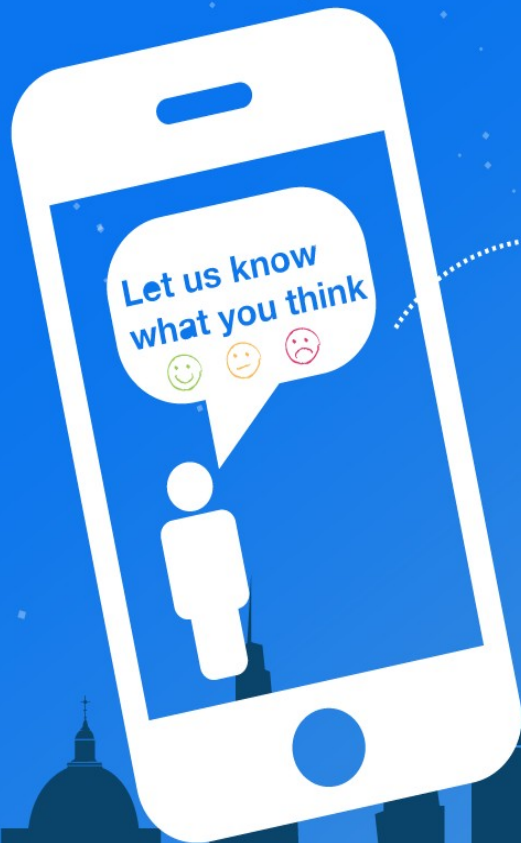




**Click 'Rate Session'
to rate session
and ask questions.**



Enjoying the conference
background music?



BachBot
BachBot.com



1 Sample-1

▶ 2,244



2 Happy Birthday

▶ 2,091



3 Twinkle Twinkle Little Star

▶ 6,524



4 Sample-2

▶ 1,304



5 Sample-3

▶ 800

View 8 tracks



Like



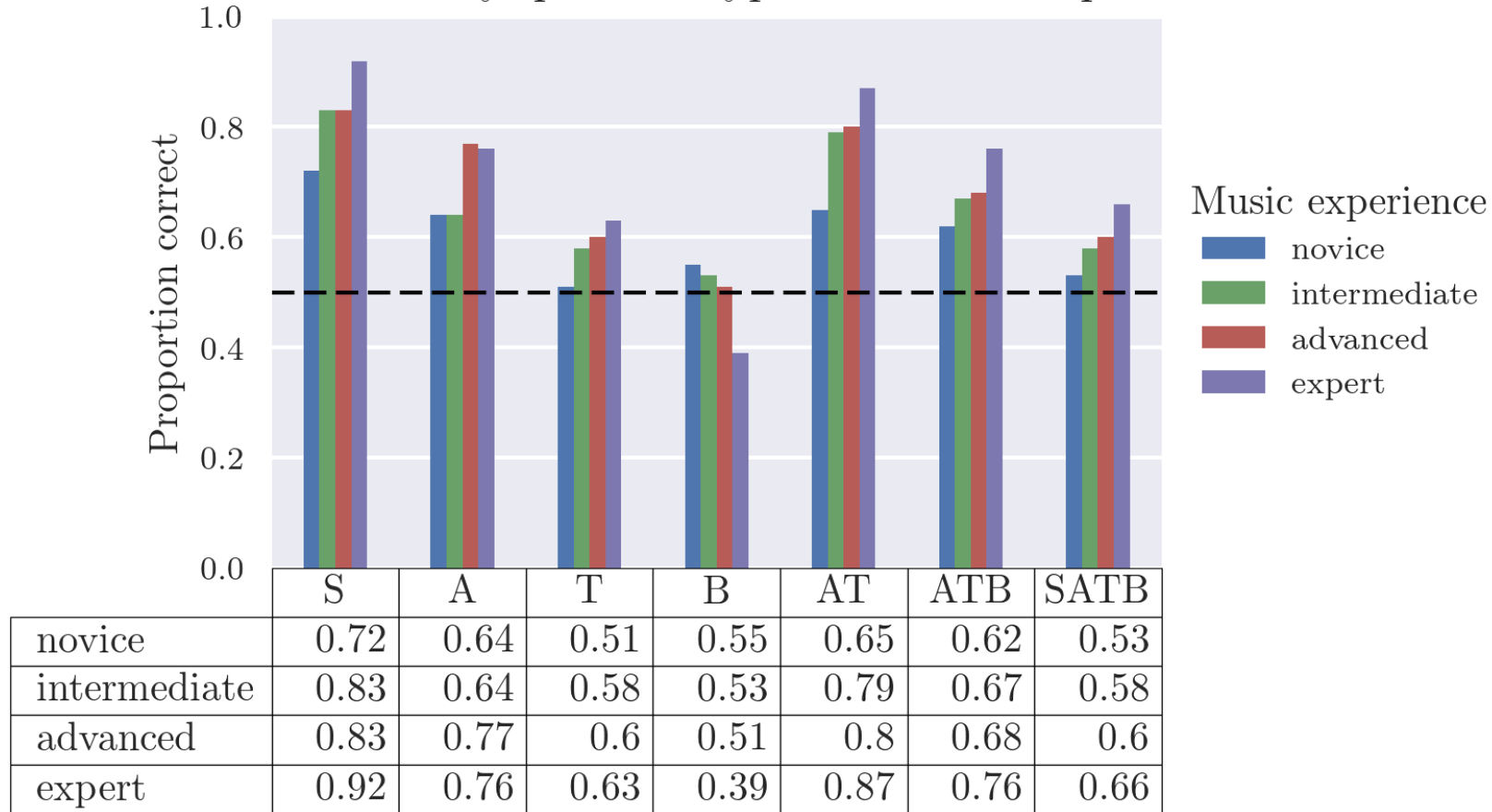
Share



More

<https://soundcloud.com/bachbot>

Performance by question type and music experience



Results from musical Turing test at bachbot.com

BachBot

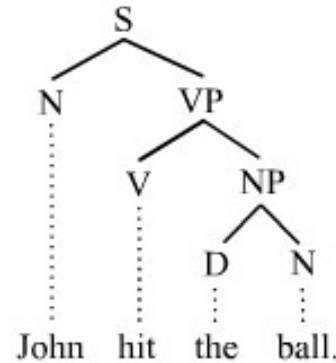
Composing Bach Chorales Using Deep Learning

Feynman Liang

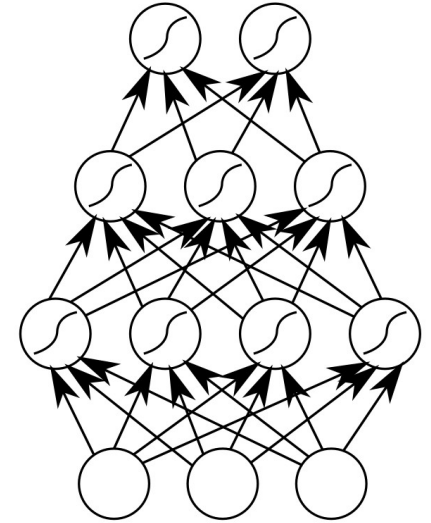
GOTO Copenhagen, 18 November 2019

The big questions

- Where is the frontier of computational creativity?
- How much has deep learning advanced automatic composition?
- How do we evaluate generative models?



vs



TL;DR

- Deep recurrent neural network model for music capable of polyphonic automatic stylistic composition and harmonization
- Learns music theory without prior knowledge
- In a musical Turing test with 1779 participants, performance is only 5% better than random chance

Open source; try it for yourself!

