**LSTM**

**Capa EMBEDDING**

La capa de embedding asigna un vector de tamaño fijo a cada palabra en el vocabulario. Los parámetros de esta capa provienen de la matriz de embeddings de tamaño:

Tamaño del vocabulario× Dimensión del embedding \ text {Tamaño del vocabulario} \ times \ text {Dimensión del embedding} Tamaño del vocabulario× Dimensión del embedding

tamaño del vocabulario es 50,000 y la dimensión del embedding es 128:

50,000×128=6,400,00050,000 \times 128 = 6,400,00050,000×128=6,400,000

**Capa LSTM**

Un LSTM tiene cuatro puertas (forget, input, output y candidate cell) y cada una necesita pesos para:

* La entrada (W): conexión desde el embedding al LSTM.
* El estado recurrente (U): conexión de la salida del LSTM consigo mismo.
* El bias (b): valores de sesgo.

Para un LSTM con 64 unidades (n = 64), la ecuación general para los parámetros es:

Parametros=(Dim. entrada+Unidades LSTM)×Unidades LSTM×4+(Unidades LSTM×4)\text{Parámetros} = ( \text{Dim. entrada} + \text{Unidades LSTM} ) \times \text{Unidades LSTM} \times 4 + (\text{Unidades LSTM} \times 4)Paraˊmetros=(Dim. entrada+Unidades LSTM)×Unidades LSTM×4+(Unidades LSTM×4)

Sustituyendo:

* Dim. entrada = 128 (del embedding)
* Unidades LSTM = 64

(128+64)×64×4+(64×4)(128 + 64) \times 64 \times 4 + (64 \times 4)(128+64)×64×4+(64×4) 192×64×4+256=49,408192 \times 64 \times 4 + 256 = 49,408192×64×4+256=49,408

**Capa Densa**

Una capa densa con 4 neuronas recibe la salida del LSTM (64 unidades), por lo que el cálculo de los parámetros es:

Paraˊmetros=(Unidades LSTM×Unidades salida)+Bias\text{Parámetros} = (\text{Unidades LSTM} \times \text{Unidades salida}) + \text{Bias}Paraˊmetros=(Unidades LSTM×Unidades salida)+Bias (64×4)+4=256+4=260(64 \times 4) + 4 = 256 + 4 = 260(64×4)+4=256+4=260

| **Capa** | **Parámetros calculados** |
| --- | --- |
| Embedding | 6,400,000 |
| LSTM | 49,408 |
| Dense | 260 |
| **Total** | **6,449,668** |

**Modelo 1: Transformer Básico con una Capa de Atención Multi-Cabeza**

**Capa Embedding**

* **Output Shape:** (None, 100, 128)
* **Parámetros:** 50,000 × 128 = 6,400,000
* Explicación: La matriz de embeddings tiene 50,000 palabras y cada palabra tiene un vector de 128 dimensiones.

**Capa MultiHeadAttention**

* **Output Shape:** (None, 100, 128)
* **Parámetros:** 263,808

**Cálculo**: MultiHeadAttention tiene varios pesos, principalmente:

* 1. Proyección de la consulta, clave y valor (Q, K, V)
  2. Capa final de proyección después de la atención

Si d\_model = 128, num\_heads = 8, cada cabeza tiene d\_head = 128 / 8 = 16.

* 1. **Pesos de Q, K, V:**  
     Cada uno requiere d\_model × d\_model, y hay 3 matrices (W\_Q, W\_K, W\_V):

3×(128×128)=49,1523 × (128 × 128) = 49,1523×(128×128)=49,152

* 1. **Proyección final:**

128×128=16,384128 × 128 = 16,384128×128=16,384

* 1. **Bias:**  
     Bias para Q, K, V y salida:

(3×128)+128=512(3 × 128) + 128 = 512(3×128)+128=512

* 1. **Total:**

49,152+16,384+512=66,04849,152 + 16,384 + 512 = 66,04849,152+16,384+512=66,048

Y como hay 8 cabezas de atención:

66,048×4=263,80866,048 × 4 = 263,80866,048×4=263,808

**Capa LayerNormalization**

* **Output Shape:** (None, 100, 128)
* **Parámetros:** 256
* Explicación: LayerNorm tiene dos parámetros aprendibles (γ y β) por cada dimensión:

2×128=2562 × 128 = 2562×128=256

**Capa GlobalAveragePooling1D**

* **Output Shape:** (None, 128)
* **Parámetros:** 0
* Explicación: No tiene parámetros aprendibles; solo calcula el promedio sobre la secuencia.

