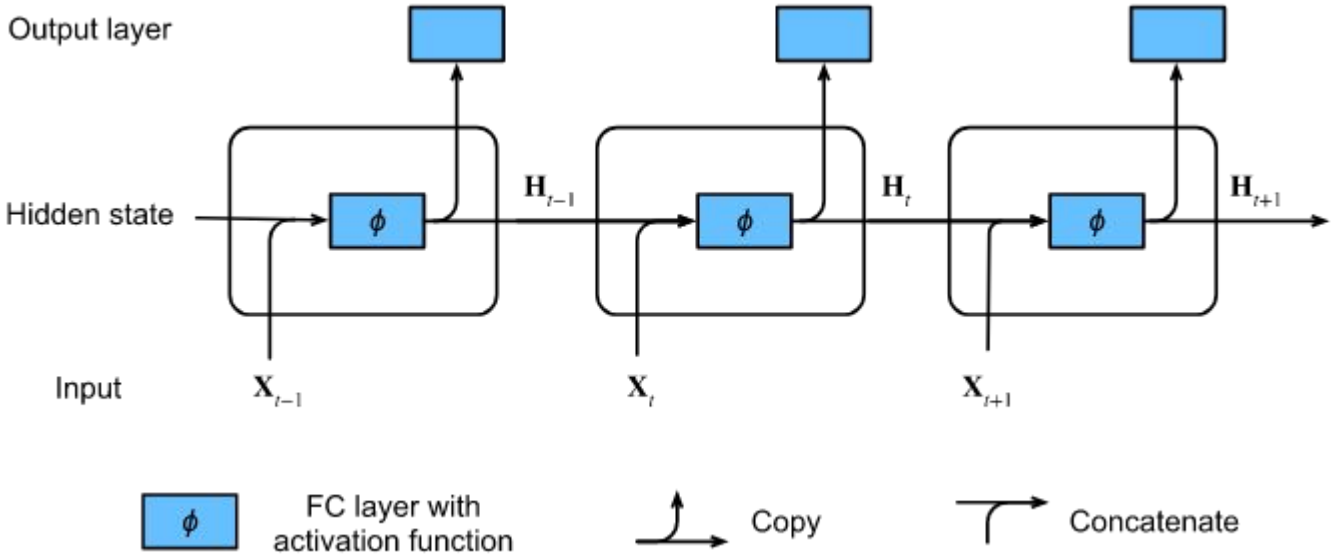
Redes Recurrentes

Orlando Ramos Flores

Contenido

- Recurrent Neural Network (RNN)
- Gated Recurrent Unit (GRU)
- Teacher Forcing

RNN



Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

RNN: Problema del desvanecimiento del gradiente

- Podríamos encontrarnos con una situación en la que una observación temprana sea muy significativa para predecir todas las observaciones futuras.
  - Nos gustaría tener algunos mecanismos para almacenar información temprana vital en una celda de memoria.
  - Sin tal mecanismo, tendremos que asignar un gradiente muy grande a esta observación, ya que afecta a todas las observaciones posteriores.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

RNN: Problema del desvanecimiento del gradiente

- Podríamos encontrarnos con situaciones en las que algunos tokens no lleven ninguna observación pertinente.
  - Por ejemplo, al analizar una página web, puede haber un código HTML auxiliar que sea irrelevante para evaluar el sentimiento transmitido en la página.
  - Nos gustaría tener algún mecanismo para omitir dichos tokens en la representación del estado latente.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

RNN: Problema del desvanecimiento del gradiente

- Podríamos encontrarnos con situaciones en las que hay una ruptura lógica entre las partes de una secuencia.
  - Por ejemplo, podría haber una transición entre los capítulos de un libro, o una transición entre un mercado de valores a la baja y al alza.
  - En este caso, sería bueno tener un medio para restablecer nuestra representación de estado interno.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Gated Recurrent Unit: GRU

