

AD18511 – DEEP LEARNING LABORATORY

DATE:

EX.NO: 13

AUTOREGRESSIVE(SEQUENCE TO SEQUENCE).

AIM:

To implement the Autoregressive (Sequence to Sequence)model for a toy example using Tensorflow.

DESCRIPTION:

- An autoregressive sequence-to-sequence model is a type of neural network architecture.
- The core idea behind an autoregressive sequence-to-sequence model is to transform one sequence into another sequence.
- Seq2Seq models are typically composed of two main components: an encoder and a decoder.
- The encoder processes the input sequence and compresses it into a fixed-size context vector, also known as the hidden state or thought vector.
- The decoder takes the context vector and generates the output sequence one step at a time.
- Many modern sequence-to-sequence models incorporate attention mechanisms, which allow the model to focus on different parts of the input sequence while generating the output.
- Autoregressive sequence-to-sequence models are applied to a wide range of tasks, including machine translation, text summarization, speech synthesis, and image captioning.

PROGRAM:

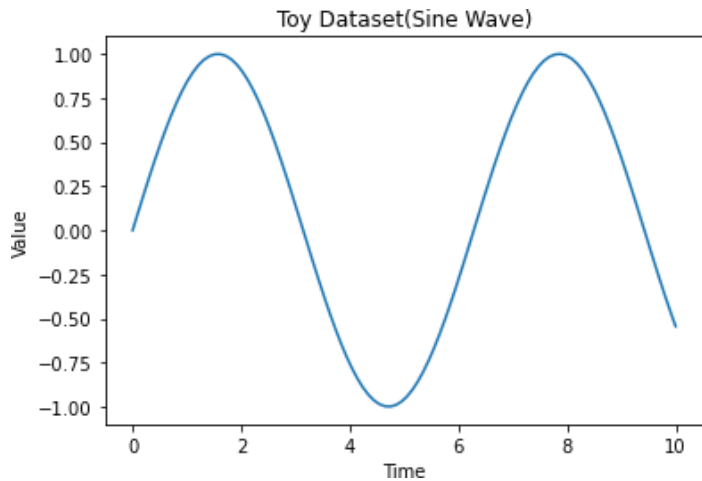
```
import numpy as np
import matplotlib.pyplot as plt

num_points=100

t=np.linspace(0,10,num_points)
data=np.sin(t)

plt.plot(t,data)
plt.title("Toy Dataset(Sine Wave)")
plt.xlabel("Time")
plt.ylabel("Value")
plt.show()
```

OUTPUT:



n=5
num_samples=len(data)

X=[]

y=[]

```
for i in range(n,num_samples):  
    X.append(data[i-n:i])  
    y.append(data[i])
```

```
X=np.array(X)  
y=np.array(y)  
split_ratio=0.8  
split_index=int(split_ratio*len(X))  
X_train,X_test=X[:split_index],X[split_index:]  
y_train,y_test=y[:split_index],y[split_index:]
```

```
class AutoregressiveModel:
```

```
    def __init__(self,n):  
        self.n=n  
        self.weights=np.random.randn(n)  
        self.bias=np.random.randn(1)  
    def predict(self,input_sequence):  
        return np.dot(self.weights,input_sequence) + self.bias  
    def train(self,X,y,learning_rate=0.01,num_epochs=100):  
        for epoch in range(num_epochs):  
            for i in range(len(X)):  
                input_sequence=X[i]  
                target=y[i]  
                prediction=self.predict(input_sequence)  
                error=target-prediction
```

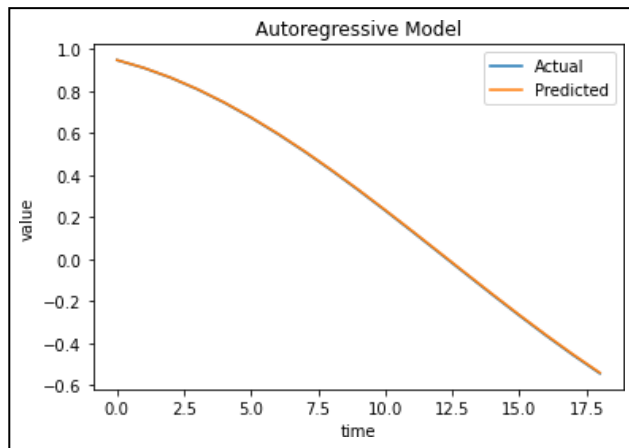
```
                self.weights+=learning_rate*error*input_sequence  
                self.bias+=learning_rate*error
```

```
model=AutoregressiveModel(n)  
model.train(X_train,y_train)
```

```
y_pred=[model.predict(input_sequence) for input_sequence in X_test]
```

```
plt.plot(y_test,label="Actual")  
plt.plot(y_pred,label="Predicted")  
plt.title("Autoregressive Model")  
plt.xlabel("time")  
plt.ylabel("value")  
plt.legend()  
plt.show()
```

OUTPUT:



RESULT:

Thus the implementation of Auto-regressive (Sequence to Sequence) model is created, trained and tested for a toy example is done successfully.