# Procesamiento del Lenguaje Natural mediante Redes Neuronales

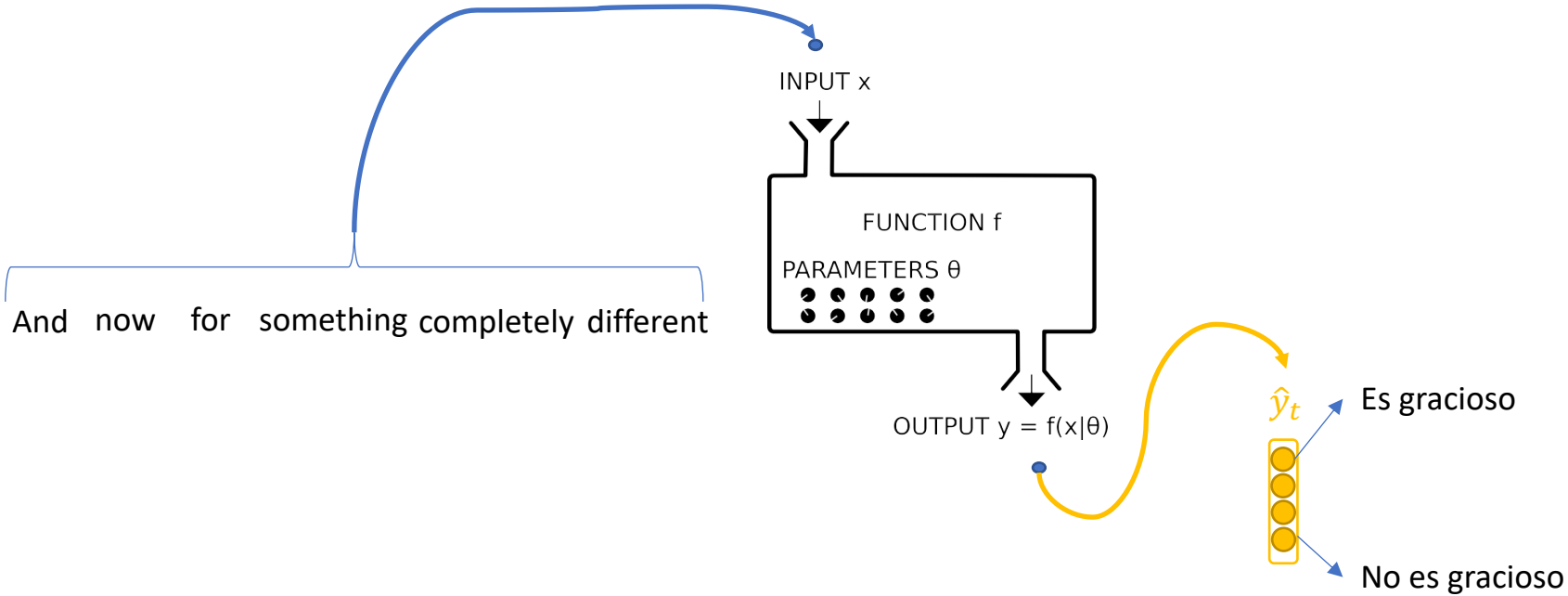
Día 5: Transformers

Germán Kruszewski
Facebook AI Research

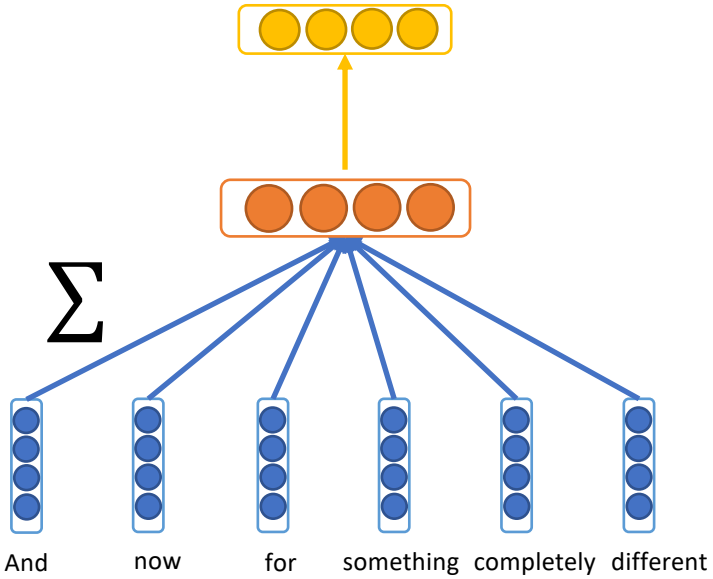
*“Young man, in mathematics you don't understand things. You just get used to them.”*

