Towards DNN-Human real-time music improvisation

Christodoulos Benetatos
Department of Electrical and Computer Engineering

Abstract

- In this project we explore the idea of human performers interacting real-time with neural network, to produce music. The last few years, we have seen a lot of research in generating music sequences using neural networks. However most of them, either generate music in offline fashion (i.e bidirectional RNN's), or in online, but in solo configuration, without incorporating humans in the generation chain. In similar systems to ours, like AI Duets [1], or the "Continuator" [2], a human performer and a computer are interacting, however not at the same time, but in a call and response configuration.
- We create a dataset of 2-voice musical pieces, and we train two different architectures to predict the next note for a voice, given the past of both voices, in an online fashion.
- The results are encouraging, and the computation time of both architectures is sufficient for implementing a real time application.

Dataset

- Since we are not aware of any dataset of 2-voice pieces (duets), where each voice is monophonic, we generated duets from Bach's Chorales, by taking all the 2-voice combinations of each chorale.

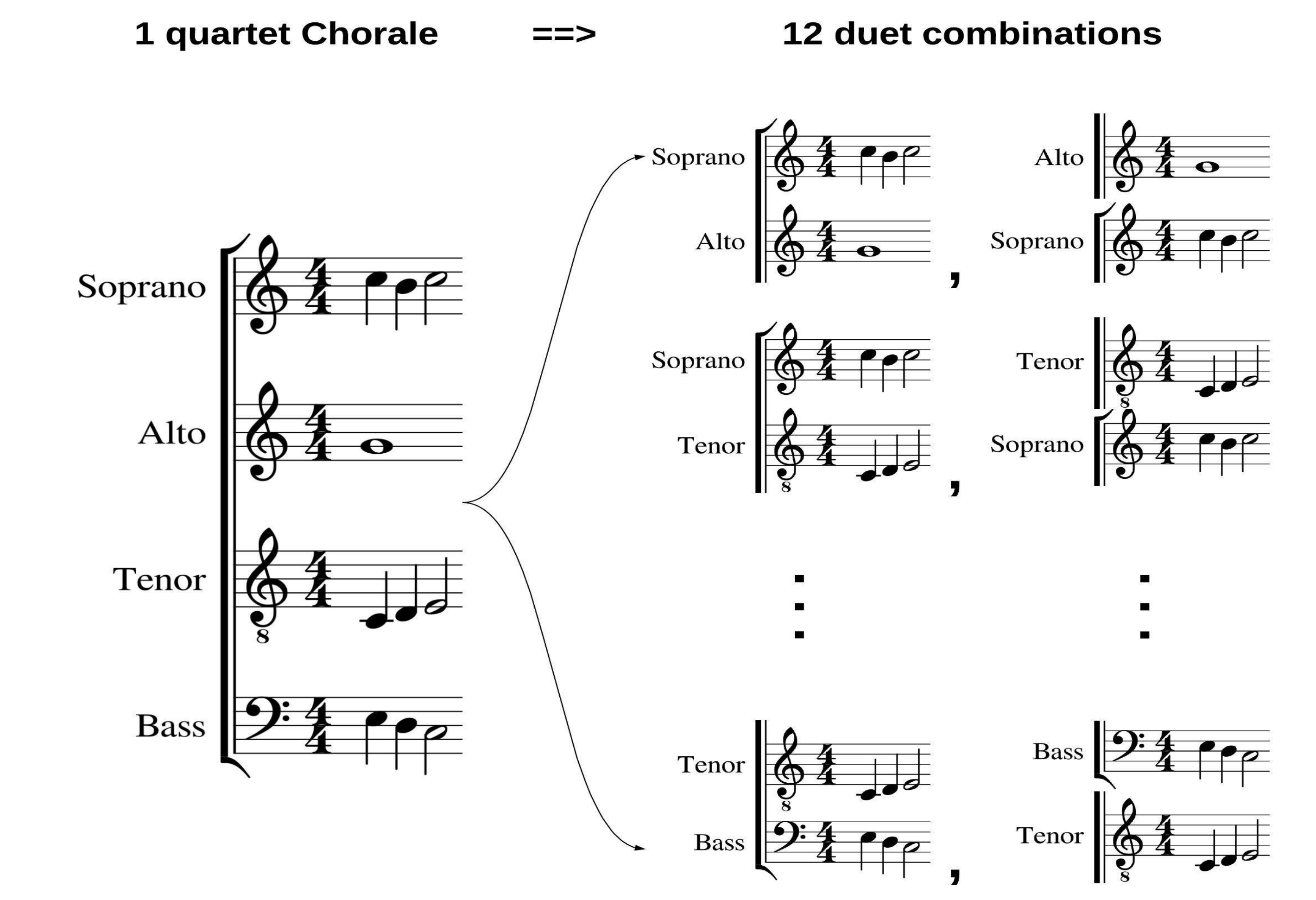

Fig. 1 : We created a dataset of duets by considering all the 12 permutations of the 4 voices in a chorale.

