INTERSPEECH 2017: Show & Tell Contribution
August 20–24, 2017, Stockholm, Sweden

Evolving recurrent neural networks that process and classify raw audio in a streaming fashion

Adrien Daniel

NXP Semiconductors, Sophia Antipolis, France
adrien.daniel@nxp.com

Abstract

The paper describes a neuroevolution-based novel approach to train recurrent neural networks that can process and classify audio directly from the raw waveform signal, without any assumption on the signal itself, on the features that should be extracted, or on the required network topology to perform the task. Resulting networks are relatively small in memory size, and their usage in a streaming fashion makes them particularly suited to embedded real-time applications.

Index Terms: machine learning, recurrent neural network, neuroevolution, evolutionary algorithm, audio processing, audio classification

1. Introduction

This work is inspired by two main observations from conventional machine learning and signal processing techniques when applied to the audio domain. First, machine learning techniques, for most of them, require to transform raw inputs to a customized lower-dimensional feature domain. For audio, Mel-frequency cepstral coefficients (MFCCs)—initially designed for speech analysis—are among the most popular features. Some techniques such as convolutional neural networks (CNNs) can work directly on lower-level features like spectrograms. Still, besides discarding phase information, they usually require further decreasing the spectral resolution to limit the network complexity. Hence, if not properly designed for the task, and because they discard a significant amount of information, handcrafted features may limit the discrimination power of machine learning techniques. Another related issue is well illustrated by the speaker authentication problem. Obviously, it is not achievable to design a separate set of features for each enrolled speaker in order to capture their own voice characteristics. However, projecting all speakers onto the same reduced feature space is likely to induce overlap between areas spanned by each of them.

Second, audio analysis and processing typically require questionable choices. Time-frequency representations—usually based on an overlap-and-add (OLA) scheme using Fourier analysis—require a careful setting of the size, shape, and overlap amount of the analysis windows and rely on independence and stationarity hypotheses which may induce artifacts. Neural networks usually have an *a priori* fixed topology, and may use special units like long short-term memory (LSTM) units to improve their performance. All these arbitrary choices potentially limit the performances the system may reach.

This paper proposes a novel approach to evolve recurrent neural networks trained to process and classify audio: (1) directly from the raw time-domain signal, (2) in a streaming fashion, and (3) with no arbitrary fixed topology. As a result, this approach circumvents the aforementioned limitations. It relies on neuroevolution techniques described in next section.

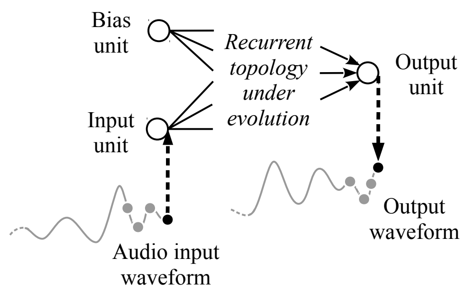

Figure 1: *Proposed generic ANN setup for audio processing.*

2. Neuroevolution

Neuroevolution is a class of machine learning techniques for training artificial neural networks (ANNs) based on evolutionary algorithms. Typically, a neuroevolution algorithm starts, at generation 0, with a population of randomly generated ANNs. At each generation, all ANNs of the current population are evaluated according to criteria defined through a *fitness* function. The best-performing individuals are selected for producing next generation's population by crossover (*i.e.* breeding) and mutation. This process stops when an individual of the current population reaches a predefined level of performance.

NeuroEvolution of Augmenting Topologies (NEAT), proposed by Stanley and Miikkulainen [1], belongs to the category of TWEANNs (Topology and Weight Evolving Artificial Neural Networks). It relies on a *complexifying* principle: initial minimal topologies are incrementally complexified over generations, while optimizing weights. To allow crossover among different topologies, it applies a gene-tracking mechanism using history markers. Finally, to preserve innovation, the population is organized in species that do not compete with each other before reaching a certain level of maturity. NEAT requires the definition of: (1) the ANN usage (input/output setup and propagation scheme), and (2) a fitness function. These are detailed next, in an approach tailored to timeseries signals like audio.

3. Proposed approach

3.1. Network usage

The proposed setup for audio processing is depicted in Figure 1. Let us assume that the task at hand involves processing one input waveform of length N to obtain one output waveform of same length. The idea is to consider audio processing as a control task. ANNs are setup with one bias unit (constantly outputting 1.0), one input unit that will successively output each of the values of the input waveform, and one output unit that will successively output each of the values of the output wave-

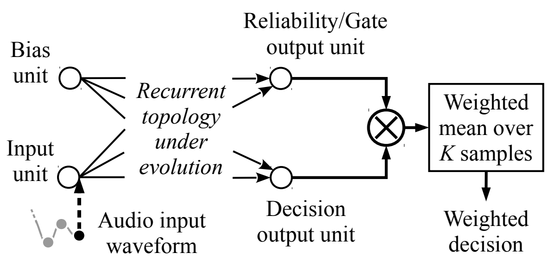
Reliability/Gate output unit
Bias unit
Recurrent topology under evolution
Input unit
Weighted mean over K samples
Audio input waveform
Decision output unit
Weighted decision

Figure 2: *Proposed ANN setup for audio classification.*

form. Let $x[i]$ and $y[i]$ be the i^{th} sample of the input and output waveforms, respectively. ANN usage is:

1. Start with pointer $i = 0$
2. Place $x[i]$ as the input unit's output
3. Perform one activation step of the ANN
4. Store the output unit's output as $y[i]$
5. If $i < N - 1$, increase i by one and go to step 2

For multichannel processing tasks, the number of inputs and outputs can be extended to the desired dimensions. Because addition of recurrent connections is allowed during the evolution process, the streaming nature of this setup, where samples are propagated one-by-one into the network, can be interpreted as a *non-linear recurrent time-domain* filter.