- John von Neumann

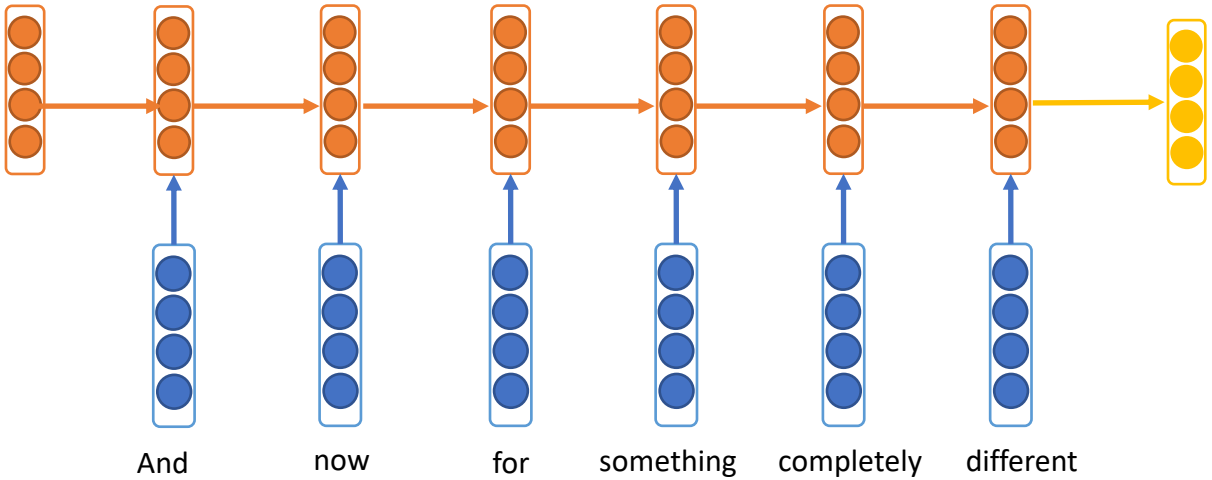
# Redes neuronales como funciones



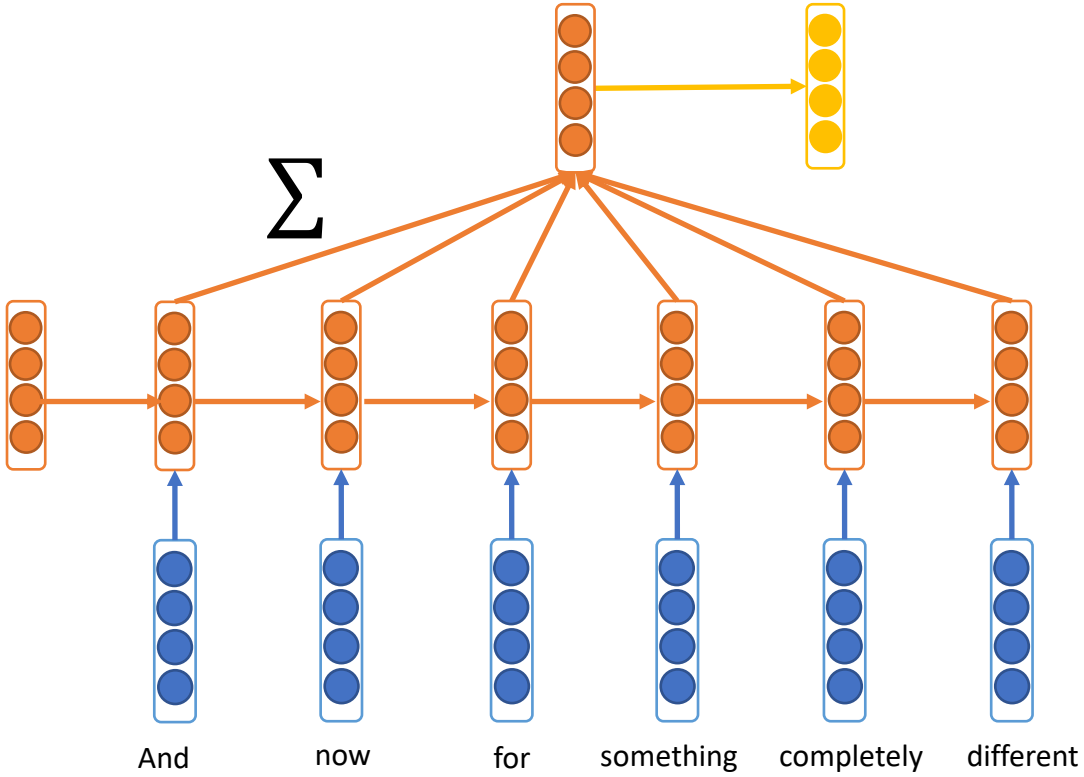
# Función
# Bag of Words/Embeddings



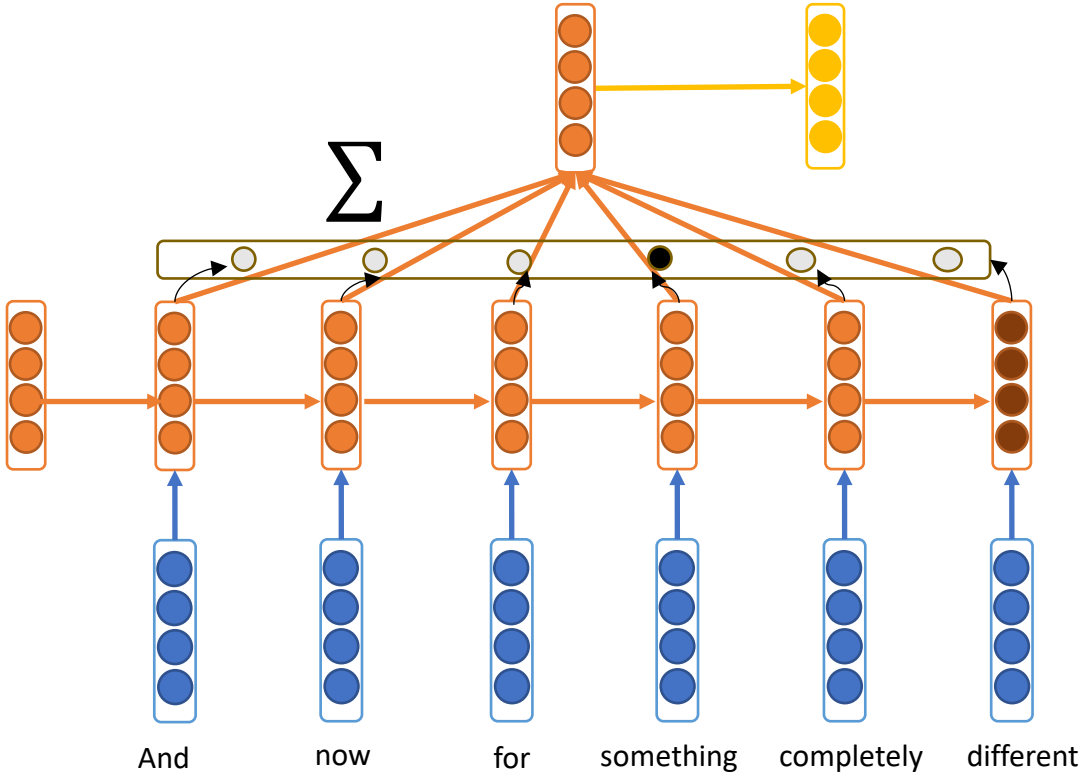
# Función
# Red Neuronal Recurrente



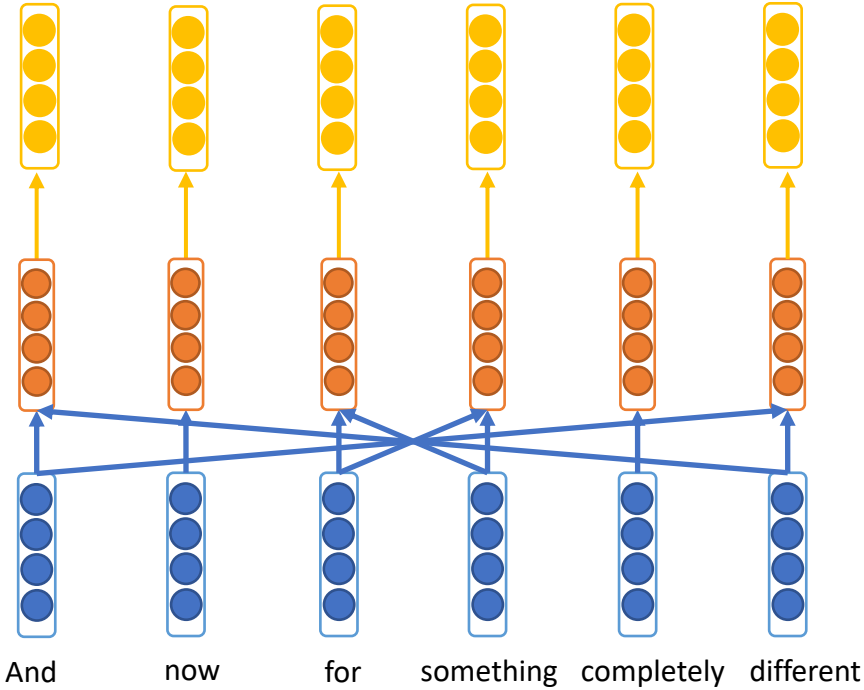
# Función
# Red Neuronal Recurrente con estados agregados