- We used 346 from the 371 chorales in the music21 corpus, and we also transposed each of them in all 12 musical keys, resulting a total number of 346*12*12 = 49824 duets.

Data Representation

- For the note representation we use 1/16th time quantization and MIDI numbers for pitch information.
- There are two possible articulations for each note, "hit" or "hold". For notes with duration greater than 1/16th, we assign the first 1/16 as "hit" and the rest as "hold. The most common way, that is found in the bibliography, is to encode all "hold" notes for all pitches with the same symbol i.e "1" (Fig. 2 top line). However this approach results very unbalanced data, and the system tends to predict mostly "1".

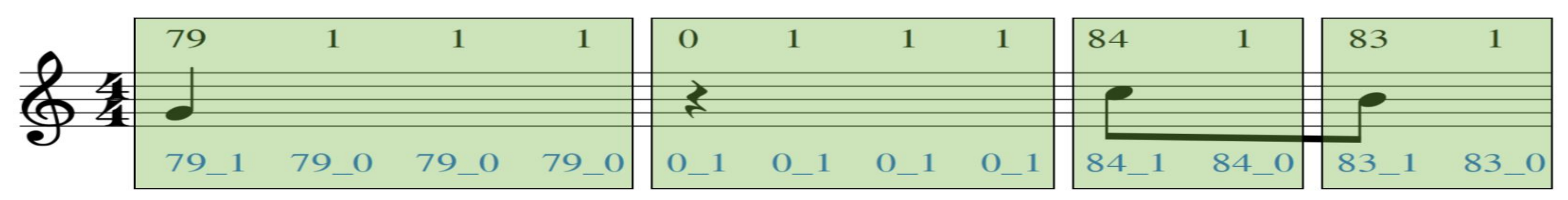

Fig. 2 : For "hold", we use new symbols that contain also the pitch information of the note that is "holded".

To overcome this problem, we used "hold" symbols, that also include the pitch information, namely which note is "holded". The form of this symbol is "MIDI_articulation" (Fig. 2 bottom line).

Model Architecture

We can see the overall structure of the system in Fig. 3. We use a sliding window over the previous notes of both voices, and the DNN predicts the next one for its corresponding voice. The other voice is the melody generated by the human performer. We vary the size of the sliding window from 16 to 64 depending on the type of DNN

