiesta el problema de enrutar billones de componentes en dispositivos nanoelectrónicos. (Utyamishev et al.⁸⁹) . explota un proceso de entrenamientos progresivos del VAE para aprender una representación latente con datos altamente dispersos y no balanceados, con el fin de generar rutas en un espacio en 2D restringido. La noción de progresividad está dada porque el VAE fue diseñado para utilizar diferentes tamaños del conjunto de entrenamiento de forma incremental. Dicho método exhibe una rápida convergencia y alto grado de rutabilidad. **¿Y al final qué es lo que hace? :-).**

89: ¿Este es el mismo [30]? No me queda claro. Si lo es, sería bueno cambiar la redacción. Si no lo es, habría que poner la referencia. Y la verdad es que ahora me pesa revisar la bibliografía :-/.

El trabajo presentado en Hottung et al. [19] en mayo de 2021 aporta interesantes ideas para el desarrollo de esta tesis. Ofrecen un enfoque de optimización basado en aprendizaje que permite guiar la búsqueda dentro de la distribución de soluciones de alta calidad para una instancia del problema. Construyen un *Conditional Variational Autoencoders* para transformar soluciones del problema de enrutamiento a puntos de un espacio continuo que, en conjunto, conforman un espacio de búsqueda. Su idea es que a partir de una solución buena s y una instancia del problema l , codificar esta información a un vector z . Posteriormente la capa de decodificación unido a un algoritmo de optimización, en este caso con una estrategia de evolución diferencial, generan una nueva solución s' a partir de z y l . En esta novedosa propuesta se emplean técnicas de *machine learning* como los modelos *sequence-to-sequence*, *Pointer Networks*, *attention mechanism* los cuales serán abordados próximamente, y por otra parte, el uso de métodos de optimización para la búsqueda en el espacio aprendido.

1.3. Arquitecturas Sequence-to-Sequence

Una solución a un problema de enrutamiento como el TSP y VRP puede ser interpretada, en su versión más simple, como un conjunto de listas de clientes donde cada lista representa una ruta. La disposición de los clientes dentro de la ruta es un factor determinante en el orden de los recorridos (~~que se deben~~(a)⁹⁰) realizar. Por tanto, una solución se compone de un conjunto de rutas y (~~estas a su vez constituyen~~ **secuencias de clientes**⁹¹) . Como parte de la revisión bibliográfica realizada se decidió indagar en los modelos *sequence-to-sequence*, ~~que(los-cuales)~~ pertenecen a la familia de modelos de *machine learning* para el procesamiento del lenguaje.

Los modelos *sequence-to-sequence* [28, 33] ~~¿han resultado?(son)~~ (~~muy~~) útiles ~~¿en?(para)~~ tareas donde se requiere (*mapear*⁹²) de una secuencia a otra. Han sido ampliamente estudiados

90: Hay una estructura que se ha hecho frecuente en la comunicación y es objeto + a + infinitivo. Por ejemplo: recorridos a realizar, problemas a resolver, objetos a ordenar, etc. Yo no estoy muy seguro de cuán aceptado esté eso por la real academia y demás, pero siempre que sea posible yo prefiero evitarlos. El motivo fundamental es que me parece "feo" :-), además de que sospecho que no existe en el español. Este es el tipo de cosas que te diría que son caprichos "de viejo", pero este en particular lo tengo desde que era recién graduado :-P. Quizás por eso siempre me han dicho que desde joven era un viejo :-P. Bueno, en fin, si puedes, te propongo evitar las construcciones a + infinitivo. Además, otro motivo para evitarlas, es que las alternativas suelen más claras y específicas.

91: Este es el tipo de elementos que se pudieran poner en la introducción del capítulo, cuando mencionas que hay una sección dedicado a esto.