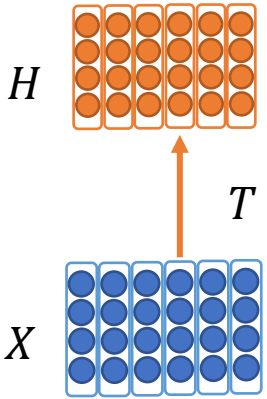


# Función
# Red Neuronal Recurrente con atención

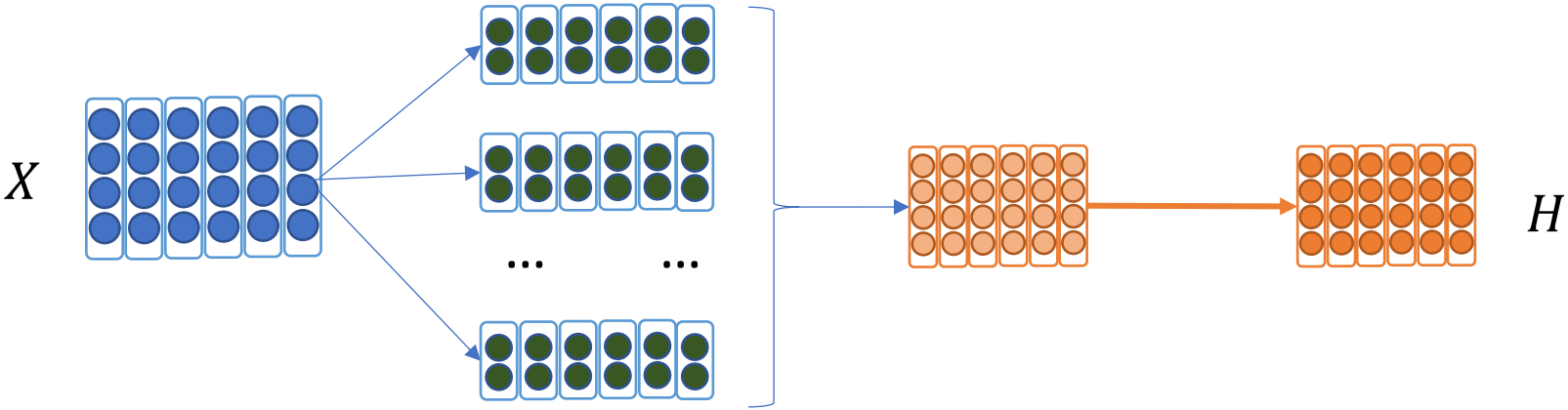


# Función de transformación

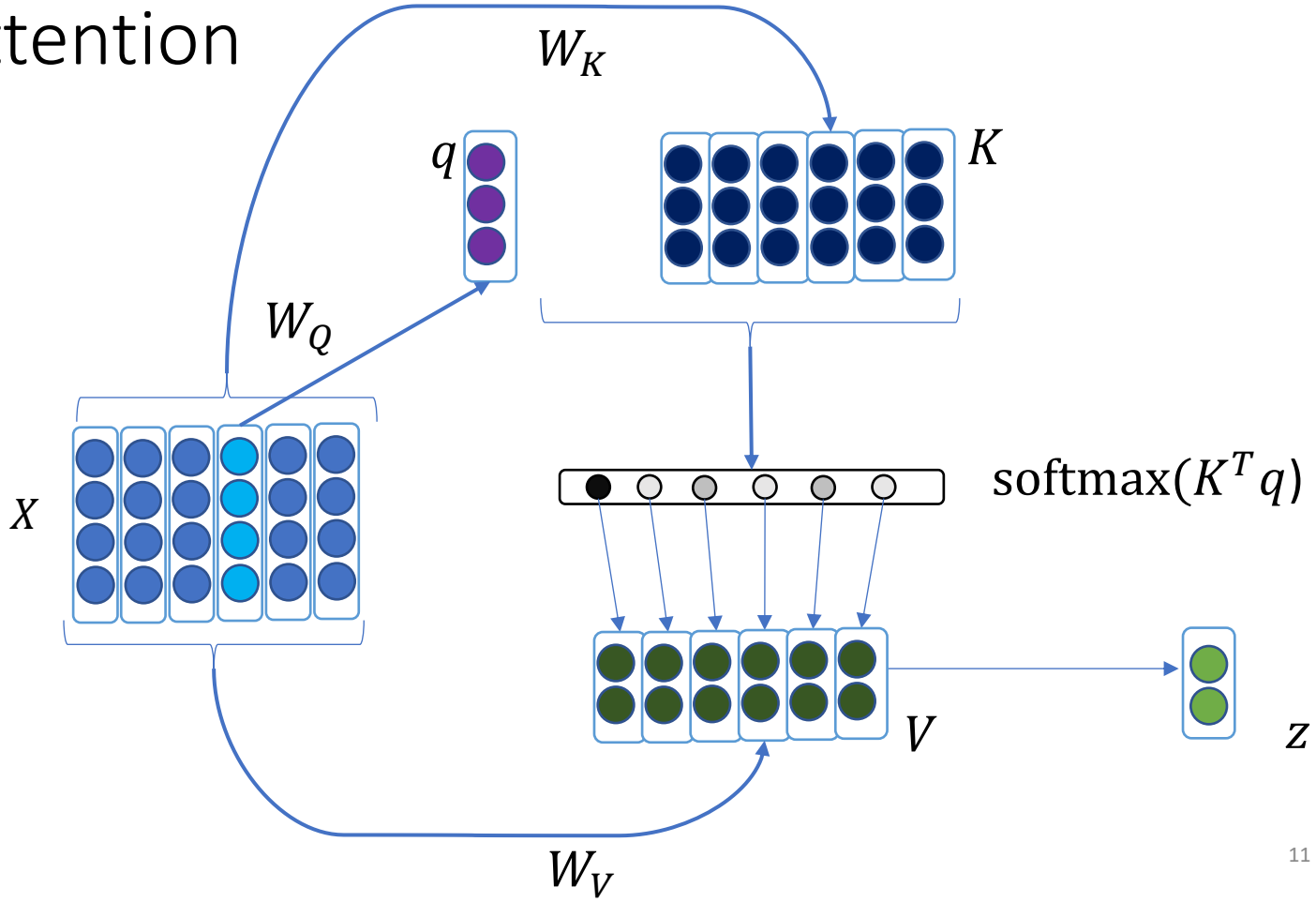


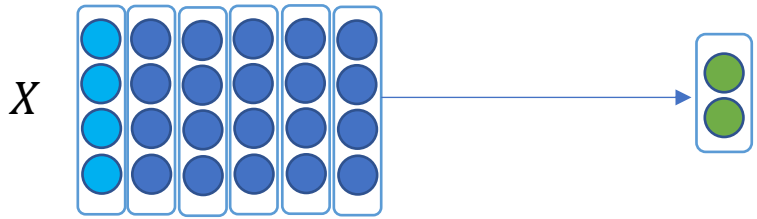