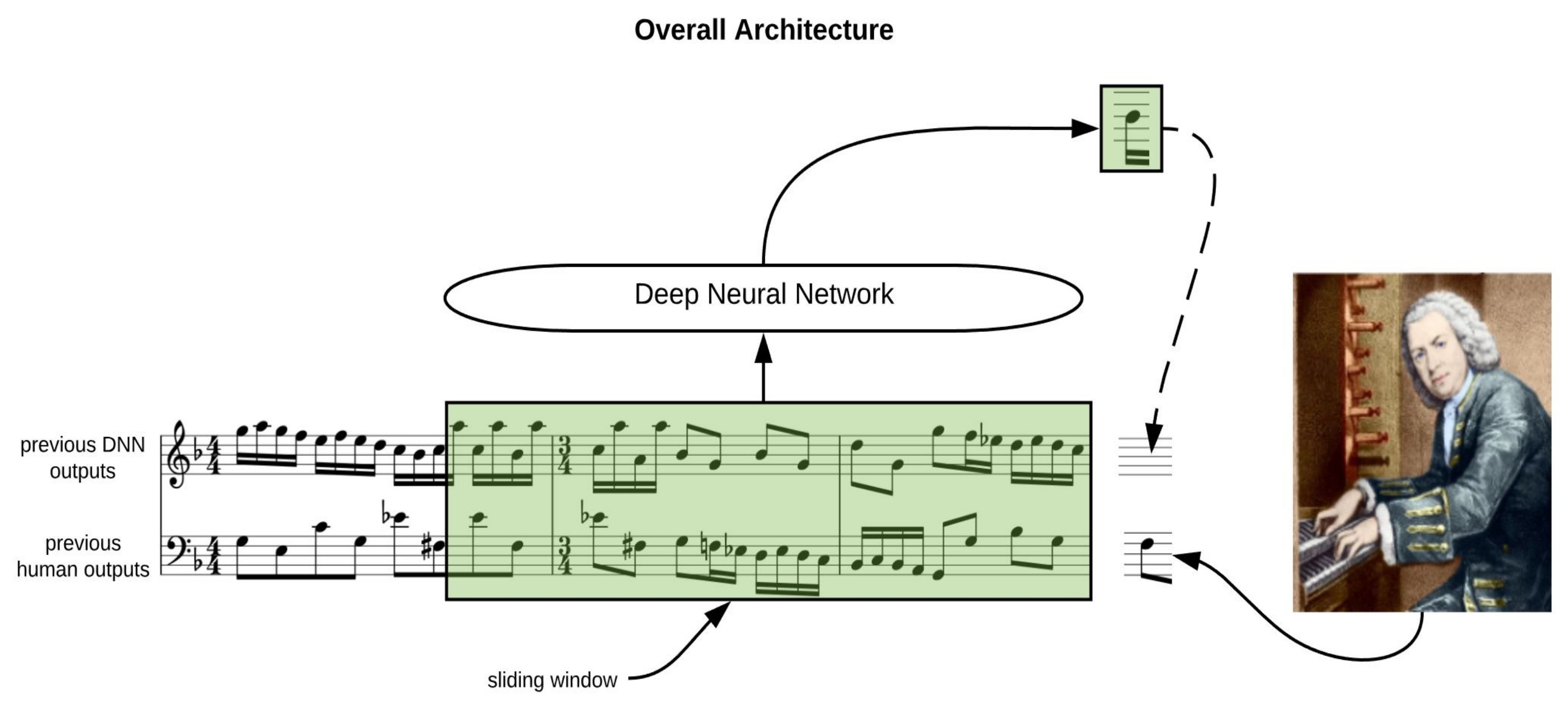

Fig. 3: General idea of the system, DNN and human performer in the same generation chain, affecting each other.

- LSTM

We used a 2 stacked layers architecture, sliding window length 16, and a dense layer for the last LSTM cell in each window. Instead of just using *teacher forcing*, we experiment with various scheduled sampling schemas [3].

