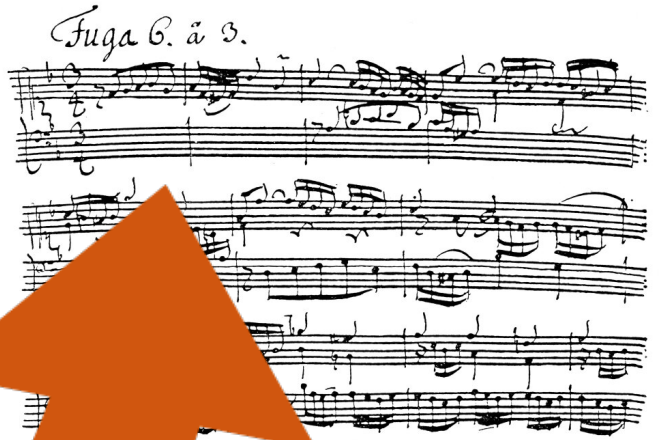
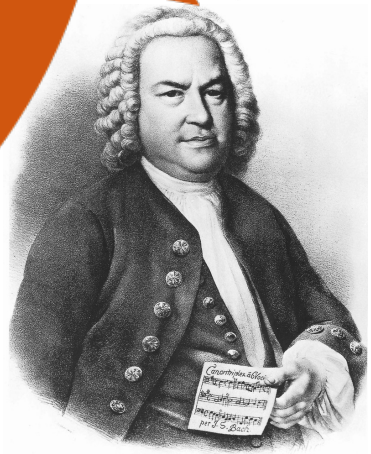
LEARNING TO GENERATE MUSIC WITH BACHPROP

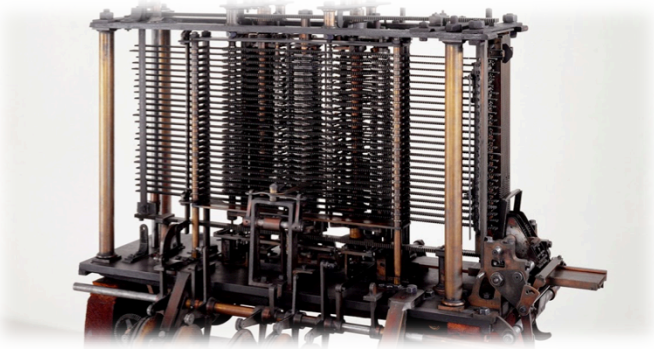
Magliaso Sommer Akademie| Florian Colombo


Fuga 6. à 3.



Ada Lovelace | First Programmer

"Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might **compose elaborate and scientific pieces of music** of any degree of complexity or extent."