feynmanliang.github.io/bachbot-slides

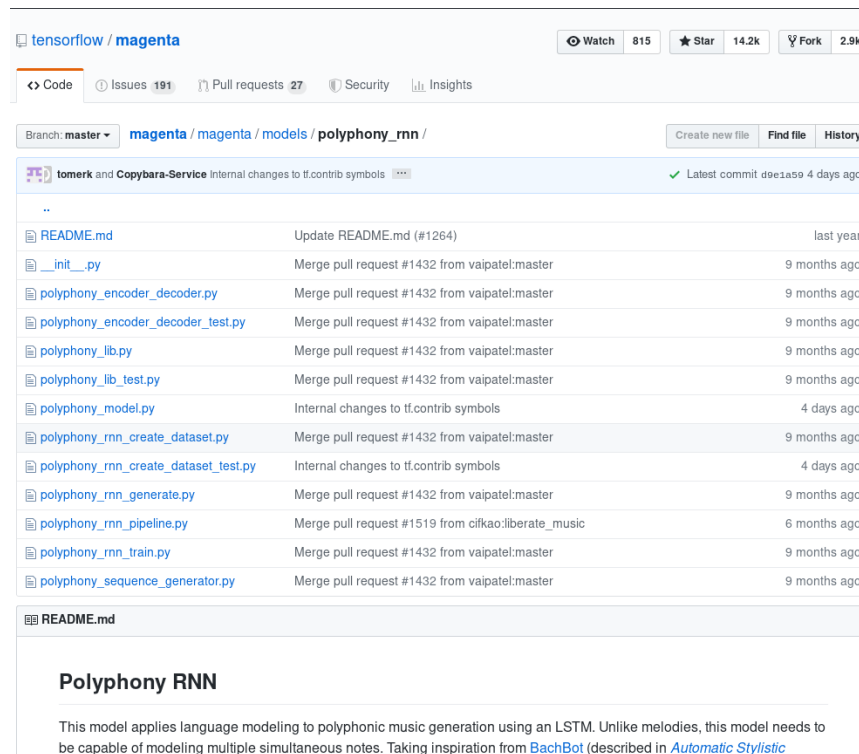
BUILD YOUR OWN BACHBOT!

- Install docker from docker.com
- Pull the image:

```
docker pull fliang/bachbot:aibtb
```

- Prepare data, train model:

```
docker run --name bachbot -it fliang/bachbot:cor  
bachbot datasets prepare  
bachbot datasets concatenate_corpus scratch/BWV-  
bachbot make_h5  
bachbot train
```



The screenshot shows the GitHub repository for tensorflow/magenta. The repository has 815 watches, 14.2k stars, and 2.9k forks. The current branch is master. The file list shows various files related to polyphony_rnn, including README.md, __init__.py, polyphony_encoder_decoder.py, polyphony_encoder_decoder_test.py, polyphony_lib.py, polyphony_lib_test.py, polyphony_model.py, polyphony_rnn_create_dataset.py, polyphony_rnn_create_dataset_test.py, polyphony_rnn_generate.py, polyphony_rnn_pipeline.py, polyphony_rnn_train.py, and polyphony_sequence_generator.py. The README.md file is selected, showing the title "Polyphony RNN" and a description: "This model applies language modeling to polyphonic music generation using an LSTM. Unlike melodies, this model needs to be capable of modeling multiple simultaneous notes. Taking inspiration from BachBot (described in Automatic Stylistic".

Overview

- Sequence modelling for music
- Deep recurrent neural networks (RNNs)
- Discoveries and demos

Motivating example for sequence modelling

Sequence modelling of text

The quick brown fox jumps _____

Question: What word comes next?

Sequence modelling of text

The quick brown fox jumps _____

The quick brown fox jumps over

The quick brown fox jumps around

The quick brown fox jumps lackadaisically

Sequence modelling of text

The quick brown fox jumps _____

P(**over** | the quick brown fox jumps) = 75%

P(**around** | the quick brown fox jumps) = 24%

P(**lackadaisically** | the quick brown fox jumps) = 1%

Sequence modelling of text

The quick brown fox jumps _____

$P(\text{over} \mid \text{the quick brown fox jumps}) = 75\%$

$P(\text{around} \mid \text{the quick brown fox jumps}) = 24\%$

$P(\text{lackadaisically} \mid \text{the quick brown fox jumps}) = 1\%$

Question: Any potential problems?

The 2-gram sequence model

The quick brown fox jumps _____

$P(\text{over} \mid \text{jumps}) = 90\%$

$P(\text{around} \mid \text{jumps}) = 8\%$

$P(\text{behind} \mid \text{jumps}) = 2\%$

Generating text using the 2-gram sequence model:

I
I am
am enjoying
enjoying GOTO
GOTO Copenhagen
I am enjoying GOTO Copenhagen

n-gram models trained on Hamlet

- 1-gram
 - To him swallowed confess hear both. Which. Of save on trail for are ay device androte life have
- 2-gram
 - Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Liveking. Follow.
- 4-gram
 - King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;

Music Primer

Soprano

Alto

Tenor

Bass

A musical score for four voices: Soprano, Alto, Tenor, and Bass. The key signature is B-flat major (two flats) and the time signature is 4/4. The Soprano part features a melodic line with a fermata on the eighth measure. The Alto part follows a similar melodic contour. The Tenor part provides harmonic support with a more active line. The Bass part provides a steady harmonic foundation with a prominent root note in the first measure.

Modern music notation



Pitch: how “high” or “low” a note is



Duration: how “long” a note is

Soprano

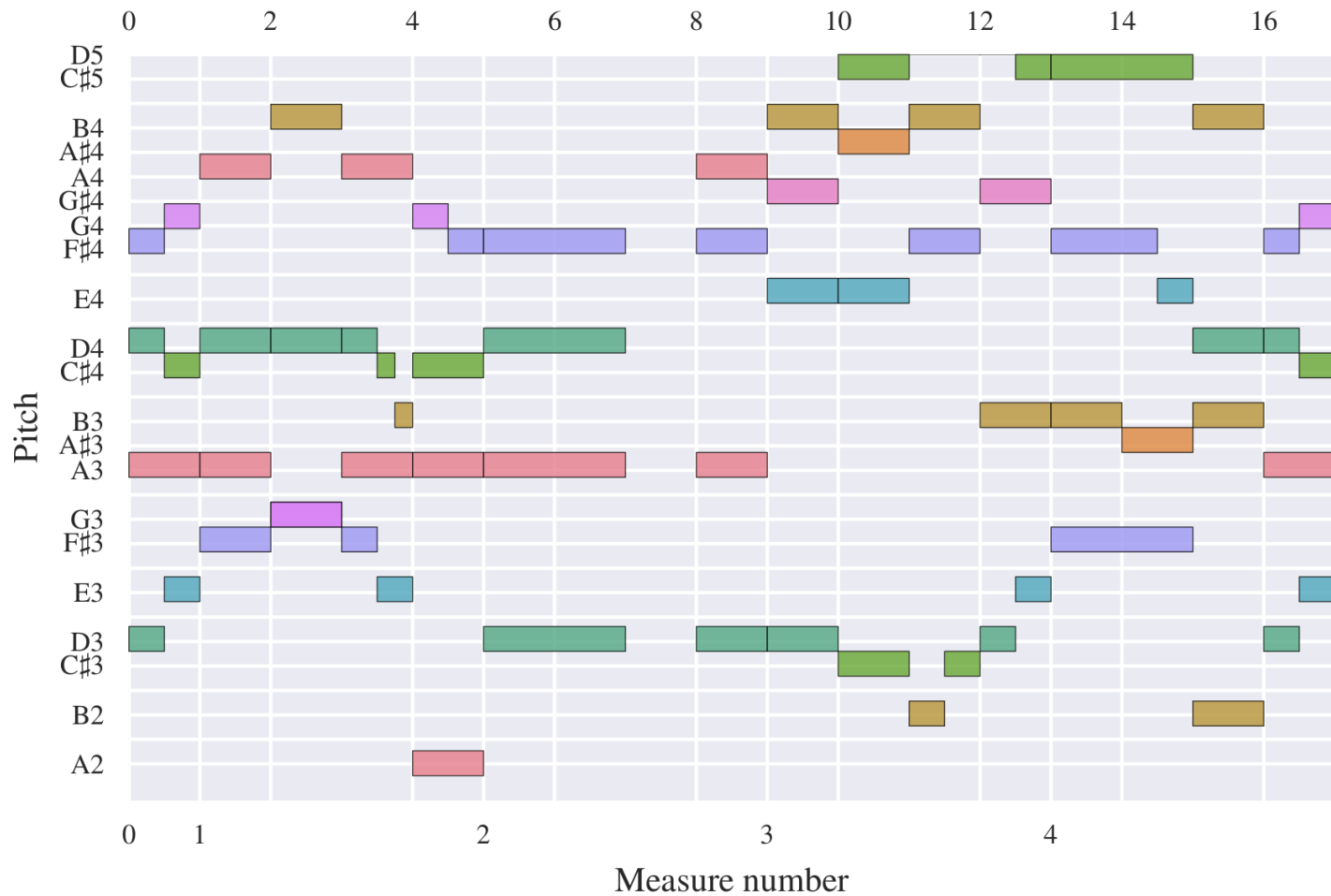
Alto

Tenor

Bass

The image displays a musical score for four voices: Soprano, Alto, Tenor, and Bass. The music is written in 4/4 time and the key of B-flat major, indicated by two flats (B-flat and E-flat) in the key signature. The Soprano part begins with a half note G4, followed by quarter notes A4, B4, and C5, then a half note D5 with a fermata. The Alto part starts with a half note F4, followed by quarter notes G4, A4, and B4, then a half note C5. The Tenor part begins with a half note D3, followed by quarter notes E3, F3, and G3, then a half note A3. The Bass part starts with a half note B2, followed by quarter notes C3, D3, and E3, then a half note F3. The score illustrates polyphony, as multiple voices are singing simultaneously.

