Redes Neuronales Recurrentes (RNN)

Saturday, August 19, 2023 7:25 PM

["Hola", "a", "todos"] → vectorizarlo → clasificador

"Hola", "como", "estas?"

→ t_0 → t_1 → t_2 (timesteps)

lenguaje → característica TEMPORAL

→ TF-IDF → conteo

(+) Word Embeddings → ventanas

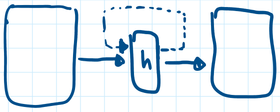
(+) CNN → Filtros

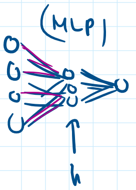
"Hola", "como", "estas"

TEMPORAL ← lenguaje → variable (longitud)

RECURRENT NEURAL NETWORKS (RNN)

RECURSIVO (REPITE)

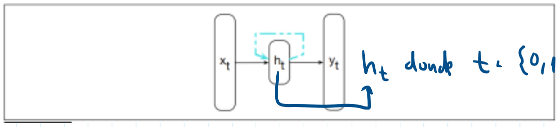




h_1 → (Entrada, h_0)

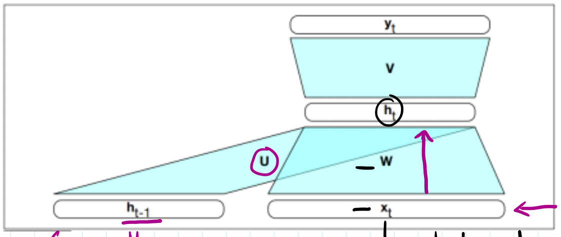
h_2 → (Entrada, h_1)

donde {0,1,2...,n} (timesteps)



h_t donde t: {0,1,2,3,4.. n)

Elman Network (Simple RNN)



NN común: W, V

RNN: W, U, V ↑ nuevo

↳ entrada = din = 5 (eg)

(Capa oculta en un tiempo anterior)

$W \in \mathbb{R}^{dh \times din}$

$U \in \mathbb{R}^{dh \times dh}$

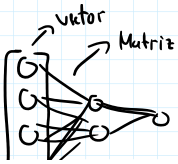
$V \in \mathbb{R}^{dout \times dh}$

dh → dimensión de la capa oculta (h_t)

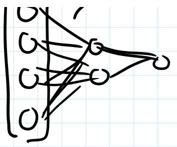
din → dimensión de entrada (x_t)

dout → dimensión de salida (y_t)

x_t, h_t y y_t son vectores



x_t, h_t y y_t son vectores

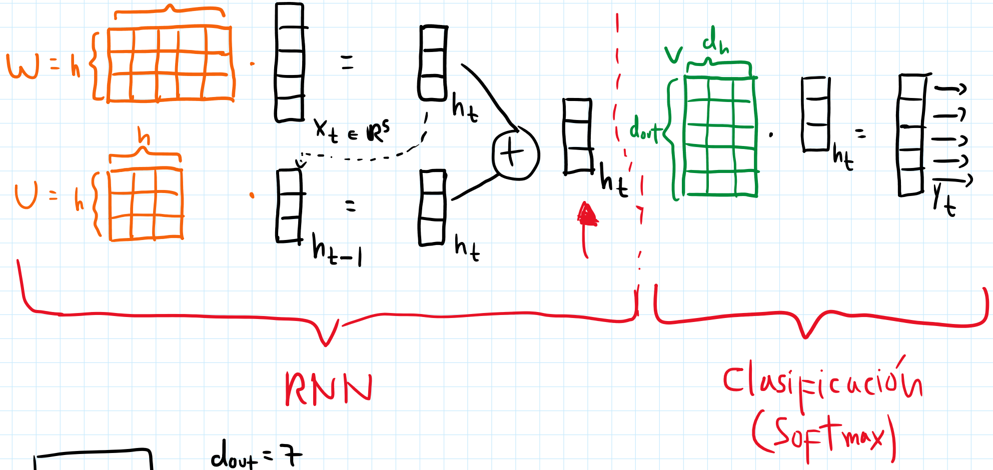


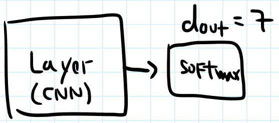
$d_{in} \in \mathbb{R}^5$ e.g $[0\ 1\ 1\ 0\ 1]$

$d_h \in \mathbb{R}^3 \longrightarrow$ (especificar)

$d_{out} \in \mathbb{R}^5$ (5 clases)

tensorflow "units" (d_h)



