Detecting sleep apnea events in snore-related sounds using a convolutional neural network

Greg Maslov <maslov@cs.unc.edu>

***Abstract*—I propose a Convolutional Neural Network model to automatically detect and classify sleep apneas and hypopneas using only unsophisticated, noisy audio recordings of snore-related sounds using a general-purpose microphone. I evaluate the model on a data set from the UNC Sleep Lab and discuss future work.**

I. Motivation

Obstructive sleep apnea (OSA) is a common condition whose symptoms can include daytime sleepiness, hypertension, cardiovascular morbidity, and impaired cognitive function. It is estimated that 5% of adults suffer from OSA, and up to 20% have mild or asymptomatic OSA. However, OSA is usually unrecognized and undiagnosed, and is likely to result in a large population-level health care burden [1].

Unfortunately, OSA is expensive to diagnose. The "gold standard" diagnosis requires several nights of instrumented sleep in a hospital and manual examination of the resulting polysomnogram (PSG) traces. It is not even easy to tell when a sleep study would be useful, as the symptoms are often dismissed or mistaken as being caused by something else. There are portable monitoring devices which can mitigate the cost and inconvenience of a sleep study, and achieve diagnostic agreement of between 91% and 75% [2], but these still represent a significant cost in sensor hardware and are not widely used.

Snoring sounds carry information about sleep apnea events [3], [4], [5]. Hardware capable of recording and processing audio is ubiquitous in the form of smartphones, tablets, and notebooks. If a reliable algorithm to detect apnea-hypopnea events based only on snoring sounds were could be developed, it could be widely deployed to allow easy self-screening for a large fraction of the population.

Convolutional neural networks (CNNs) are a type of deep neural architecture which has been successfully applied to many different classification and recognition tasks. I propose the use of such a network to recognize sleep apnea events in snore-related sound recordings.

II. Related Work

Current approaches to detecting sleep apnea events are mostly based on spectral analysis; they include:

- linear predictive coding to model formant frequencies (88% sensitivity, 82% specificity); [3]
- logistic regression on snore waveform parameters: pitch, power spectral density, formant frequency and amplitude, 1st derivatives of these, etc. (>93% sensitivity, 73-88% specificity); [5]

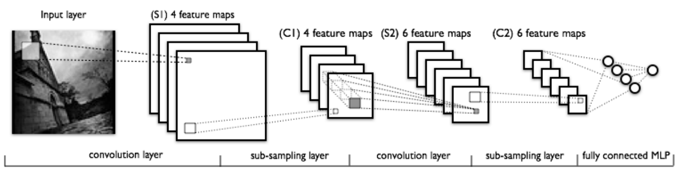

Fig. 1. An example of a LeNet model for image recognition.

- ad hoc analysis of power spectrum peaks (93% sensitivity; 67% specificity) [6].

Higher specificity would be desirable to reduce false-positive costs in a method intended for wide application. Furthermore, the above cited methods all use the signal from a tracheal microphone, the use of which is not feasible in the intended application (since a tracheal microphone is not a common household item).

III. Model

I used a simplified LeNet-like[7] convolutional neural network. The input is a 300x300 real-valued image computed using a Short-Time Fourier Transform (STFT) of the raw audio (see section §IV). The first four hidden layers alternate between convolution and subsampling layers. Above these is one fully-connected sigmoidal layer followed by a final logistic regression layer. A similar model is shown in figure 1.

Part of my motivation for choosing this type of model was that it works well for image recognition, and a spectrogram (i.e., an STFT window) is an image that can be used to isolate spectral features of an audio recording. The other part was that this type of deep architecture seemed the easiest to understand and implement given my background (as opposed to, say, RBM-based models).

Each convolution layer consists of a small fixed number of feature maps. Each feature map produces an output image by convolving its input image(s) by a parameter kernel W, adding a parameter bias vector b, and applying a sigmoidal (tanh) nonlinearity. The convolution kernel of a feature map is thus 3-dimensional, having the axes <input image index, x-position, y-position>, and the weights of a layer (composed of several feature maps) form a 4-dimensional tensor.

The sub-sampling layers have no parameters. They merely implement a "max-pooling" operation. Each subsampling layer has one hyperparameter, the pool size, which in most models is simply two. Max-pooling consists of dividing each input image into a nonoverlapping uniform grid, each cell being pool-size wide in both dimensions. The maximum of each cell's elements becomes the output of that cell. With a pool size of m, this effectively scales down each input image by a factor of $1/m$.