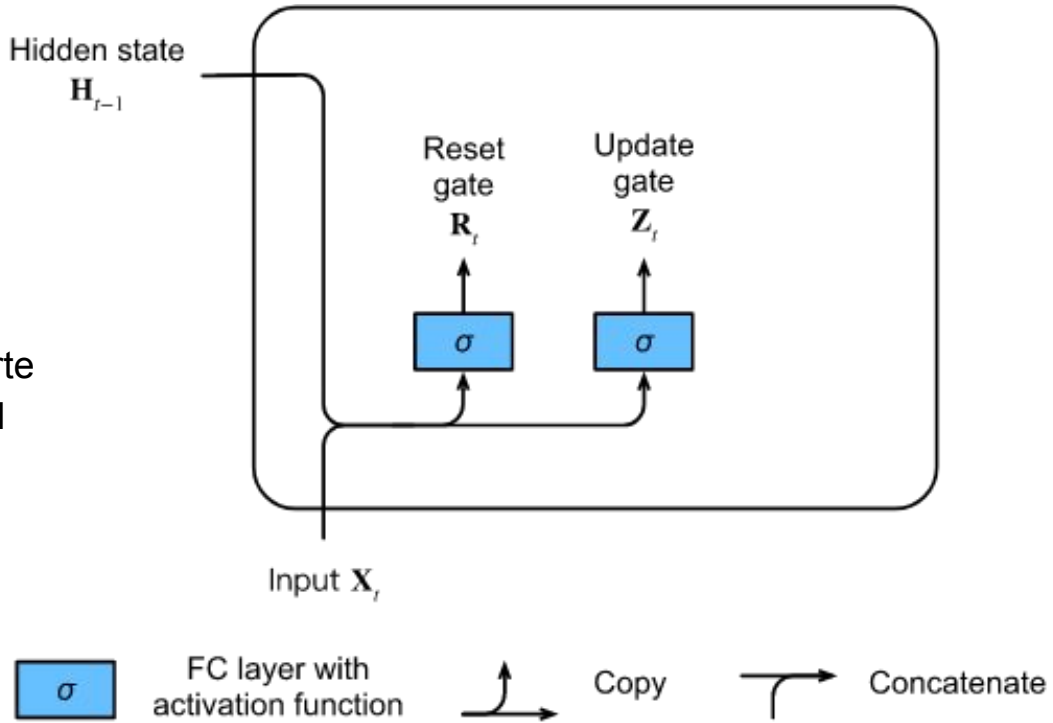
GRU: Unidad Recurrente Cerrada

- La distinción clave entre los RNN estándar y las GRU es que estas últimas admiten la activación del estado oculto.
- Esto significa que tenemos mecanismos dedicados para cuándo se debe ***actualizar*** un estado oculto y también cuándo se debe ***restablecer*** (reiniciar).
- Estos mecanismos se aprenden y abordan las preocupaciones enumeradas anteriormente.
- Por ejemplo, si el primer token es de gran importancia, la GRU aprende a no actualizar el estado oculto después de la primera observación.
- Asimismo, aprende a saltarse las observaciones temporales irrelevantes.
- Por último, aprende a restablecer el estado latente cuando sea necesario.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

GRU: Restablecer y actualizar la puerta

- Se introduce la puerta de reinicio y la puerta de actualización.
- Por ejemplo, una puerta de reinicio permite controlar cuánto del estado anterior aún se desea recordar.
- Del mismo modo, una puerta de actualización permite controlar qué parte del nuevo estado es solo una copia del estado anterior.



Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

GRU: Restablecer y actualizar la puerta

- Para un paso de tiempo dado ***t***, suponemos que la entrada es un mini batch de $X \epsilon \mathbb{R}^{n \times d}$, donde **n** es el número de muestras y **d** es el número de entradas.
- El estado oculto del paso de tiempo anterior es $H_{t-1} \epsilon \mathbb{R}^{n \times h}$, donde **h** es el número de unidades ocultas.
- La puertas de reinicio $R_t \epsilon \mathbb{R}^{n \times h}$ y actualización $Z_t \epsilon \mathbb{R}^{n \times h}$ son calculadas:

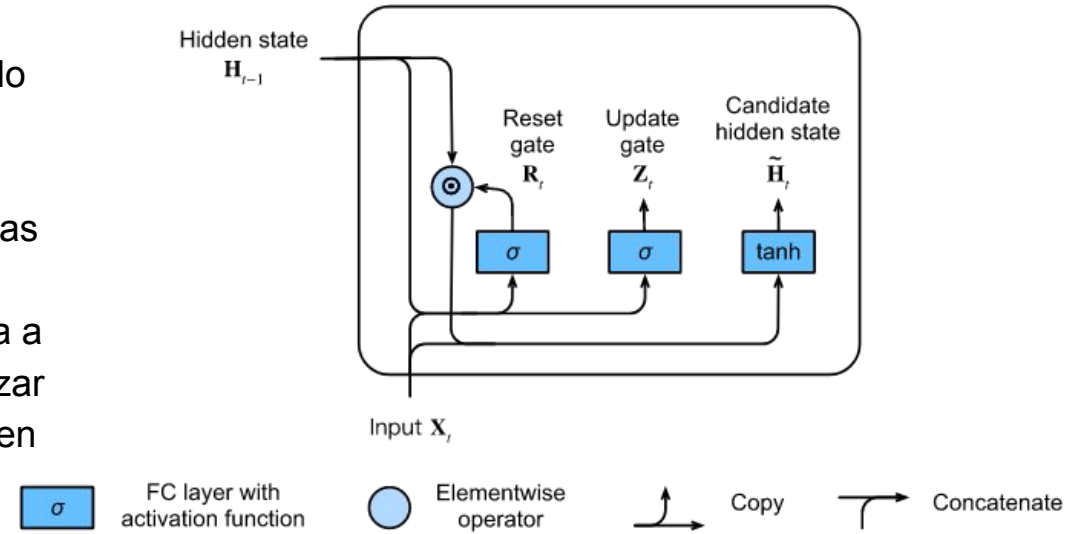
$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r)$$
$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)$$

$$W_{xr}, W_{xz} \epsilon \mathbb{R}^{d \times h}$$
$$W_{hr}, W_{hz} \epsilon \mathbb{R}^{h \times h}$$
$$b_r, b_z \epsilon \mathbb{R}^{1 \times h}$$

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

GRU: Candidato de estado oculto

- Dado esto, definimos un vector de estado "candidato" usando $\tilde{\mathbf{H}}$:
- Esto combina los estados anteriores *ocultos* que no se reinician con las nuevas *entradas*.
- La combinación lineal resultante se pasa a través de una función ***tanh*** para garantizar que las unidades ocultas permanezcan en el intervalo (−1, 1).
- Si las entradas de la puerta de reinicio $\mathbf{R}_t$ están cerca de 1, recuperamos la regla de actualización RNN estándar.

