Deep Learning para NLP - Fundamentals

Mariano Crosetti

Rosario, Argentina
Universidad Austral
Maestría en Explotación de Datos y Gestión del Conocimiento

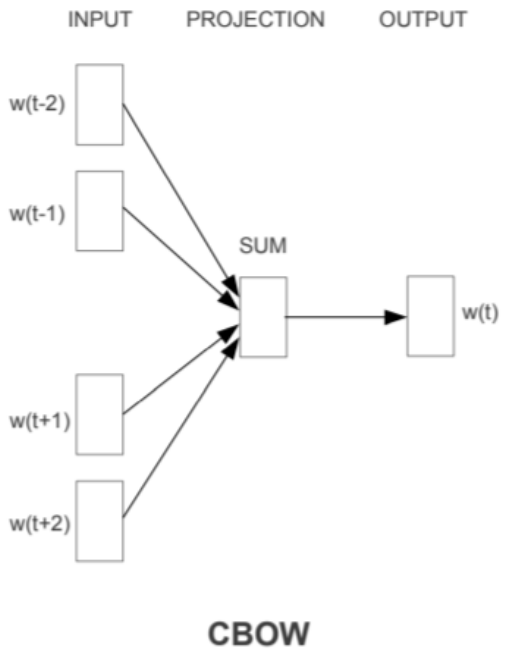
"Si he logrado ver más lejos ha sido porque he subido a hombros de gigantes" - Isaac Newton

Contenidos I

Repaso

- **Word Embedding:** vector $\mathbb{R}^d$ que "codifica el significado de las palabras"
  - Palabras semánticamente similares están cerca en el espacio.
  - Tiene otras propiedades útiles como que "direcciones con significados".
- Los word embeddings se entrenan: empezamos un vector random y los actualizamos (utilizando descenso por gradiente) para predecir:
  - **CBOW:** Una palabra dado las palabras de su contexto.
  - **Skip-gram:** Las palabras del contexto dado su centro.

$$P(w_{t+1} = x | w_t = y) = \frac{\exp(v_x . v_y)}{\sum_{z \in V} \exp(v_z . v_y)}$$



Repaso

- **Text Embedding:** vector $\mathbb{R}^d$ que "codifica el significado de un texto".
  - Textos similares están cerca en el espacio.
  - El embedding de una pregunta está cerca del embedding de las respuestas.
  - Se puede usar para tareas como clasificación de texto.
  - Para calcularlo se puede tomar la media de los word embeddings. Es simple pero no tiene en cuenta el orden de las palabras (es un modelo *bag of words*), por lo que no es muy efectivo para casos que requieren un entendimiento profundo.
- **Language Model:** modelo que predice la probabilidad de la siguiente palabra:

$$P(x_{t+1}|x_1 x_2 x_3 \ldots x_t)$$

Modelo de lengaujes

Cómo se implementa un modelo de lenguaje?

- Intento 1 (rudimentario):

Modelo de lengaujes

Cómo se implementa un modelo de lenguaje?

- Intento 1 (rudimentario): contar para cada n-grama cuál es la siguiente palabra más probable (para un n elegido).
- Intengo 2:

Modelo de lengaujes

Cómo se implementa un modelo de lenguaje?

- Intento 1 (rudimentario): contar para cada n-grama cuál es la siguiente palabra más probable (para un n elegido).
- Intengo 2: usemos una red neuronal que dado los últimas n palabras prediga la siguiente palabra. Podemos usar todas capas densas.
  - Si tenemos embeddings entrenados de dimensión d, la entrada de la red podría ser un vector de dimensión

Modelo de lengaujes

Cómo se implementa un modelo de lenguaje?

- Intento 1 (rudimentario): contar para cada n-grama cuál es la siguiente palabra más probable (para un n elegido).
- Intengo 2: usemos una red neuronal que dado los últimas n palabras prediga la siguiente palabra. Podemos usar todas capas densas.
  - Si tenemos embeddings entrenados de dimensión d, la entrada de la red podría ser un vector de dimensión $n \times d$.

$$\mathbb{R}^{n \times d} \to \mathbb{R}^{h_1} \to \mathbb{R}^{h_2} \to \ldots \to \mathbb{R}^{|V|}$$

  - Sigue teniendo el problema de contexto acotado.
  - Otro problema: la manera en la que interactúan las palabras depende completamente de la posición, cuando en realidad sólo importa la posición relativa.

