Neural Networks

Piña Colada: Kritim Bastola, Jagat Subedi, Fernanda Lozoya

Course: CS 491/591

December 5, 2025

Introduction

Machine learning methodology is composed of various different types of algorithms that each have different types of capabilities. The aim of the project this time was to further experiment with different models. Specifically, this project focuses on implementing multilayer perceptrons, and convolutional neural networks. As done before, the goal is multiclass classification, through feature extraction of audios. Data preprocessing was completed in the same manner as before. However, in addition to this, spectrogram data was extracted as a means to train our networks. Transfer learning was utilized as a method of further comparison, as it called for pre-trained models.

MLP Implementation

The first approach at neural networks to classify the given music data, is an implementation of multilayer perceptrons(MLP). The team utilized data from the previous project, which meant a classification of six extracted features. These being MFCC, delta, delta-delta, STFT,chroma, and spectral contrast. As the classifying the same data would not have been an ideal comparison, we utilized cross entropy and adam, as the base loss and optimizer functions for both MLP and CNN implementations. The MLP model is composed of 64 neurons and one hidden layer, with dropout and weight decay within Adam optimizer to regularize data. Network architecture is depicted in Figure 1 for further explanation. Making this a straightforward implementation, and focusing on demonstrative experimentation of MLP capabilities.

As dictated, the team looked for ways to determine the best hyperparameter settings. Ultimately, deciding on random breadth search to iterate through different settings in order to find the optimal configurations, as depictured in Figure 2. Such were determined by a validation accuracy at each run. The model was successful in scoring higher with the utilized data than the previous model implemented of logistic regression. Scoring an accuracy of 0.79 on kaggle, an improvement from the logistic model, scoring only 0.64 on its accuracy.