# Función de transformación



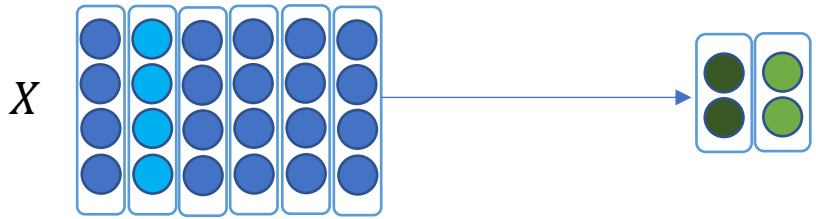
# Self-attention



# Self-attention

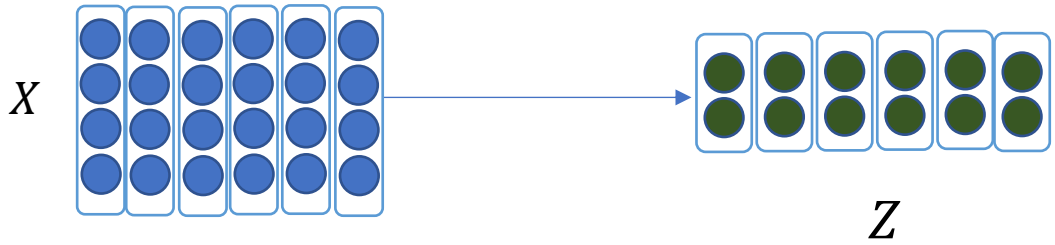


# Self-attention

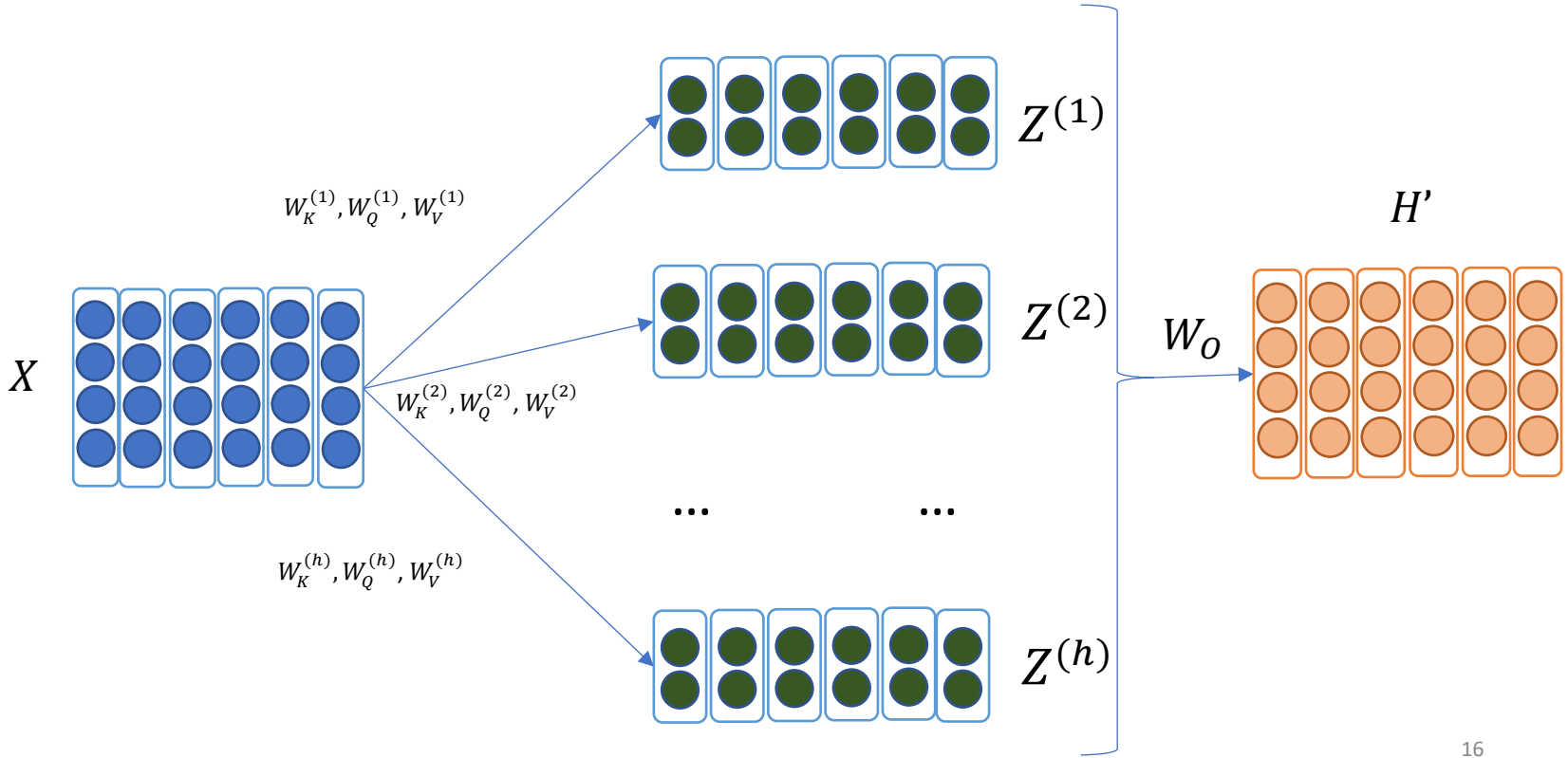


# Self-attention

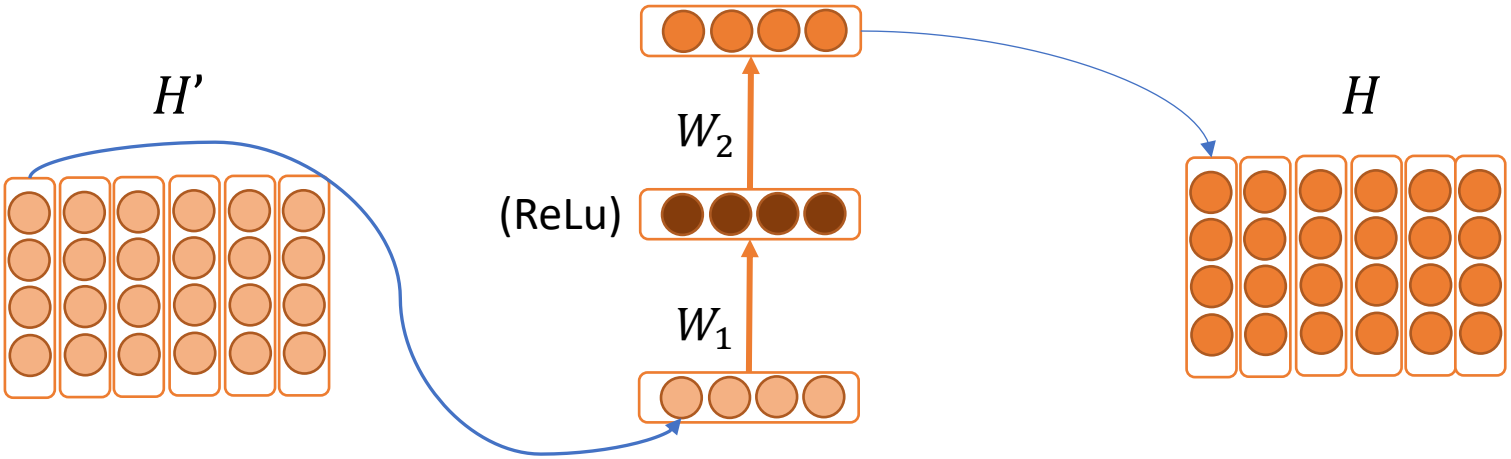
# Self-attention



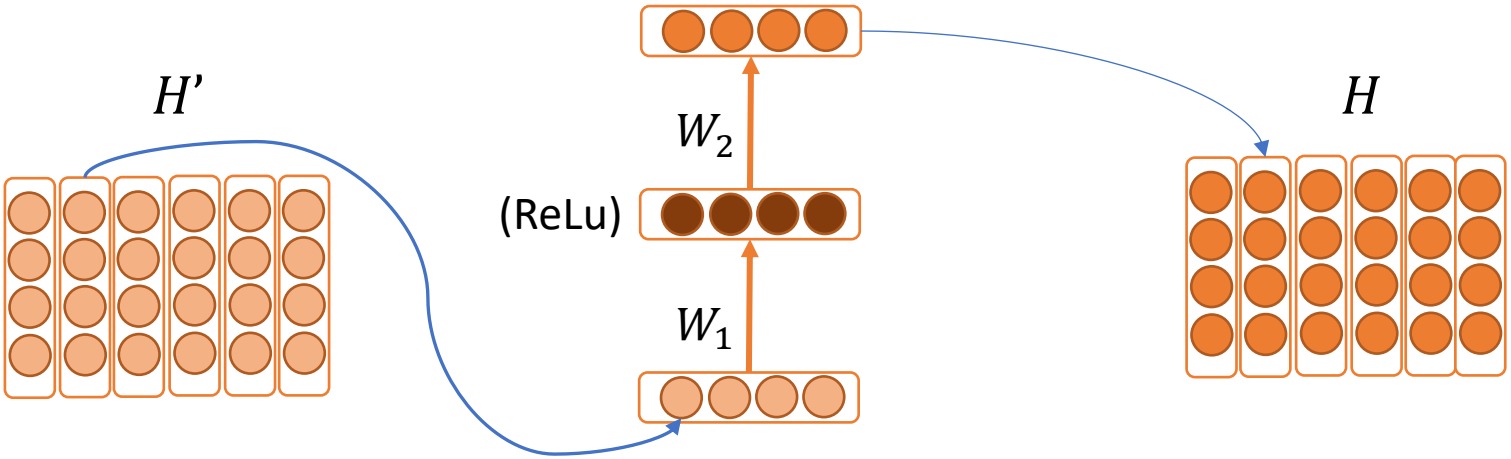