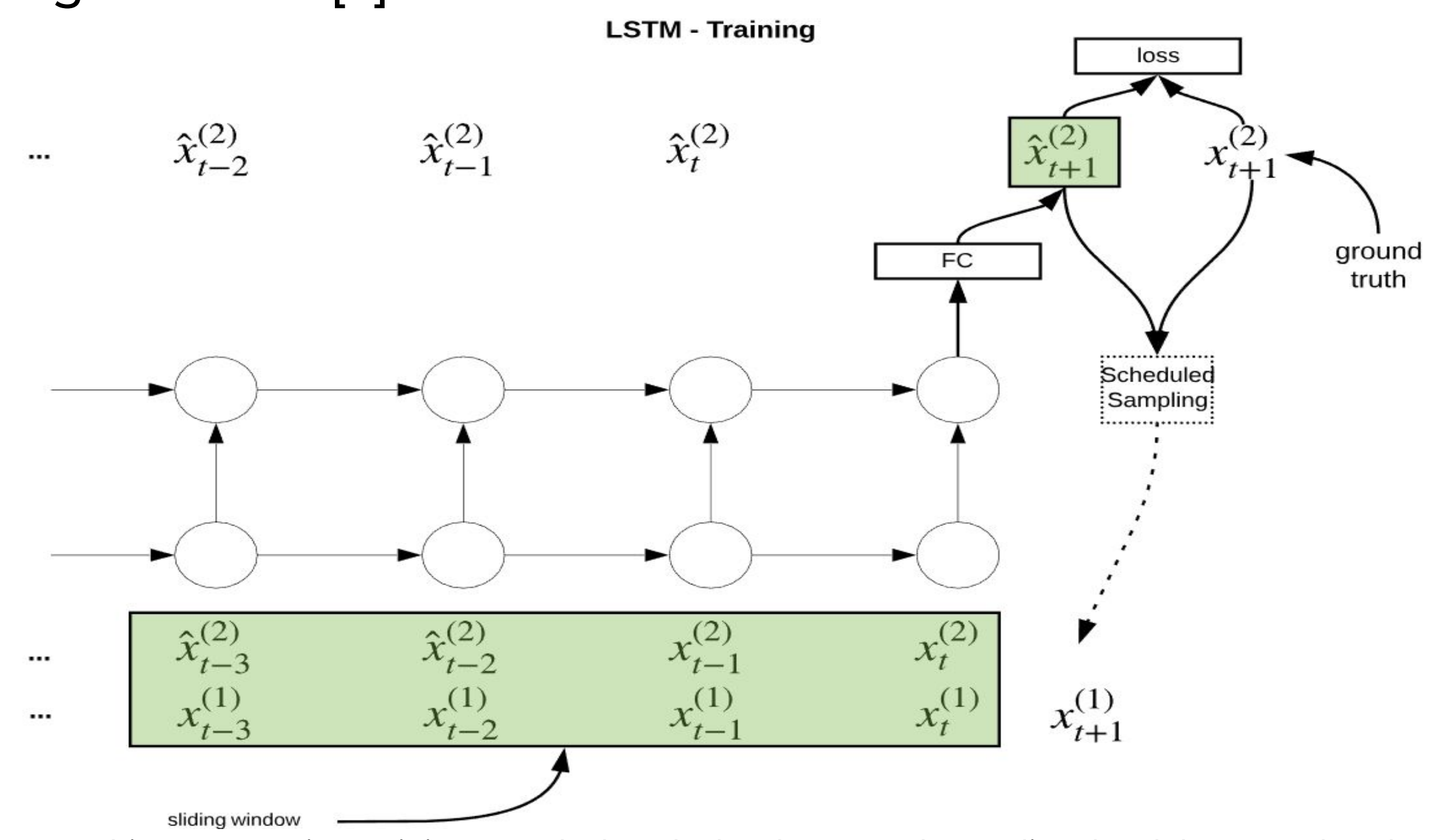
LSTM - Training
loss
FC
ground truth
Scheduled Sampling
sliding window
$\hat{x}^{(2)}_{t-2}$ $\hat{x}^{(2)}_{t-1}$ $\hat{x}^{(2)}_{t}$ $\hat{x}^{(2)}_{t+1}$ $x^{(2)}_{t+1}$
$x^{(2)}_{t-3}$ $x^{(2)}_{t-2}$ $x^{(2)}_{t-1}$ $x^{(2)}_{t}$
$x^{(1)}_{t-3}$ $x^{(1)}_{t-2}$ $x^{(1)}_{t-1}$ $x^{(1)}_{t}$ $x^{(1)}_{t+1}$

Fig. 4 : LSTM architecture. During training we calculate the loss between the predicted and the ground truth note, but instead of pushing the ground truth to the window (teacher forcing), we push the predicted with a varied probability (scheduled sampling)

- Causal Dilated Convolution Network (cd-CNN)

These networks are found in the core of WaveNet architecture [4]. Here, we do not use zero padding, in order to reduce the length of the sequence after each layer. We assume that the input sequence length is $N = 2^i$, then the number of layers will be $i - 1$. We use exponential increased dilation factor for the first $i - 2$ layers, and dilation of 1 for the last. Because CNNs are faster to train, we could use sliding window length up to 64.