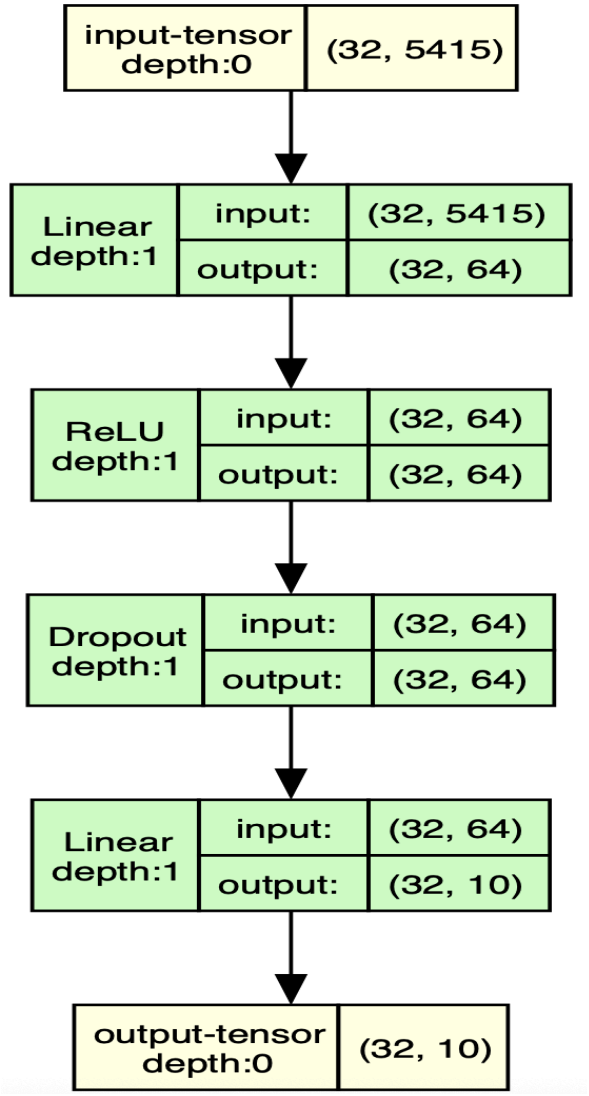

Figure 1: Network Architecture

	hidden_size	dropout_rate	lr	weight_decay	batch_size	epochs	trial	val_acc
0	512	0.020747	0.000419	0.000001	16	15	1	0.820225
19	64	0.057048	0.000247	0.000008	32	30	20	0.797753
16	256	0.145783	0.000553	0.000047	64	15	17	0.786517
7	256	0.347283	0.000896	0.000031	64	30	8	0.786517
12	64	0.060379	0.000401	0.000365	64	20	13	0.775281
8	256	0.495529	0.001181	0.000002	32	20	9	0.775281
2	512	0.276270	0.008042	0.000898	128	30	3	0.764045
18	256	0.395131	0.000120	0.000003	32	15	19	0.764045
9	512	0.453609	0.001047	0.000068	16	20	10	0.764045
13	64	0.353788	0.000424	0.000355	16	30	14	0.764045

Figure 2: Random Breadth Search Results

Convolutional Neural Networks

The approach for CNN implementation was that the team decided to select two networks to compare these using an F1-score. The team opted for VGG and ResNet implementations to begin testing and analysis. A simplistic model of VGG scored way too low on an accuracy measurement utilizing F1-score. Therefore and ultimately, the team opted on the ResNet algorithm as best suited for these classification problems. This was likely due to the ResNet's use of skip connections, which allow the program to take different routes[5]. In simpler terms, ResNet allows greater adaptability, which seemed ideal for this specific classification problem. The dataset was different to that of the MLP and previous logistic regression model, as this utilized spectrogram data. The extraction was conducted utilizing an advised method by the instructor [6]. Research of ResNet implementations was conducted in order to find a method that worked best for training purposes. The network is composed of 4 layers with filter sizes of 64, 128, 256, and 512. Thus, following a ResNet-18 pattern[1]. Then, the model utilizes average pooling, but now before utilizing maxpooling, an area of additional tuning for the team.

The training function itself was written in reliance on PyTorch tools,constructed in a similar structure as the MLP structure. As tensors and datasets were constructed, the team selected cross entropy as the algorithm's loss function. After research, the team selected two optimizers based on their similar functionality [7], to test for this model. The functions being Adam and RMSProp, both being torch tools. Adam was selected in accordance with the MLP model, but after research, the team decided to test a second optimizer. The use of the adam function also included a weight decay regularization, as done within the MLP. The learning rate had varied values to identify an optimal setting for model learning and minimal loss. Though kernel sizes were experimented with, along with spectrogram generation settings. The highest

accuracy scores produced experimentation between optimizer and learning rate experimentation. With a learning rate of 0.00001, figures 3 and 4 represent the difference between the RMS and Adam optimizers Figure 4 shows a more steady learning process, while figure 3 is only slightly more varied. However, the RMS run produced the highest kaggle score of the CNN experimentation, though Adam was not far behind.