Polyphony: multiple simultaneous voices



Piano roll: convenient computational representation

Soprano

Alto

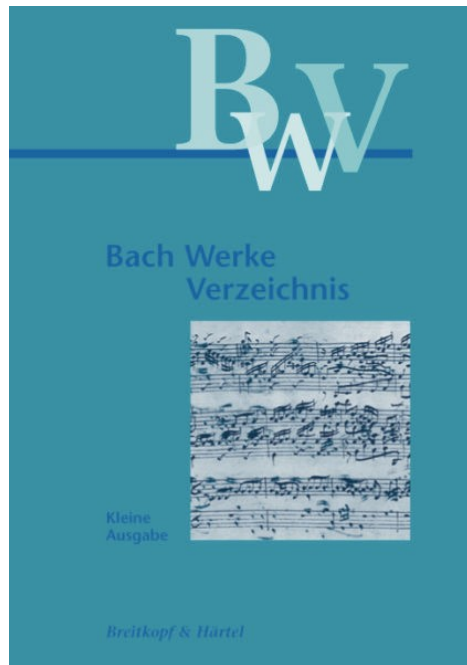
Tenor

Bass

The image displays a musical score for four voices: Soprano, Alto, Tenor, and Bass. The key signature is B-flat major (two flats) and the time signature is 4/4. The Soprano part begins with a treble clef and a key signature of two flats. It features a fermata on the final note of the second measure. The Alto part also uses a treble clef and a key signature of two flats. The Tenor part uses a bass clef and a key signature of two flats. The Bass part uses a bass clef and a key signature of two flats, with a sharp sign on the first line of the staff. The score is written in a standard musical notation style with notes, rests, and a fermata.

Fermatas and phrasing

From Bach Chorales to Sequences



← → ↻ ⓘ web.mit.edu/music21/ ☆

music21: a toolkit for computer-aided musicology

What is music21?

Music21 is a set of tools for helping scholars and other active listeners answer questions about music quickly and simply. If you've ever asked yourself a question like, "I wonder how often Bach does *that*" or "I wish I knew which band was the first to use these chords in this order," or "I'll bet we'd know more about Renaissance counterpoint (or Indian ragas or post-tonal pitch structures or the form of minuets) if I could write a program to automatically write more of them," then music21 can help you with your work.

How simple is music21 to use?

Extremely. After starting Python and typing "from music21 import *" you can do all of these things with only a single line of music21 code:

```
Display a short melody in musical notation:
converter.parse("tinynotation: 3/4 c4 d8 f g16 a g f#").show()

Print the twelve-tone matrix for a tone row (in this case the opening of Schoenberg's Fourth String Quartet):
print (serial.rowToMatrix([2,1,9,10,5,3,4,0,8,7,6,11]) )

or since all the 2nd-Viennese school rows are already available as objects, you can type:
print (serial.getHistoricalRowByName('RowSchoenbergOp37').matrix() )

Convert a file from Humdrum's **kern data format to MusicXML for editing in Finale or Sibelius:
converter.parse('/users/cuthbert/docs/composition.krn').write('musicxml')
```

```
def closedPosition(self):
    """
    returns a new Chord object with :

    >>> chord1 = Chord(["C#4", "G5",
    >>> chord2 = chord1.closedPosition
    >>> print(chord2.lily.value)
    <cis' e' g'>4
    """
    newChord = copy.deepcopy(self)
    tempChordNotes = newChord.pitches
    chordBassPS = self.bass().ps
    for thisPitch in tempChordNotes:
        while thisPitch.ps > chordBassPS:
            thisPitch.octave = thisPitch.octave - 1
    newChord.pitches = tempChordNotes
```

- [Get Started with music21](#)
- [Browse the music21 documentation](#)
- [Download music21](#) from Google Code
- [Get our latest news](#) and updates at the music21
- [Read the Frequently Asked Questions](#) list
- [Sign up](#) for the music21list mailing list at Google Groups.

<http://web.mit.edu/music21/>

Soprano

Alto

Tenor

Bass

4/4

Soprano

Alto

Tenor

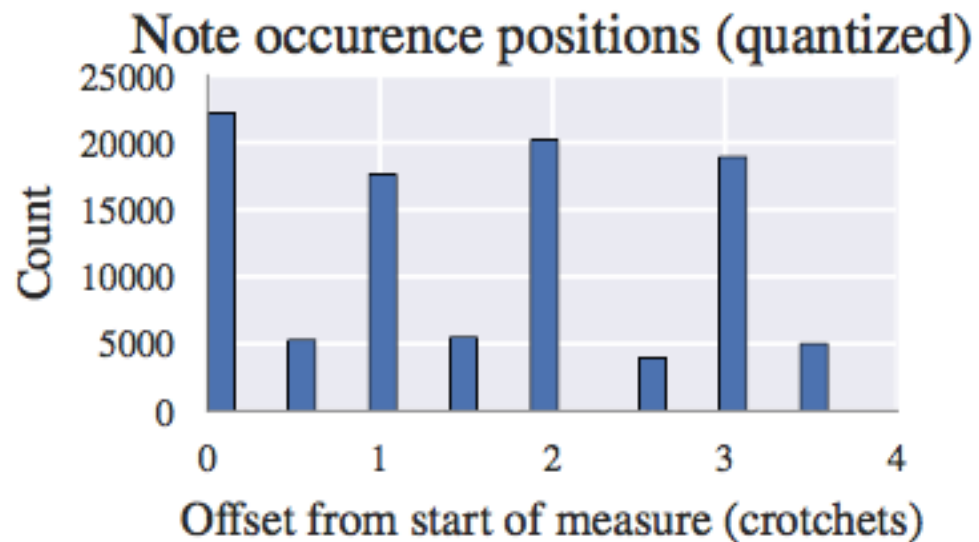
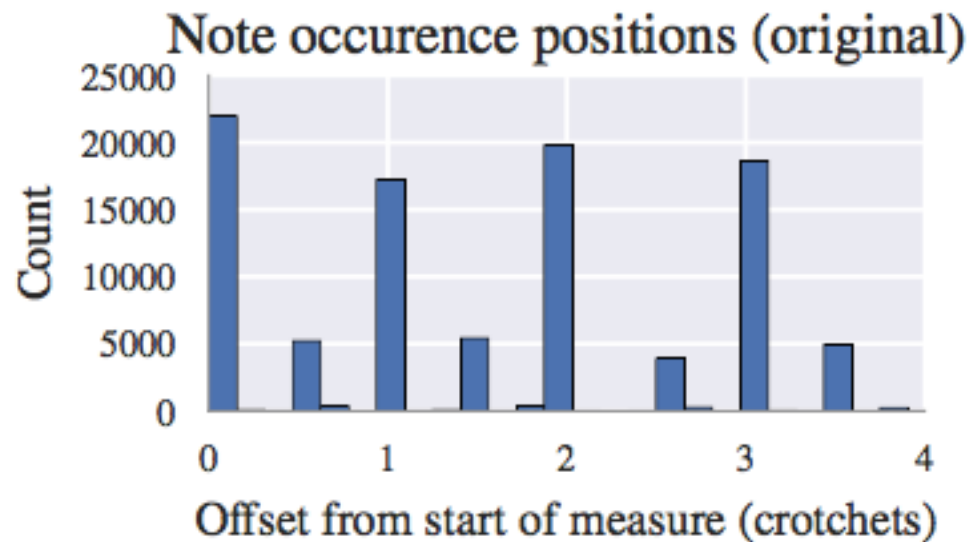
Bass

4/4

Transpose to Cmaj/Amin (convenience)
Quantize to 16th notes (computational)



Transposition preserves relative pitches



Quantization to 16th notes: affects less than 0.2% of dataset

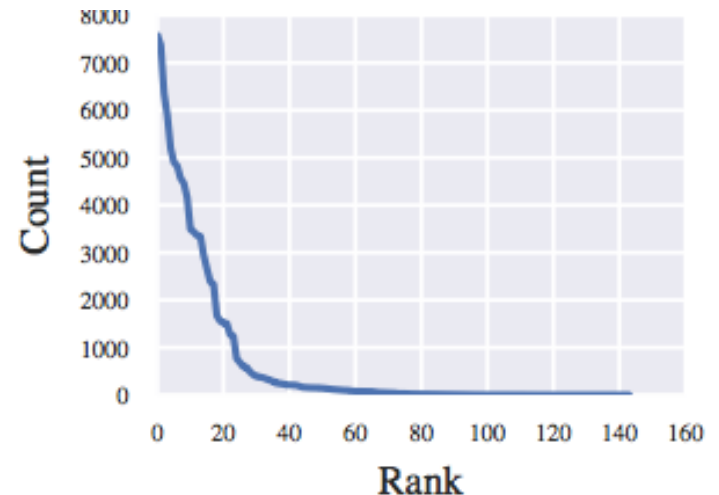
Handling polyphony

Question: How many chords can be constructed from 4 voices, each with 128 pitches?