Redes Neuronales Recurrentes

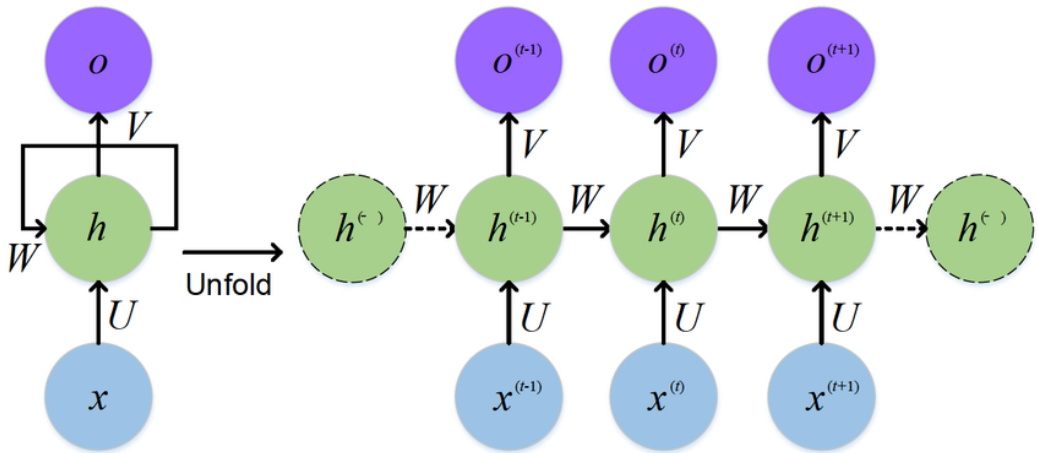
Fueron inventadas para procesar secuencia de valores x_t o embeddings.

$$h_t = tanh(b + Wh_{t-1} + Ux_t)$$

$$o_t = softmax(c + Vh_t)$$

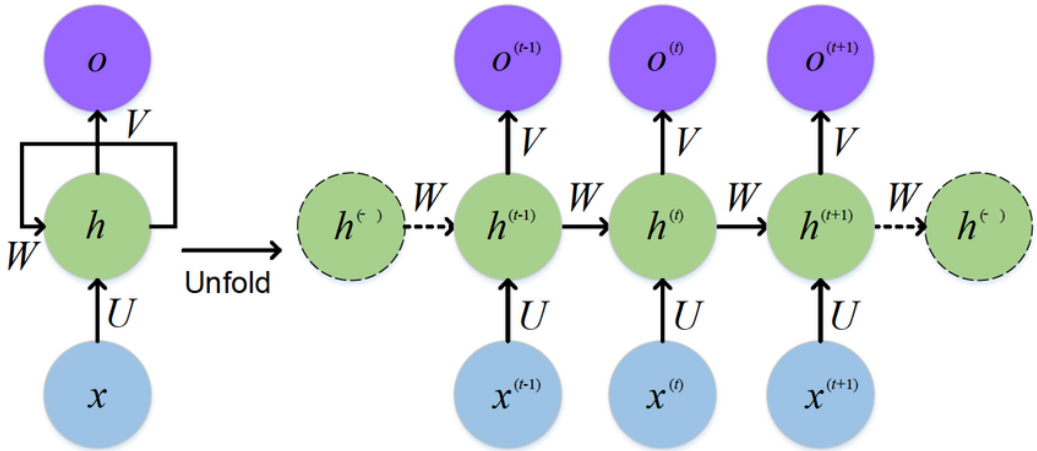
La formula nos indica como calcular el nuevo estado con el estado anterior y el input. W, U, V son matrices con parámetros entrenables.

- Tomaon como input:
  - Un embedding o valor actual.
  - Un embedding del “estado anterior” (inicialmente cero).
- Y producen:
  - Un nuevo “estado” (para aplicar sucesivamente la RNN).
  - Un output. A veces se usa sólo el último output.