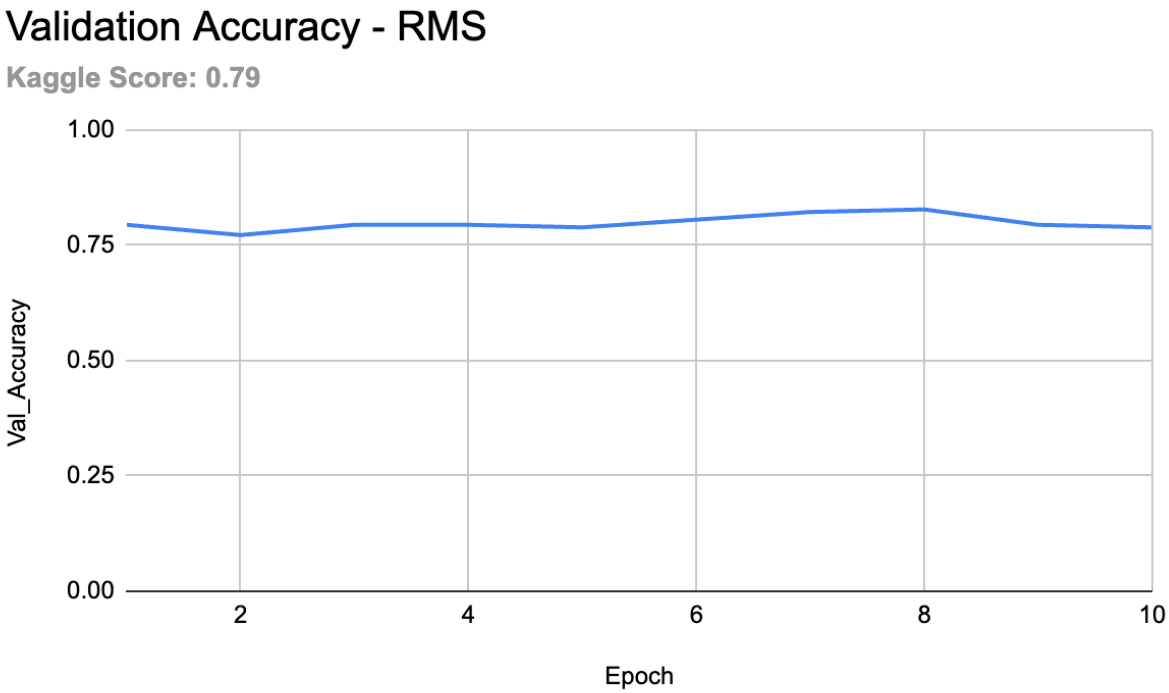

Figure 3: RMS Learning Curve

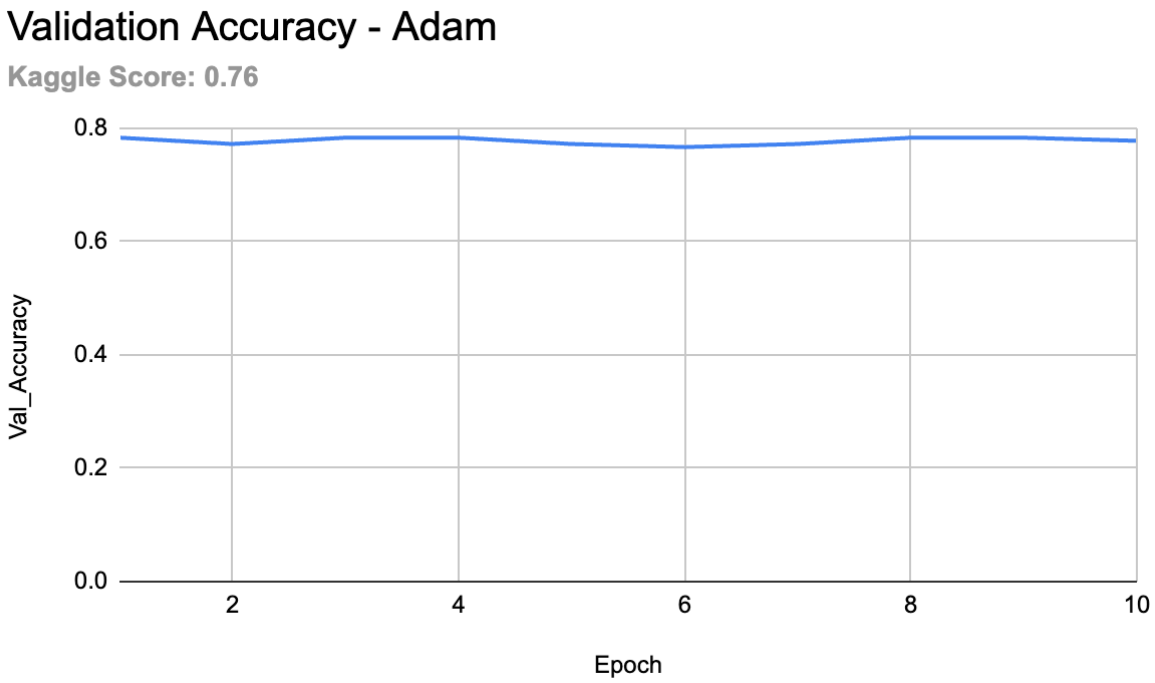

Figure 4: Adam Learning Curve

Transfer Learning

The implementation of transfer learning utilized pre-trained models. The dataset was a series of spectrograms, just as the CNN's dataset. The chosen model is also a ResNet, however, this is a ResNet-50. As opposed to the previous implementation of a ResNet-18 structure. The selected optimizer for this model, SGD, was selected due to its stability in such a large network[9]. However, this model ran a much larger dataset than the ResNet-18. The team decided to run a large model in hopes of topping the Kaggle score accuracy, as both the previous models scored the same. This however utilized more data and a deeper structured network. As such, each epoch took a long time to converge. In addition, the process was due to model tuning and trying to find accurate results before allowing final convergence.

Comparison

Each model had something unique to it, making it somewhat hard to do a direct comparison of performance. In addition, from the experimentation the team gathered that each model can be harshly affected by many factors. Whether it was optimizers, loss function, regularization, data preprocessing, data normalization, weight decay, or learning rate. Each of these could either bring accuracy numbers to a steady convergence each time, or plummet them to unstable and inaccurate outputs. I believe the team had most to learn from the CNN implementation, as it had more adjustable parameters and factors. Interestingly enough, for each

of the models, accuracy scores were all very close. MLP was interesting and even matched the CNN score on Kaggle, however, the perceptron method was not much of a deep network. Therefore, the team drew the conclusion that the data was the one that influenced this. As the extracted features contained far more information than the spectrogram data extraction. The ResNet-50, though large, struggled initially to produce meaningful results. The team was just able to extract more from the ResNet-18.