The max-pooling step is actually implemented using a softmax function, which permits the use of gradient descent through the subsampling layers.

After the final subsampling layer is a tanh-activation layer fully connected with the outputs of the layer below it, just as you would see in any multilayer perceptron. On top of that is a logistic regression layer, which performs the final classification.

IV. Data

The training data consists of a set of audio recordings of snoring and sleep sounds, annotated with the approximate time of each apnea or hypopnea event. The duration of an event is not provided. The audio is formatted as a stream of 16-bit signed integer samples at a rate of 12000Hz.

I performed some preprocessing to extract (hopefully) useful features from the raw audio data. The first step is to compute a Short-Time Fourier Transform (STFT) of the audio signal. This turns the one-dimensional time-domain data into a two-dimensional time- and frequency-domain image, known as a spectral waterfall or spectrogram. The first Fourier coefficient (the DC offset) is discarded from each row, as well as all of the high-frequency coefficients above the frequency range of snoring and breathing sounds. This truncated image is then subdivided into sequential, overlapping windows of around 15 seconds each. Each window image is then padded on the left and right sides (along the time axis) with 10 rows of zeroes. This ensures that features near the edge of the image can still be detected by a convolution kernel. See section §V for the exact parameters of these steps; I tried to choose them so that the resulting windows are square, reasonably small, and informative.

Finally, the windows are classified according to which type of sleep apnea event, if any, occurs inside them. Although windows overlap, events never occurred close enough together in the data set to cause ambiguity. See figure 2 on page 5 for an example of what they look like.

A. UNC Sleep Lab

The UNC-CH Sleep Lab performs routine sleep studies and collects polysomnograms (PSGs) with audio and video channels. The PSGs are manually annotated for apnea-hypopnea events based on the criteria listed in table I. Dr. Heidi Roth kindly made available one month's worth of data, comprising 29 full-night recordings with a total of 218 hours of audio, of which 103 are annotated. I did not make use of the unannotated recordings.

Most subjects are recorded for two nights: one to adjust to sleeping in the hospital environment, and one used for an actual diagnosis. Some subjects are healthy, and others have varying types and severities of sleep apnea. The audio data was recorded using a microphone (of unknown specifications) mounted in the ceiling 56 inches directly above the pillow, pointing down. In addition to sleep sounds, the recordings also contain ambient noise from air conditioning, hospital systems, television, and conversations.

Apnea	90% or more decrease on NASAL channel for at least 10 seconds • *Central* - absence of airflow and effort. • *Obstructive* - absence of airflow and decreased effort. • *Mixed* - initial absence of airflow and effort followed by effort in Abdomen and Chest channels.
Hypopnea	Decrease in NASAL, but partly reduced (at least 30%) in PNASAL and 4% desaturation in SpO2 channel within 10-20 seconds after the respiratory event.
RERA	Not flat in NASAL, 50% reduction in PNASAL, end either an arousal or 3% desaturation. In REM, arousal is a chin EMG change; in NREM, need 3 seconds of alpha waves in EEG.

TABLE I
RULES FOR SCORING RESPIRATORY EVENTS

Class	Count
(no event)	131725
RERA	4263
OAa	7707
CAa	774
OHa	3954
CHa	78
OA	142
CA	41
OH	754
CH	0

TABLE II
NUMBER OF TRAINING EXAMPLES IN EACH CLASS

The annotations consist of several different types of events: obstructive apnea (OA), obstructive apnea with arousal (OAa), obstructive hypopnea (OH), obstructive hypopnea with arousal (OHa), central apnea (CA), central apnea with arousal (CAa), central hypopnea (CH), central hypopnea with arousal (CHa), respiratory effort related arousal (RERA), and body position changes (Supine, Upright, Left, Right).

The annotated data set did not contain any CH events. The body position changes were irrelevant. This leaves the distribution of training examples (after preprocessing and windowing) shown in table II. An example of each class is shown in figure 2 on page 5.

B. Synthetic

I used freely available audio processing software to construct a simple synthetic data set for testing. The construction was as follows:

- A background of Brownian noise at amplitude 0.1. This type of noise was chosen because it sounds similar to the background noise in the real data.
- DTMF tones, amplitude 0.01, duty cycle 30%, pattern "1,2,1,2,1,2,1,...". This is intended to approximately imitate soft breathing sounds: a slow repeating pattern, alternating between two frequencies, with pauses. The chosen amplitude is between 0-10dB above the noise floor in the DTMF frequency range. DTMF tones were used because they were easy to generate with this particular software package.
- To simulate apnea events, I chose some arbitrary sections of the "breathing" pattern to delete.