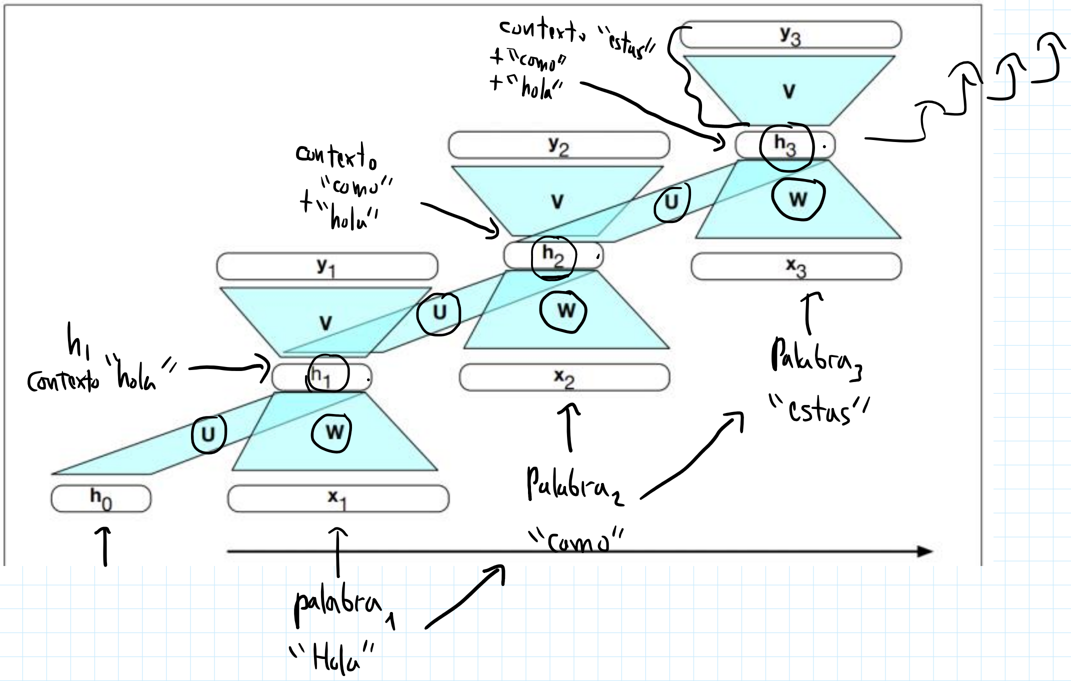


d_{out} => cantidad de clases a predecir (e.g binaria, $d_{out} = 2$
multi-clase (7) $d_{out} = 7$

$$\mathbf{h}_t = g(\mathbf{U}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{x}_t)$$
$$\mathbf{y}_t = f(\mathbf{V}\underline{\mathbf{h}_t})$$

g → función de activación

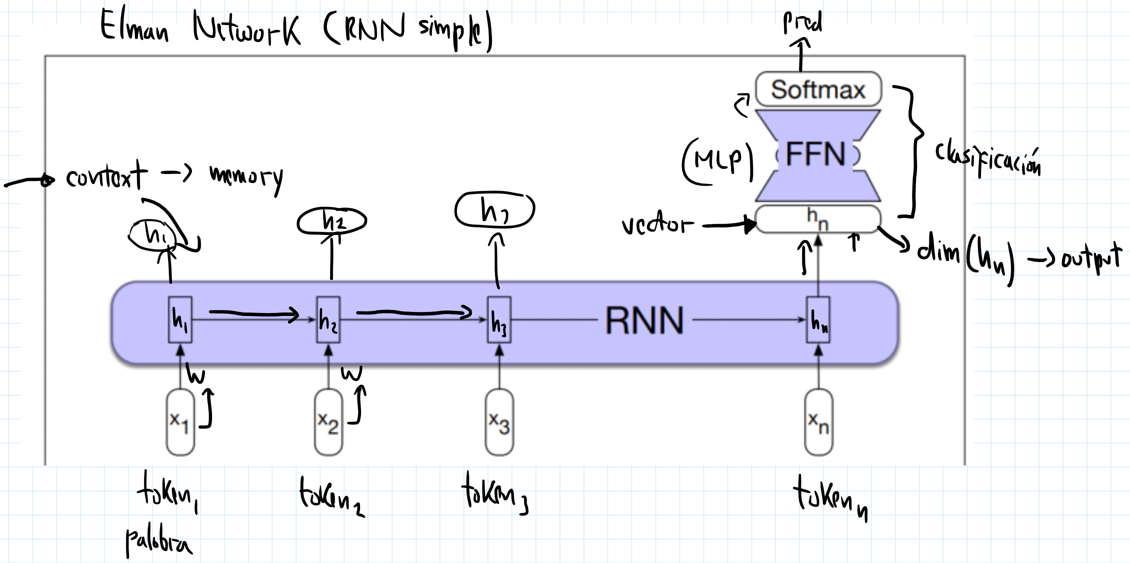
f → softmax



Elman Network (RNN simple)

