Analog to Digital Conversion

Submitted By:
Vaibhav Thakur (15MI412)
Komal Mehra (15MI413)

An analog-to-digital converter, or ADC as it is more commonly called, is a device that converts analog signals into digital signals. Analog information is transmitted by modulating a continuous transmission signal by amplifying a signal's strength or varying its frequency to add or take away data. Digital information describes any system based on discontinuous data or events. Computers, which handle data in digital form, require analog-to-digital converters to turn signals from analog to digital before it can be read. One example is a modem which turns signals from digital to analog before transmitting those signals over communication lines such as telephone lines that carry only analog signals. The signals are turned back into digital form (demodulated) at the receiving end so that the computer can process the data in its digital format.

Methodology:

A dataset has been generated in the .ipynb file using Python Language. Inputs are taken as X(analog) and Output values are given as Y(digital).
The model used is Sequential. Layers like LSTM and Dense have been added in the model and then the model has been compiled using the Optimizer- adam and Loss Function as Mean Square Error (MSE).

Data Generation and Collection:

Dataset has been generated using the code:

```
X= [[[0+j] for j in np.arange(i-1,i,0.1)] for i in np.arange(1,32)]
print(*X, sep="\n")

Y= [(0+i) for i in range(1,32)]
print (Y)
```

Here, X denotes the input analog values and Y denotes output values.

The code is as follows:

```
X= [[[0+j] for j in np.arange(i-1,i,0.1)] for i in np.arange(1,32)]
print(*X, sep="\n")

Y= [(0+i) for i in range(1,32)]
print (Y)
```

