How Are You Feeling?
Inferring Mood from Audio Samples

Joel Haynie
Department of Computer Sciences
University of Wisconsin, Madison
email@wisc.edu

Ankit Vij
Department of Computer Sciences
University of Wisconsin, Madison
email@wisc.edu

Amanpreet Singh Saini
Department of Computer Sciences
University of Wisconsin, Madison
email@wisc.edu

Eric Brandt
Department of Computer Sciences
University of Wisconsin, Madison
ebrandt@wisc.edu

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Background

Works we plan to cite (to make sure bibtex is working):

- CNN Architectures for Large-Scale Audio Classification, Hershey[6]
- What's wrong with CNNs and spectrograms for audio processing?, Rothmann[7]
- Inside the spectrogram: Convolutional Neural Networks in audio processing, Dorfler[3]
- Getting Started with Audio Data Analysis using Deep Learning, Shaikh[8]
- Hearing AI: Getting Started with Deep Learning for Audio on Azure, Zhu[9]
- How do deep convolutional neural networks learn from raw audio waveforms?, Gong[4]
- AudioSet [5]
- TensorFlow [1]
- Keras [2]

2 Implementation

2.1 Data Acquisition and Extraction

We collected our data from Google's AudioSet [5] which consists of an expanding ontology of 632 audio event classes and a collection of 2,084,320 human-labeled 10-second sound clips drawn from YouTube videos. The ontology is specified as a hierarchical graph of event categories, covering a wide range of human and animal sounds, musical instruments and genres, and common everyday environmental sounds. From the dataset, we focussed on music mood samples and extracted data points with 4 mood classes- Happy, Sad, Angry, and Scary. The dataset had two groups, 'unbalanced' and 'evaluation' set (We used the 'unbalanced' data set because the 'balanced' data set was too small

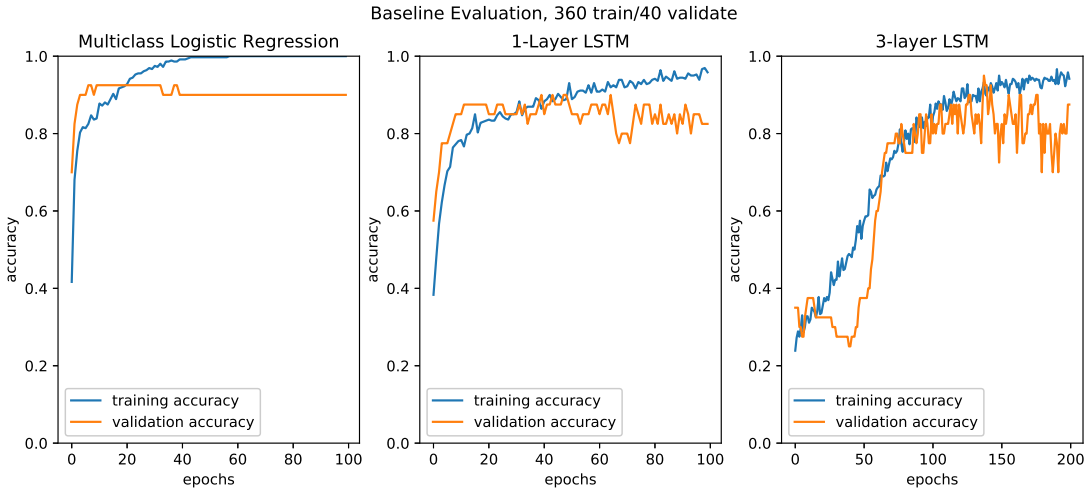

Figure 1: Baseline training performance for 3 models.

for our needs). We created a `.csv` of 223 data points with their labels from the 'evaluation' set with $\tilde{5}6$ entries for each mood class. Additionally, we created a `.csv` of 400 data points with their labels from the 'unbalanced' set with 100 entries for each mood class. These CSVs were then used to for two purposes:

1. Download the corresponding Google-produced spectrographs of each of the samples for use in establishing a baseline classification accuracy.
2. Download the audio samples in .WAV format directly from their source (YouTube) which we used in the pre-processing step described in the next section.