92: ¡Geeeee! No, no. Vamos a buscar otra alternativa para esto. Transformar, convertir, vaya, que acepto hasta emparejar

en el campo de la traducción automática (en los últimos años⁹³), y existen (numerosas⁹⁴) variantes de los mismos. La arquitectura general consiste en dos (redes neuronales recurrentes⁹⁵) ESPACIO(RNN) nombradas (encoder y decoder⁹⁶). La red *encoder* lee la secuencia de entrada y encapsula la información **extraída** en un vector de tamaño fijo ESPACIO(o una secuencia de vectores); luego, el *decoder* convierte la información codificada a una secuencia de salida. **falta transición a lo que sigue**

1.3.1. RNN Encoder-Decoder

Este framework que sirve como base y guía para los modelos *encoder-decoders* de secuencias está compuesto por un *encoder* que lee la secuencia de entrada $X = (x_1, \dots, x_T)$ y la (**mapea**⁹⁷) a un vector c . La mayoría de las propuestas utilizan una RNN tal que:

$$\begin{aligned} h_t &= f(x_t, h_{t-1}) \\ c &= q(h_1, \dots, h_T) \end{aligned}$$

donde h_t es el (estado oculto⁹⁸) de la RNN en el paso t , y c es un vector de contexto generado a partir de la secuencia de estados ocultos. f y q son funciones no lineales, generalmente un caso de redes neuronales recurrentes nombradas como LSTM ESPACIO(Long Short-Term Memory).

El *decoder* es entrenado para predecir la próxima (palabra⁹⁹) y_t dado el vector de contexto c y todas las palabras predichas anteriormente. Define la probabilidad de la (traducción¹⁰⁰) y a través de la probabilidad condicional conjunta DOS PUNTOS(÷)

$$p(y) = \prod_{t=1}^T p(y_t | \{y_1, \dots, y_{t-1}\}, c) \text{ PUNTO.} \quad (1.15)$$

Con una RNN cada probabilidad condicional (es modelada¹⁰¹) como

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c) \text{ COMA} \quad (1.16)$$

donde g es una función no lineal que (computa¹⁰²) la probabilidad de y_t , y s_t es el estado oculto de la RNN

93: ¿Cuáles son los últimos años? Mientras mayor sea la persona a la que le preguntes mayor será su rango. Por ejemplo, para muchas personas el mp3 es un formato reciente :-o, mientras que para ti, el mp3 es "de toda la vida" :-D. Por eso sería bueno precisar bien este período.

94: Esto es peligrosamente descriptivo, casi como si fuera un adjetivo, además que de provoca preguntas como ¿exactamente cuántas?

95: ¿Para la sección de introducción al ML?

96: ¿Esto no es así en todos los auto-encoders? ¿Por qué esto es diferente? Quizás sea bueno explicar eso.

97: Esto no es un error ortográfico, pero no tengo una forma de poner un emoji "de pánico", así que cada vez que vea uno de estos lo marcaré como un error ortográfico... sobre todo porque tiene unos colores chillones :-P. Convierte, transforma

98: ¿El estado oculto es lo mismo que la capa oculta? Sospecho que esto debería estar en la sección de introducción al ML.

99: ¿qué palabra? :-o. Creo que faltó decir que a los elementos de la secuencia se les llama palabras, o que este tipo de modelo se usa mucho en el lenguaje natural y por eso procesa palabras, o algo así. Pero esta palabra "aquí ahora de pronto" pudiera sacar de paso a la persona que esté leyendo. Quizás la solución sea definir o precisar bien los términos básicos de estos modelos como secuencia, palabras, estado oculto, traducción y eso. Quizás el mejor lugar sea la sección de introducción a la ML. Recuerda que esto para ti es "lo de todos los días", pero no tiene por qué ser así para quienes se lean el documento.

100: Me pasó lo mismo que con palabra. Así que aquí vendría la misma explicación

101: Voz pasiva

102: ¿calcula? ¿estima?

