Universidad de sonora

Seminario de física computacional

Tercera evaluación:

Redes neuronales recurrentes

Minjares Neriz Victor Manuel

Mayo 2021

Índice

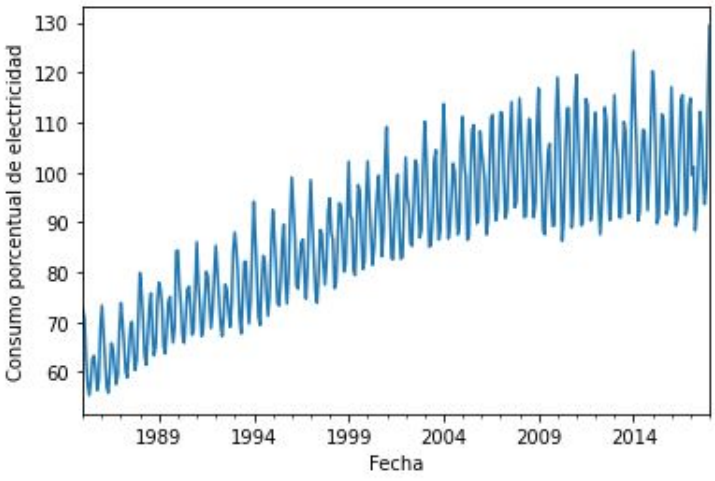
1. Introducción

En este último proyecto del seminario practicaremos como crear y configurar las redes neuronales recurrentes (RNN). La diferencia entre estas redes y las que hemos visto anteriormente es que el tiempo es la variable independiente en estas redes, a diferencia de las otras que podia ser cualquier parámetro o variable. Particularmente en este trabajo entrenaremos nuestra red neuronal para que prediga el consumo porcentual de electricidad que tiene cierta industria, usando datos de su consumo entre los años 1985 y 2018.El dataset cuenta con 397 datos y dos columnas, la primera con la fecha y la segunda con el consumo. Dataset obtenido de kaggle:
https://www.kaggle.com/kandij/electric-production.

Dataset:

	DATE	Value
0	1985-01-01	72.5052
1	1985-02-01	70.6720
2	1985-03-01	62.4502
3	1985-04-01	57.4714
4	1985-05-01	55.3151
...	...	...
392	2017-09-01	98.6154
393	2017-10-01	93.6137
394	2017-11-01	97.3359
395	2017-12-01	114.7212
396	2018-01-01	129.4048

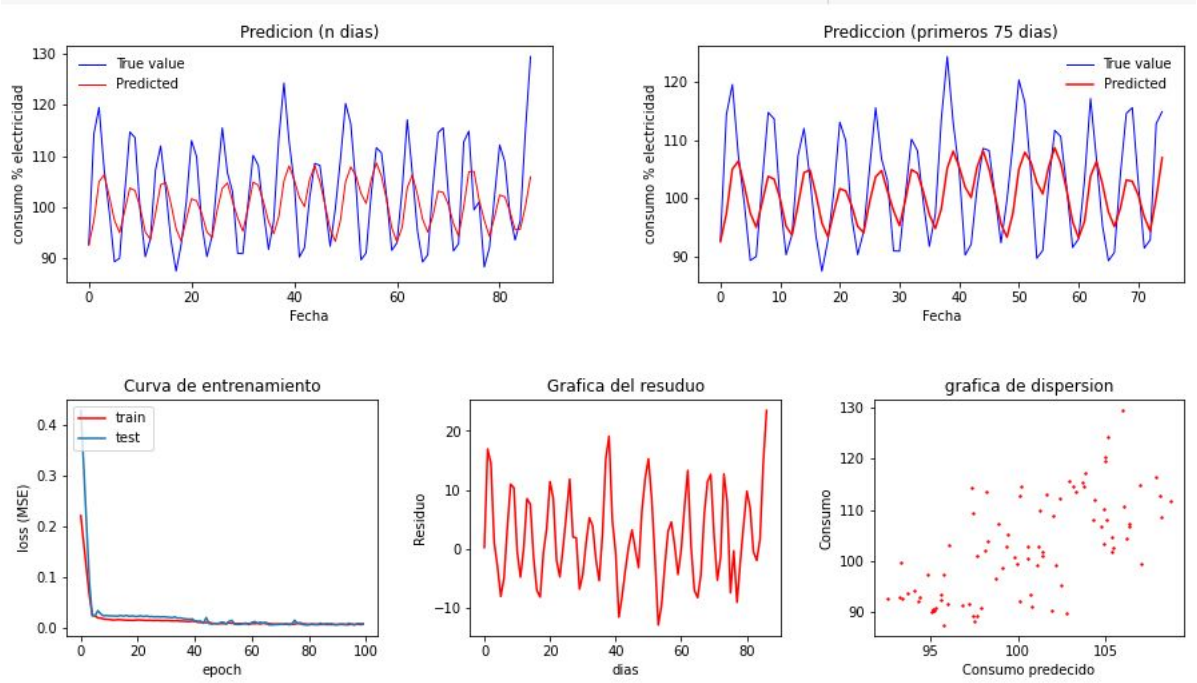
Gráfica de los datos:



2. Resultados

2.1. Configuración incial

- Tipo de RNN: LSTM normal
- Dropout: no
- Optimizer: adam
- Activation function: relu
- Batch size: 32
- Epochs: 100



Error cuadratico medio: 61.7

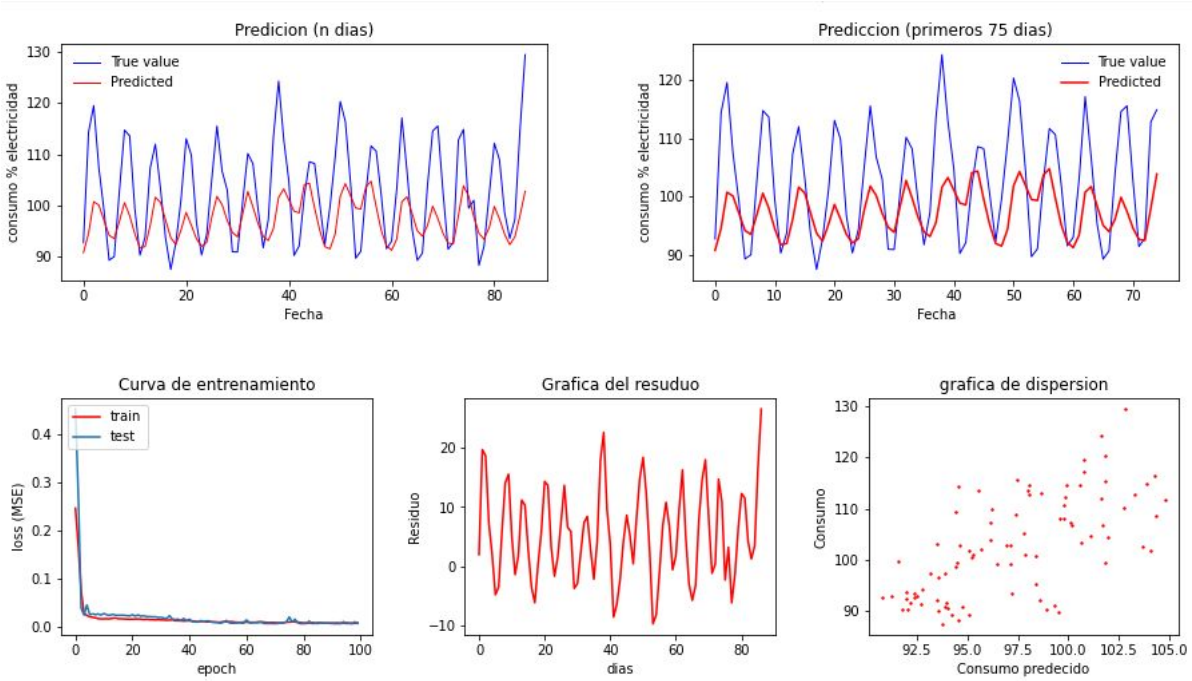
R^2: 0.35

Podemos ver en la grafica de entrenamiento que no hay overfitting ya que las curvas del loos test y train entre más iteraciones más se juntan por lo tanto no es necesario aplicar "dropout". Las variación se aplicaran individualmente, es decir sera la configuración inicial + variación #