# Multi-head attention



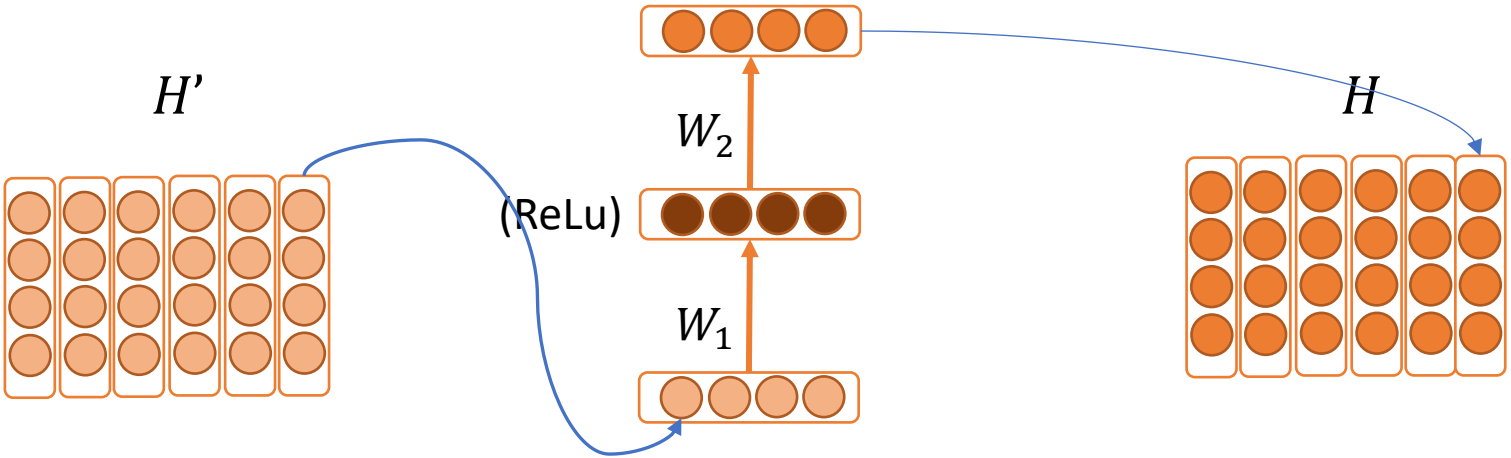
# Feed-forward



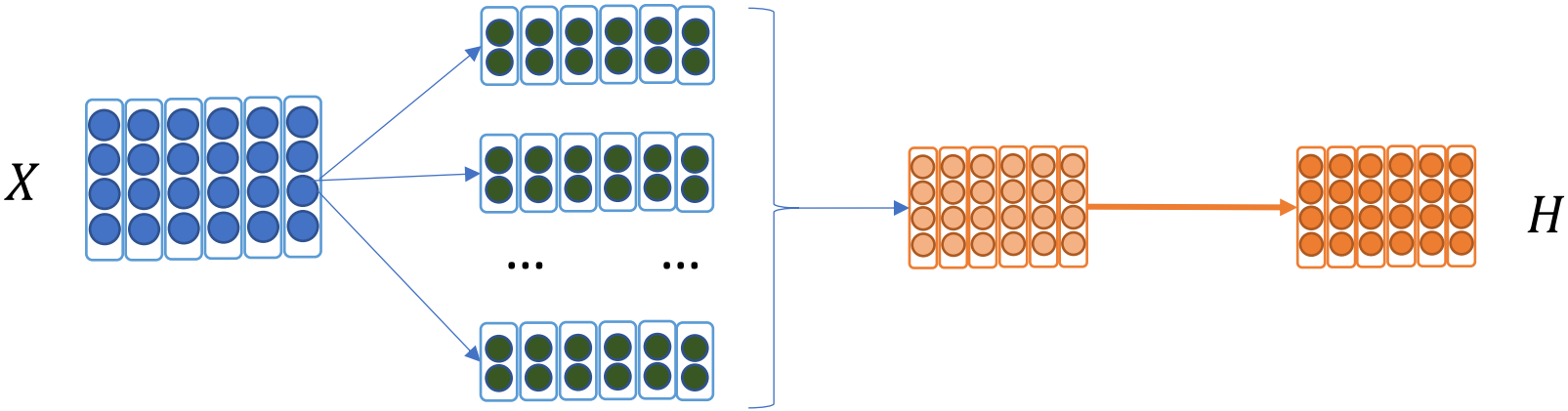
# Feed-forward



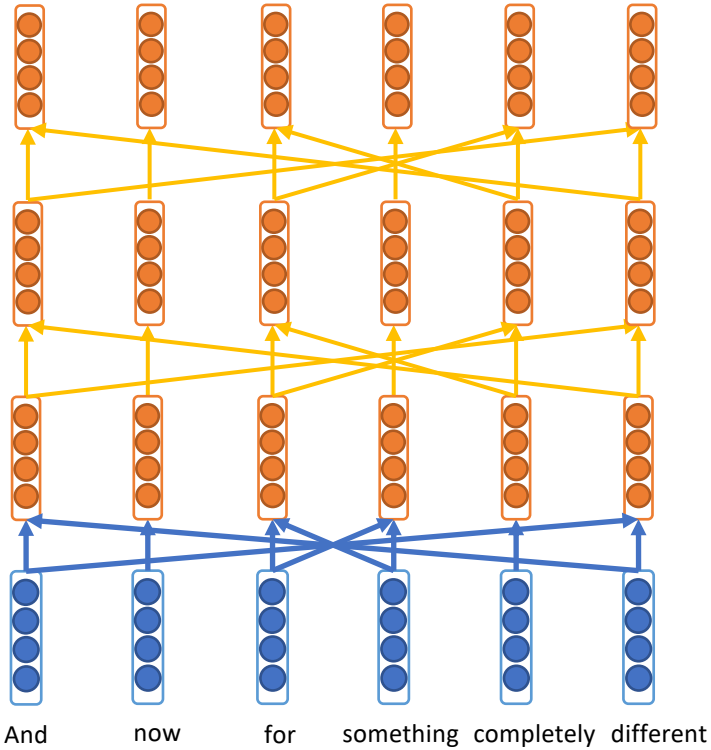
# Feed-forward

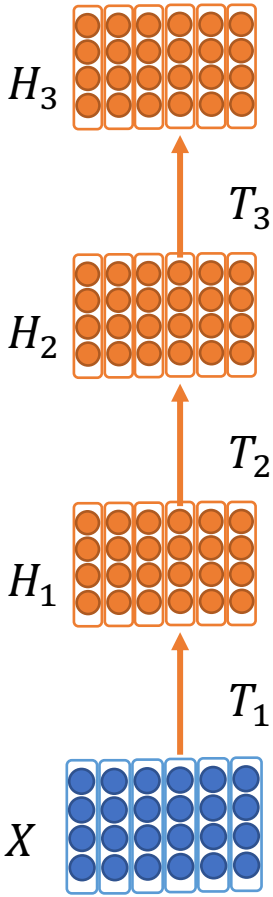