This scheme can be directly applied to audio classification tasks as well, reading the output stream as decision scores. However, a specific setup is proposed as depicted on Figure 2, assuming a binary classification task. ANNs now have two outputs: the optimization process will be driven to evolve a *reliability* output that reflects how reliable the current *decision* output is. To do so, let $y_d[i]$ be the decision output and $y_r[i]$ be the reliability output for input sample $x[i]$. A weighted decision $y_w[i]$ is computed over a sliding window of K samples:

$$y_w[i] = \frac{\sum_{j=0}^{K-1} y_d[i-j] \times y_r[i-j]}{\sum_{j=0}^{K-1} y_r[i-j]}. \tag{1}$$

For tasks involving more than two classes, the number of decision outputs can be extended to the number of classes to detect. Optionally, the reliability output can be transformed into a *gate* output, by thresholding its output value: $y_r[i] > \tau \mapsto 1$, otherwise 0, where τ is the threshold parameter. In this particular case, Eq. 1 acts as a mean over output decisions that are selected by the output gate. These classification setups allow the optimization process to evolve behaviors that dynamically adapt the output rate to react to specific parts of the input stream (*e.g.*, highly discriminative features of a target class) while ignoring others (*e.g.*, silence, low signal-to-noise ratio).

3.2. Fitness function

During evaluation, each ANN of the current population is fed the same input x to obtain, depending on the task, either an audio waveform y, or weighted decisions y_w. A generic fitness function is used for comparison with a groundtruth signal g:

$$F(y, g) = \frac{1}{1 + \sum_{i=0}^{N-1} (g[i] - y_{(w)}[i])^2}. \tag{2}$$

This function reaches 1 when y and g are equal and tends to 0 as they diverge. Note that for audio processing tasks, translating this function to the spectral domain—by means of a short-term Fourier Transform (STFT) for instance—leads to better results.

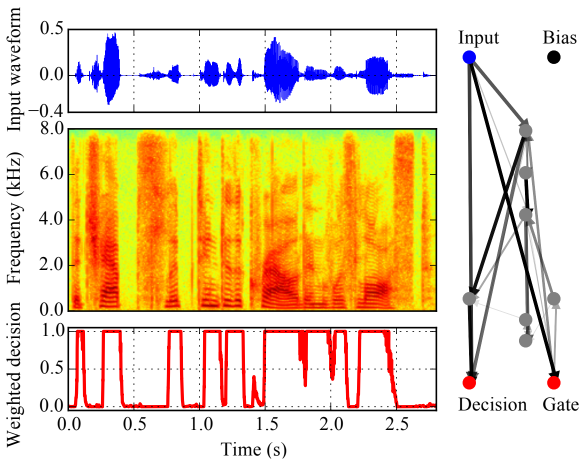

Figure 3: *Left: Example of output for voicing detection on speaker and sentence not seen during training: "The chap slipped into the crowd and was lost". The network was trained on 28 minutes of speech (6 female and 6 male speakers), with $K = 160$ (10 ms), and $\tau = 0$. All units have ReLU activation functions, except the two output units with identity activations. Right: Evolved topology (11 units, 22 connections).*

3.3. Example and demonstrations

As a simple but illustrative example, Figure 3 shows the topology and output of a network evolved to detect voicing in speech. Other applications, including speech/music discrimination and speaker authentication, will be demonstrated at the conference.

4. Discussion and conclusions

This paper briefly describes a novel approach to evolve recurrent neural networks for audio processing and classification with notable characteristics. First, taking raw audio as an input, no feature handcrafting is necessary, this being left up to the learning algorithm. For classification tasks, this aspect is essential for the system to automatically learn the most discriminative features. Taking again the speaker authentication example, evolving a network to recognize a target speaker against a set of background speakers leaves the optimization process free to catch the discriminative characteristics of this target voice.

Second, its sample-by-sample streaming nature is well suited to real-time applications. It avoids arbitrary choices and potential artifacts inherent to time-frequency representations. NEAT's complexifying principle allows to discover relatively simple but nonetheless efficient recurrent topologies. As an example, evolved ANNs for a speaker authentication task require only a few hundred connections, which is several orders of magnitude below that of deep neural networks like CNNs. From a practical standpoint, such low memory requirements are well adapted to embedded applications.

Extended studies of this approach including other applications and comparison with state-of-the-art methods are under preparation.

5. References

[1] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.