UIUC SST DiCOVA 2021 Challenge System Report

John Harvill[1], *Yash Wani*[2], *Mark Hasegawa-Johnson*[1], *Narendra Ahuja*[1], *David Beiser*[2], *David Chestek*[3]

[1]University of Illinois at Urbana-Champaign
[2]University of Chicago
[3]University of Illinois at Chicago

harvill2@illinois.edu, yashwani@uchicago.edu, jhasegaw@illinois.edu, n-ahuja@illinois.edu, dbeiser@medicine.bsd.uchicago.edu, dchest2@uic.edu

Abstract

The recent outbreak of the COVID-19 pandemic has caused much loss of life and economic damage over the past year. In order to help speed up the process of bringing our society back to a normal state, it is desirable to search for testing methods that are cheap, reliable and fast. These tests would allow people to know sooner if they have been infected, allowing these individuals to access medical care and quarantine faster. Due to the effectiveness of cough for determining many kinds of diseases, we explore the ability of audio recordings of cough samples to inform us about COVID-19 status in this challenge. We explore the applicability of recent techniques such as autoregressive predictive coding pretraining and spectral augmentation to improve the performance of a neural cough classification system. We use uni-directional long short-term memory (LSTM) networks for pretraining and bi-diretional LSTM, or BLSTM, networks for classification.

Index Terms: COVID-19, acoustics, machine learning, respiratory diagnosis, healthcare

1. System Description

1.1. Methodology Overview

Due to the small size of the DiCOVA dataset [1], we choose to pretrain on the larger COUGHVID dataset [2] in order to train relevant feature extractors for cough. We then train a classifier to determine presence of COVID-19 on DiCOVA data using the features from the lower layers of the pretrained network as input.

1.2. Pre-processing

We first downsample all audio recordings to 16kHz. We then compute the Mel log spectrogram using a window of 1024 samples and a hop length of 160 samples (10ms). The spectrograms are then clipped by setting any components less than or equal to -120dB to -120dB. Then we normalize the spectrogram such that the minimum value is 0 (corresponding to -120dB) and the maximum value is 1 (corresponding to 0dB).

1.3. Feature Description

Pretraining has shown incredible success in natural language processing with the advent of BERT [3]. Inspired by this success, Chung et al. [4] adapt pretraining to audio in a way similar to BERT. The key similarity to BERT is that the pretraining is predictive. No external labels are required, but rather structure is learned by trying to fill in missing parts of the original signal. Chung et al. explore two types of pretraining (1) Autoregressive predictive coding (APC), (2) Contrastive predictive coding (CPC). The authors find superior performance of APC over CPC for both phone classification and speaker verification tasks, so we choose to implement APC.

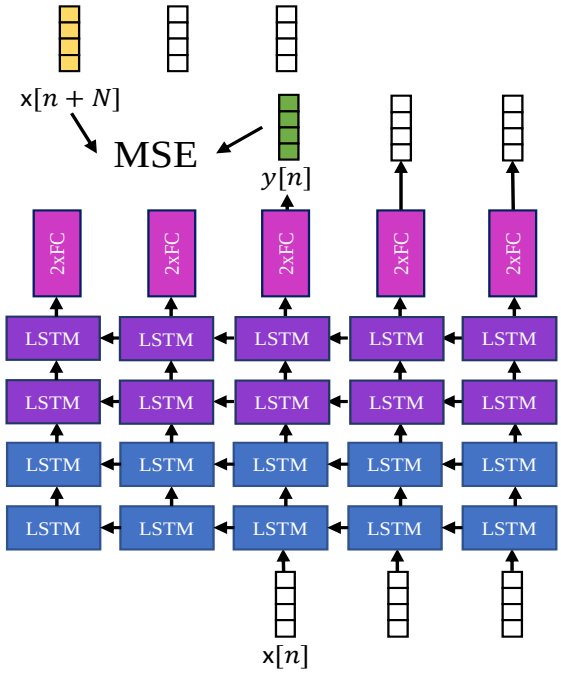

Figure 1: *Autoregressive Predictive Coding*

APC is a simple yet effective form of pretraining for audio. The objective is for the model to predict a future spectral frame given previous frames. If the goal is to predict the N'th future frame, the error for any audio clip becomes:

$$E = \sum_{n=1}^{T-N} (y[n] - x[n+N])^2 \quad (1)$$

By forcing the model to predict the future spectral frame, underlying structure of cough is learned. We use a uni-directional long short-term memory (LSTM) model for pre-training (bi-directional would break causality). We use 4 LSTM layers with a hidden size of 400 and a dropout of $p = 0.1$. Then we use two linear layers with 500 nodes. We apply the hyperbolic tangent nonlinearity after the first linear layer. The output dimension of the model is the same as that of the input since we compute mean-squared error (MSE) loss. See Figure 1 for a visual de-