# Función de transformación

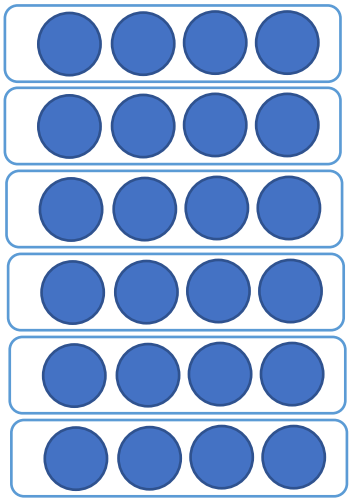


# Función de transformación

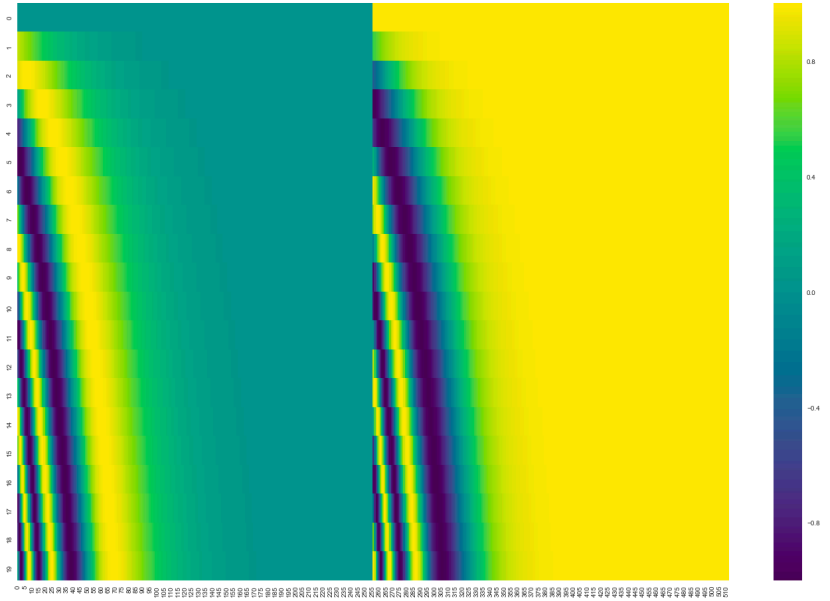


# Positional embeddings



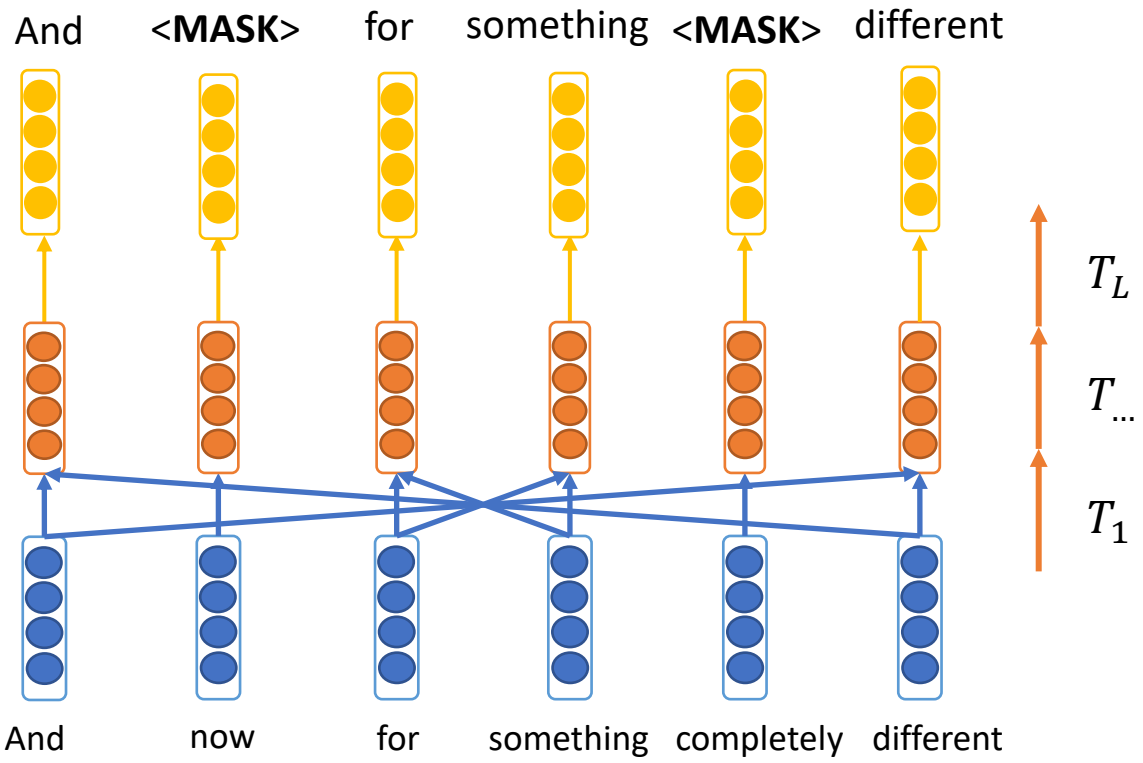
+



http://jalammar.github.io/illustrated-transformer/