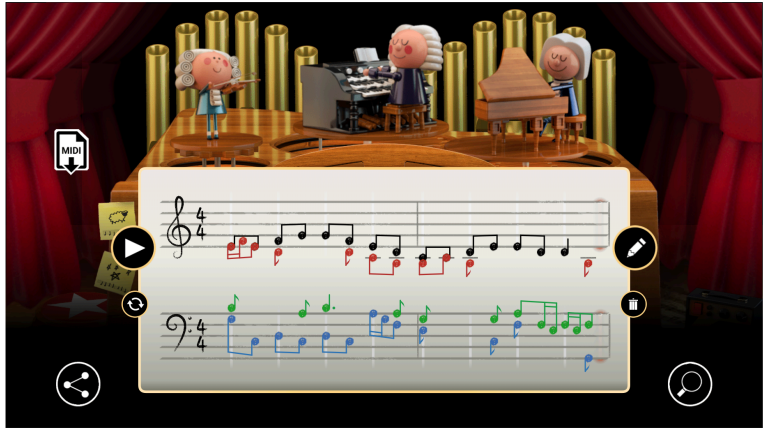


Deep Generative Models | Images

Karras et al. ICLR18

Deep Generative Models | Music

magenta



Music Composition | Automated approaches

Grammars

Markov Chains

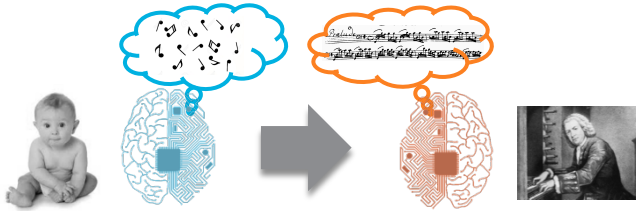
Evolutionary algorithms

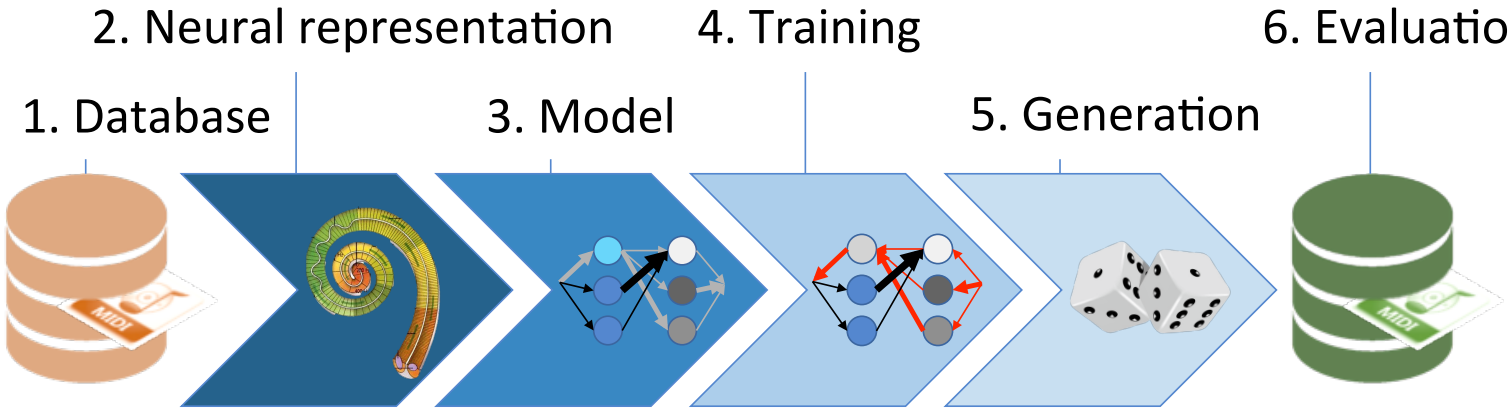
Cellular automata

Neural Networks

What is the next note?