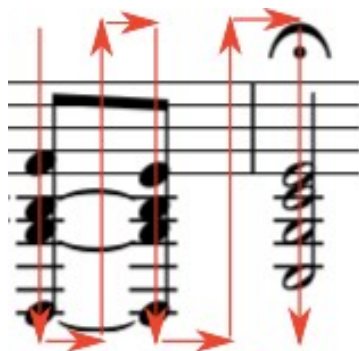
Handling polyphony

Question: How many chords can be constructed from 4 voices, each with 128 pitches?

Answer: $O(128^4)!$ Data sparsity issue...



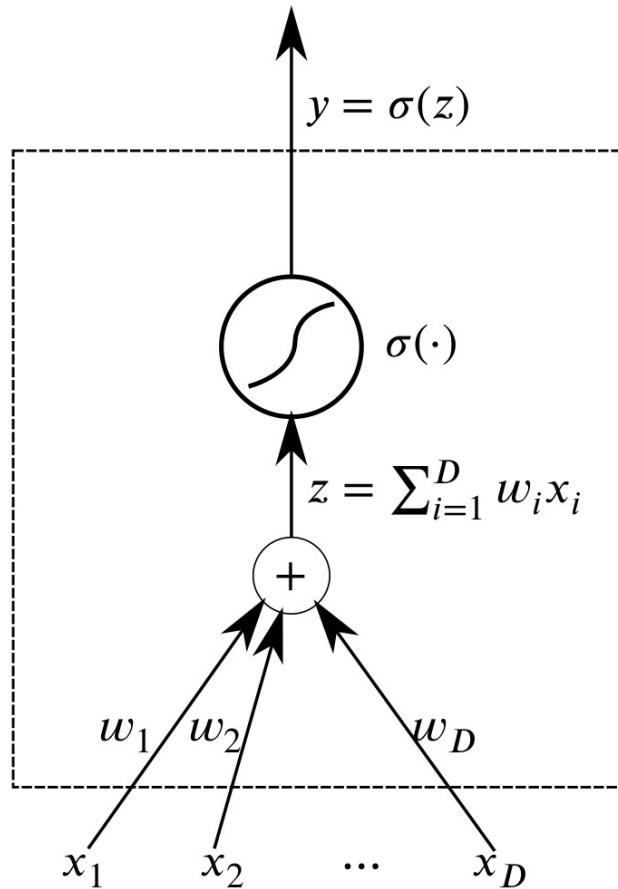
Serialize in SATB order



START	(.)
(62, True)	(60, False)
(56, True)	(57, False)
(52, True)	(52, False)
(47, True)	(50, False)
(60, False)	(.)
(56, True)	(60, True)
(52, True)	(57, True)
(47, True)	(52, True)
	(50, True)

$O(128^4) \Rightarrow O(128)$ vocab. size!

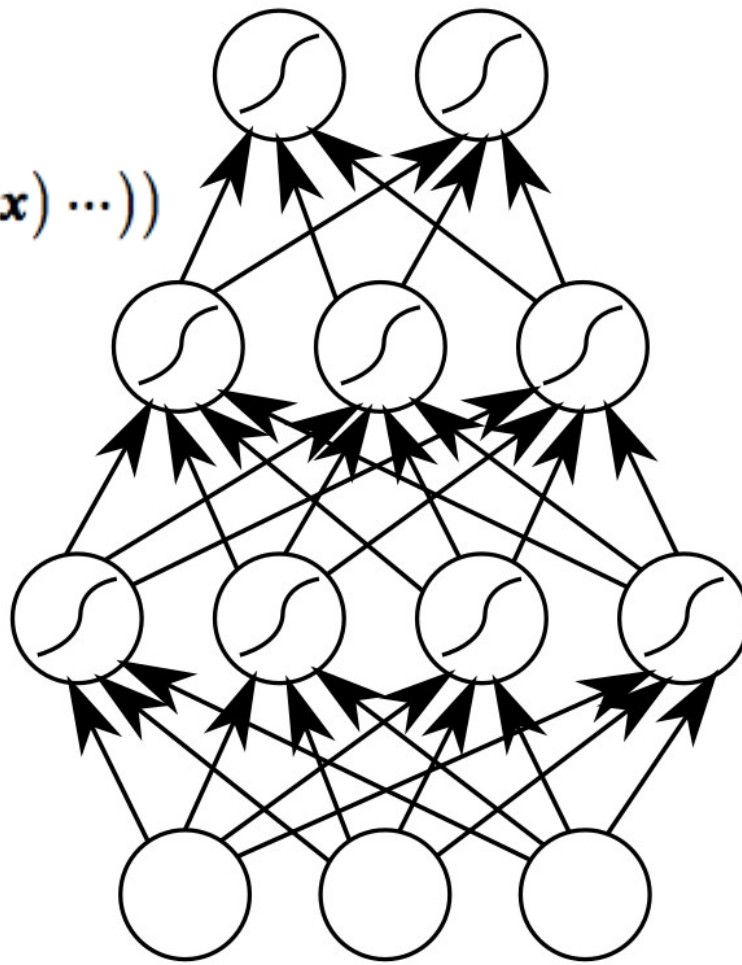
Recurrent neural networks (RNNs)



Neuron

Input **x**, output **y**, parameters **w**, activations **z**

$$\sigma \left(\mathbf{W}_{L,L-1} \sigma \left(\mathbf{W}_{L-1,L-2} \cdots \sigma \left(\mathbf{W}_{2,1} \mathbf{x} \right) \cdots \right) \right)$$