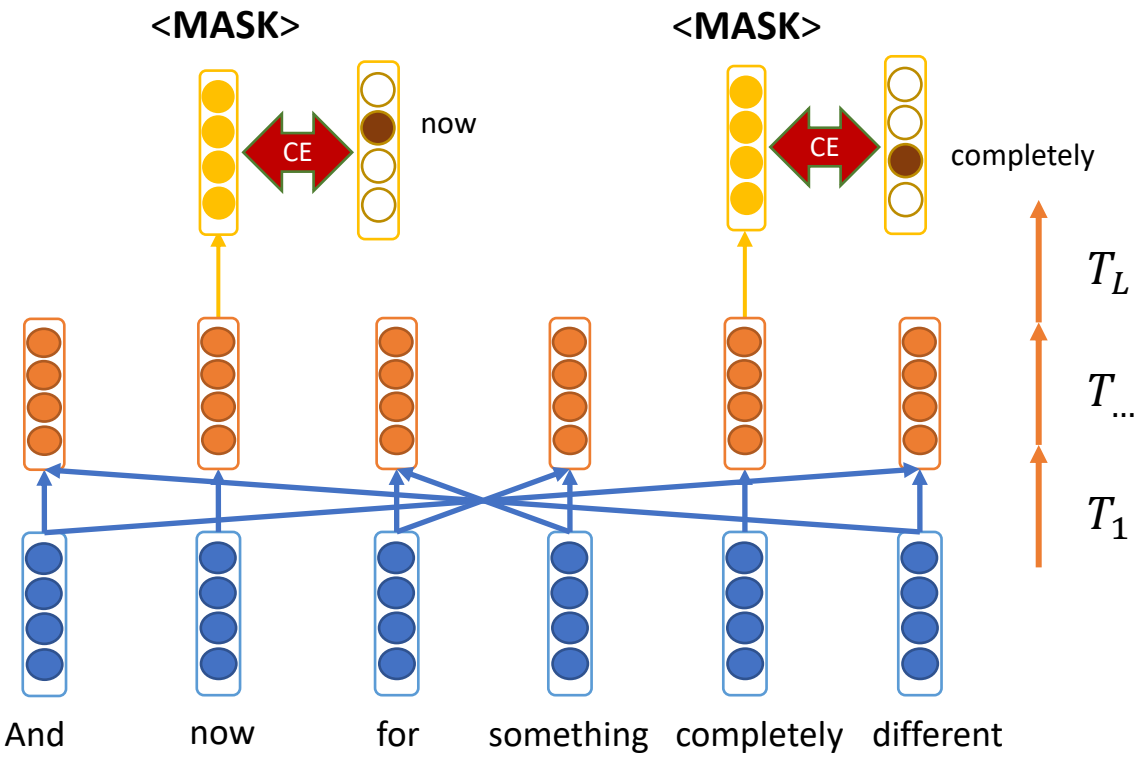
# Entrenamiento por predicción *Cloze*

Devlin et al. (2018)



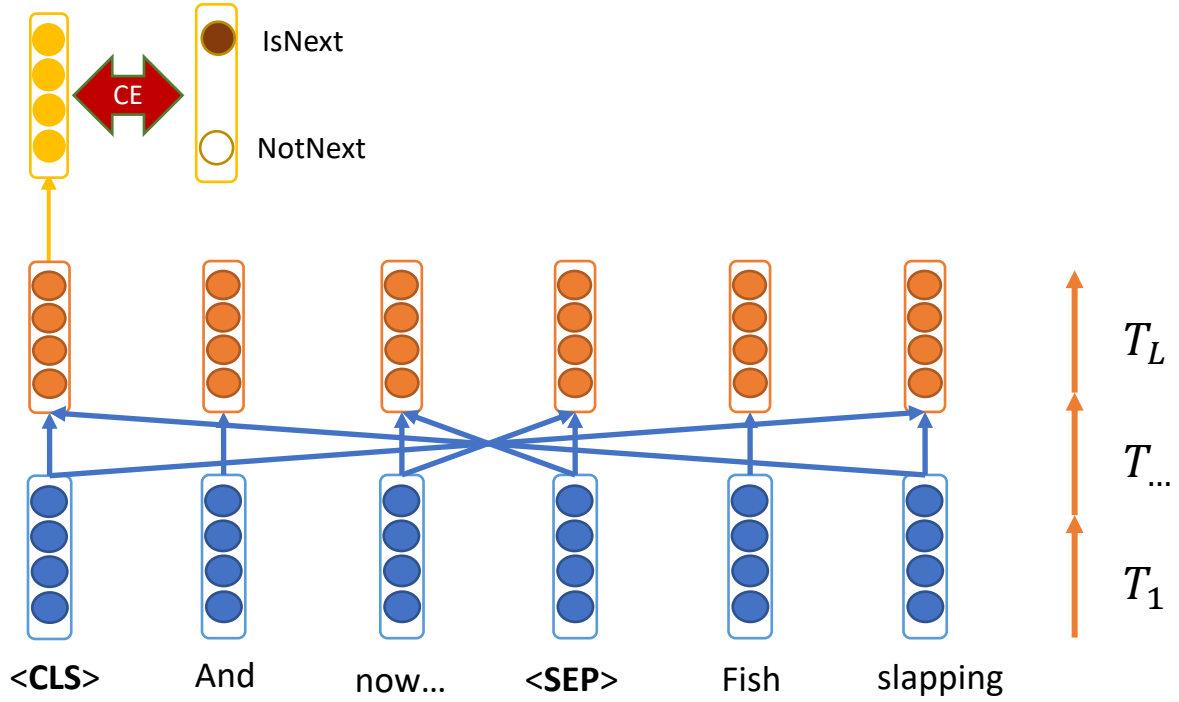
# Entrenamiento por predicción *Cloze*

Devlin et al. (2018)



# Entrenamiento por predicción de la siguiente oración

Devlin et al. (2018)