En la arquitectura ("*vanilla*"¹⁰³) Sequence-to-Sequence [28], la secuencia de entrada aparece solo una vez en el *encoder* y la salida del mismo (es generada¹⁰⁴) a partir de un único vector **ESPACIO** (por ejemplo, el último estado oculto del *encoder RNN*). Otras extensiones como (por ejemplo) Bahdanau et al. [4], *ilustran?*(*ilustra*) que la información de entrada puede (ser usada¹⁰⁵) de forma más explícita para aumentar la cantidad de información durante los pasos de decodificación. (En este sentido¹⁰⁶) , emplean otra red neuronal denominada(*nombrada como*) *attention mechanism* (que "*atiende*" todos los estados¹⁰⁷) del *encoder RNN*. Este mecanismo permite que el *decoder* se enfoque en zonas importantes de la secuencia de entrada y use la información relevante para producir mejores secuencias de salida. (Recientemente¹⁰⁸) , el concepto de *attention* ha sido una idea ((*muy*)¹⁰⁹) (*popular*¹¹⁰) en la investigación debido a su capacidad de (alinear diferentes objetos¹¹¹) [33, 24, 25, 32].

El objetivo por el que te pongo algunos comentarios de ¿qué significa esto? es que igual que me surge la pregunta a mí, le puede surgir a cualquiera que lea el documento, y por lo general eso se convierte después en preguntas del tribunal, del oponente y eso :-/. La idea es evitar la mayor cantidad de esas ;-). Al menos las que sabemos que se pueden evitar. Por eso es que hay muchos comentarios así :-/.

Pon una transición a lo que sigue.

1.4. Reinforcement learning en problemas de enrutamiento.

Debido a su (naturaleza combinatoria y complejidad¹¹²) , el VRP y sus variantes han sido candidatos naturales para problemas de optimización resueltos por métodos de ML ¿Por qué?. Por sus características secuenciales de decidir en qué orden se deben visitar los clientes, la dinámica del tiempo de demanda, y la falta de una (distribución¹¹³) subyacente detrás de las rutas, estos métodos se basan principalmente en técnicas de (*deep reinforcement learning*¹¹⁴) **ESPACIO**(DRL, aprendizaje profundo por refuerzo).

(**RL**¹¹⁵) es un proceso de aprendizaje que ocurre por diferentes interacciones con el entorno y en el que cada acción se traduce en una recompensa. ¿Esto es "*en la vida real*" o solo en el contexto de ML? Aborda los problemas donde se desconoce (el modelo subyacente¹¹⁶) y en el que cada acción (impacta sobre el ambiente¹¹⁷) . (Su objetivo¹¹⁸) es determinar una política, es decir, qué acción tomar en cada estado, con el fin de obtener la mayor ganancia. Te propongo reescribir este párrafo por todas las preguntas que provoca :-/.

103: ¿básica? ¿elemental? No estoy seguro de cuán familiarizados estén los posibles miembros del tribunal con esta terminología :-). Yo creo en inglés si se usa mucho y es casi cultura popular, pero no estoy seguro de que en español sea así. ¿Quizás tenga que ver con que esos países el sabor de helado que más se vende sea el de Vainilla? No estoy seguro de que en Cuba se cumpla eso... No, ¿qué pregunta tan interesante!, pero concéntrate que eso no tiene nada que ver con la tesis... (eso último es conmigo, no contigo)

104: Voz pasiva

105: Voz pasiva

106: ¿Para ello?

107: ¿qué significa que "atienda" los estados?

108: ¿cuán reciente?

109: Es "muy" probable que te sugiera quitar todos los "muy" que me encuentre, aunque me imagino que ya lo estás sospechando

110: Esta es una palabra "complicada". ¿no podemos limitarnos a decir que ha sido usada por varios investigadores?

111: ¿qué significa esto?

112: Creo que esto se lee más fácil si cambias el orden: debido a su complejidad y naturaleza combinatoria... El criterio para determinar eso es leerlo en voz alta, que es un ejercicio que hacen los escritores.

113: Esta distribución se refiere a las distribuciones de probabilidad, pero recuerda que en los VRP son problemas "de distribución" (de "cosas", pero distribución también). Por ese motivo, y sobre todo porque es probable que tu tesis se la lean personas que se dedican al VRP sería bueno especificar que esta es probabilística

114: Para la sección de introducción

115: ¿sin D?

116: ¿qué es esto?

117: ¿qué significa esto? ¿A qué contexto se refiere?

118: ¿el del RL?