The dataset generated is as follows:

```
[[0.0], [0.1], [0.2], [0.30000000000000004], [0.4], [0.5], [0.6000000000000001], [0.7000000000000001], [0.8], [0.9]]
[[1.0], [1.1], [1.2000000000000002], [1.3000000000000003], [1.4000000000000004], [1.5000000000000004], [1.6000000000000005], [1.7000000000000006], [1.8000000000000007], [1.90000000
[[2.0], [2.1], [2.2], [2.3000000000000003], [2.4000000000000004], [2.5000000000000004], [2.6000000000000005], [2.7000000000000006], [2.8000000000000007], [2.900000000000001]]
[[3.0], [3.1], [3.2], [3.3000000000000003], [3.4000000000000004], [3.5000000000000004], [3.6000000000000005], [3.7000000000000006], [3.8000000000000007], [3.900000000000001]]
[[4.0], [4.1], [4.199999999999999], [4.299999999999999], [4.399999999999999], [4.499999999999998], [4.599999999999998], [4.6999999999999975], [4.799999999999997], [4.89999999999999
[[5.0], [5.1], [5.199999999999999], [5.299999999999999], [5.399999999999999], [5.499999999999998], [5.599999999999998], [5.6999999999999975], [5.799999999999997], [5.89999999999999
[[6.0], [6.1], [6.199999999999999], [6.299999999999999], [6.399999999999999], [6.499999999999998], [6.599999999999998], [6.6999999999999975], [6.799999999999997], [6.89999999999999
[[7.0], [7.1], [7.199999999999999], [7.299999999999999], [7.399999999999999], [7.499999999999998], [7.599999999999998], [7.6999999999999975], [7.799999999999997], [7.89999999999999
[[8.0], [8.1], [8.2], [8.299999999999999], [8.399999999999999], [8.499999999999998], [8.599999999999998], [8.699999999999998], [8.799999999999997], [8.899999999999997]]
[[9.0], [9.1], [9.2], [9.299999999999999], [9.399999999999999], [9.499999999999998], [9.599999999999998], [9.699999999999998], [9.799999999999997], [9.899999999999997]]
[[10.0], [10.1], [10.2], [10.299999999999999], [10.399999999999999], [10.499999999999998], [10.599999999999998], [10.699999999999998], [10.799999999999997], [10.899999999999997]]
[[11.0], [11.1], [11.2], [11.299999999999999], [11.399999999999999], [11.499999999999998], [11.599999999999998], [11.699999999999998], [11.799999999999997], [11.899999999999997]]
[[12.0], [12.1], [12.2], [12.299999999999999], [12.399999999999999], [12.499999999999998], [12.599999999999998], [12.699999999999998], [12.799999999999997], [12.899999999999997]]
[[13.0], [13.1], [13.2], [13.299999999999999], [13.399999999999999], [13.499999999999998], [13.599999999999998], [13.699999999999998], [13.799999999999997], [13.899999999999997]]
[[14.0], [14.1], [14.2], [14.299999999999999], [14.399999999999999], [14.499999999999998], [14.599999999999998], [14.699999999999998], [14.799999999999997], [14.899999999999997]]
[[15.0], [15.1], [15.2], [15.299999999999999], [15.399999999999999], [15.499999999999998], [15.599999999999998], [15.699999999999998], [15.799999999999997], [15.899999999999997]]
[[16.0], [16.1], [16.200000000000003], [16.300000000000004], [16.400000000000006], [16.500000000000007], [16.60000000000001], [16.70000000000001], [16.80000000000001], [16.90000000
[[17.0], [17.1], [17.200000000000003], [17.300000000000004], [17.400000000000006], [17.500000000000007], [17.60000000000001], [17.70000000000001], [17.80000000000001], [17.90000000
[[18.0], [18.1], [18.200000000000003], [18.300000000000004], [18.400000000000006], [18.500000000000007], [18.60000000000001], [18.70000000000001], [18.80000000000001], [18.90000000
[[19.0], [19.1], [19.200000000000003], [19.300000000000004], [19.400000000000006], [19.500000000000007], [19.60000000000001], [19.70000000000001], [19.80000000000001], [19.90000000
[[20.0], [20.1], [20.200000000000003], [20.300000000000004], [20.400000000000006], [20.500000000000007], [20.60000000000001], [20.70000000000001], [20.80000000000001], [20.90000000
[[21.0], [21.1], [21.200000000000003], [21.300000000000004], [21.400000000000006], [21.500000000000007], [21.60000000000001], [21.70000000000001], [21.80000000000001], [21.90000000
[[22.0], [22.1], [22.200000000000003], [22.300000000000004], [22.400000000000006], [22.500000000000007], [22.60000000000001], [22.70000000000001], [22.80000000000001], [22.90000000
[[23.0], [23.1], [23.200000000000003], [23.300000000000004], [23.400000000000006], [23.500000000000007], [23.60000000000001], [23.70000000000001], [23.80000000000001], [23.90000000
[[24.0], [24.1], [24.200000000000003], [24.300000000000004], [24.400000000000006], [24.500000000000007], [24.60000000000001], [24.70000000000001], [24.80000000000001], [24.90000000
[[25.0], [25.1], [25.200000000000003], [25.300000000000004], [25.400000000000006], [25.500000000000007], [25.60000000000001], [25.70000000000001], [25.80000000000001], [25.90000000
[[26.0], [26.1], [26.200000000000003], [26.300000000000004], [26.400000000000006], [26.500000000000007], [26.60000000000001], [26.70000000000001], [26.80000000000001], [26.90000000
[[27.0], [27.1], [27.200000000000003], [27.300000000000004], [27.400000000000006], [27.500000000000007], [27.60000000000001], [27.70000000000001], [27.80000000000001], [27.90000000
[[28.0], [28.1], [28.200000000000003], [28.300000000000004], [28.400000000000006], [28.500000000000007], [28.60000000000001], [28.70000000000001], [28.80000000000001], [28.90000000
[[29.0], [29.1], [29.200000000000003], [29.300000000000004], [29.400000000000006], [29.500000000000007], [29.60000000000001], [29.70000000000001], [29.80000000000001], [29.90000000
[[30.0], [30.1], [30.200000000000003], [30.300000000000004], [30.400000000000006], [30.500000000000007], [30.60000000000001], [30.70000000000001], [30.80000000000001], [30.90000000
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]
```

Link to code :
https://colab.research.google.com/drive/1BxkubvRwMrVcyCjD1ggP8Ol0iNGsIbq_#scrollTo=L9Nzb0O9AyDY

Layers:

The layers used were LSTM and Dense.

1. Long short-term memory is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data.
2. A **dense layer** is just a regular **layer** of neurons in a neural network. Each neuron receives input from all the neurons in the previous **layer**, thus

densely connected. The **layer** has a weight matrix W, a bias vector b, and the activations of previous **layer** a.