BachProp | Algorithm

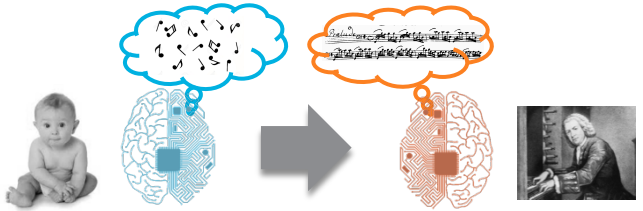


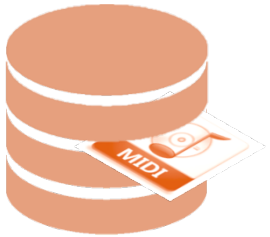


Original corpus

Generated corpus

Datasets





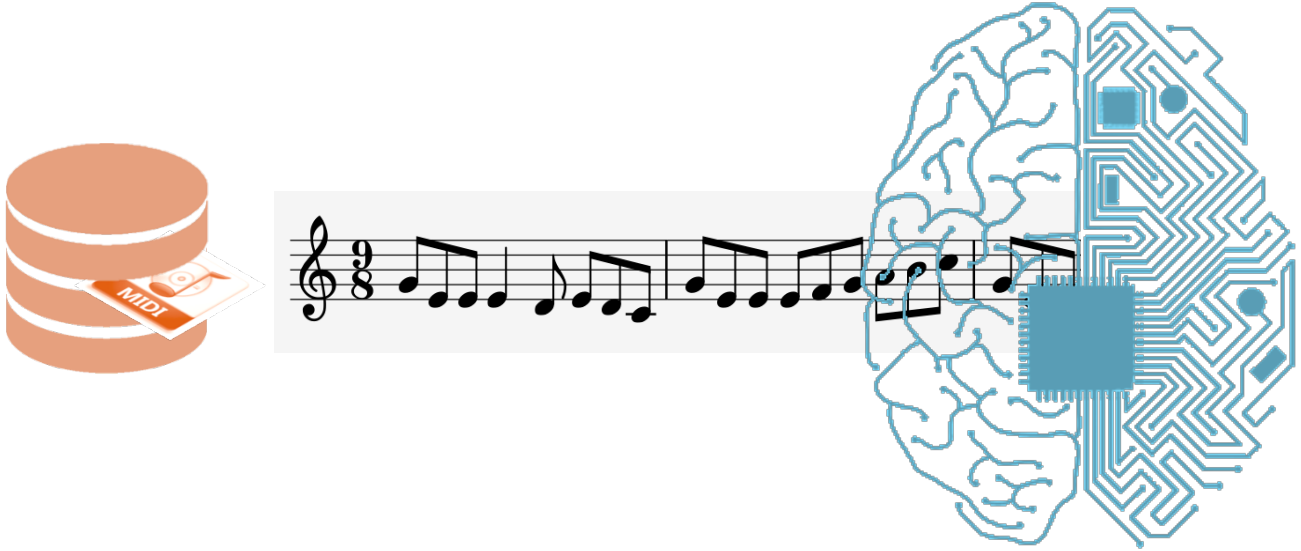
Original corpus

- **Nottingham**
  - 1035 folk tunes with homogeneous structure
- **Chorales**
  - 381 4-voices chorales harmonized by J.S. Bach
- **John Sankey**
  - 135 keyboard pieces from J.S. Bach recorded live by John Sankey
- **String quartets**
  - 215 Haydn and Mozart string quartets

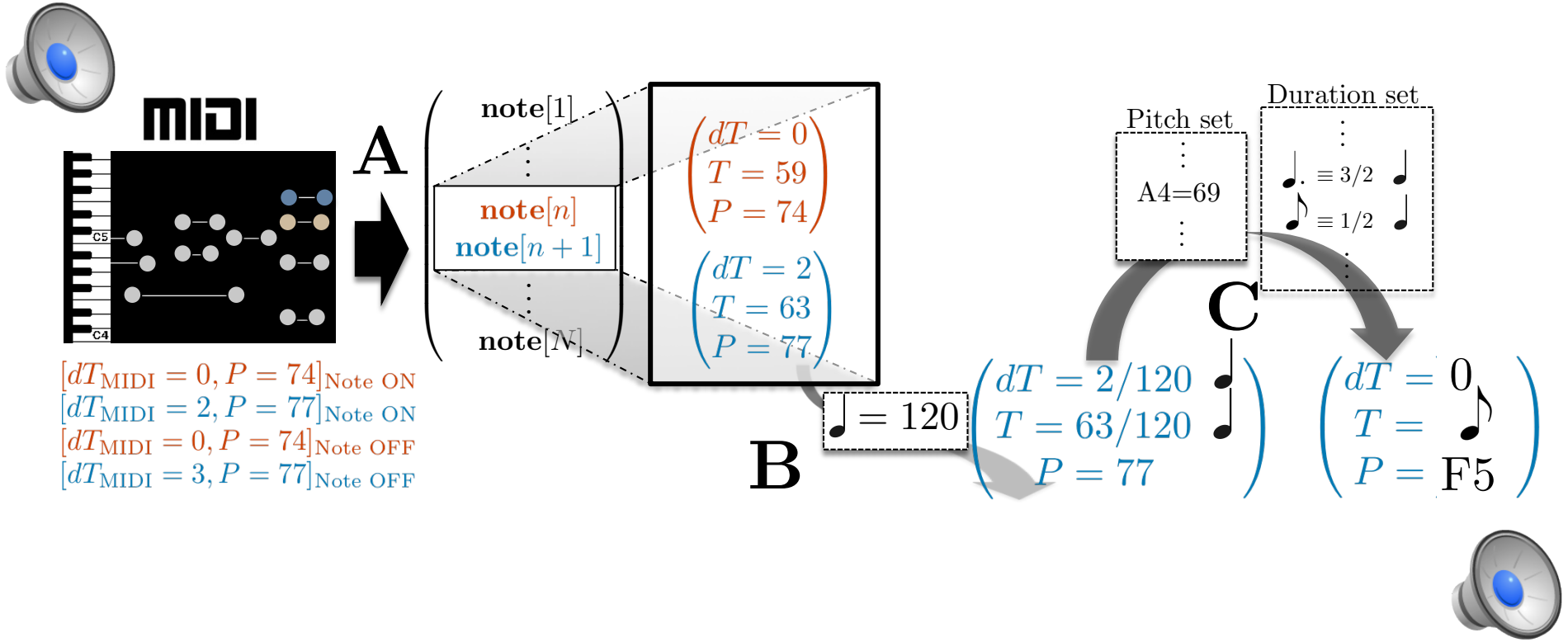
Original and generated examples| BachProp