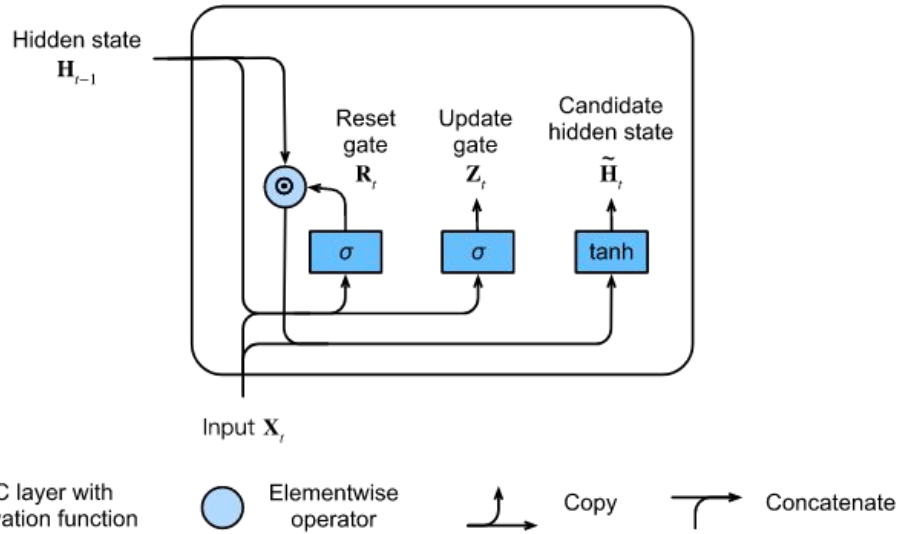


$$\tilde{H} = tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h)$$

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

GRU: Candidato de estado oculto

- Si las entradas están cerca de **0**, el modelo actúa más como un MLP aplicado a $\mathbf{X_t}$. Por lo tanto, la puerta de reinicio puede capturar información nueva a corto plazo.
- El resultado es un candidato, ya que todavía necesitamos incorporar la acción de la puerta de actualización.



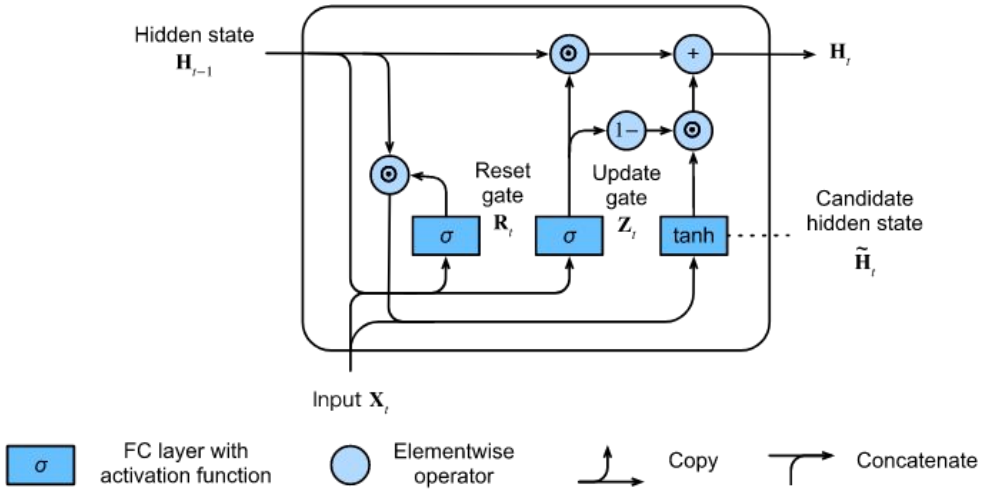
$$W_{xh} \epsilon \mathbb{R}^{d \times h}$$
$$W_{hh} \epsilon \mathbb{R}^{h \times h}$$
$$b_h \epsilon \mathbb{R}^{1 \times h}$$

$$\tilde{H} = tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h)$$

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

GRU: Estado Oculto

- Una vez que hemos calculado el nuevo estado candidato, el modelo calcula el nuevo estado real usando las dimensiones del estado candidato $\tilde{\mathbf{H}}_t$ elegido por la puerta de actualización, $\mathbf{1} - \mathbf{Z}_t$, y manteniendo las dimensiones restantes en sus valores anteriores de $\mathbf{H}_{t-1}$
- Cuando $\mathbf{Z}_{td}$ es cercana a **1**, pasamos $\mathbf{H}_{t-1,d}$ sin cambios e ignoramos $\mathbf{X}_t$. Por lo tanto, la puerta de actualización puede capturar dependencias a largo plazo.