2.2. Variación 1

Cambio de tipo normal de la RNN

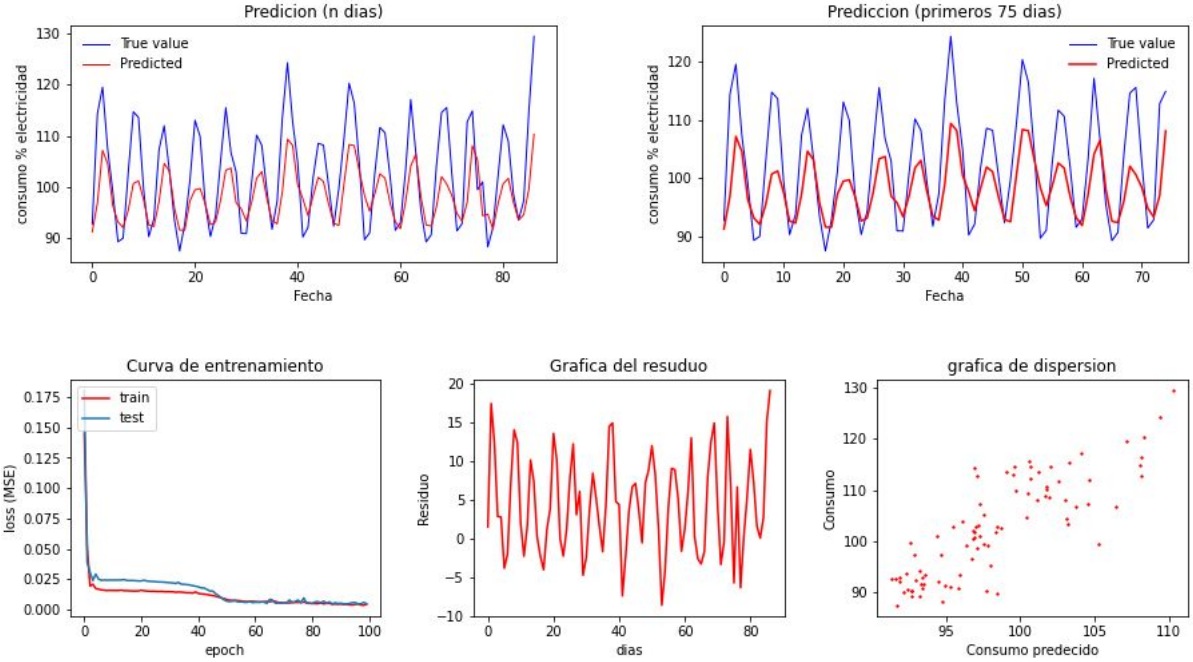
2.2.1. Stacked



Error cuadratico medio: 89.27

R^2: 0.05

2.2.2. Bidireccional

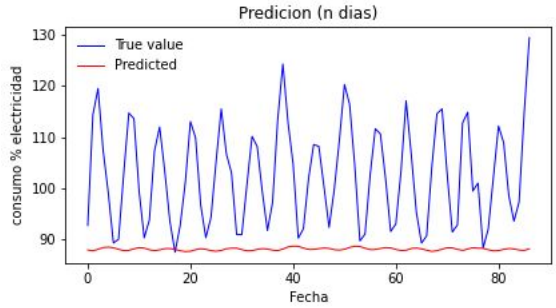


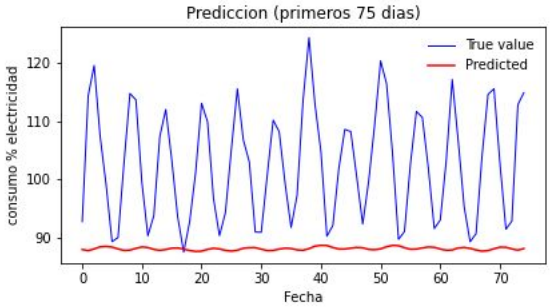
Error cuadratico medio: 57.79

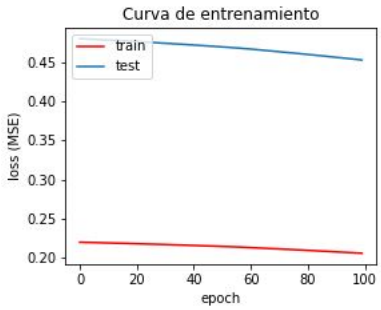
R^2: 0.39

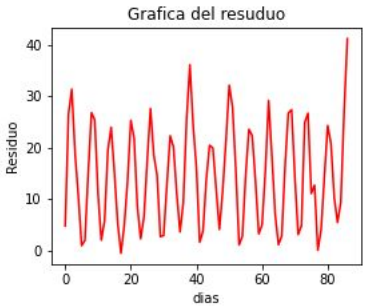
2.3. Variación 2

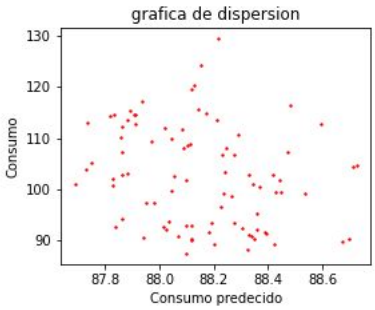
Optimizer utilizado **Adadelta**









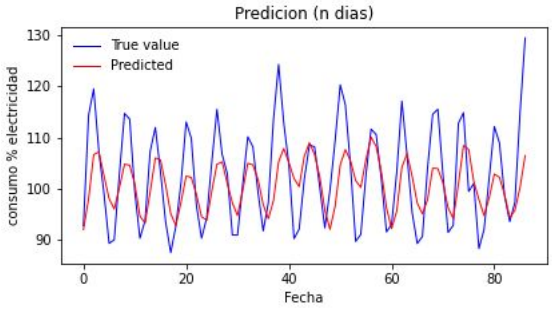


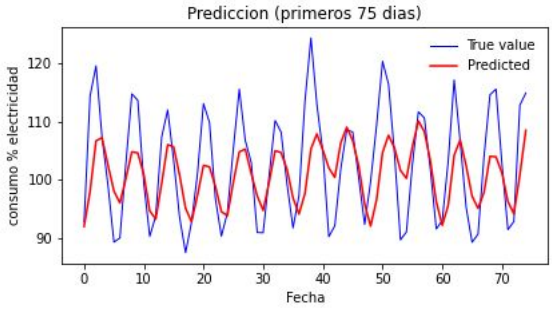
Error cuadratico medio: 305.63

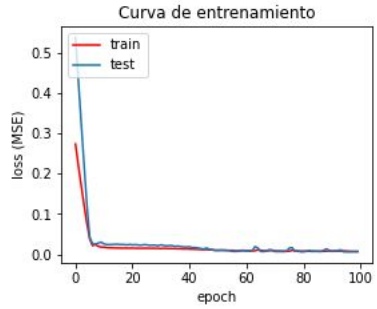
R^2: 2.24

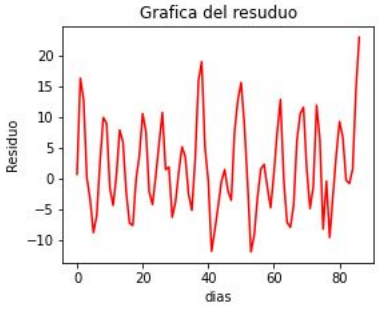
2.4. Variación 3

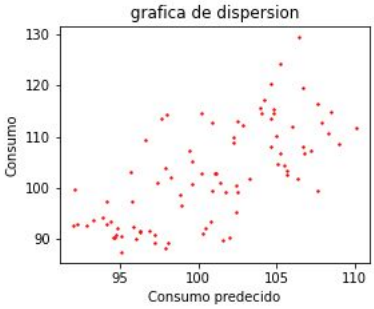
Batch size utilizado **50**









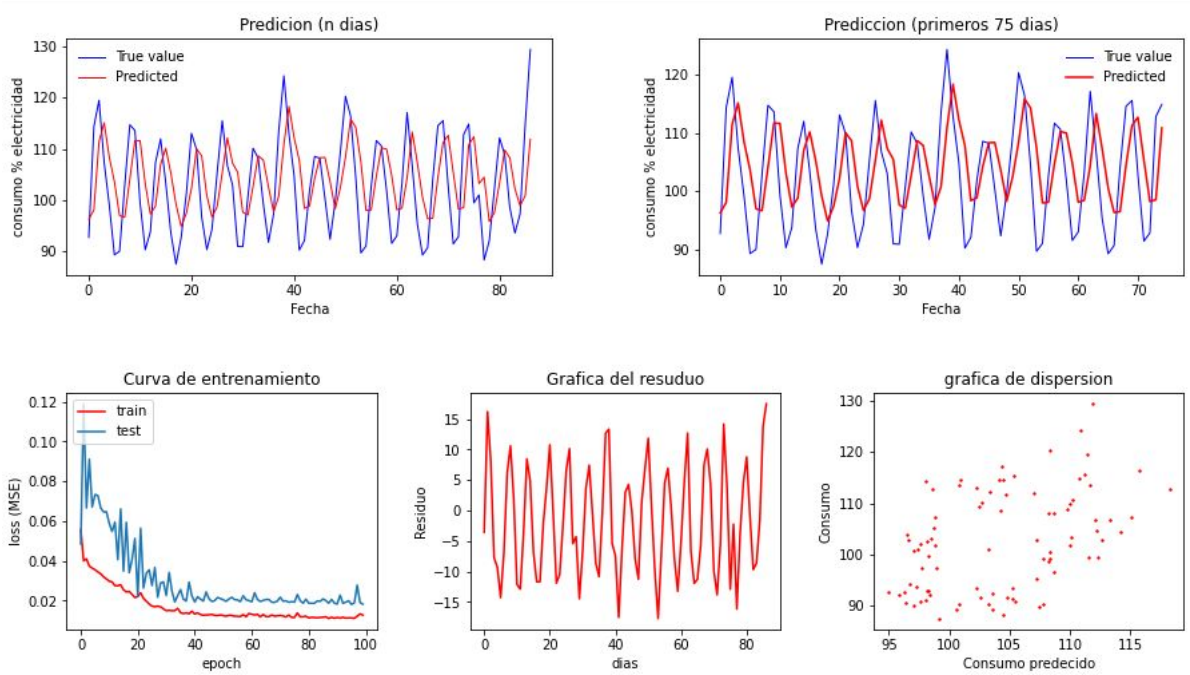


Error cuadratico medio: 58.06

R^2: 0.38

2.5. Variación 4

Función de activación utilizada **sigmoid**



Error cuadratico medio: 88.22