sites.google.com/view/bachprop/

Neural representation | Ear of an ANN

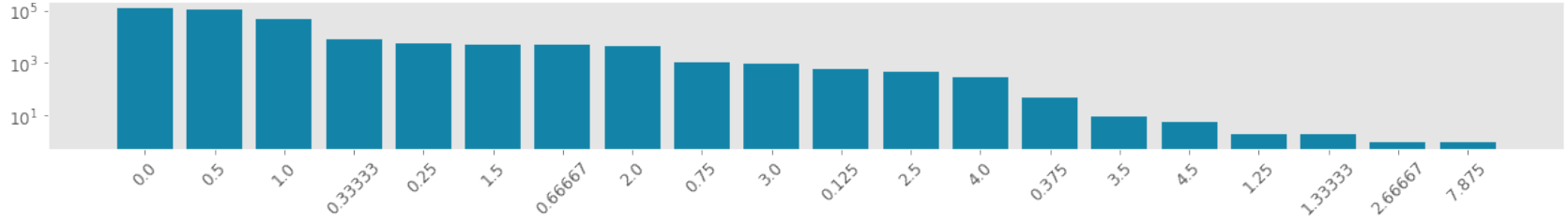


Normalization | MIDI to note sequence

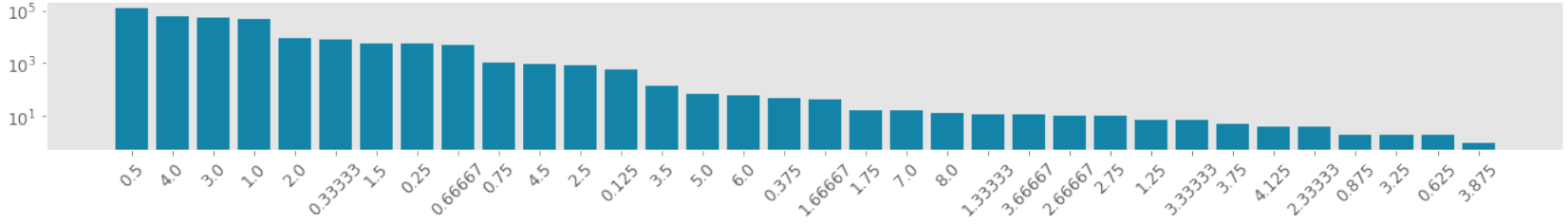


Feature dictionaries | Distributions

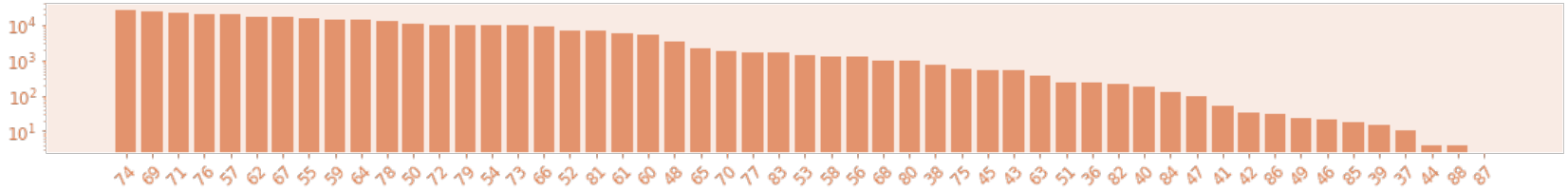
dT



T

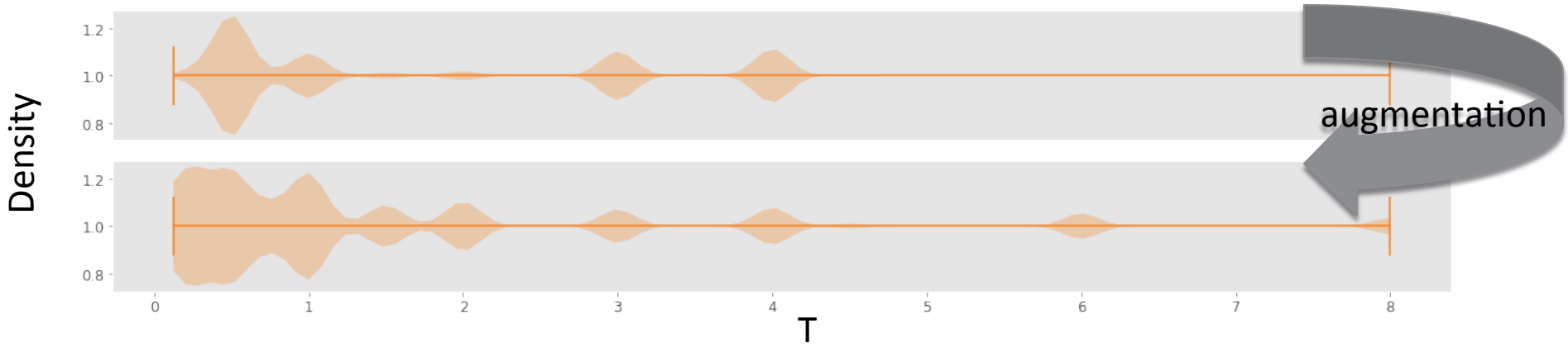


P



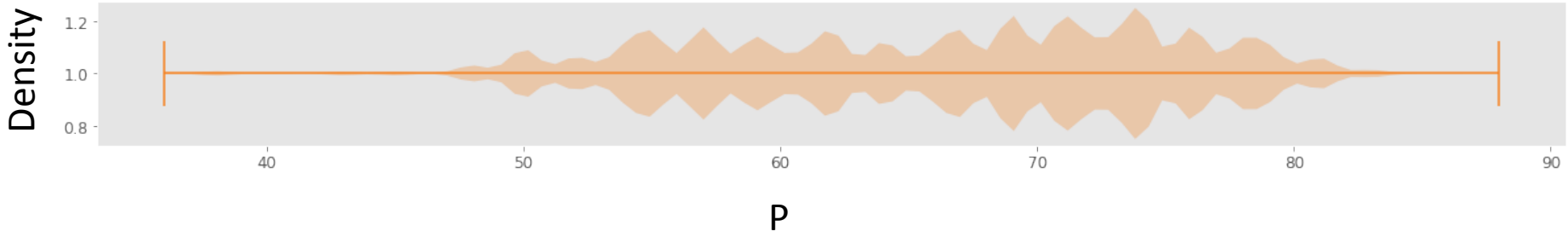
Data Augmentation | Rhythm

- Dilate and contract durations
  - Factors: [1/4, 1/3, 1/2, 2/3, ¾, 1,5/4, 3/2, 2, 3, 4]
  - Constrain on the set of possible durations
- Nottingham dataset
  - 1037 songs → 3334 songs