$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t$$

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

GRU

- Los diseños de la GRU ayudan a lidiar con el problema del gradiente descendiente en las RNN, así como ayudar a capturar mejor las dependencias para secuencias con grandes distancias (pasos) de tiempo.
- Las puertas de reinicio ayudan a capturar dependencias a corto plazo en secuencias.
- Las puertas de actualización ayudan a capturar dependencias a largo plazo en secuencias.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Teacher Forcing

Teacher Forcing

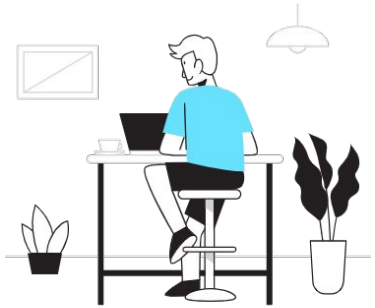
a. **Parte**

b. **Parte**

c. **Parte**

calcular cada parte, su respuesta será usada en la siguiente parte

Objetivo



Teacher Forcing

a. **Parte**

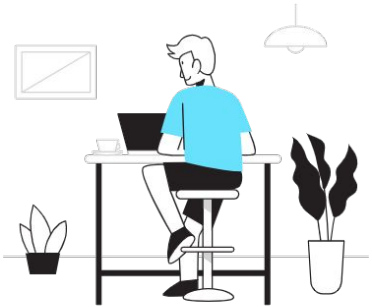
b. Parte

c. Parte

Resp A

Objetivo

compara tu respuesta con la correcta



y proporciona la respuesta correcta para que se pueda usar en la parte (**b**).

registra la puntuación de la respuesta de la parte (**a**)

Teacher Forcing

a. Parte

b. Parte

c. Parte

Resp A

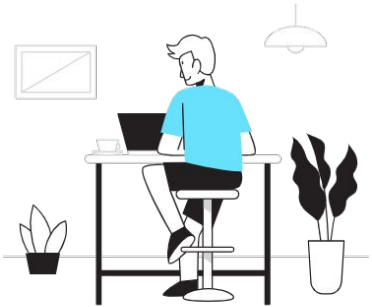
Resp B

Objetivo

compara tu respuesta con la correcta

registra la puntuación de la respuesta de la parte (**b**)

y proporciona la respuesta correcta para que se pueda usar en la parte (**c**).



Teacher Forcing

a. Parte

b. Parte

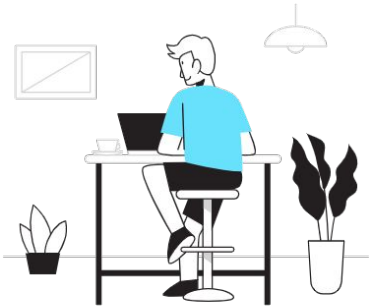
c. Parte

Resp A → Resp B → Resp C → **Objetivo**

compara tu respuesta con la correcta

registra la puntuación de la respuesta de la parte (**c**)

y proporciona la respuesta correcta



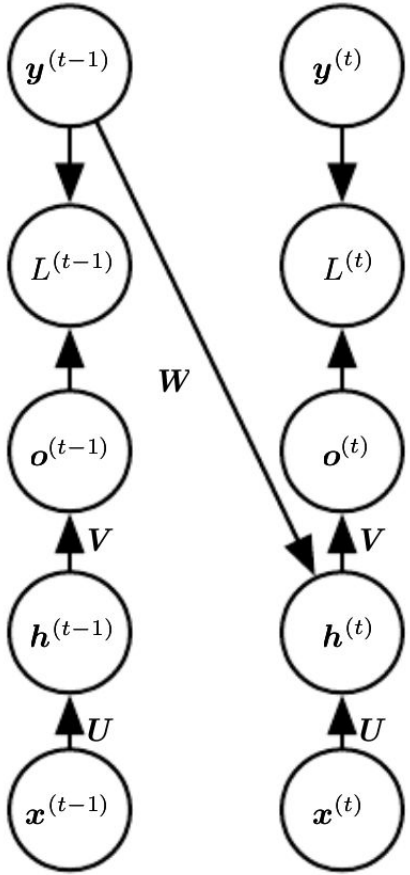
Teacher Forcing

- Los modelos que tienen conexiones recurrentes desde sus salidas que conducen de regreso al modelo pueden ser entrenados con **Teacher Forcing**.
- **Teacher forcing** es un procedimiento que surge de la función de pérdida de máxima verosimilitud (maximum likelihood), durante el entrenamiento el modelo recibe la salida real verdadera (ground truth) $\mathbf{y}^{(t)}$ como entrada en el tiempo **t+1**.
- Por lo tanto, la máxima verosimilitud especifica que durante el entrenamiento, en lugar de retroalimentar la propia salida del modelo, estas conexiones deben alimentarse con los valores objetivo que especifican cuál debería ser la salida correcta.