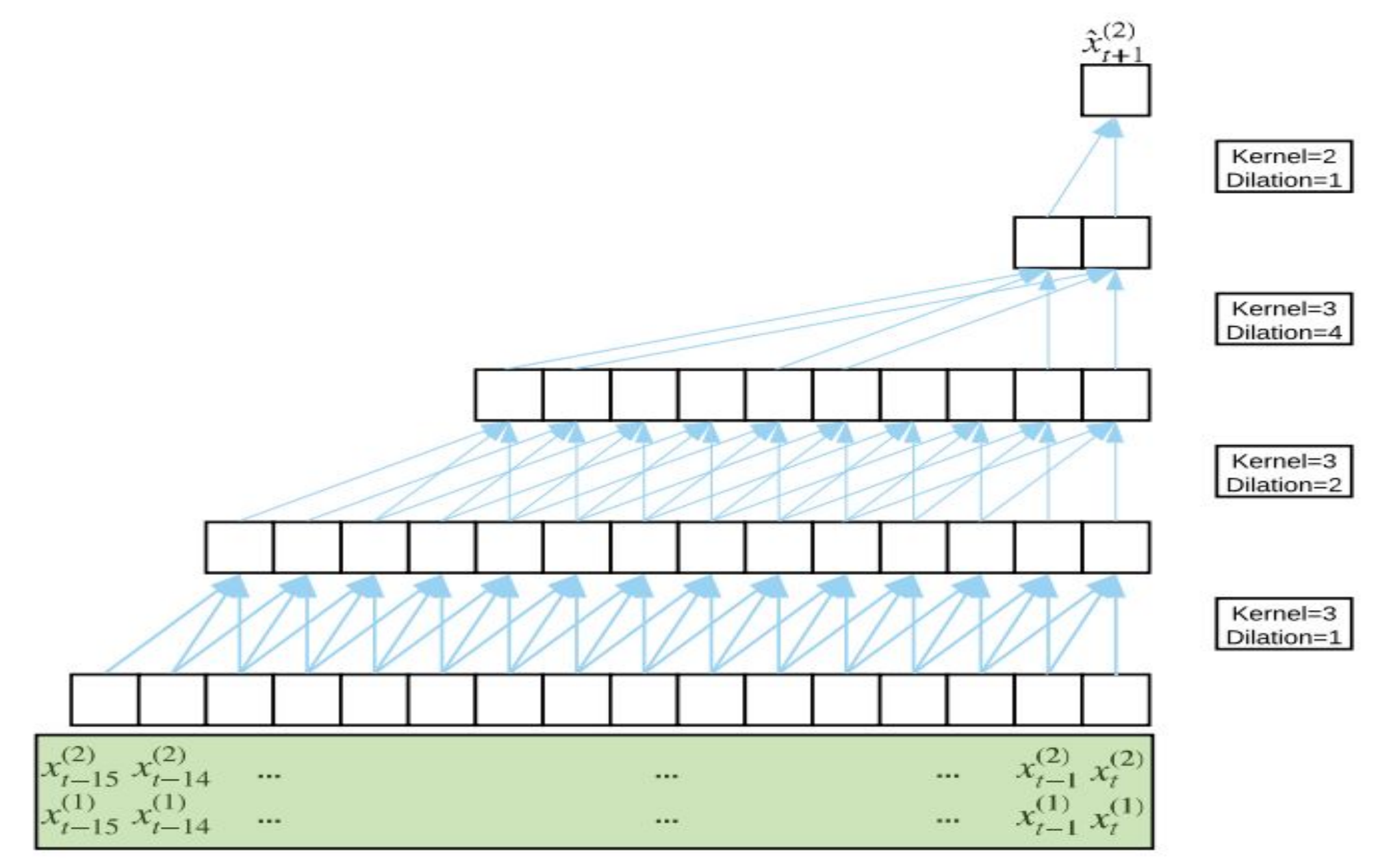
$\hat{x}^{(2)}_{t+1}$
Kernel=2 Dilation=1
Kernel=3 Dilation=4
Kernel=3 Dilation=2
Kernel=3 Dilation=1
$x^{(2)}_{t-15}$ $x^{(2)}_{t-14}$ ... $x^{(2)}_{t-1}$ $x^{(2)}_{t}$
$x^{(1)}_{t-15}$ $x^{(1)}_{t-14}$ ... $x^{(1)}_{t-1}$ $x^{(1)}_{t}$

Fig. 5 : cd-CNN, in this figure, N=16 -> i=4, so we have 3 layers of exponential increased dilations, and 1 at the top, to get the single predicted token-note.

Results - Future Work

- Both architectures give similar results regarding the loss and the accuracy over training, validation and test dataset. Using cd-CNN with $N = 32$, we achieve the best accuracy in test dataset, 91.3% versus 88.9% of the LSTM. In Fig. 6a are the training results of cd-CNN.
- In Fig. 6b, you can see some overfitting problems we experienced with LSTM, even after using dropout.
- More experiments should be done with scheduled sampling, since we had difficulties to stabilize the training.
- Forward calculation time for a 1/16th note was under 10ms for both NNs, meaning that we can design a real time application and achieve high BPM speeds