La mayoría de los enfoques de RL para resolver el VRP, lo interpretan como un proceso de decisión de Markov, en el cual la solución óptima puede verse como una secuencia de acciones que deciden qué cliente visitar una vez ha sido atendido el cliente anterior. Estos algoritmos se enfrentan a dos dificultades principales. Primero, el VRP es un problema altamente combinatorio: codificar y decodificar el estado actual y la acción son un paso crucial. En segundo lugar, su estado (*depende del tiempo*¹¹⁹) y se ve afectado por cada decisión, de modo que la representación del problema evolucione con el tiempo. Debido a estos inconvenientes, (*gran parte*¹²⁰) de los autores se han concentrado en (*enfoques autorregresivos*¹²¹), es decir, modelos que se basan en la información del estado previamente observado.

Vinyals et al. [33] fueron los primeros en introducir las *Pointer Networks*, un modelo inspirado en la arquitectura *sequence-to-sequence*. Estas redes (*son aplicadas*¹²²) a problemas de optimización combinatoria, donde (*la longitud de la secuencia de salida está determinada por la longitud de las secuencias de entrada*¹²³). La arquitectura aprende la probabilidad condicional de una secuencia de salida formada por valores discretos que se corresponden a las posiciones de los elementos en la secuencia de entrada. Los autores entrenan su modelo para resolver instancias del TSP de hasta 50 clientes, usando un *attention mechanism* para (*computar*¹²⁴) una permutación de la secuencia de entrada que constituye una ruta para el TSP. Durante la inferencia, seleccionar la secuencia con mayor probabilidad entre todas las opciones posibles es computacionalmente costoso e impracticable, por lo cual se realiza un procedimiento de (*beam search*¹²⁵) para encontrar la mejor secuencia posible. De esta forma alcanzaron resultados (*prometedores*¹²⁶) para instancias de hasta 30 clientes.

Mientras que en [33] se obtiene una solución del TSP de una forma supervisada, Bello et al. [6] lo extiende con un aprendizaje no supervisado que emplea un método (*critic actor*¹²⁷). Aplican una función de penalidad a partir de la distancia total recorrida en cada estado. Además, adaptaron su trabajo para hacer frente a las limitaciones del VRP, usando un término de alta penalidad sobre las soluciones no factibles. Nazari et al. [24] amplía aún más este último para resolver el VRP, reemplazando la *Pointer Networks* por un *decoder RNN* basado en un proceso de *attention*. Con ello logran capturar (*las dependencias de tiempo y los cambios del ambiente en el VRP*¹²⁸), principalmente cómo evoluciona la demanda con el tiempo. Su modelo es invariante en el orden de la entrada *ESPACIO* (el orden de los clientes) por tanto puede manejar eficientemente elementos dinámicos en la entrada *ESPACIO* (por ejemplo, modificación de la demanda de un cliente una vez *que* este (*es visitado*¹²⁹)). Obtienen resultados similares a las propuestas de última generación en problemas de pequeña y mediana dimensión. Múltiples extensiones a este trabajo (*han sido introducidas*¹³⁰) para resolver variantes del VRP, como [31, 9] para el HVRP, y [34] para el VRPTW.

Otra alternativa a las *Pointer Networks* son los (*graph encoding*¹³¹), los cuales manifiestan mejor la naturaleza combinatoria del problema que los modelos *sequence-to-sequence*. En Khalil et al. [20] codifican las instancias usando un grafo constituido mediante una red neuronal, siendo así invariante en el orden de los nodos. Posteriormente ellos lo resuelven a través de un algoritmo de aprendizaje profundo (*Q-learning*¹³²) [23].

Hottung et al. [18] propone un nuevo enfoque llamado *neural large neighborhood search*

119: ¿del tiempo en el problema de VRP? ¿del tiempo de corrida del entrenamiento? ¿otro?

120: ¿qué porciento? Realmente, ¿cuán grande es esa parte?

121: ¿Para la sección de introducción?

122: Voz pasiva

123: Estas secuencias de entrada y de salida ¿"pertenecen" a la red neuronal o son entradas y salidas del VRP que quieren resolver?

124: calcular, obtener

125: ¿qué es esto?

126: ¿qué significa prometedores? ¿Llegan al óptimo?, ¿a qué distancia del óptimo se quedan?

127: ¿qué es esto?

