Few-Shot Distribution Learning for Music Generation

Hugo Larochelle
Google Brain
hugolarochelle@google.com

Chelsea Finn
UC Berkeley
cbfinn@eecs.berkeley.edu

Sachin Ravi
Princeton University
sachinr@princeton.edu

Abstract

Few-shot distribution learning refers to the problem of learning a generative model in the few-shot learning regime. We propose to investigate this problem in the context of generating music data, such as lyrics or MIDI sequences, using ideas from recent developments in adaptive language models, few-shot learning and meta-learning. We plan to collect and construct benchmarks for this problem and evaluate various solutions.

1 Introduction

There is currently a large gap between humans and machines in their respective ability to learn from few examples, referred here as the problem of few-shot learning. This makes few-shot learning an important subject for AI research, and recently there has been promising progress in the specific context of few-shot classification [16, 10, 13, 4, 9, 7, 14]

A less explored problem is that of few-shot distribution learning. In this context, we must design or learn a procedure which, given a small number of samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, from some distribution $p(\mathbf{x})$, will produce a good estimator of $p(\mathbf{x})$, as evaluated on unseen samples $\{\mathbf{x}'_1, \mathbf{x}'_2, ..., \mathbf{x}'_m\}$ from that distribution. In this project, we propose to explore the setting where observations $\mathbf{x}_t$ correspond to music sequences, such as lyrics (sequences of words) or MIDI formated songs (sequences of notes). More fundamentally, this problem allows us to study approaches for a machine to quickly learn a generative model of new objects or concepts.

2 Related Work

Lake et al. [7] popularized the problem of learning a generative model from few examples using ideas from probabilistic programming. Later, Rezende et al. [12] set out to explore a deep neural network approach to this problem, motivated by the ability for this approach to more easily scale and not require as much prior knowledge on the form of $p(\mathbf{x})$. Edwards and Storkey [3] also proposed a deep variational model that learns to extracts statistics that characterize the distribution of a potentially small set of samples. However [12, 3] don't provide a likelihood evaluation for few-shot distribution learning, relying mostly instead on a subjective evaluation of sample quality. Moreover, they mostly rely on the Omniglot dataset [7] as a benchmark for few-shot learning, which corresponds to small handwritten symbols. There thus seems room for further pushing this line of research towards more complex distributions and more systematic quantitative benchmarking.

The closest work to this project is probably that of Bartunov and Vetrov [2] and Reed et al. [11]. The former lay out an episode-based framework similar to what we are proposing to follow here, as well as a deep variational model inspired by Vinyals et al. [16] for few-shot distribution learning, focusing also on the Omniglot dataset. The later explores more challenging datasets of natural images and explores a variety of ways to perform meta-learnign with an autoregressive generative model of images. We propose a similar line of inquiry, but focused on the music domain.

This work also relates to research on adaptive language models, i.e. models that, at test time, adapt their estimate of the conditional disttribution of future words based on the words processed so far during evaluation. In the context of autoregressive neural networks, Grave et al. [5] proposed a neural version of cache-based language models, by augmenting a neural language model with a memory of recent hidden layer values from previous time steps. Matching the current hidden layer with that memory can then be used to identify similar contexts and change the distribution of the next word based on the words seen after these matching contexts. Krause et al. [6] also recently studied the use of so-called dynamic evaluation [8], where a language model is updated at evaluation time using an online gradient descent step on the negative log-probability of processed words.

3 Few-Shot Distribution Learning

We propose two approaches for adapting a generative model to the few-shot distribution learning task. In both cases, models are (meta-)trained based on episodes that correspond to pairs of a training (also called support) set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and a corresponding test set $\{\mathbf{x}'_1, \mathbf{x}'_2, ..., \mathbf{x}'_m\}$.

We take inspiration from the recent literature on few-shot classification and meta-learning in general[1], from which two general paradigms for tackling a few-shot learning problem can be observed:

- *Few-shot learning by jointly training an initialization and fine-tuning procedure* [10, 4]: In this context, one would pick a particular distribution estimator family to represent $p(\mathbf{x})$ (e.g. an autoregressive LSTM language model) and learn its initialization as well as how to perform a few steps of updates from this initialization based on the samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$.
- *Few-shot learning by training a generic sequential neural architecture*: Here, there are no separation into an estimator family and an adaptive procedure to produce its parameters. There is simply a sequential model (e.g. an RNN with memory [13, 5] or a temporal convolution network [9]) that sequentially runs over $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and then can compute $p(\mathbf{x}_m|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ (e.g. by conditioning an output LSTM language model).

