Milestone 2 Report

Current State of the Project

Milestone 2 goal: Modifying a pre-trained VGG model and creating a Python notebook to have it classify between 8 different genres of music based on features found in the Million Song Dataset as well as extensively testing for optimal parameters to maximize accuracy.

Adjustments to Proposal

- For training/testing purposes we are still using the Million Song Subset rather than the full dataset so the sample data is more limited.
    - To ensure equal representation for each genre in training/testing, we are using n samples for each genre, where n is the number of samples for our smallest genre.
        - We have 48 samples for Folk music and so our testing/training uses only 48 samples from each genre. (384 samples total for training/testing)

Current Challenges / Bottlenecks

- Google Colab does not support simultaneous collaboration
    - Testing separately was done by each member making a copy of the notebook and testing out different parameters with their copies
- The full Million Song Dataset is available at https://aws.amazon.com/datasets/million-song-dataset/
    - This is meant to be hosted on Amazon Drive and getting it onto Google Cloud is a bottleneck we are currently dealing with.
- We were able to get a working model but had difficulty with the input to the first layer.
    - The model currently only looks at one feature of the data, which is the timbre of different sections within each song
    - In addition to our original goal for Milestone 3, we will also factor other features into the classifications, including: tempo, key, and mode.
    - As a result of using only one feature at this time, we were not able to meet our initial goal of 75% accuracy. Instead, our accuracy after optimizations is **[INSERT NUMBER HERE AFTER FINAL TESTING]**.
        - We expect to have much higher accuracy once we get the model to consider four features rather than just one.

Team Member Contributions

- Carroll, Quinn
    - Hosted zoom sessions for two paired-programming sessions
    - Helped debug and modify our genreClassificationDataset's __getitem__ function output to match input size required by the VGG-19 model
    - Helped create training and testing code for the ML model
- Jung, Cassiel
    - Add collected .h5 files of genres, folk and blues, to shared Google Drive and GitHub.
    - Collaborated with other team members through zoom to modify input size of VGG19 and remove errors for testing.

    - Did testing the model by changing step size to get the best accuracy and minimum loss.
        - Kept gamma and momentum as its original value but used the best result that Matthew and Aswin got for the learning rate and batch size to decrease running time.
- Poon, Matthew
    - Handled compiling genre subsets from Milestone 1, putting them on Google Drive, and setting up the Google Drive downloader so Colab notebook could access the dataset
    - Worked on the Colab notebook with the rest of the team in a paired-programming fashion over Zoom
        - Did the typing for one of two paired-programming sessions. In this session, we worked on finding the shape of the tensors that VGG19 accepts as inputs.
    - Experimented with different inputs for learning rate to optimize the model (results in table at the end of this report)
- Sai Subramanian, Aswin
    - Work found in genreClassifier.ipynb:
        - Downloaded the pretrained VGG-19 model from PyTorch and replaced the output layer with an 8-output-feature layer.
        - wrote and tested the initial version of our custom PyTorch dataset class "genreClassificationDataset." Later, in our first paired programming session of two, we found that the sample tensor returned by the __getitem__ function needed modification in order to be accepted by the VGG-19 model.
        - Experimented on moulding the shape of samples given by genreClassificationDataset's __getitem__ method to abide by shape constraints required by VGG19's first layer.
        - Moved sample moulding from the __getitem__ to the __init__ method of genreClassificationDataset. This increased the space occupied by the dataset class, and its initialization time, but decreased the testing and training time for 10 epochs from about 50 minutes to less than 2 minutes.
    - Worked on the Colab notebook with the rest of the team in a paired-programming fashion over Zoom
        - Did the typing for one of two paired-programming sessions. In this session, we worked on modifying the shape of tensors fed into VGG19 to mimic the shape of the images it was originally intended for, [3, 224, 224].
    - Experimented with training and testing over a range of batch sizes.

* First sample is the base model with no modifications

Testing for Learning Rate

* Bolded learning rate produced the best results, first row was the base model

Learning rate	momentum	Step size	gamma	minibatch	Test loss	Test acc	Best epoch
0.001	0.9	5	0.1	4	0.5083	0.2853	3
0.1	0.9	5	0.1	4	0.6193	0.1299	0
0.0001	0.9	5	0.1	4	0.4764	0.3117	1
0.00001	0.9	5	0.1	4	0.4786	0.2857	10
0.0005	0.9	5	0.1	4	0.4841	0.2587	9
0.0002	0.9	5	0.1	4	0.4750	0.3377	15
0.0004	0.9	5	0.1	4	0.4850	0.2727	7

Testing for Batch Size

Training log for varying **batch size**. Learning rate (0.001), gamma (0.1), step size (5) and momentum (0.9) held constant.

Batch Size	Min. Test Loss	Max. Test Acc.	Epochs	Elapsed Time (s)
1	1.9902	0.2468	20	366
2	0.9207	0.2987	20	248
3	0.6475	0.2338	20	204
4	0.4542	0.2468	20	190
5	0.3681	0.3117	20	174
6	0.3102	0.3506	20	168
7	0.2701	0.2208	20	163
8	0.2389	0.3247	20	153
16	0.1241	0.2208	20	111
32	0.0694	0.3377	20	103

64	0.0464	0.2857	20	103
128	0.0247	0.2597	20	105

It looks a batch size of around 32 to 64 results in the best loss values and fastest epochs with 384 samples in the full dataset.

E.g. for a batch size of 48, we got:

Batch Size	Min. Test Loss	Max. Test Acc.	Epochs	Elapsed Time (s)
48	0.0489	0.3247	20	101

Testing for Step Size

Training the model by changing step size while others are constant. Will use 0.002 for learning rate and batch size of 48 which were the best result that other team members got. Gamma and momentum will be the initialized value of 0.1 and 0.9 each.

Best result written in bold letters.

Step Size	Test Loss	Maximum Test Accuracy	Number of Epochs	Elapsed Time(s)
1	0.0483	0.3117	20	111
2	0.1910	0.3247	20	111
3	0.0433	0.3766	20	111
4	0.1946	0.3377	20	111
5	**0.0437**	**0.3766**	**20**	**111**
7	0.0502	0.3766	20	111
11	0.0738	0.3247	20	111
16	0.0655	0.3636	20	111
24	0.1844	0.3636	20	111