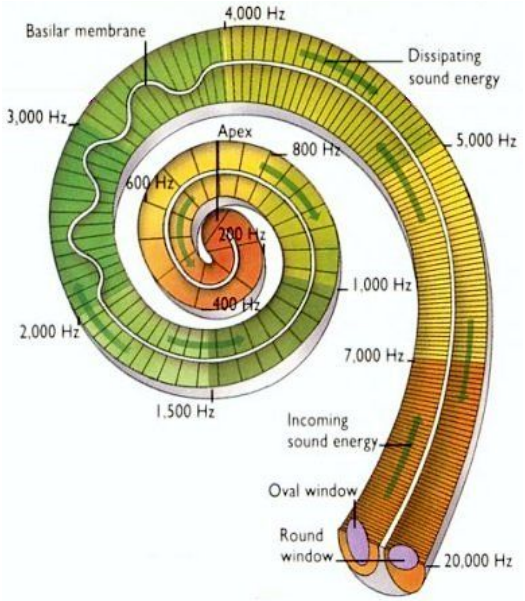


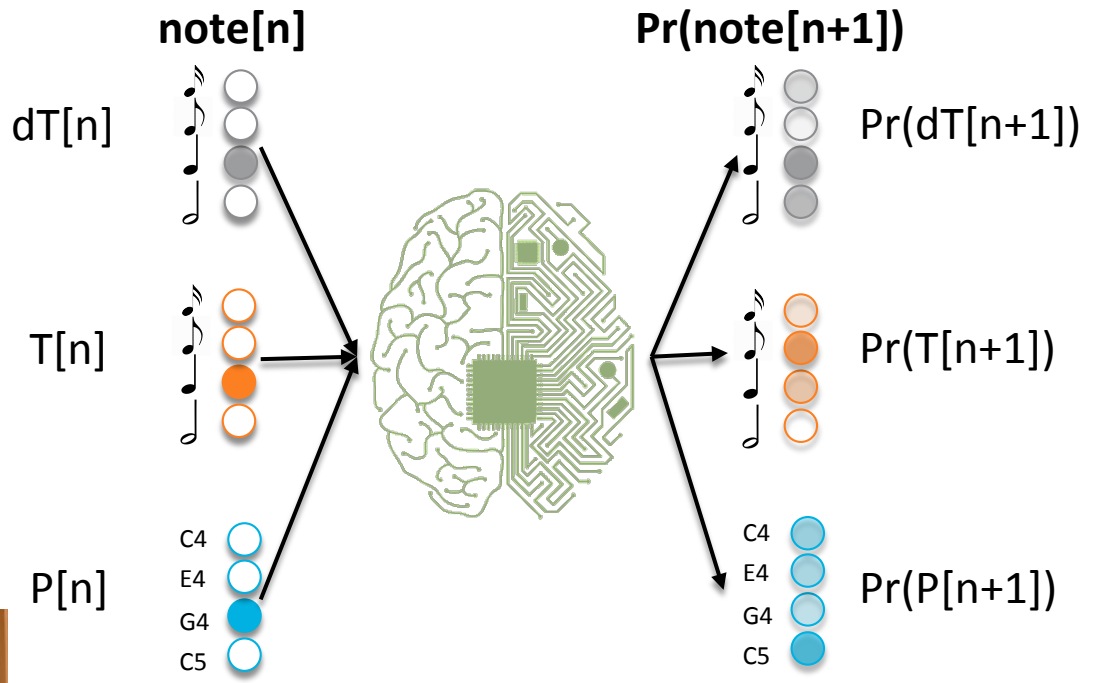
Data Augmentation | Pitch

- Music is transposition invariant
  - Transpose in every possible keys greatly increase the number of training samples