Output Layer

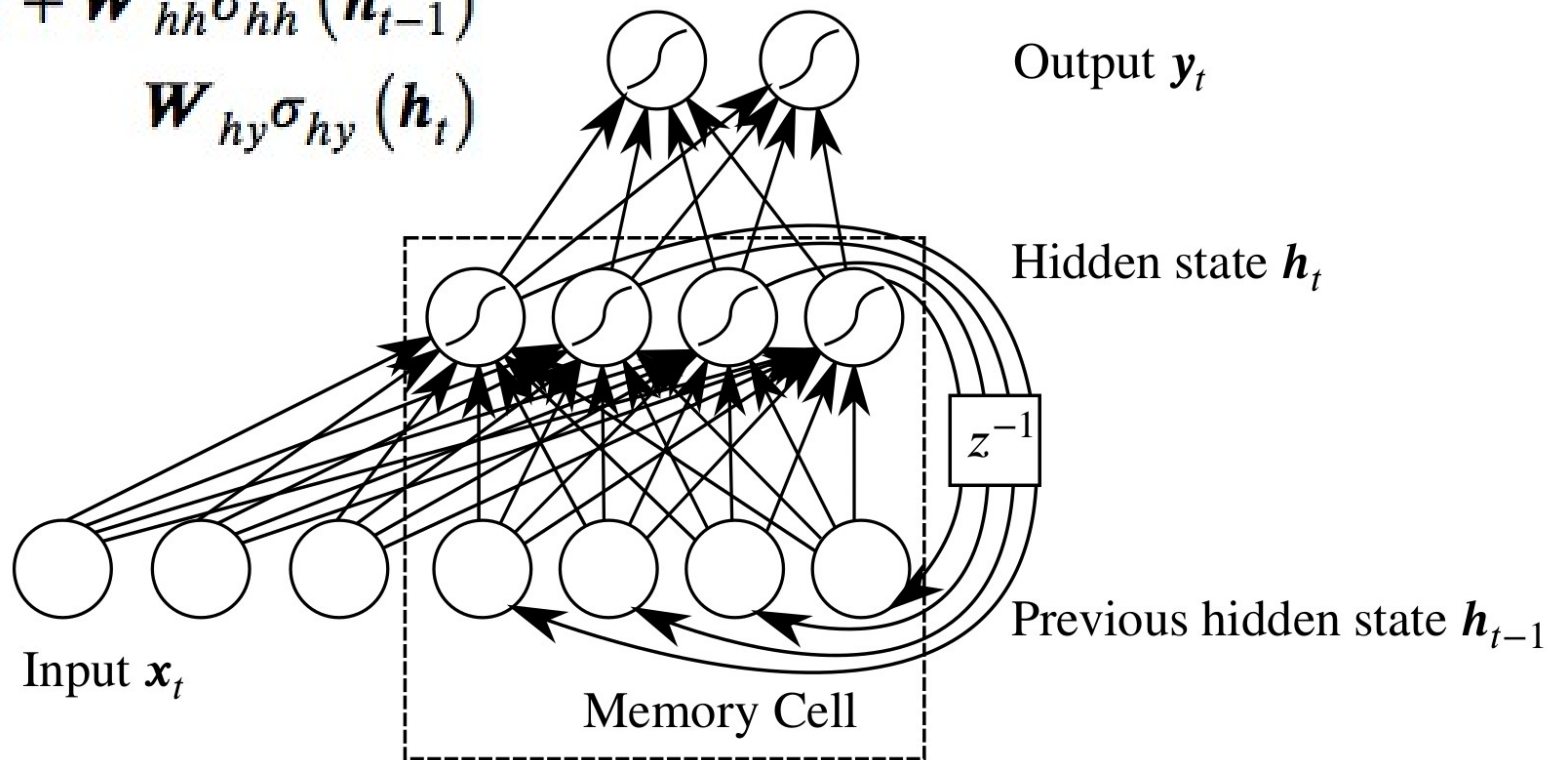
Hidden Layers

Input Layer

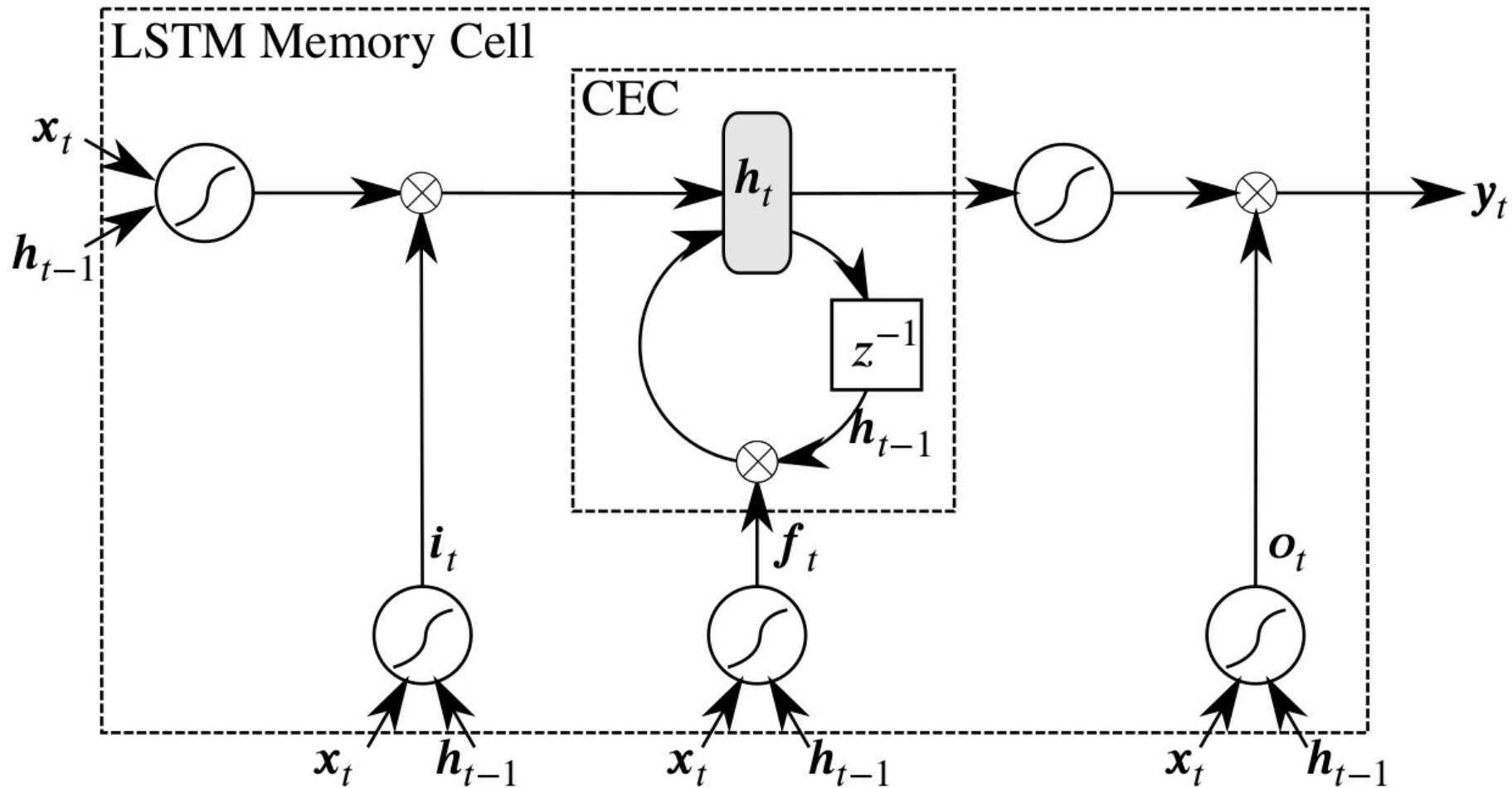
Feedforward neural network

$$\mathbf{h}_t = \mathbf{W}_{xh} \sigma_{xh}(\mathbf{x}_t) + \mathbf{W}_{hh} \sigma_{hh}(\mathbf{h}_{t-1})$$

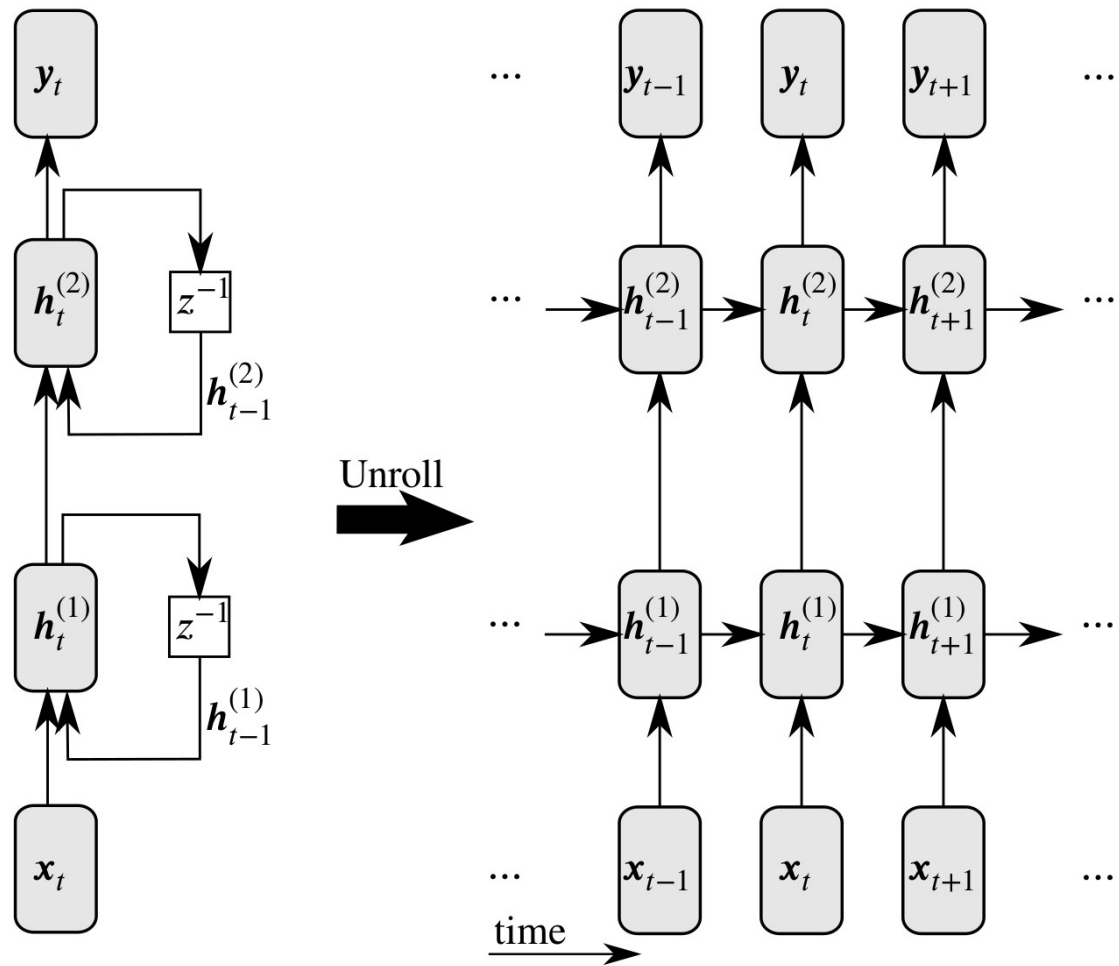
$$\mathbf{y}_t = \mathbf{W}_{hy} \sigma_{hy}(\mathbf{h}_t)$$