4 Experiments

We propose to focus the experiments of this paper on data from the music domain. One motivation is that this type of data hasn't been explored as much as in previous work on few-shot distributional learning, which has focused more on images instead.

But more importantly, we believe that successful few-shot learning for music data could consitute a much more natural way for artists to use an ML-based assistant for composing music. Indeed, instead of having to train their own models on various types of music in order to influence the music being generated, they could simply change the music data provided as the support set and immediately obtain a generative model for that type of music. We think this sort of short-loop interaction between the artist and the ML system is likely to be more appealing.

4.1 Datasets

Part of this project will require collecting various types of music datasets. We propose constructing 2 benchmarks, for 2 types of music song data: lyrics and songs in MIDI format. We also require that meta-data be collected about these songs, so as to allow to generate episodes. For example, if each song is labeled by its artist, then episodes can be generated by randomly choosing an artist and then randomly splitting his/her songs into a training (support) and test set. In this example, we would train to adapt to a specific artist's distribution (i.e. his/her style). Artists would be split into meta-training, meta-validation and meta-test sets of episodes (much like in few-shot classification, meta-training, meta-validation and meta-test sets use non-overlapping sets of classes).

[1] See `http://bair.berkeley.edu/blog/2017/07/18/learning-to-learn/` for a recent overview

The metric for performance evaluation would be average negative log-probabilites $-\log p(\mathbf{x}'_t)$ (or an approximate estimate if untractable) of all samples in the episodes' test sets.

4.2 Baselines

We propose to treat the following methods as baselines, that we would expect the methods of the next section to improve uppon:

- Training a regular LSTM language model on all songs in the meta-training set. At evaluation time, ignore each episode's training (support) set and evaluate on its test set. This is thus a model that is non-adaptive. This baseline is meant to make sure that adaptive methods are indeed better.
- Training a regular LSTM language model on sequences corresponding to the simple concatenation of all sequences in the training set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ followed by the concatenation of the test set sequences $\{\mathbf{x}'_1, \mathbf{x}'_2, ..., \mathbf{x}'_m\}$.

4.3 Proposed methods

There is certainly a lot of room to innovate in terms of methods. We provide here two more specific proposals, but imagine new ones could emerge later.

4.3.1 Joint training of an initialization and fine-tuning procedure

In this paradigm, there are two main dimensions to explore

- *Choice of the distribution family* $p(\mathbf{x})$: simplest would be a regular LSTM language model.
- *Choice of the finetuning procedure*: We could either use a well known optimizer (e.g. ADAM) that we unroll [4], a meta-LSTM that effectively learns learning rate and weight decay gates [10] or an LSTM that directly outputs the parameter updates [1].

4.3.2 Generic sequential neural architecture

Here, the two main dimensions to explore are:

- *Architecture for processing the training set sequentially to produce some representation of it*: Following the continuous cache model of Grave et al. [5], we could have an LSTM process each song in the episode's training set and store them in a memory. We could also design a Transformer architecture [15] for a set of sequences to obtain the memory.
- *Model for* $p(\mathbf{x}'|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ *conditioning on the training set's representation*: The memory representation could be combined with an LSTM output model as in Grave et al. [5]. The Transformer architecture [15] could also be used.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, 2016.

[2] Sergey Bartunov and Dmitry P Vetrov. Fast adaptation in generative models with generative matching networks. *arXiv preprint arXiv:1612.02192*, 2016.

[3] Harrison Edwards and Amos Storkey. Towards a neural statistician. *ICLR*, 2017.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 2017.

[5] Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. *ICLR*, 2017.

[6] Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. Dynamic evaluation of neural sequence models. *arXiv preprint arXiv:1709.07432*, 2017.

[7] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[8] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*.

[9] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Meta-learning with temporal convolutions. *arXiv preprint arXiv:1707.03141*, 2017.

[10] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*.

[11] Scott Reed, Yutian Chen, Thomas Paine, Aäron van den Oord, S. M. Ali Eslami, Danilo Rezende, Oriol Vinyals, and Nando de Freitas. Matching networks for one shot learning. *arXiv preprint arXiv/1710.10304*, 2017.

[12] Danilo Rezende, Ivo Danihelka, Karol Gregor, Daan Wierstra, et al. One-shot generalization in deep generative models. In *ICML*, 2016.

[13] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.

[14] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[16] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. One-shot generalization in deep generative models. In *NIPS*, 2016.