References

[1]
"7.6. Residual Networks (ResNet) — Dive into Deep Learning 0.14.3 documentation," *d2l.ai*. https://d2l.ai/chapter_convolutional-modern/resnet.html
[2]
YasinShafiei, "ResNets fully explained with implementation from scratch using PyTorch. | @YasinShafiei | Medium," *Medium*, Aug. 15, 2024. https://medium.com/@YasinShafiei/residual-networks-resnets-with-implementation-from-scratch-713b7c11f612
[3]
"normalizing mel spectrogram to unit peak amplitude?," *Stack Overflow*. https://stackoverflow.com/questions/54432486/normalizing-mel-spectrogram-to-unit-peak-amplitude
[4]
P. Nandi, "CNNs for Audio Classification | Towards Data Science," *Towards Data Science*, Mar. 24, 2021. https://towardsdatascience.com/cnns-for-audio-classification-6244954665ab/
[5]
N. Ahmed and S. Mukherjee, "Writing ResNet from Scratch in PyTorch | DigitalOcean," *Digitalocean.com*, 2025. https://www.digitalocean.com/community/tutorials/writing-resnet-from-scratch-in-pytorch
[6]
alifrahman, "Spectrograms extraction using librosa," *Kaggle.com*, May 25, 2021. https://www.kaggle.com/code/alifrahman/spectrograms-extraction-using-librosa (accessed Dec. 06, 2025).
[7]
Musstafa, "Optimizers in Deep Learning," *Medium*, Feb. 12, 2022. https://musstafa0804.medium.com/optimizers-in-deep-learning-7bf81fed78a0
[8]
"pytorchMLP.ipynb," *Google.com*, 2024. https://colab.research.google.com/drive/134NRf4PAR1CuQejf6q3DTDZNaFklR8NY?usp=sharing (accessed Dec. 07, 2025).
[9]
X. Li, Z. Yang, and Y. Wang, "Stochastic gradient accelerated by negative momentum for training deep neural networks," *Applied Intelligence*, vol. 55, no. 15, Sep. 2025, doi: https://doi.org/10.1007/s10489-025-06900-9.