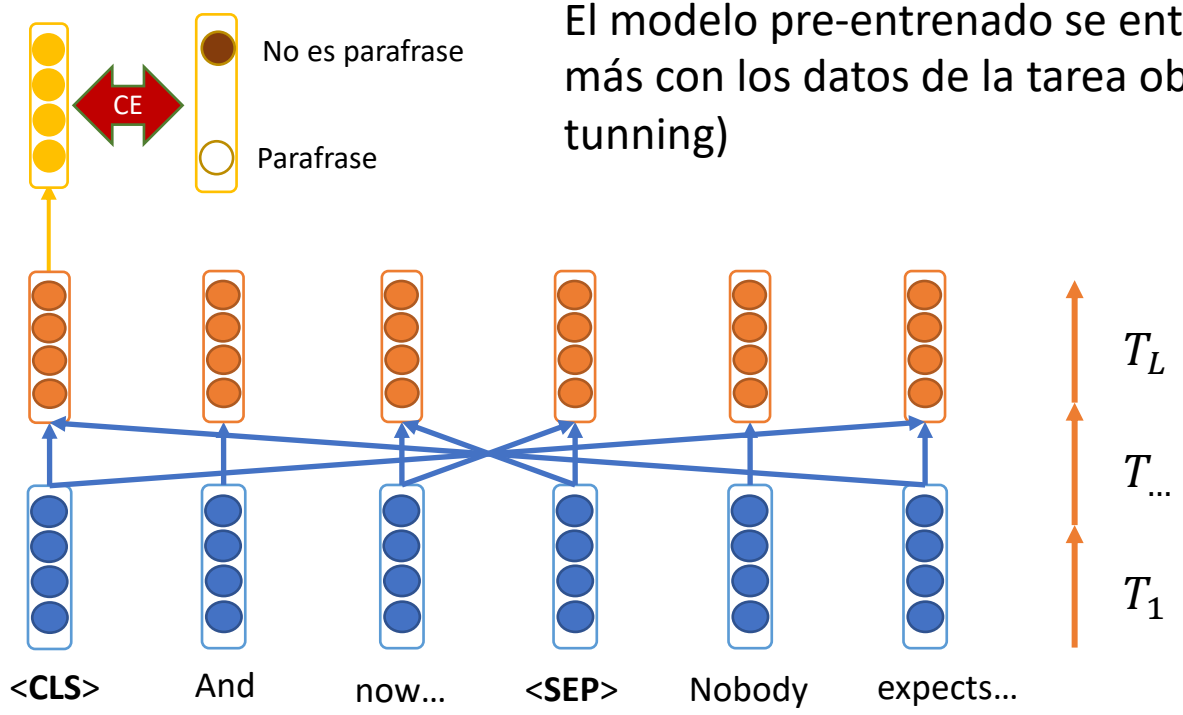
# GLUE: Evaluando un modelo para todo

Wang et al. (2019)

- Clasificación de oraciones:
  - Aceptabilidad gramatical
  - Sentiment analysis
- Similitud y parafrases
- Inferencia

# Aplicando BERT

Devlin et al. (2018)




El modelo pre-entrenado se entrena un poco más con los datos de la tarea objetivo (fine-tunning)

# Resultados de BERT

Devlin et al. (2018)

| System | MNLI-(m/mm) 392k | QQP 363k | QNLI 108k | SST-2 67k | CoLA 8.5k | STS-B 5.7k | MRPC 3.5k | RTE 2.5k | **Average** - |
|---|---|---|---|---|---|---|---|---|---|
| Pre-OpenAI SOTA | 80.6/80.1 | 66.1 | 82.3 | 93.2 | 35.0 | 81.0 | 86.0 | 61.7 | 74.0 |
| BiLSTM+ELMo+Attn | 76.4/76.1 | 64.8 | 79.8 | 90.4 | 36.0 | 73.3 | 84.9 | 56.8 | 71.0 |
| OpenAI GPT | 82.1/81.4 | 70.3 | 87.4 | 91.3 | 45.4 | 80.0 | 82.3 | 56.0 | 75.1 |
| $\text{BERT}_{\text{BASE}}$ | 84.6/83.4 | 71.2 | 90.5 | 93.5 | 52.1 | 85.8 | 88.9 | 66.4 | 79.6 |
| $\text{BERT}_{\text{LARGE}}$ | **86.7/85.9** | **72.1** | **92.7** | **94.9** | **60.5** | **86.5** | **89.3** | **70.1** | **82.1** |

# Modelos basados en Transformers

- Vanilla Transformer: Vaswani et al. (2017). Attention is All you Need. https://arxiv.org/abs/1706.03762

- GPT/GPT-2: Radford et al . (2018). Improving Language Understanding by Generative Pre-Training. https://openai.com/blog/language-unsupervised/

- BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/pdf/1810.04805.pdf

- Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. https://arxiv.org/abs/1901.02860

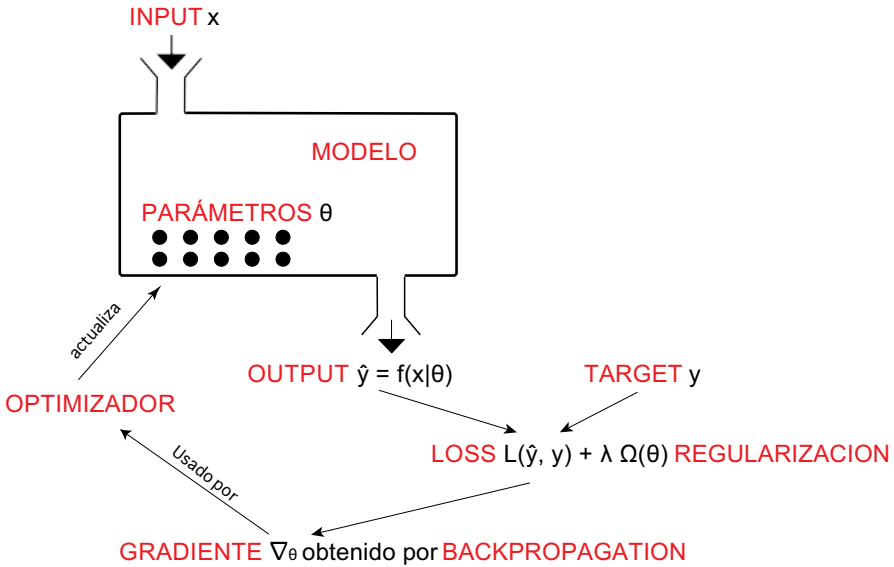
(Imágenes de los Transformers asignadas completamente al azar)

# Herramientas

- Clasificador rápido: https://fasttext.cc/
- Vectores pre-entrenados: https://fasttext.cc/docs/en/crawl-vectors.html
- Librerías para deep learning:
  - Pytorch https://pytorch.org
  - Chainer https://chainer.org/
  - TensorFlow http://tensorflow.org/
  - Dynet https://github.com/clab/dynet
- Librería para Seq2seq: https://github.com/IBM/pytorch-seq2seq
- Librería para transformers: https://github.com/huggingface/pytorch-transformers

# Repaso y concusiones

- Una red neuronal es una función con parámetros ajustables.
- Como producto del entrenamiento, la red aprende representaciones de la entrada que no fueron explícitamente programadas.
- Sin embargo, puedo dotar a la red de mecanismos que mejoran su capacidad de descubrir buenas representaciones.



# Repaso y concusiones

- Marco general de un problema de Machine Learning
  - Reconocer mis datos
  - Plantear un modelo (lo más simple posible)
  - Entender qué falta. Iterar.
- Consideraciones al trabajar con lenguaje natural
  - El lenguaje natural es complejo.
  - Podemos encontrar atajos para resolver muchos problemas prácticos.
- Consideraciones al trabajar con redes neuronales
  - Algunos modelos requieren enormes cantidades de cómputo (GPU servers, etc.)
  - Pero también, modelos simples pueden correr sin problemas en CPUs (p. ej. Fasttext)

# Desafíos y problemas abiertos

- Cantidad de datos.
- Composicionalidad sistemática.
- Reconciliación entre la práctica y la teoría lingüística.
- Eficiencia.
- Aprendizaje basado en la comunicación.