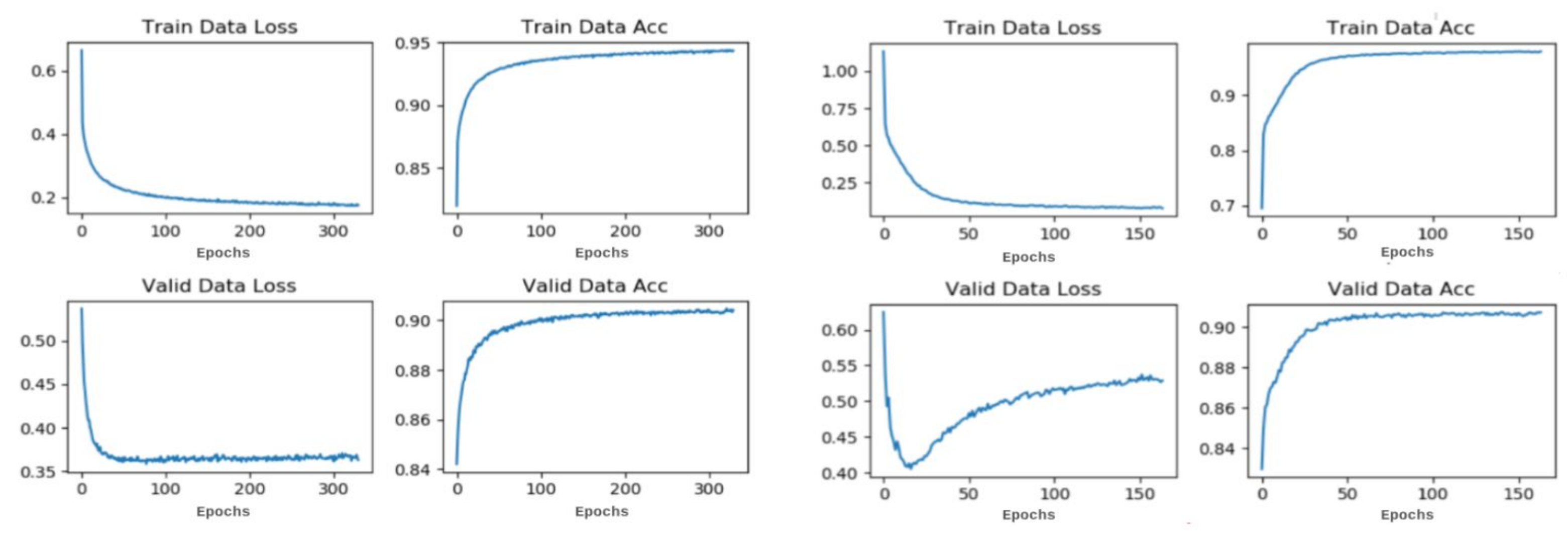

Fig. 6 : Training loss and accuracy for train and validation dataset for a) cd-CNN, N=32 and b) LSTM, N=16

- In Fig. 7, we can see an excerpt from the generated result. The reference melody (black notes) is the Soprano part of a chorale from the test dataset. We can see that the NN learned to generate music in the right key, in different range from the given melody, and following basic music theory rules.

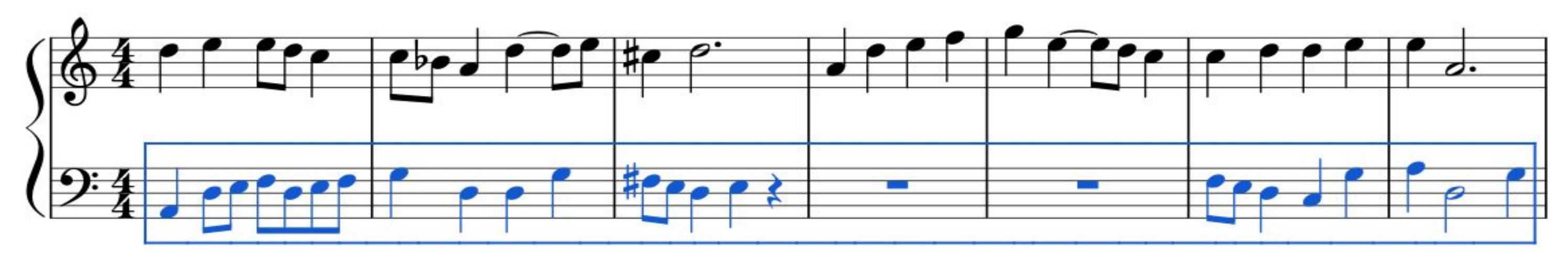

Fig. 7 : Excerpt from chorale BWV 658. The top voice is the Soprano from the original piece, while the bottom, highlighted in blue, is the generated from the cd-CNN, with N=32

References

[1] Yotam Mann, "AI Duet", https://experiments.withgoogle.com/ai-duet, May 2017
[2] Pachet, Francois. "The continuator: Musical interaction with style." *Journal of New Music Research* 32.3 (2003): 333-341.
[3] Bengio, Samy, et al. "Scheduled sampling for sequence prediction with recurrent neural networks." *Advances in Neural Information Processing Systems*. 2015.
[4] Van Den Oord, Aäron, et al. "WaveNet: A generative model for raw audio." *SSW*. 2016.