This data should be relatively easy to model, since the "breathing" tones are above the noise floor, and an apnea event can be detected simply by finding gaps in the pattern. Unlike the real data, there is no variation in the breathing pattern's rate, pitch, timbre, or whatever unknown quality might denote an imminent apnea event. There is no structured environmental noise and no tossing-and-turning sound. There are only two classes of event: apnea and normal breathing, as opposed to the nine classes in the real data set.

V. Training

The overall training algorithm is stochastic gradient descent (SGD), with the cost function being the negative log-likelihood of a batch of predictions from the logistic regression output layer; $LL(\theta) = -\sum_i \log p_\theta(y_i|x_i)$. I made use of the symbolic differentiation capabilities of Theano[8] to derive the gradient of each parameter with respect to this cost throughout the model.

In my initial tests using simple gradient descent (also known as backpropagation), I found that the performance of this model on the synthetic data set was extremely sensitive to the learning rate and batch size parameters, so much so that I was having trouble getting reliable convergence at all. Implementing the Rprop algorithm [9] solved my convergence issues and eliminated the need to carefully tune the learning rate.

The complete set of hyperparameters is listed below.

Data preprocessing parameters

- STFT width: 150 milliseconds.
- STFT stride: 50 milliseconds.
- STFT windowing function: Hamming (raised cosine). This was chosen to maximize frequency resolution.
- Number of high-frequency STFT coefficients to discard: 600 (that is, two-thirds of them). This step combined with the wide STFT has the effect of magnifying all the features which occur in the low- to mid-frequency range. Hopefully those are the interesting ones.
- Image downsampling factor: 1 (no downsampling).
- Image window width: 14 seconds. Chosen so that most apnea events can fit completely inside the window.
- Image window stride: 2.5 seconds. One second or even half a second would have been better, but I did not have the computational resources to support further doubling of the data set.
- Image window padding: 10 pixels on the left and right (the time axis), none on the top and bottom (the frequency axis).

These shape parameters result in windows of size 300x300, each overlapping by a large margin with its neighbors. The total number of examples (windows) across all annotated audio files is 149,448 (see table II).

Network topology parameters

- CNN kernel sizes: [15,6,5]. These were guessed to hopefully match the feature size at each layer.
- CNN max-pool sizes: [2,2,2].
- Number of kernels per layer: [4,16,64]. These were chosen so that the total number of parameters in each layer would be approximately constant.
- Sigmoid layer size: 500. Of uncertain provenance and significance.

Training parameters

- Initial Rprop learning rates: $\Delta \leftarrow 0.005$. Not critical, but should be kept reasonably small.
- Rprop parameters: $\eta^- = 0.5$, $\eta^+ = 1.2$, $\Delta_{min} = 10^{-6}$, $\Delta_{max} = 50$. All noncritical and as suggested in [9].
- Training set: 6000 total examples, distributed equally between classes but otherwise selected at random. Some classes do not have enough examples to meet this quota; the shortfall is made up with no-event examples.
- Validation and test sets: 1000 examples each, selected in the same way as the training set.
- Minibatch size: 500. The more the better, but memory usage increases. The training set is randomly reshuffled out to the minibatches every epoch.
- Maximum training epochs: 100.

VI. Implementation

The implementation was done in Python2.7, making use of the Theano[8][1] library and some code from deeplearning.net. Although Theano has the ability to transparently move computations to a GPU, mine unfortunately lacked sufficient video memory, so all of my processing was done on a four-core 3.4GHz AMD processor, using approximately 8GB of RAM.

VII. Results

For lack of easy access to anything more sophisticated, I compared my model to a basic logistic regression + Rprop. The logistic regression model achieved 8% validation error on the synthetic data set, but only 68% on the real data (figure 3). My CNN model achieved 1.5% validation error on the synthetic data set (figure 4), and 70% on the real data after 18 training epochs. Alas, at the time of this writing, I cannot know how it might perform with a longer training period, because I only got the last major bugs ironed out 12 hours ago and the training process is still running.

VIII. Discussion

Working with the assumption for now that my model will only ever achieve 70% error, I suspect that the culprit is my network topology hyperparameters. If I had another week I'd either make some more guesses or do an exhaustive search.

Maybe the problem is that this data set really doesn't support making predictions of sleep apnea events.