2.2 Baseline

To establish a baseline for the achievable accuracy in learning a multi-class classification task, we began by using the spectrographs for the 400 + 223 data instances we identified as a) being music audio, and b) having one of the 4 mood labels we chose for analysis.

To evaluate this baseline, Python was used, enlisting the libraries TensorFlow [1] and Keras [2].

The 400 samples were evenly divided by class for stratified k-fold cross validation and used to train 3 different neural networks:

1. Simple multi-class logistic regression classifier
2. 1-Layer LSTM (Long-short term memory) recurrent neural network
3. 3-Layer LSTM (Long-short term memory) recurrent neural network

In each case, the model was trained using batches of 40 samples, randomized at each presentation, for a sufficient number of epochs to infer steady-state accuracy.

We evaluated the performance of each of the 3 models by two methods:

1. Validation set accuracy over an increasing number of epochs.
2. Evaluation on a Test Set of 223 never-seen-before data instances (`audioset/eval`).

The performance of the training sessions is shown in figure 1.

After training, the evaluation on the 220-instance balanced test set, we observed the accuracies shown in table 1.

Finally, to make sure that our four chosen classes do not have an abnormal correlation between any combinations of classes, we also computed confusion matrices for the models. The confusion matrix for Logistic Regression (arguably the best performing classifier) is shown in table 2.

Table 1: Baseline accuracy on held-aside test set of 220 instances for 3 models.

Model	Accuracy
Logistic Regression	0.803
1-Layer LSTM	0.830
3-Layer LSTM	0.731

Table 2: Confusion matrix for baseline logistic regression classifier of 223 test instances.

	Happy	Sad	Angry	Scary
Happy	45	9	2	1
Sad	11	39	2	4
Angry	0	1	53	4
Scary	0	7	3	42

From the baseline evaluation we can draw some conclusions and inferences:

- The input feature sets must be nearly linearly separable, as evidenced by the strong performance of the simple multiclass logistic regression classifier.
- Google's preprocessing of the raw audio waveforms into 10-frame spectrographs, including processing by Google's own CNN and PCA reduction clearly has produced data that is well separated without significant further processing.
- Evidence of the linearly separable feature data is supported by much more complicated non-linear classifiers (1-Layer LSTM and 3-Layer LSTM) not yielding better performance.
- There is evidence that the LSTM models are subject to overtraining at higher numbers of epochs.
- More complicated models with more parameters, particularly the 3-Layer LSTM, take significantly more time to train.
- The confusion matrix suggests, surprisingly, that 'Happy' and 'Sad' are the most often confused classifications, and that 'Scary' and 'Angry' are comparatively easy to predict.

2.3 Preprocessing

2.4 CNN Training

3 Discussion

4 Conclusion

4.1 Future Work

References

[1] Google Brain. Tensorflow: An open source machine learning framework for everyone, 2018. URL `https://www.tensorflow.org`.

[2] François Chollet. Keras: The python deep learning library, 2015. URL `https://keras.io`.

[3] M. Dörfler, R. Bammer, and T. Grill. Inside the spectrogram: Convolutional neural networks in audio processing. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 152–155, July 2017. doi: 10.1109/SAMPTA.2017.8024472.

[4] Yuan Gong and Christian Poellabauer. How do deep convolutional neural networks learn from raw audio waveforms?, 2018. URL `https://openreview.net/forum?id=S1Ow_e-Rb`.

[5] Google. Audioset: A large-scale dataset of manually annotated audio events, 2018. URL `https://research.google.com/audioset/index.html`.

[6] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017. URL https://arxiv.org/abs/1609.09430.

[7] Daniel Rothmann. What's wrong with cnns and spectrograms for audio processing?, 2018. URL https://towardsdatascience.com/whats-wrong-with-spectrograms-and-cnns-for-audio-processing-311377d7ccd.

[8] Faizan Shaikh. Getting started with audio data analysis using deep learning (with case study), 2017. URL https://www.analyticsvidhya.com/blog/2017/08/audio-voice-processing-deep-learning/.

[9] Xiaoyong Zhu, Max Kaznady, and Gilbert Hendry. Hearing ai: Getting started with deep learning for audio on azure, 2018. URL https://blogs.technet.microsoft.com/machinelearning/2018/01/30/hearing-ai-getting-started-with-deep-learning-for-audio-on-azure/.