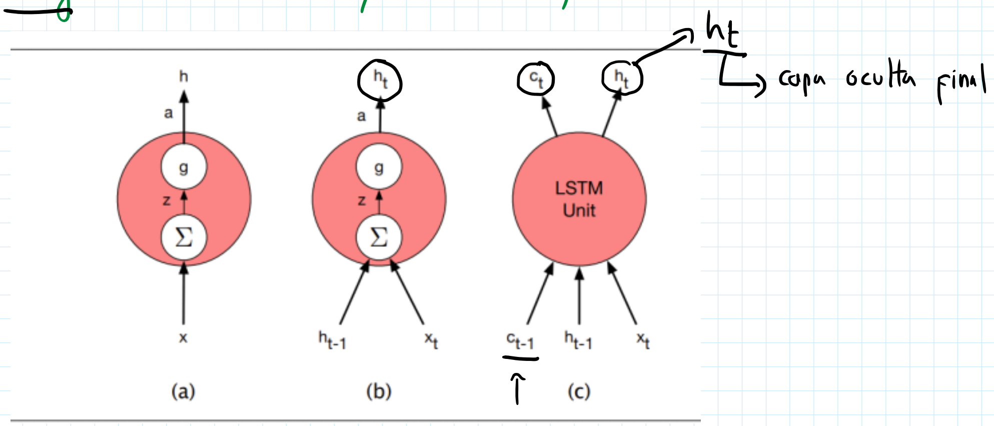
Pred
↑

Elman Network (RNN simple)

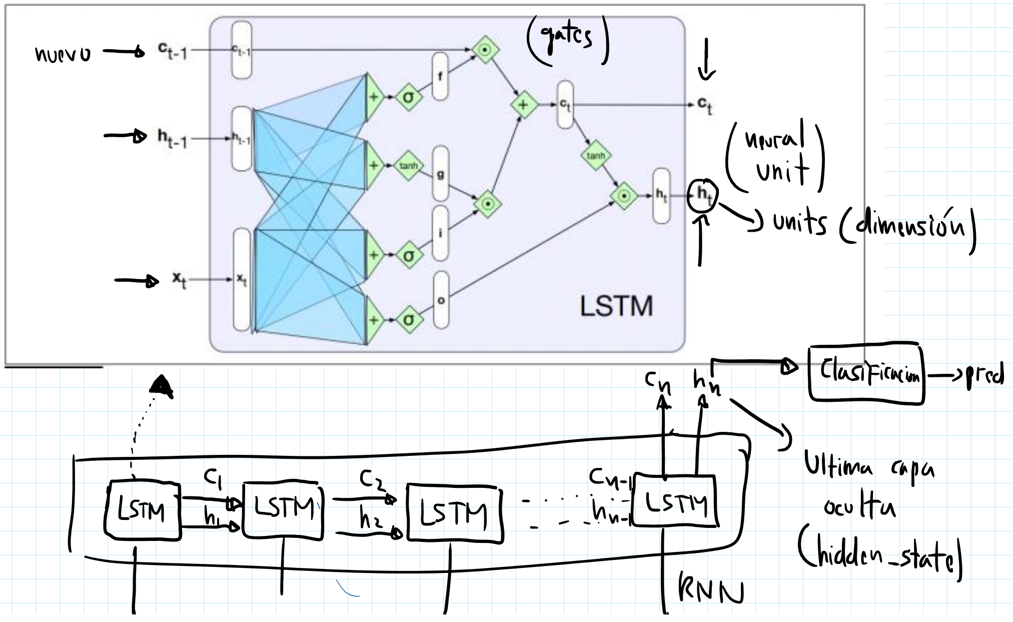


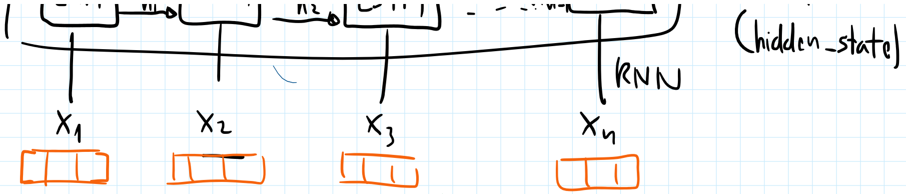
donde $\{x_1 \dots x_n\} \rightarrow$ documento

Long-short Term Memory (LSTM)



(a) MLP (b) clasica RNN (c) LSTM





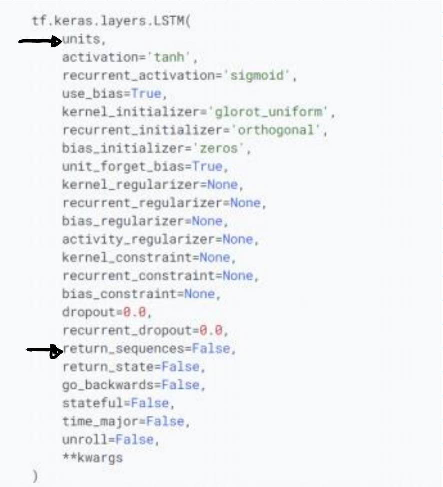
Gated-Recurrent Unit (GRU)