Regardless, it fairly clear that my synthetic data set is too easy.

[1]Available at http://deeplearning.net/software/theano/

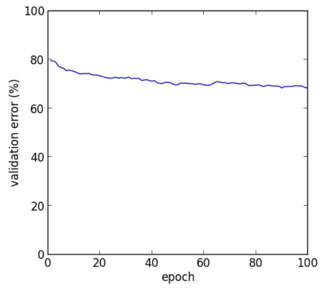

Fig. 3. Logistic regression + RPROP trained on the real data set. The final validation error is 68.5%.

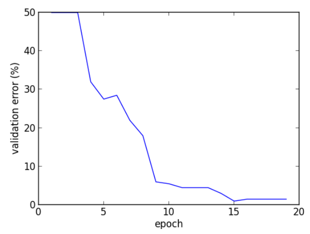

Fig. 4. CNN model trained on 1000 examples from the synthetic data set (section IV-B). The final validation error is 1.5%.

IX. Future Work

The model is currently very large in terms of the number of parameters to be learned. Mostly this is because I wasn't sure how to design an appropriate CNN topology for this problem, and went with a liberal guess. A grid search over the hyperparameters to find the smallest effective model may prove fruitful, if time-consuming.

A smaller model would also be faster to evaluate, and vital if a mobile application is in the future.

It may have been helpful to make use of the unlabeled data as well as the annotated recordings. In [10] Lee describes a "Convolutional DBN" architecture for semi-supervised learning of subtle spectrogram features such as the gender of a speaker, which may be well-applicable to this task.

All of the recordings in the data set were taken in the same few identical rooms, with the same type of microphone, in the same hospital's ambient noise environment. It remains to be seen how well or poorly the trained model would perform in a novel environment. Perhaps augmenting the data set with noisy or cleverly re-filtered versions of the audio signals would be effective, if performance is poor.

The Apnea-Hypopnea Index is the actual diagnostic measure used in rating severity of sleep apnea. It is a count of apnea and hypopnea events per hour. An algorithm still needs to be designed to transform the output of the trained model into a single AHI value.

173	0	4	18	0	2	0	0	8
104	0	0	4	0	1	0	0	2
59	0	10	33	0	0	0	0	9
33	0	7	69	0	2	0	0	0
86	0	6	3	0	0	0	0	16
12	0	4	36	0	23	0	0	3
98	0	5	6	0	0	0	0	2
23	0	0	24	0	4	0	0	0
84	0	3	0	0	0	0	0	24

Fig. 5. Confusion matrix of the CNN model after 18 training epochs. Each row represents the predictions made for that class.

X. Acknowledgments

I thank Dr. Heidi Roth and Chayla Hart at the UNC Sleep Lab for their efforts in collecting this data set and their assistance in grooming and deciphering it.

References

[1] T. Young, "Epidemiology of Obstructive Sleep Apnea: A Population Health Perspective," *American Journal of Respiratory and Critical Care Medicine*, vol. 165, pp. 1217–1239, May 2002.

[2] R. Santos-Silva, D. E. Sartori, V. Truksinas, E. Truksinas, F. F. F. D. Alonso, S. Tufik, and L. R. A. Bittencourt, "Validation of a portable monitoring system for the diagnosis of obstructive sleep apnea syndrome.," *Sleep*, vol. 32, pp. 629–36, May 2009.

[3] A. K. Ng, T. S. Koh, E. Baey, T. H. Lee, U. R. Abeyratne, and K. Puvanendran, "Could formant frequencies of snore signals be an alternative means for the diagnosis of obstructive sleep apnea?," *Sleep medicine*, vol. 9, pp. 894–8, Dec. 2008.

[4] J. A. Fiz, R. Jané, J. Solà-Soler, J. Abad, M. A. García, and J. Morera, "Continuous analysis and monitoring of snores and their relationship to the apnea-hypopnea index.," *The Laryngoscope*, vol. 120, pp. 854–62, Apr. 2010.

[5] J. Solà-Soler, R. Jané, J. A. Fiz, and J. Morera, "Automatic classification of subjects with and without sleep apnea through snoring analysis.," *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2007, pp. 6094–7, Jan. 2007.

[6] H. Nakano, M. Hayashi, E. Ohshima, N. Nishikata, and T. Shinohara, "Validation of a new system of tracheal sound analysis for the diagnosis of sleep apnea-hypopnea syndrome.," *Sleep*, vol. 27, pp. 951–7, Aug. 2004.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[8] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[9] M. Riedmiller, "Rprop - Description and Implementation Details," tech. rep., 1994.