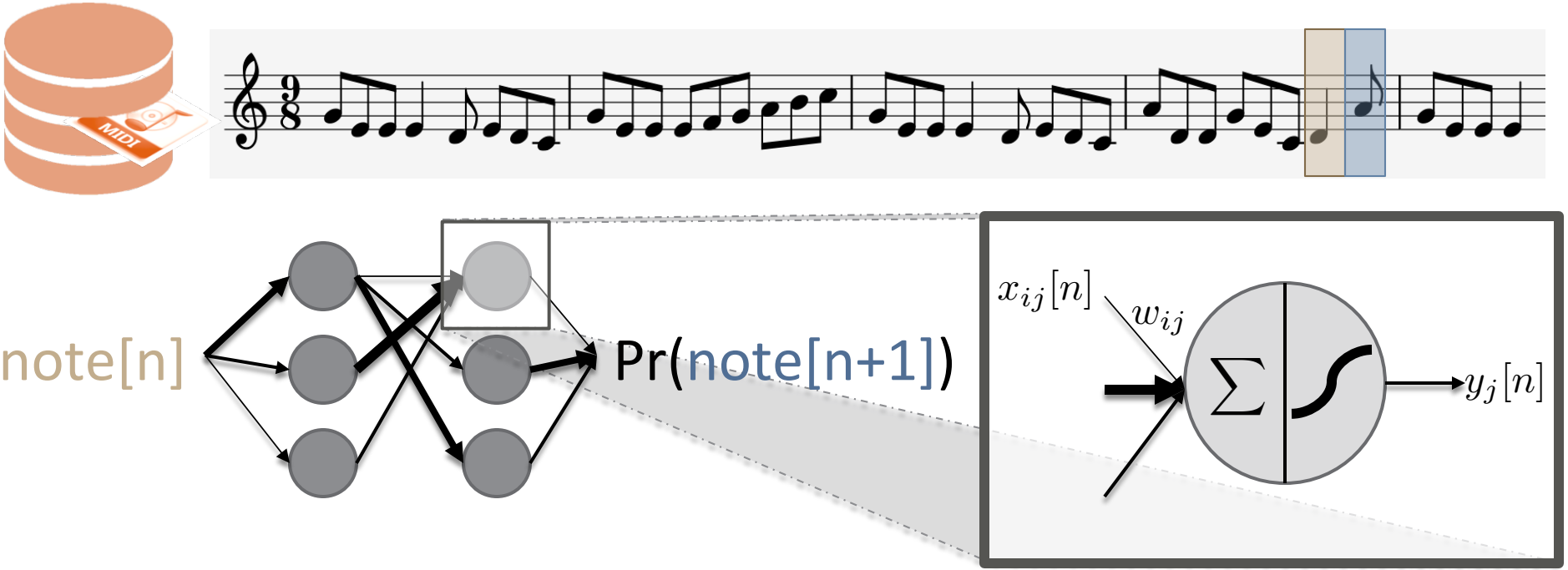


Data Representation | I/O

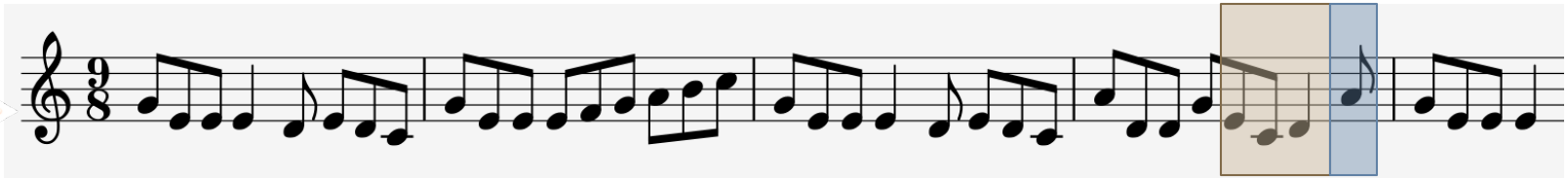




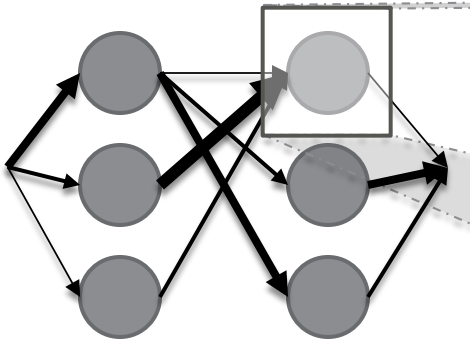
Model | Feedforward



