

TP7

Tomás Hüttenbräucker
Aprendizaje Profundo y Redes neuronales artificiales
(11 de noviembre de 2020)

1. MANEJO DE DATOS

Usando la base de datos `airline-passengers` que muestra la cantidad de pasajeros de una aerolínea en función de los 144 meses de 12 años se creó el set de datos X y la predicción Y . Llamando a la cantidad de pasajeros en función del mes $x[n]$, el set de datos es:

$$X[k] = (x[k], x[k+1], x[k+2], \dots, x[k+l-1])$$

$$Y[k] = x[k+l]$$

con la idea de que el modelo, usando $X[k]$, pueda predecir el valor $Y[k]$. El parámetro l es el parámetro de lag, que define la cantidad de valores anteriores usados para predecir. Una vez definidos los datos, se normalizaron los valores entre 0 y 1 y se le sumó a la cada muestra de X ruido gaussiano de varianza 0.02 y media 0 y se dividieron en train y test.

2. MODELO CON CAPA LSTM.

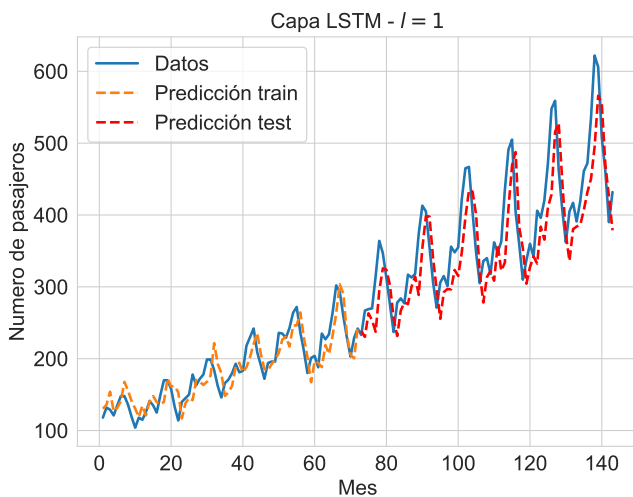


Figura 1: Predicción del modelo con capa LSTM.

Se creó un modelo formado por una capa tipo LSTM de 4 neuronas y una capa densa de salida de 1 neurona. Como función de costo se usó MSE. A continuación se muestra la arquitectura del modelo usado:

Model: "sequential_1"

Layer (type)	Output Shape
LSTM_1 (LSTM)	(None, 4)
dense_1 (Dense)	(None, 1)

```
=====
Total params: 101
Trainable params: 101
Non-trainable params: 0
-----
```

En la figura 2 se muestra la evolución del modelo en el entrenamiento, evaluando para los datos de test y train. Se ve que el modelo converge bien tanto para los datos de test como para train. En la figura 1 se muestra la predicción del modelo sobre los datos de test y de train, comparado con el valor real de los datos. Se ve que el modelo hace un buen trabajo al predecir.

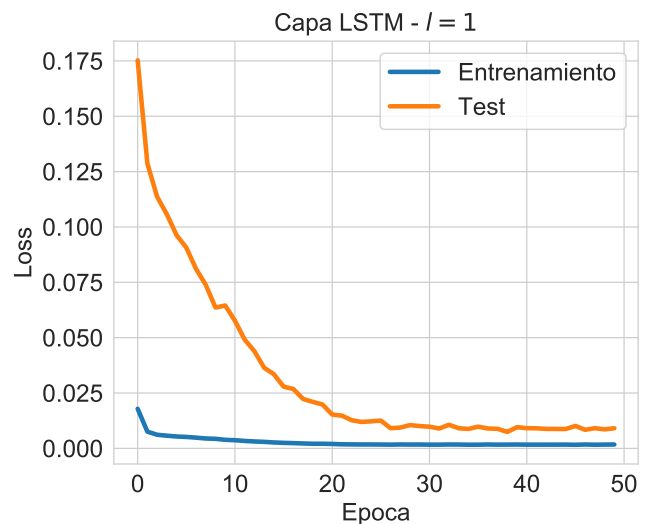


Figura 2: Costo del modelo con capa LSTM en función de la época de entrenamiento.

3. DEPENDENCIA CON EL PARÁMETRO DE LAG

Se buscó definir la dependencia de los resultados obtenidos por el modelo con el parámetro de lag l . Para esto, se realizó un barrido para valores de l entre 1 y 24, promediando en cada caso 20 veces para obtener resultados más precisos.