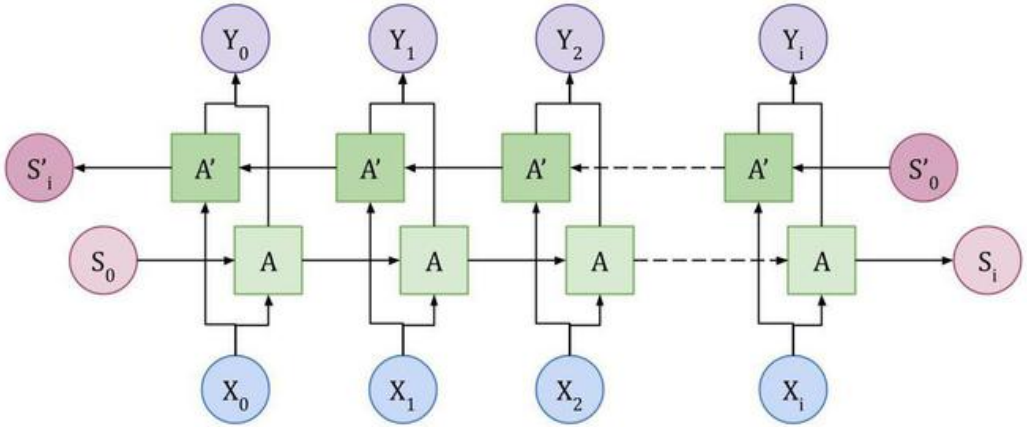
Seq2Seq

- Permite modelar problemas que toman una secuencia y/o producen una secuencia (seq2seq). Ejemplos:
  - Dado la secuencia de caracteres de un nombre determinar el origen étnico: chino, español, alemán o inglés. (seq2value).
  - Dado un origen étnico generar un nombre que luzca de dicho origen. (value2seq).
  - Dado un texto en inglés, producir texto en español (seq2seq).

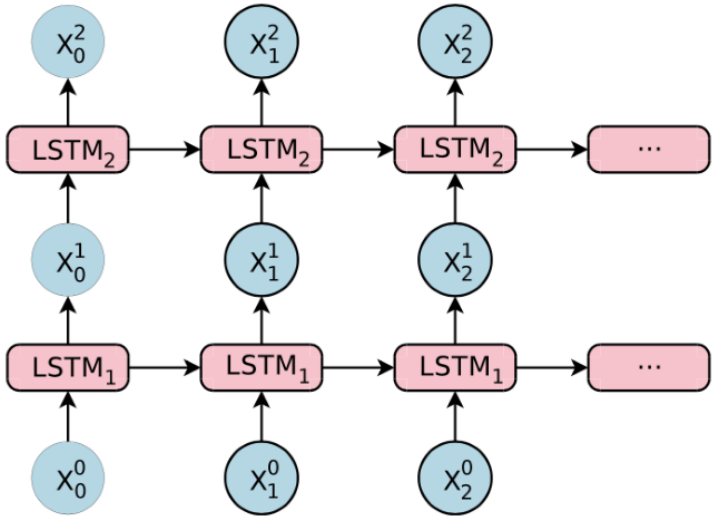


RNN

- Bidirecionalidad: a veces se entrena un modelo en la secuencia y otro en la secuencia invertida y finalmente se combinan los outputs.