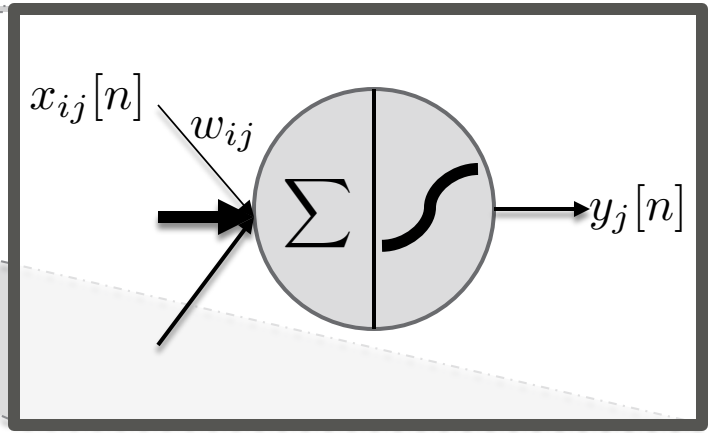
Model | Feedforward



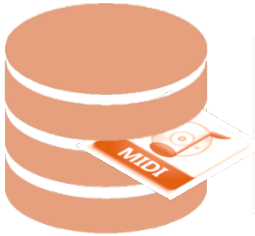
note[n-N:n]

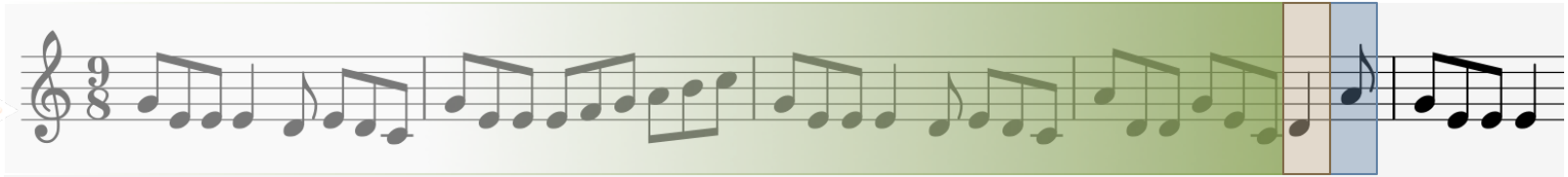


Pr(note[n+1])