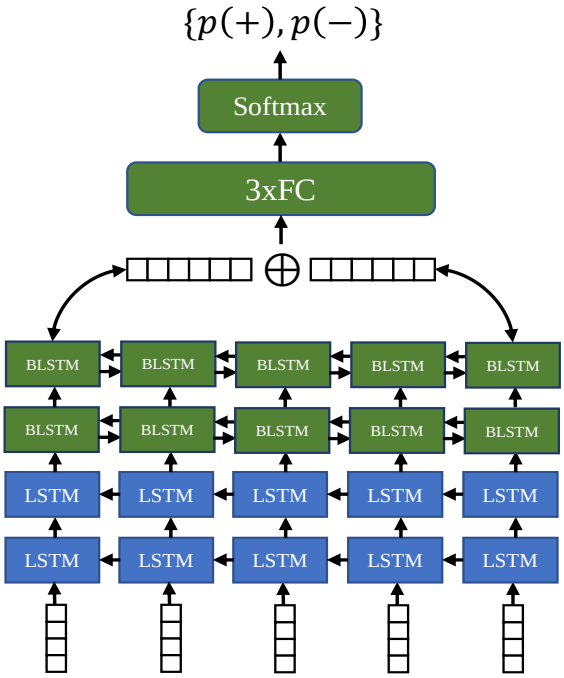

Figure 2: *Fine-tuning: We denote frozen layers from pre-training in blue and trainable layers in green.*

scription. We split the 4 LSTM layers into "upper" (purple) and "lower" (blue) layers because we use the output of the lower layers as extracted features during fine-tuning, discussed next.

1.4. Classifier Description

We pass the output from the lower layers of the APC model as input features for our classification network. We use a network composed of 2 bi-directional long short-term memory (BLSTM) layers followed by three fully-connected layers. The forward and backward summaries are taken from the BLSTM layers and concatenated before being fed through the fully-connected layers. We apply dropout with $p = 0.1$ in both the BLSTM and fully-connected layers. To predict the probability of the cough sample coming from a COVID-19 positive patient, we take the softmax at the output and use the cross-entropy loss for training. During training, we apply SpecAugment from Park et al. [5] to the cough samples. We find that spectral augmentation is critical for generalization to the test data. See Figure 2 for a visualization of the fine-tuning model.

1.4.1. Ensembling

During training we validate with the area-under-curve (AUC) metric. AUC measures the performance of a classification system with imbalanced data better than the accuracy metric. AUC is the area underneath the true positive rate (TPR) vs. false positive rate (FPR) curve for different classification thresholds ranging from zero to one. Since we directly optimize the cross-entropy loss and not AUC, we find the AUC varies substantially throughout training. We hypothesize that different sets of model parameters may classify particular samples better or worse than one another, and that an ensemble of several high-performing validation checkpoints may improve test performance. We find this to be the case, and choose the best three validation checkpoints and take the mean of their output probabilities for final validation scores for each fold. For the blind test data, we take the mean of the scores from each of the five folds. Thus our test predictions are an average of $5 \times 3 = 15$ model probability scores.

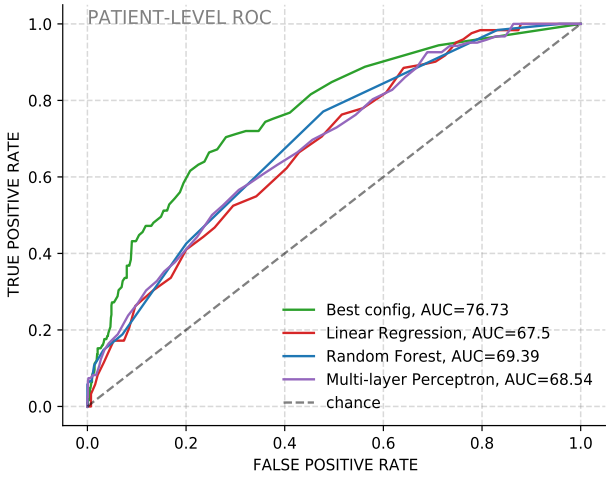

Figure 3: *Best config: We plot ROC curves for our best-performing configuration against the three baselines provided for the DiCOVA challenge on the validation data. The curves given are the mean for each approach over five folds.*

1.5. Results

Muguli et al. [1] provide three baseline system implementations for the DiCOVA challenge, described briefly below:

- Linear Regression: Classifier is trained for a maximum of 25 iterations with `liblinear` optimizer, regularization strength of 0.01 and l_2 penalty.
- Multi-layer Perceptron: Classifier is composed of one layer of 25 hidden units with the `tanh` nonlinearity applied to the output. l_2 regularization is used with weight 0.001. Examples are sampled during training such that the model is equally exposed to positive and negative samples.
- Random Forest: Classifier uses 50 trees and Gini impurity.

A comparison between the three baselines and our proposed system is given in Figure 3. We notice over 7 percentage points improvement for our method over the random forest approach, which is the best-performing baseline method. The improved performance of our approach over the baseline methods appears to come from a combination of three factors:

- Improved features: Pretraining on COUGHVID [2] audio samples helps us find features directly tuned for extracting relevant information from coughing sounds
- Improved capacity: Our classifier contains over 2 million parameters
- Data augmentation: Creating small variations from the existing audio clips leads to better generalization

We achieve an AUC of 85.35 on the test data provided in the DiCOVA challenge. This demonstrates that for the provided train/val/test split, our approach generalizes well.

2. References

[1] A. Muguli, L. Pinto, N. R., N. Sharma, K. Prashant, P. Ghosh, R. Kumar, S. Ramoji, S. Bhat, S. Chetupalli, S. Ganapathy, and V. Nanda, "Dicova challenge: Dataset, task, and baseline system for covid-19 diagnosis using acoustics," *arXiv preprint arXiv:2103.09148*, 2021.

[2] L. Orlandic, T. Teijeiro, and D. Atienza, "The coughvid crowdsourcing dataset: A corpus for the study of large-scale cough analysis algorithms," *arXiv preprint arXiv:2009.11644*, 2020.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv preprint arXiv:1904.03240*, 2019.

[5] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.