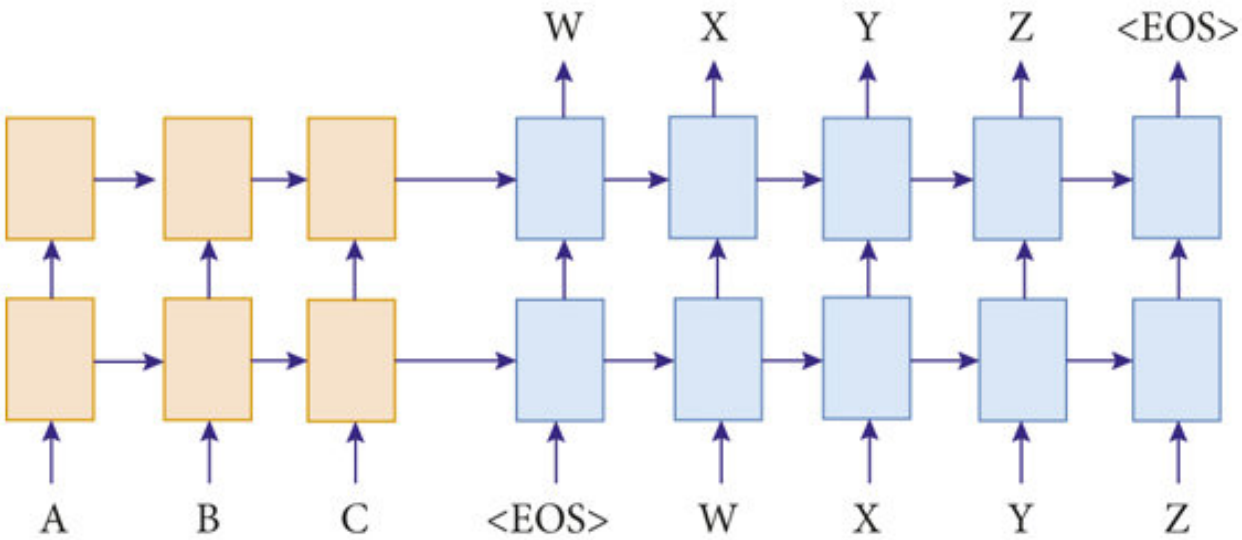


- Stacked: a veces se “apilan” redes recurrentes. El output de una es el input de la siguiente “capa”



RNN

- Text Embedding: Como los h_i guardan información de la secuencia, tomar el último h_i puede ser una buen embedding del texto.
- Problemas como los de Machine Translation donde la secuencia de input es muy distinta que la de output se usan distintas RNN (distintos pesos) para ambas partes:

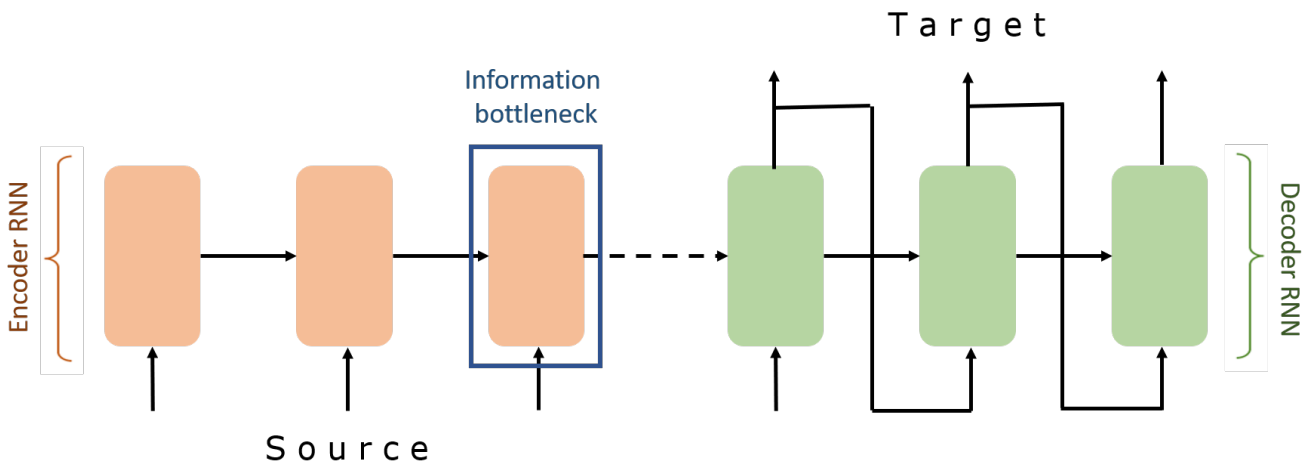


  - Se entrenan ambas partes a la vez.
  - Durante entrenamiento en el input del decoder se usa el source of truth como input y el output sólo se usa para calcular el costo (“teacher forcing”).
  - Durante inferencia no tenemos source of truth. El output del decoder alimenta al mismo decoder.