Memory cell

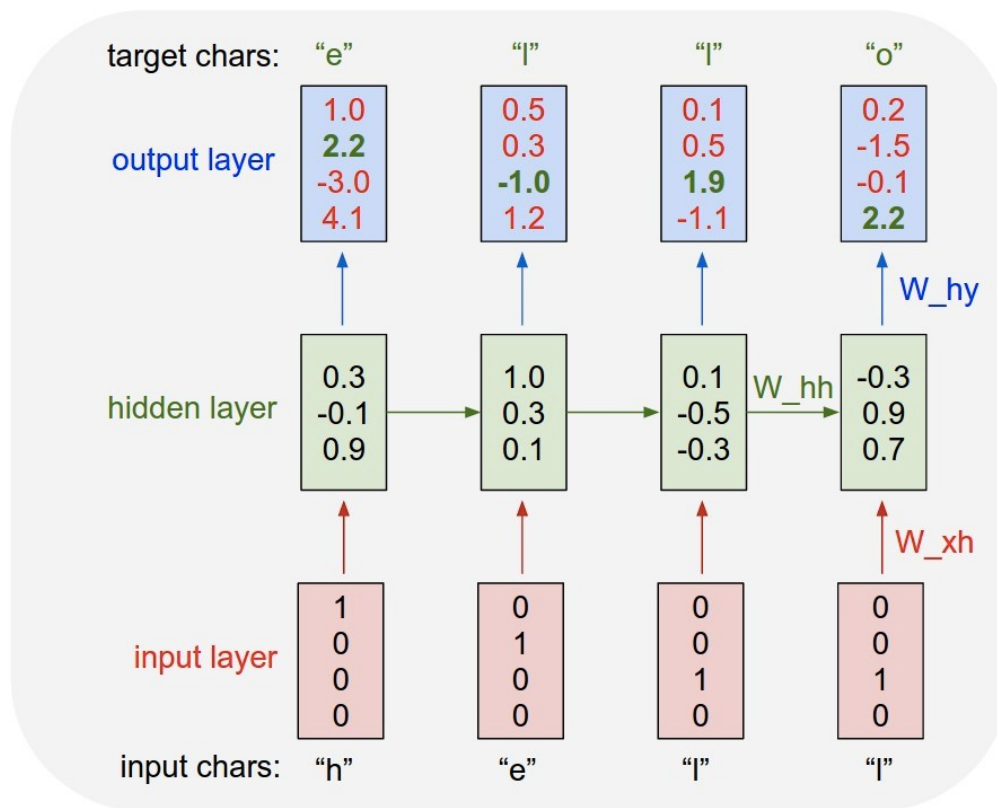


Long short-term memory (**LSTM**) cell



Stacking memory cells to form a deep RNN
Unrolling for training

Sequential prediction training criteria



Training and optimizing BachBot

Training BachBot

RNN dynamics

$$\tilde{P}(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{h}_{t-1})$$

Initial state (all 0s)

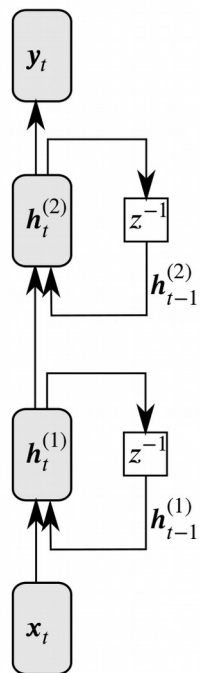
$$+ \mathbf{h}_0$$



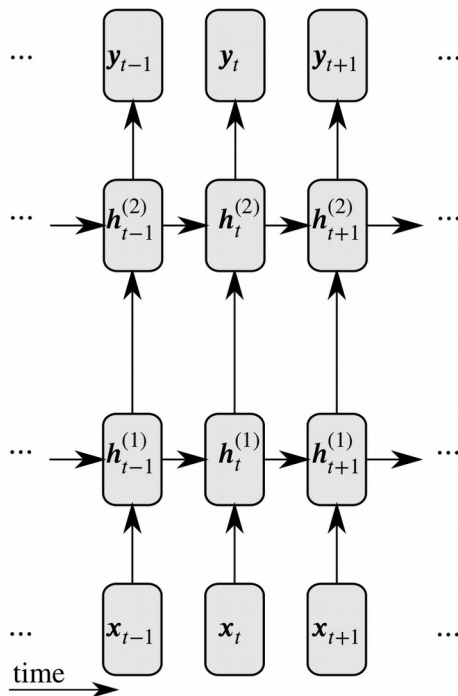
Prob. distr. over sequences

$$\mathbf{y}_t = \tilde{P}(\mathbf{x}_t | \mathbf{x}_{1:t-1})$$

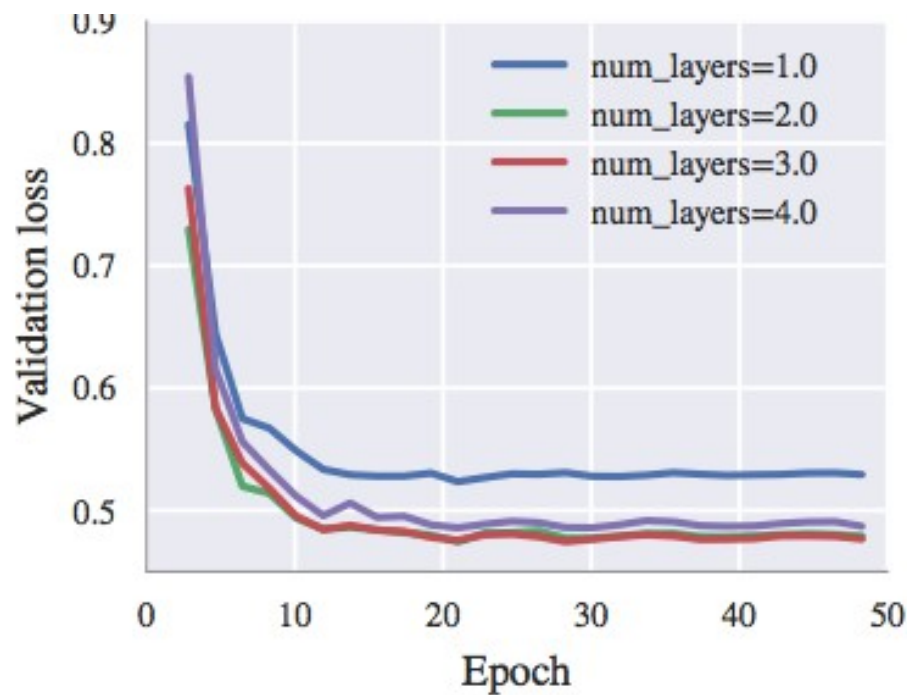
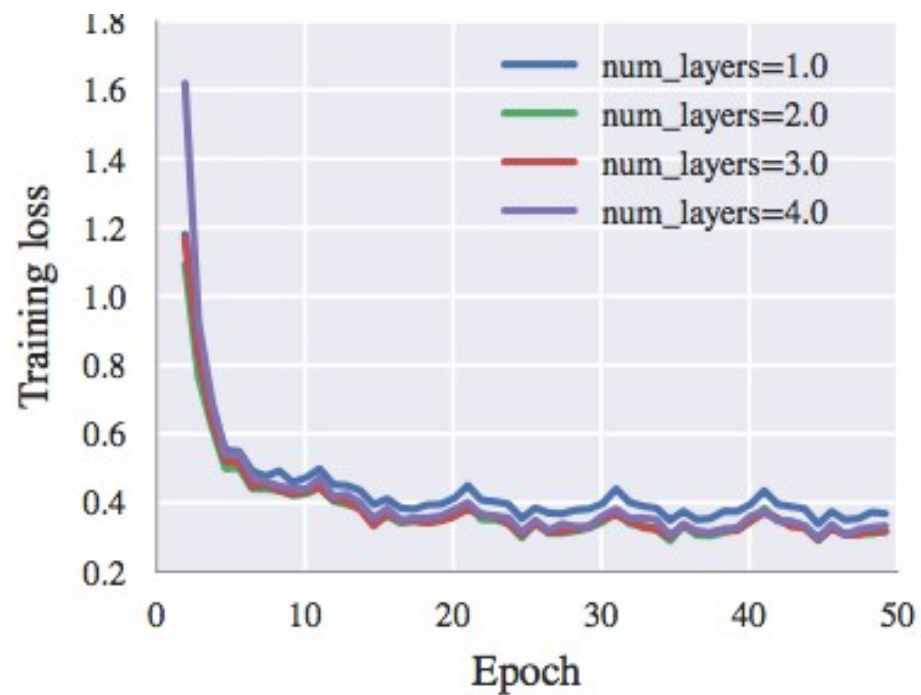
Optimize the
RNN parameters...