**Capa Dense (128 unidades)**

* **Output Shape:** (None, 128)
* **Parámetros:** 16,512

**Cálculo**:

* + Pesos: 128 × 128 = 16,384
  + Bias: 128
  + Total: 16,384 + 128 = 16,512

**Capa Dense (4 unidades, salida)**

* **Output Shape:** (None, 4)
* **Parámetros:** 516

**Cálculo**:

* + Pesos: 128 × 4 = 512
  + Bias: 4
  + Total: 512 + 4 = 516

**📌 Resumen Final**

* **Total de parámetros entrenables:** 6,681,092
* **Parámetros no entrenables:** 0
* **Parámetros del optimizador:** 13,362,186 (estos dependen del algoritmo usado, como Adam, que almacena momentos para la actualización de pesos).

🔹 **Conclusión:** Este modelo usa Embedding + Multi-Head Attention + Pooling + Densas para clasificar en 4 clases. Es un modelo basado en Transformers para NLP.

**Transformer Avanzado con Varias Capas de Atención Multi-Cabeza**

**Capa Embedding**

* **Output Shape:** (None, 100, 128)
* **Parámetros:** 50,000 × 128 = 6,400,000
* **Explicación:** Asigna un vector de 128 dimensiones a cada palabra del vocabulario de 50,000 palabras.

**Capa MultiHeadAttention**

* **Output Shape:** (None, 100, 128)
* **Parámetros:** 263,808

**Cálculo:**

* + Se utilizan 8 cabezas (num\_heads = 8), cada una con un tamaño de 16 (128 / 8).
  + Cada una tiene matrices W\_Q, W\_K, W\_V (3 × d\_model × d\_model), una matriz de salida (d\_model × d\_model) y bias.

3×(128×128)+(128×128)+(3×128)+128=66,0483 × (128 × 128) + (128 × 128) + (3 × 128) + 128 = 66,0483×(128×128)+(128×128)+(3×128)+128=66,048

* + Como hay 8 cabezas:

66,048×4=263,80866,048 × 4 = 263,80866,048×4=263,808

**Capa LayerNormalization**

* **Output Shape:** (None, 100, 128)
* **Parámetros:** 256
* **Explicación:** Tiene dos parámetros (γ y β) por cada dimensión.

2×128=2562 × 128 = 2562×128=256

**Capa Reshape**

* **Output Shape:** (None, 100, 128)
* **Parámetros:** 0
* **Explicación:** Solo reorganiza la forma de los datos sin agregar parámetros.

**Capa GlobalAveragePooling1D**

* **Output Shape:** (None, 128)
* **Parámetros:** 0
* **Explicación:** Calcula el promedio de cada vector a lo largo de la dimensión de secuencia.

**Capa Dense (64 unidades)**

* **Output Shape:** (None, 64)
* **Parámetros:** 8,256

**Cálculo:**

* + Pesos: 128 × 64 = 8,192
  + Bias: 64
  + Total: 8,192 + 64 = 8,256

**Capa Dense (4 unidades, salida)**

* **Output Shape:** (None, 4)
* **Parámetros:** 260

**Cálculo:**

* + Pesos: 64 × 4 = 256
  + Bias: 4
  + Total: 256 + 4 = 260

**📌 Resumen Final**

* **Total de parámetros entrenables:** 6,672,580
* **Parámetros no entrenables:** 0
* **Conclusión:** Este modelo usa Embedding + Multi-Head Attention + Pooling + Densas con una capa oculta de 64 neuronas en vez de 128, reduciendo el número de parámetros en comparación con el modelo anterior.