National University of Sciences and Technology

School of Electrical Engineering and Computer Science

Department of Computing

CS-405 Deep Learning

Fall 2023

# Lab 10

**Improvise a jazz solo with LSTM**

# Date: 23rd November 2023

# Name: Alina Nasir

# CMS ID: 342350

Instructor: Dr Daud Abdullah

**Section 1:**

In the first section we explored the dataset that is a corpus of Jazz music and played a snippet of the audio from the training set. Then we pre-processed the raw music data into values. And the overview of model to be used was discussed.

**Section 2:**

In this section we build and train a model that will learn musical patterns. The model takes input X of shape (m,Tx,90) and labels Y of shape (Ty,m,90) and is an LSTM with hidden states of 64 dimensions. The following concepts were used in defining the model.

**For Loop for Sequence Generation:**

We used for-loop for sequence generation because at test-time we won't know all the values of Xt in advance. So we implemented our own for loop to iterate over the time steps. In the next step we implemented the djmodel.

**Sharable Weights:**

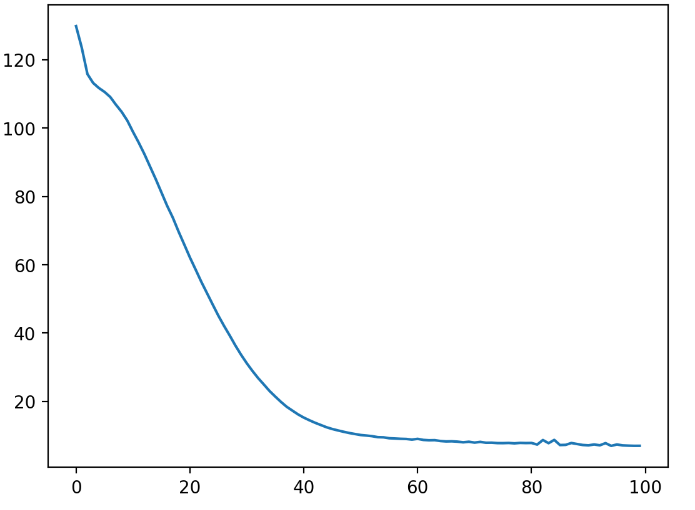
We then used the sharable weights because that all Tx copies of the LSTM layer have the same weights. This is because the weights capture the learned patterns from the data, and consistency in weights across time steps ensures that the model generalizes well and learns meaningful dependencies.

**3 types of layers:**

There are three types of layers that we worked with.

1. **Reshape():** This layer is used to reshape the output to a specific shape.
2. **LSTM():** The Long Short-Term Memory layer is a type of recurrent neural network layer designed for handling sequential data and capturing long-term dependencies.
3. **Dense():** This layer represents a regular fully-connected neural network layer, connecting every neuron from the previous layer to every neuron in the current layer."

We then created the model and trained the model for which the graph of loss vs epochs is given below:



**Section 3:**

In this section we used the trained model to generate music. We implemented the music inference model by using the LSTM Cell to predict the next hidden and cell states, computing the output, appending it to the list, and then preparing the output for the next time step by extracting the maximum value index and converting it into a one-hot encoding. This process is repeated for each time step. We then predicted the output for our inputs, converting the predictions into indices, and then converting these indices into their corresponding one-hot vector representations. These actions were crucial for generating a sequence of musical values using the provided inference model. Our model of RNN generated a sequence of values which are then post-processed into musical chords (meaning that multiple values or notes can be played at the same time).

.