LSTMs

- Como el “estado oculto” es lo único que mantiene una memoria sobre los valores anteriores, las RNN tienen una capacidad limitada de memoria.
- Por ejemplo les cuesta el simple problema de detectar si una secuencia de paréntesis está bien parenteseada.
- Las Long Short-Term Memory son RNN dónde la formulación matemática de cómo se calcula el estado oculta mitiga este problema.
- Todas las cosas que podíamos hacer en una RNN se pueden hacer en LSTM: las podemos apilar y podemos hacer LSTM bidireccionales.

Pero siguen teniendo el bottle-neck problem:

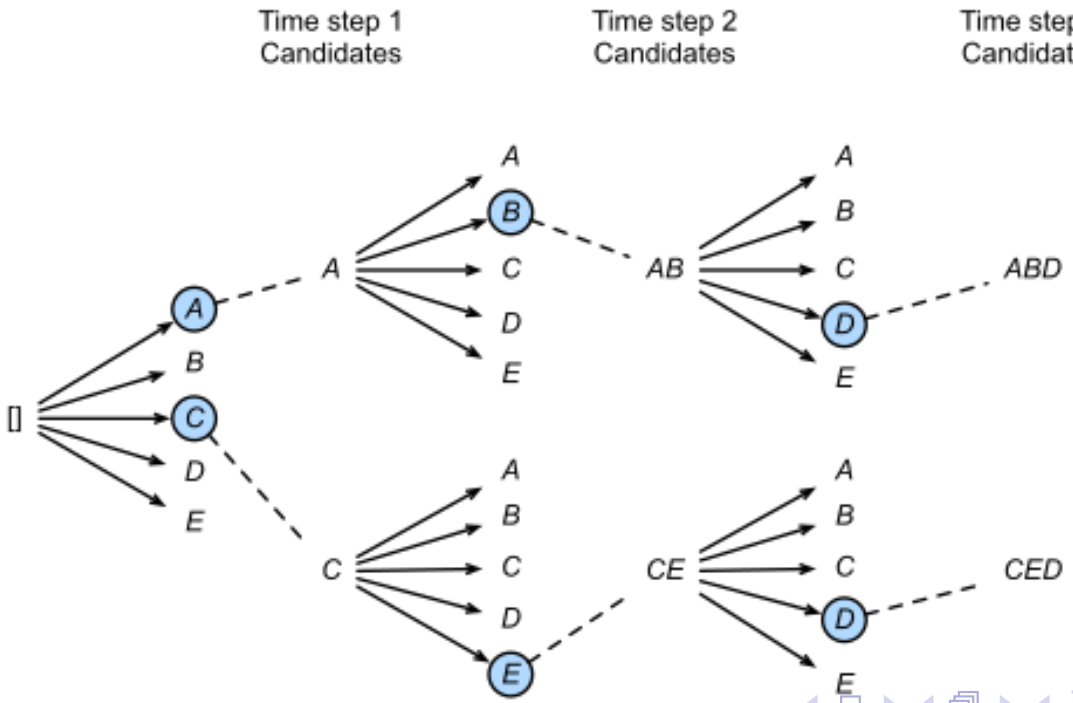


Por ejemplo en Machine Translation, el hidden state entre el Encoder y el Decoder tiene que guardar toda la información de la oración a traducir.

Decoding strategy

Nuestro modelo calcula $P(w_{t+1}|w_1 w_2 ... w_t)$ como usamos esto para generar texto:

- Greedy: elegir siempre la palabra más probable. El problema es que la mejor decisión local no necesariamente produce la secuencia más probable.
- Beam-search: Siempre quedarme con los k mejores candidatos. Para cada uno de ellos explorar como se modifica su score agregando una palabra.



Métricas

Qué métricas podemos usar?

- BLEU: que proporción de los n-gramas generados aparecen en el source of truth (cuántos n-gramas son correctos o explicados). Además incluye un término de penalización para evitar una generación vacía.
- ROUGE: que proporción de los n-gramas del source of truth fueron generados.
- BLEU mide precision mientras que ROUGE mide recall
- Como hay muchas maneras respuestas correctas para los problemas de machine translation, question answering o summarization, no hay una métrica automática que sea perfecta.
- La inspección humana manual es la mejor métrica.

Attention Mechanism