Unroll



...in order to maximize the
probability of the real
Bach chorales.



Deeper RNNs do better (to a certain point)

Samples from various points during training

100 minibatches:



500 minibatches:



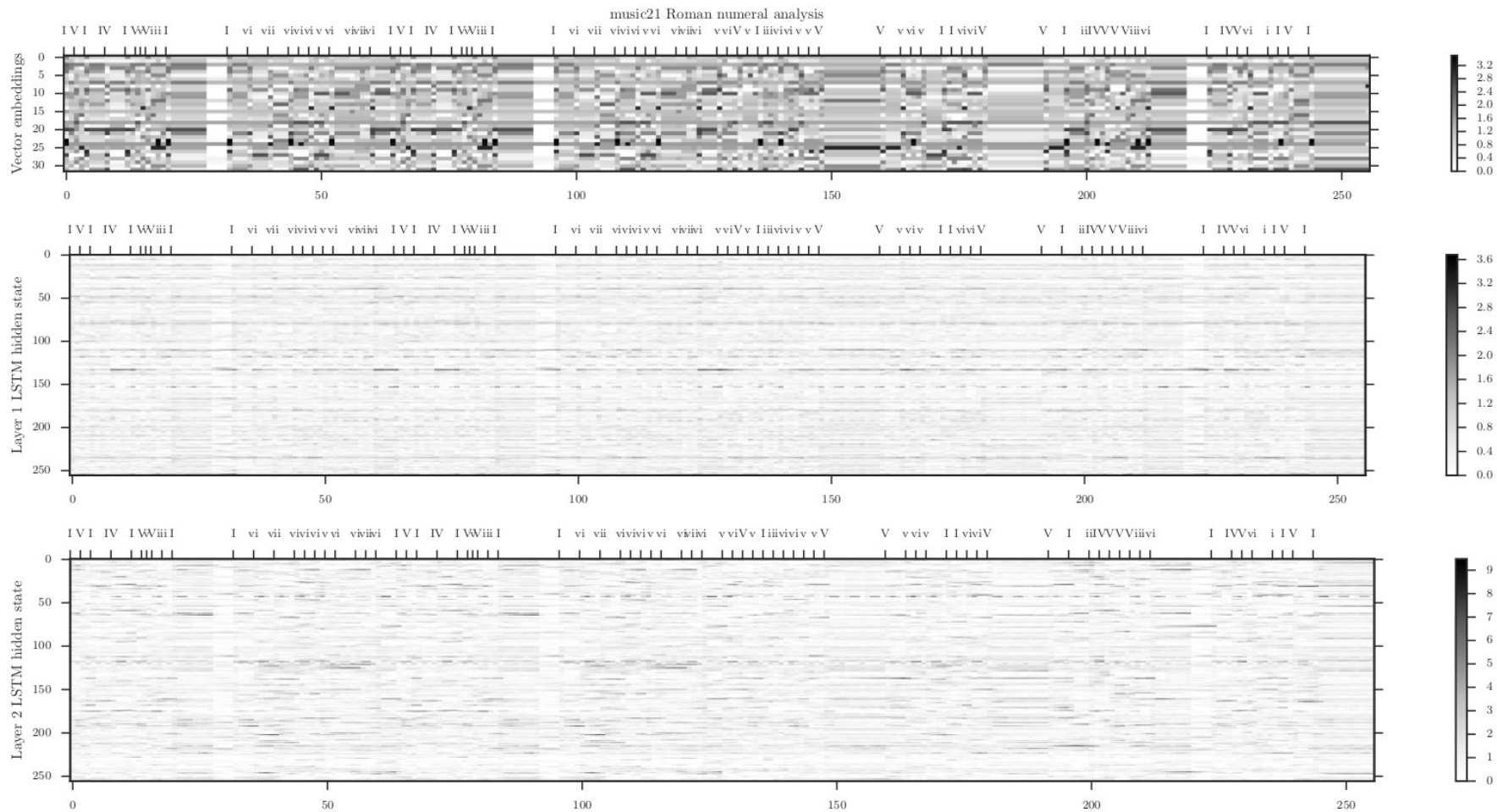
2000 minibatches:



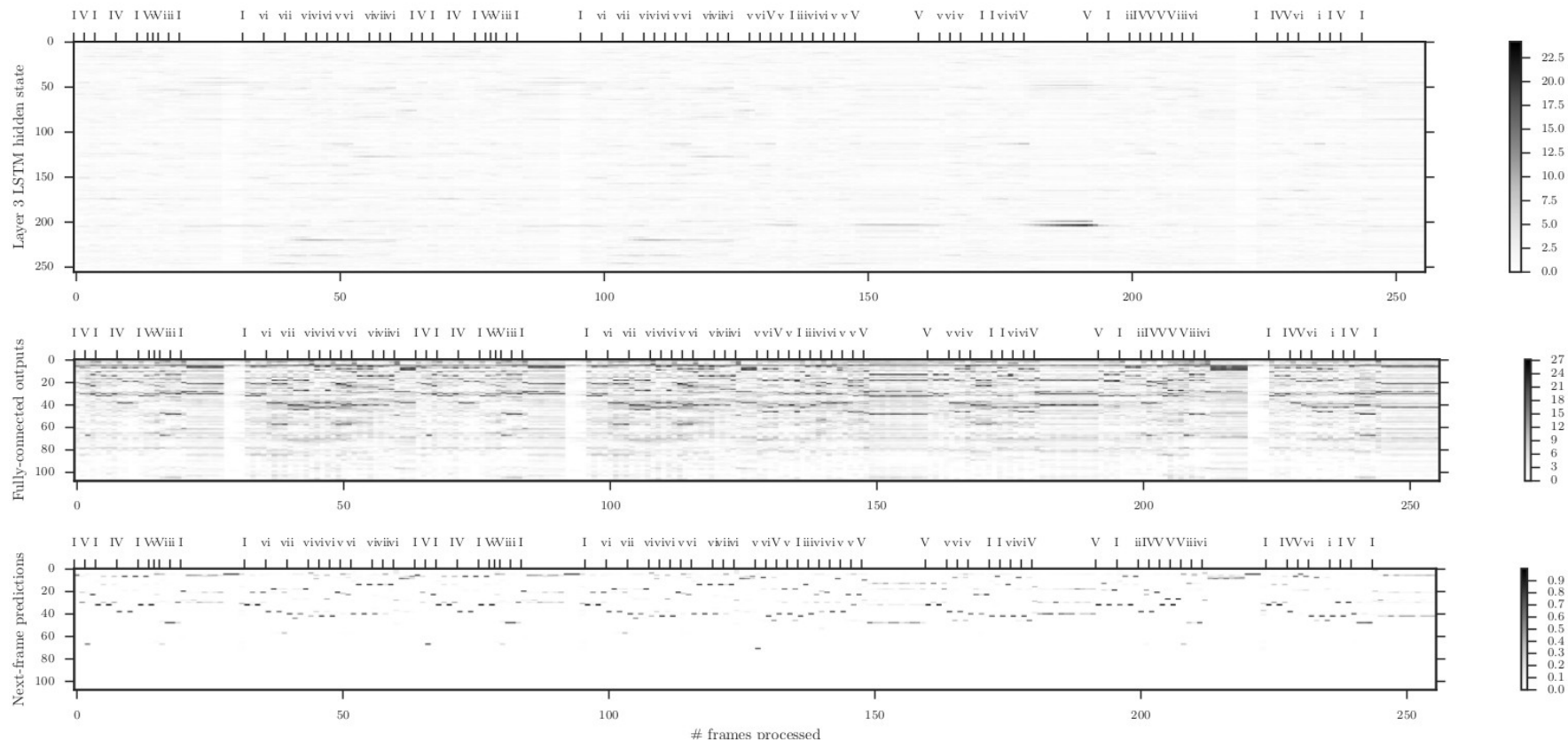
5000 minibatches:



Discoveries and Demos



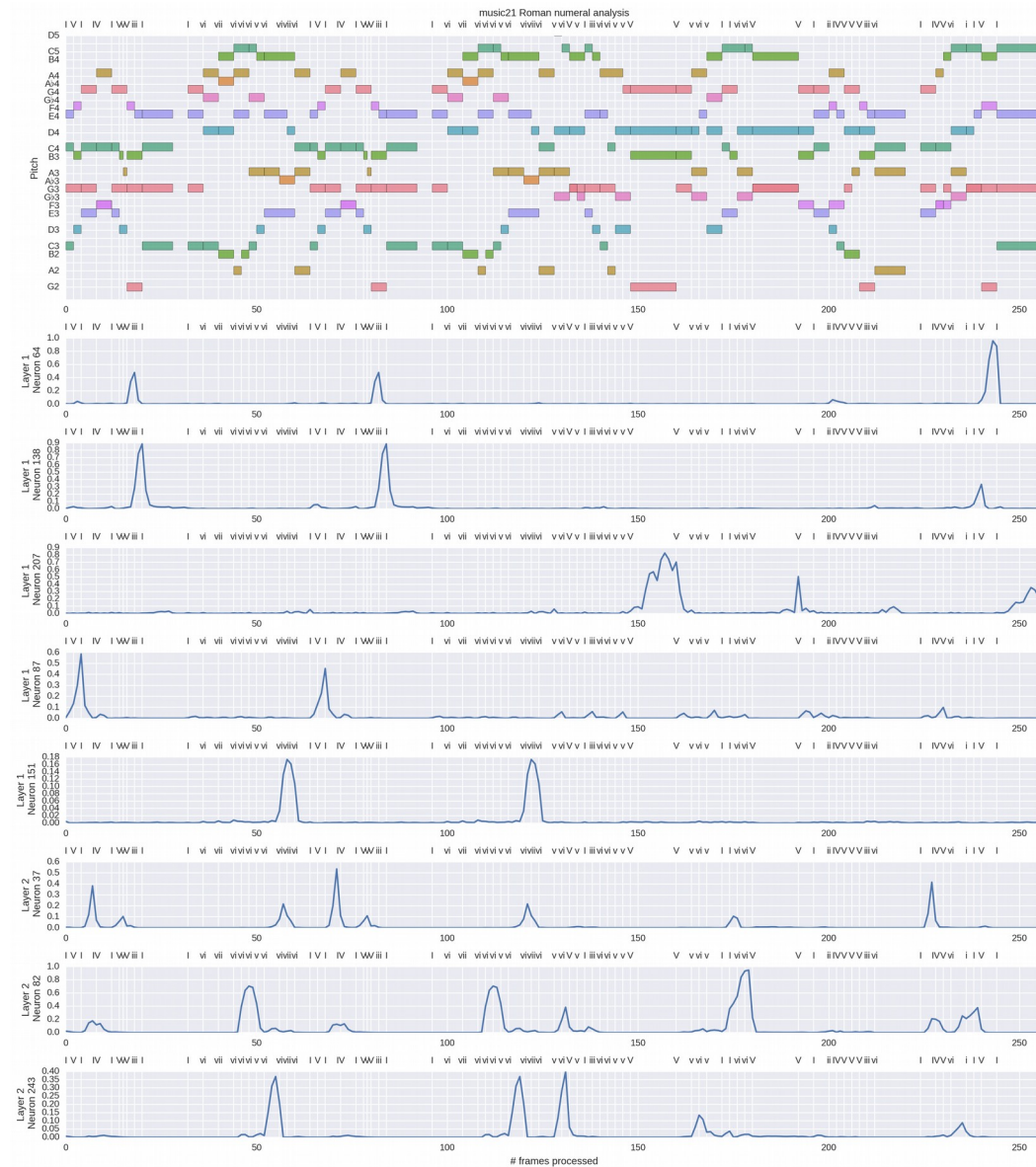
Hidden state is difficult to interpret!
Input and memory cell (layer 1 and 2)



Layers closer to the output resemble piano roll (consequence of sequential training criteria)
Memory cell (layer 3), outputs, and predictions

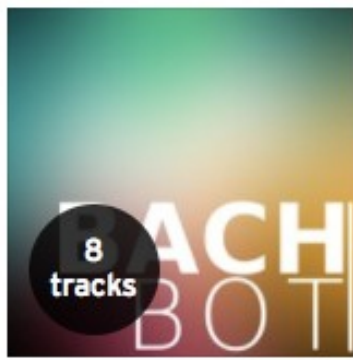
Model learns music theory!

- L1N64 and L1N138: Perfect cadences with root position chords in tonic key
- L1N151: A minor cadences ending phrases 2 and 4
- L1N87 and L2N37: I^6 chords





Harmonization: given a melody (here C major scale)



BachBot
BachBot.com



 1 Sample-1	▶ 2,244
 2 Happy Birthday	▶ 2,091
 3 Twinkle Twinkle Little Star	▶ 6,524
 4 Sample-2	▶ 1,304
 5 Sample-3	▶ 800
 6 Sample-4	▶ 558
 7 Sample-5	▶ 538
 8 C major scale	▶ 828

View fewer tracks

Like Share More

Harmonization: produce the accompanying parts
<https://soundcloud.com/bachbot>



BachBot
BachBot.com



3 Twinkle Twinkle Little Star

▶ 6,524



4 Sample-2

▶ 1,304



5 Sample-3

▶ 800



6 Sample-4

▶ 558



7 Sample-5

▶ 538

View 8 tracks



Like



Share



More

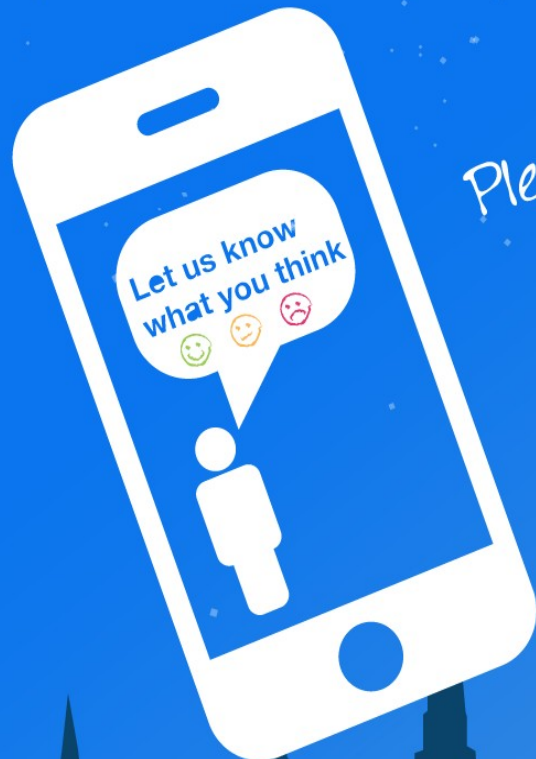
What if Bach remixed twinkle twinkle little star?

<https://soundcloud.com/bachbot>

Stand up if it's your birthday



Thank You!



Please

**Remember to
rate this session**

Thank you!