128: cuando hablas de tiempo ¿te refieres a que el VRP que analizan es un problema con ventanas de tiempo o algo así? Lo que me pasa es que no estoy seguro de qué significa hablar de tiempo en el contexto general del VRP, a no ser que sea la variante del VRP que incluye tiempo en su formación.

129: Voz pasiva

130: Voz pasiva

131: ¿qué es esto? :-)

132: ¿qué es esto?

(NLNS) que integra aprendizaje de heurísticas a la metaheurística de alto nivel *large neighborhood search* (LNS). El mecanismo de aprendizaje está basado en una red neuronal profunda con un *attention mechanism* cuyo funcionamiento consiste en aprender heurísticas para reparar soluciones incompletas del VRP y emplearlas después como guía en la búsqueda de nuevas soluciones.

1.5. Conclusiones

Bibliografía

- [1] Distribución normal multivariada — Wikipedia, la enciclopedia libre, 2021. [Internet; descargado 26-octubre-2021].
- [2] Helmholtz machine — Wikipedia, the free encyclopedia, 2021. [Online; accessed 26-October-2021].
- [3] Rowel Atienza. *Advanced Deep Learning with Keras*. Packt Publishing Ltd., 2018.
- [4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [5] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.
- [6] Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.
- [7] Yoshua Bengio. Learning deep architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, 2009.
- [8] Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 226–234. JMLR.org, 2014.
- [9] Xinwei Chen, Marlin W. Ulmer, and Barrett W. Thomas. Deep q-learning for same-day delivery with a heterogeneous fleet of vehicles and drones. *CoRR*, abs/1910.11901, 2019.
- [10] Francois Chollet. *Deep Learning with Python*. Manning Publications Co., 2018.

- [11] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.
- [12] José Fernando Gonçalves and Mauricio G. C. Resende. A parallel multi-population biased random-key genetic algorithm for a container loading problem. *Comput. Oper. Res.*, 39(2):179–190, 2012.
- [13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2018.
- [14] Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and R M Neal. The “wake-sleep” algorithm for unsupervised neural networks. *Science*, 1995.
- [15] Geoffrey E. Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504 – 507, 2006.
- [16] Geoffrey E. Hinton and Richard S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *NIPS*, 1993.
- [17] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79 8:2554–8, 1982.
- [18] André Hottung and Kevin Tierney. Neural large neighborhood search for the capacitated vehicle routing problem. *CoRR*, abs/1911.09539, 2019.
- [19] André Hottung, Bhanu Bhandari, and Kevin Tierney. Learning a latent search space for routing problems using variational autoencoders. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [20] Elias B. Khalil, Bistra Dilkina, George L. Nemhauser, Shabbir Ahmed, and Yufen Shao. Learning to run heuristics in tree search. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 659–666. ijcai.org, 2017.
- [21] Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [22] Yann LeCun. Phd thesis: Modeles connexionnistes de l’apprentissage (connectionist learning models). 1987.
- [23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King,

- Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015.
- [24] MohammadReza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, and Martin Takác. Reinforcement learning for solving the vehicle routing problem. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9861–9871, 2018.
 - [25] Malte Probst. The set autoencoder: Unsupervised representation learning for sets. 2018.
 - [26] Olivér Rákos, Szilárd Aradi, Tamás Bécsi, and Zsolt Szalay. Compression of vehicle trajectories with a variational autoencoder. *Applied Sciences*, 10:6739, 2020.
 - [27] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020.
 - [28] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.
 - [29] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM Monographs on Discrete Mathematics and Applications, 2002.
 - [30] Dmitry Utyamishchev and Inna Partin-Vaisband. Progressive VAE training on highly sparse and imbalanced data. *CoRR*, abs/1912.08283, 2019.
 - [31] José Manuel Vera and Andres G. Abad. Deep reinforcement learning for routing a heterogeneous fleet of vehicles. *CoRR*, abs/1912.03341, 2019.
 - [32] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *CoRR*, abs/1511.06391, 2016.
 - [33] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700, 2015.
 - [34] Ke Zhang, Meng Li, Zhengchao Zhang, Xi Lin, and Fang He. Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach. *CoRR*, abs/2002.05513, 2020.