R^2: 0.13

3. Conclusiones

Las corridas del programa las efectue en googlecolab, ya que el tamaño del dataset que utilice era pequeño por lo que no tuve que utilizar el gpu de la escuela. Probe la configuración inicial dos 4 diferentes epochs, 25, 50, 100 y 200. De las primeras 3 la mejor predicción fue la de 100 y la de 200 fue un poco mejor que la de 100 pero presentaba overfitting por lo que me decidi por quedarme con 100 epochs.

Tanto en la configuración incial como en las variación tenemos error cuadratico medio y R^2 muy malos lo cual es comprensible si vemos las graficas de las prediciones (curvas rojas) con las reales (azules), pero en este tipo de redes neuronales minimizar en el caso del error cuadratico medio o que sea $R^2 = 1$ no es importante, lo importante es la tendencia, o cambio, sean lo más parecida posible entre la predicción con los originales, lo cual en la mayoria de nuestras configuraciones ocurre.

La peor variación entrenada es la 2, optimizer = Adadelta. Totalmente inservible para las redes neuronales tipo RNN. Las demás gráficas tienen un desempeño bueno, tienen loss bajo tanto en los datos de entrenamiento como los de test, y sus tendencias son muy parecidas a las originales, solo una pequeña fase de diferencia. Entre las configuraciones la que más destaca es la variación 1, bidireccional. Tiene error practicamente 0, sus tendencias son las más parecidas a las originales, además su gráfica de residuo tiene los valores más pequeños por lo que esta predicción es la que tiene magnitud más parecida a la original. También en su gráfica de dispersión es la que presenta una tendencia lineal más marcada entre los datos predichos y los reales. Por lo tanto para estos datos esta variación es la mejor de las probadas en este trabajo.

Link del programa:
https://colab.research.google.com/drive/1ZFRWQ75LWOZseCuByBTI2l67btsHlOdq#scrollTo=ddFR0CTVTu7K