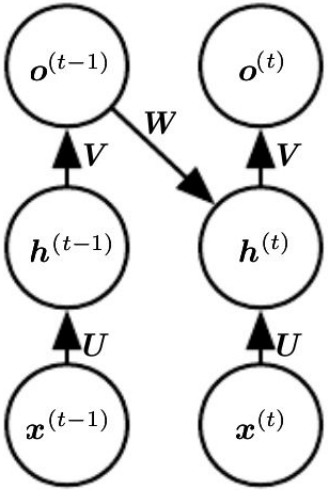
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Teacher Forcing

- Teacher Forcing es una técnica de entrenamiento que es aplicable a RNNs que tienen conexiones desde su salida a sus estados ocultos en el siguiente paso de tiempo.
- En el *Train time*, se alimenta la salida correcta $\mathbf{y^{(t)}}$ extraída del conjunto *train* como entrada en $\mathbf{h^{(t+1)}}$.
- Cuando se implementa (deploy) el modelo, generalmente no se conoce el verdadero resultado.
- En este caso, se aproxima la salida correcta $\mathbf{y^{(t)}}$ con la salida del modelo $\mathbf{o^{(t)}}$ y se alimenta la salida nuevamente al modelo.



Train time



Test time

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

Teacher Forcing

- La motivación original de usar Teacher Forcing fue para evitar la propagación hacia atrás a través del tiempo (BPTT) en modelos que carecen de conexiones de *hidden*-to-hidden.
- Teacher Forcing aún se puede aplicar a los modelos que tienen conexiones *hidden-to-hidden*, siempre que tengan conexiones desde la salida en un paso de tiempo a los valores calculados en el siguiente paso de tiempo.
- Sin embargo, tan pronto como las unidades ocultas se vuelven una función de pasos de tiempo anteriores, el algoritmo BPTT es necesario.
- Por lo tanto, algunos modelos pueden ser entrenados tanto con Teacher Forcing como con el BPTT.

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

Teacher Forcing

- Pro:
  - El entrenamiento converge más rápido.
  - En las primeras etapas del entrenamiento, las predicciones del modelo son muy malas. Así, al aplicar Teacher Forcing, se evita que los estados ocultos del modelo se actualicen mediante una secuencia de predicciones incorrectas, se acumule errores y con ello el modelo pueda generalizar mejor.
- Contra
  - Durante la inferencia, dado que generalmente no hay datos reales disponibles, el modelo RNN deberá retroalimentarse con su propia predicción anterior para la próxima predicción.
  - Por lo tanto, existe una discrepancia entre el entrenamiento y la inferencia, y esto podría provocar inestabilidad y un rendimiento deficiente del modelo.
  - Esto se conoce como sesgo de exposición (Exposure Bias) en la literatura.

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

Teacher Forcing

- Dado que pasamos toda la secuencia de la verdad fundamental a través del modelo RNN, ¿Es posible que el modelo "haga trampa" simplemente memorizando el objetivo real (*target*)?
  - **No**. En el paso de tiempo **t**, la entrada del modelo es el *target* en el paso de tiempo **t-1**, y los estados ocultos del modelo han sido actualizados por los *targets* desde el paso de tiempo **t-2**. El modelo nunca puede mirar hacia el futuro.
- ¿Se utiliza Teacher Forcing fuera del procesamiento del lenguaje natural?
  - **Si**. Se puede usar en cualquier modelo que genere secuencias, por ejemplo en el pronóstico de series de tiempo.
- ¿Se utiliza Teacher Forcing fuera de las redes neuronales recurrentes?
  - **Si**. Se utiliza en otros modelos autorregresivos como el Transformer.

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