En las figuras 3 y 4 se muestra el costo final sobre los datos en función del parámetro de lag l luego de promediar 20 veces, no parece haber una dependencia clara.



Figura 3: Loss sobre datos de test en función del lag l

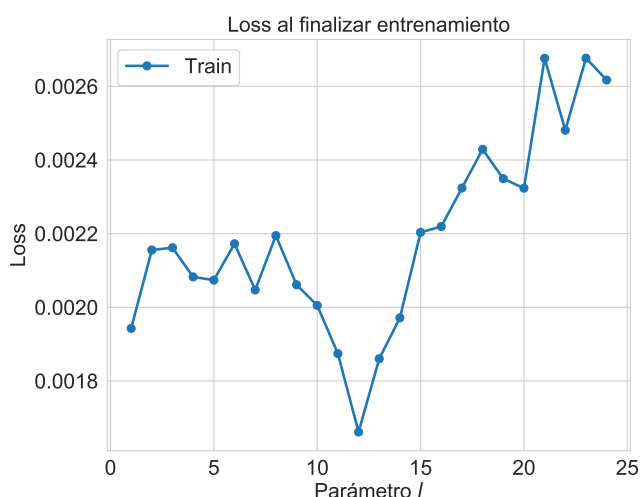


Figura 4: Loss sobre datos de train en función del lag l

4. RED CON CAPA DENSA

Para comparar con el resultado obtenido usando LSTM, se repitió la misma arquitectura para la red, pero usando ahora una capa densa de 4 neuronas en vez de la capa LSTM. La arquitectura obtenida fue:

Modelo con capas densas
Model: "sequential_2"

Layer (type)	Output Shape
dense_1 (Dense)	(None, 1, 4)
output (Dense)	(None, 1, 1)

```
=====
Total params: 13
Trainable params: 13
Non-trainable params: 0
=====
```

En la figura 5 se muestra la función de costo del modelo sobre los datos de entrenamiento y testing en función de las épocas. Parece tener un comportamiento similar al obtenido con la capa LSTM, con una los para los datos de test mayor. En la figura 6 se muestra la predicción del modelo, no parece muy diferente de los resultados obtenidos con la capa LSTM.

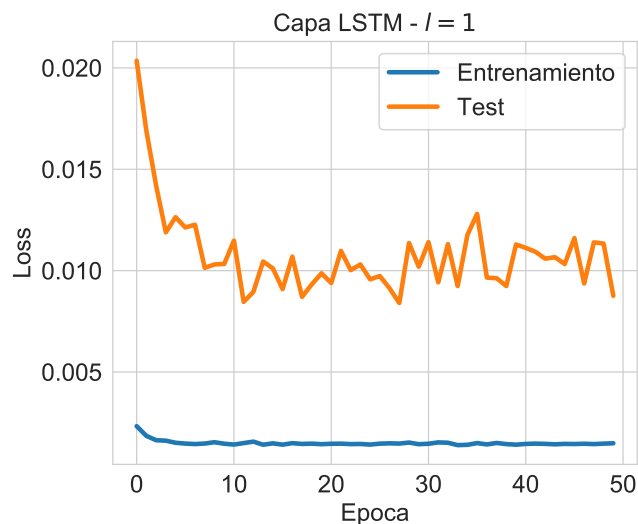


Figura 5: Costo en función de las épocas para el modelo con capa densa.

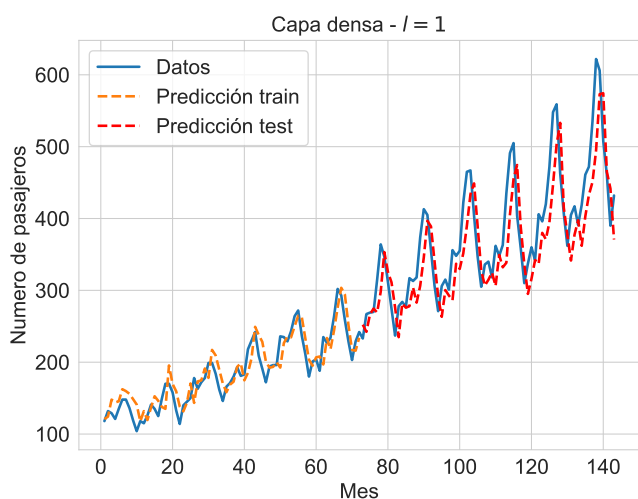


Figura 6: Predicción realizada por el modelo con capa densa.