LSTM v. GRU

↓ + paramatras > ↓ − parametros

GRU > veloz entrenamiento

tensorflow:



```
tf.keras.layers.LSTM(
    units,
    activation='tanh',
    recurrent_activation='sigmoid',
    use_bias=True,
    kernel_initializer='glorot_uniform',
    recurrent_initializer='orthogonal',
    bias_initializer='zeros',
    unit_forget_bias=True,
    kernel_regularizer=None,
    recurrent_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    recurrent_constraint=None,
    bias_constraint=None,
    dropout=0.0,
    recurrent_dropout=0.0,
    return_sequences=False,
    return_state=False,
    go_backwards=False,
    stateful=False,
    time_major=False,
    unroll=False,
    **kwargs
)
```

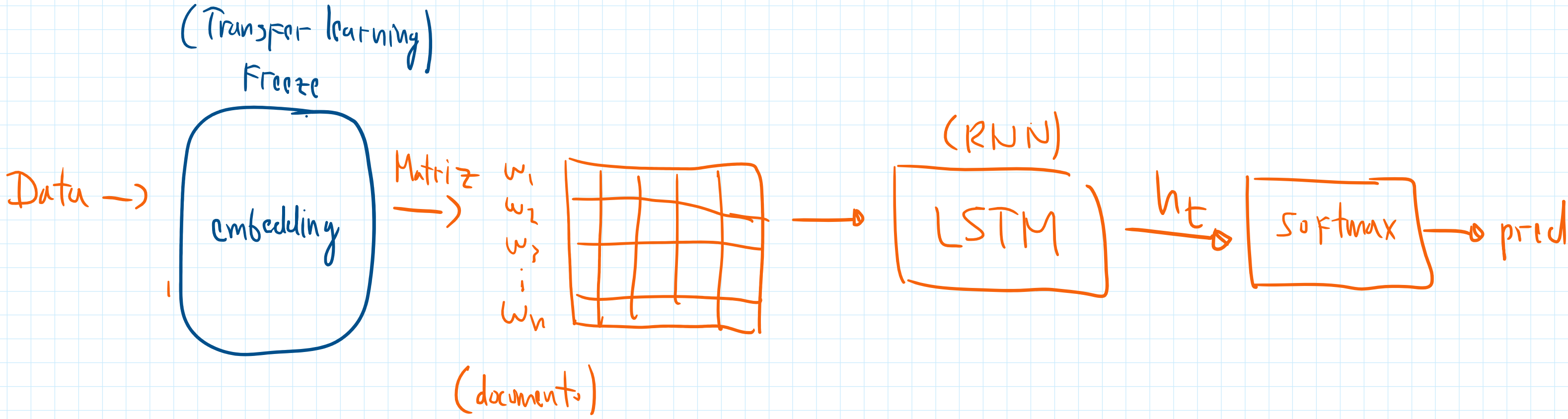
units: dimensión de capa oculta h_t donde $t:\{0,1,2...n\}$

(e.g. units=32 → vector de 32 dimensiones)

Data → $\boxed{\text{LSTM}}$ → Vector de "units" dimensiones (32)

units $\boxed{\text{NO ES}}$ el número de neural units de LSTM

return_sequences:
- False → última capa oculta (h_n) → vector
- True → todas las capas ocultas ($h_{0...n}$) → matriz

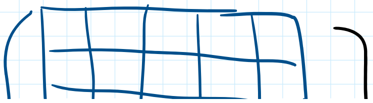


corpus → documento_{56} → 120 palabras

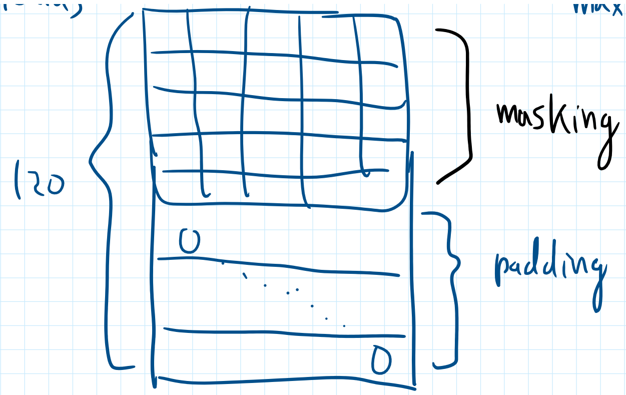
embedding_matrix = (120, 300)

↑ max ↑ preentrenado

documento_5 → 5 palabras



[illegible]


120
0
0
masking
padding