Optimizer = 'adam'
Adam is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum.

Activation = ReLU
In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input.
The rectified linear activation function is a piecewise linear function that will output the input directly if is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

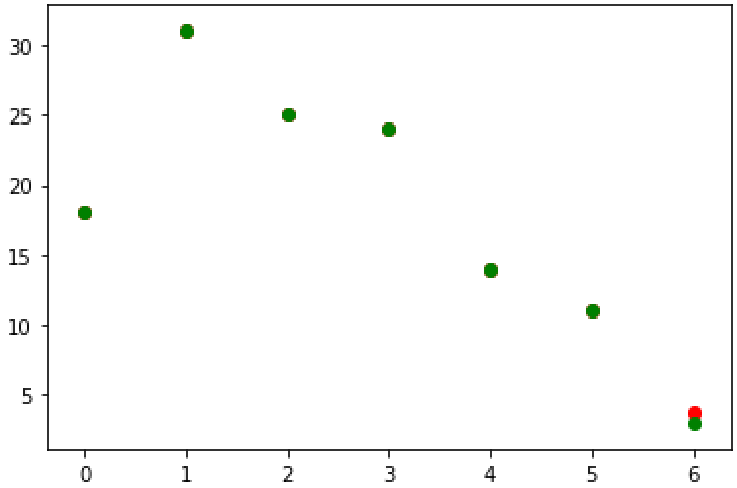
Loss = 'mse'
MSE is the average of the squared error that is used as the loss function for least squares regression:

$$\boxed{\sum_{i=1}^{n} \frac{\left(w^T x(i) - y(i)\right)^2}{n}}$$

It is the sum, over all the data points, of the square of the difference between the predicted and actual target variables, divided by the number of data points. RMSE is the square root of MSE. MSE is measured in units that are the square of the target variable, while RMSE is measured in the same units as the target variable. Due to its formulation, MSE, just like the squared loss function that it derives from, effectively penalizes larger errors more severely.

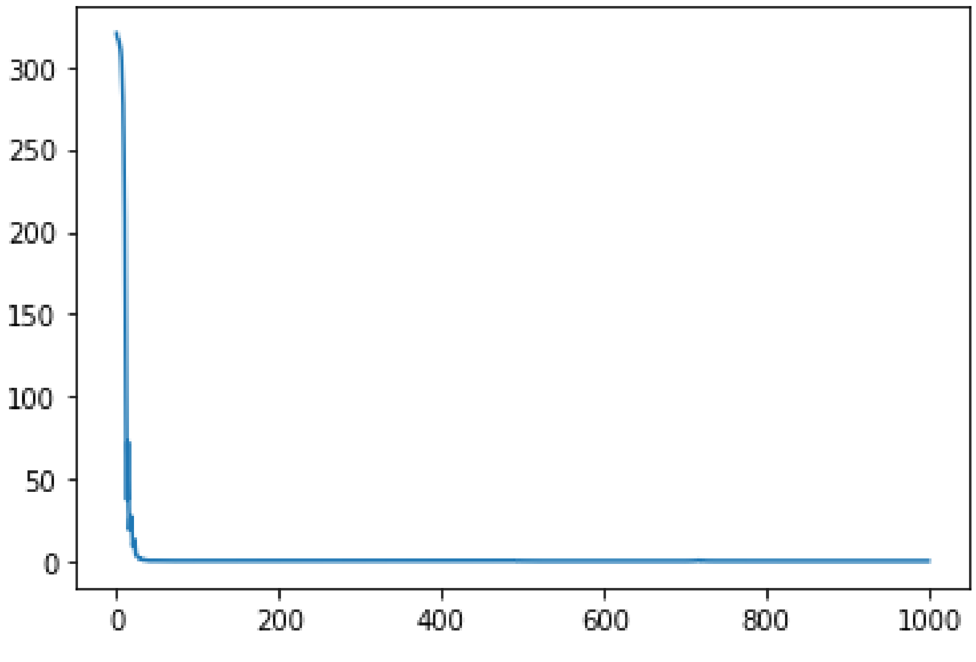
Results:

1.



30
25
20
15
10
5
0
1
2
3
4
5
6

2.



300
250
200
150
100
50
0
0
200
400
600
800
1000