[10] H. Lee, Y. Largman, P. Pham, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in neural information . . .* , pp. 1–9, 2009.

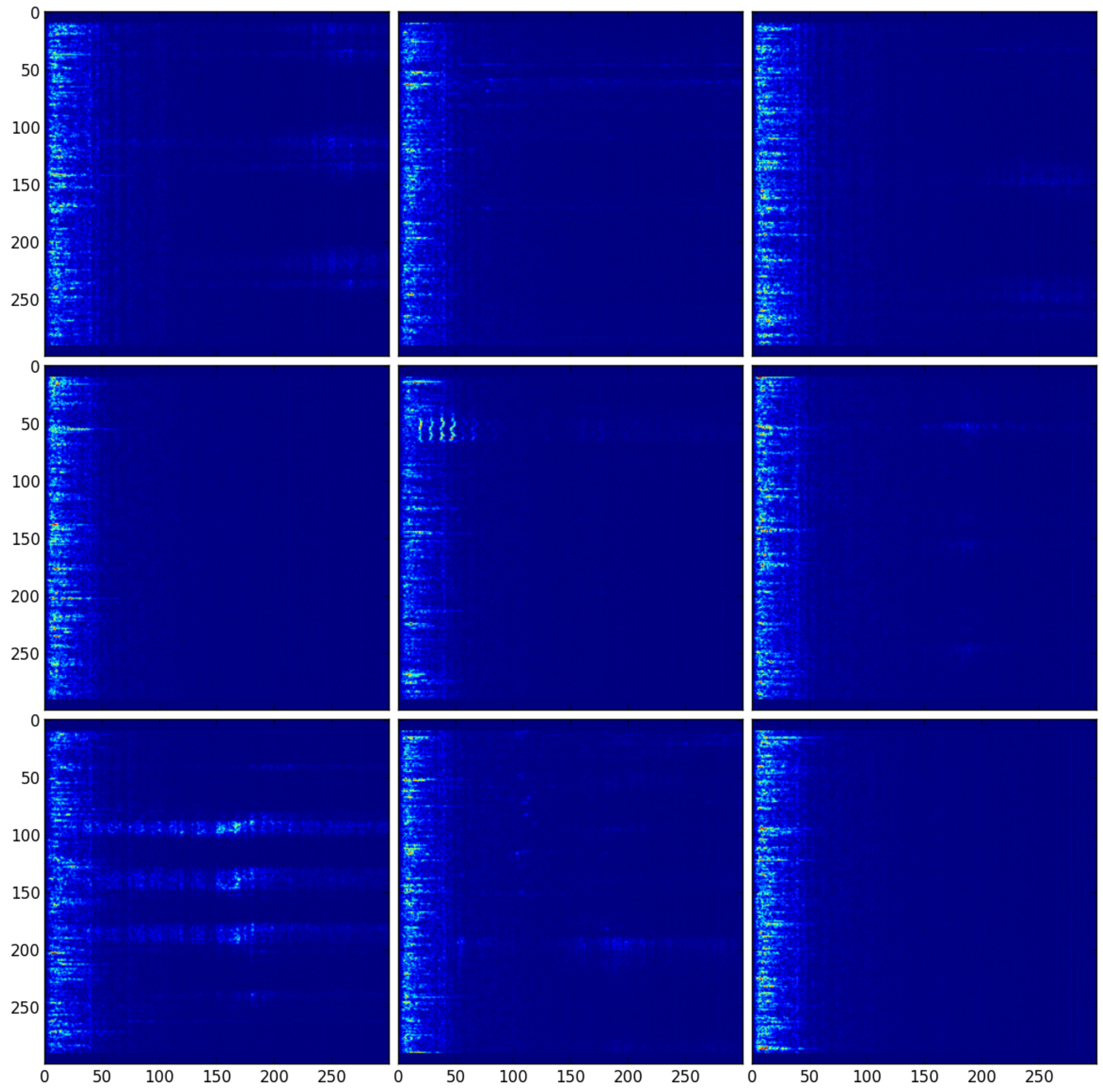

Fig. 2. One randomly chosen example from each class (except CH, which was not present in the data). Starting from the upper left and proceeding by rows, the classes are: no event, RERA, OAa, CAa, OHa, CHa, OA, CA, OH. The horizontal axis is frequency. The vertical axis is time.