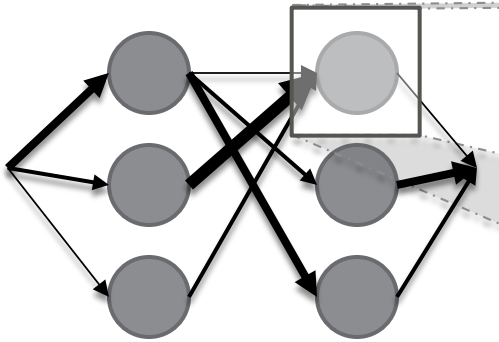


Model | Recurrent

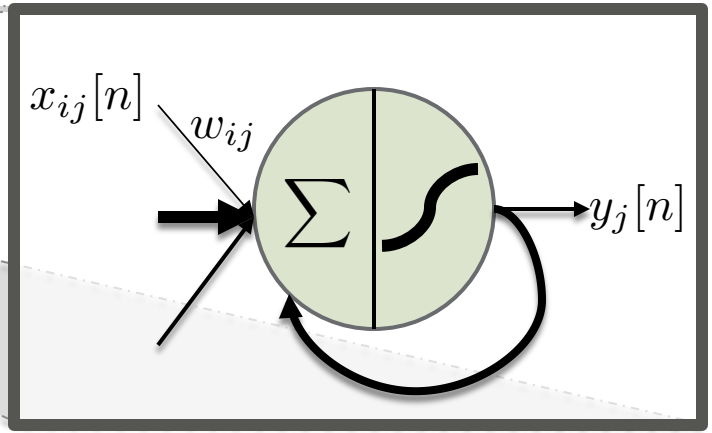




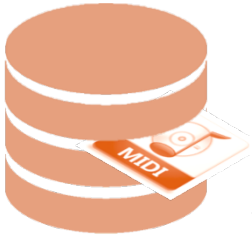
note[n]

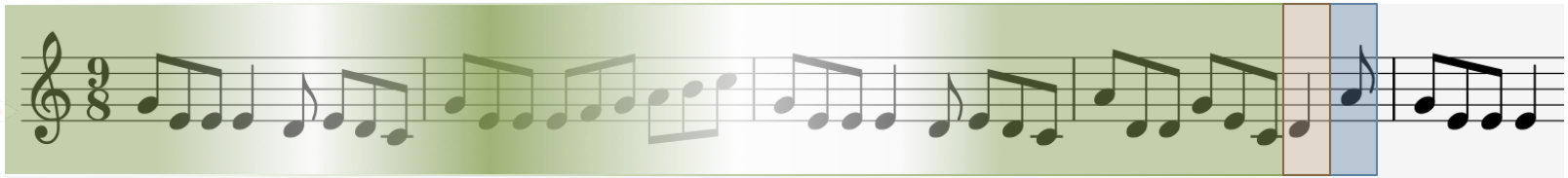


Pr(note[n+1])

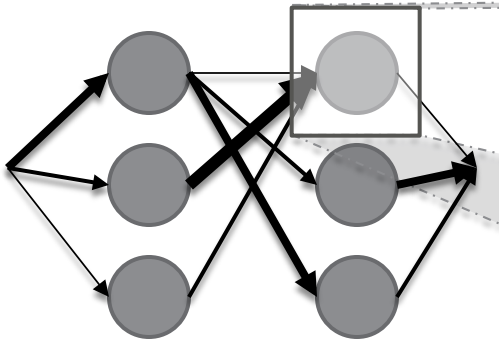


Model | LSTM (GRU)

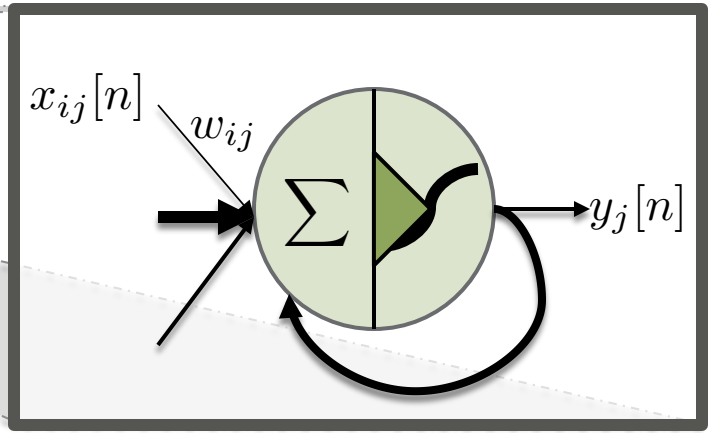




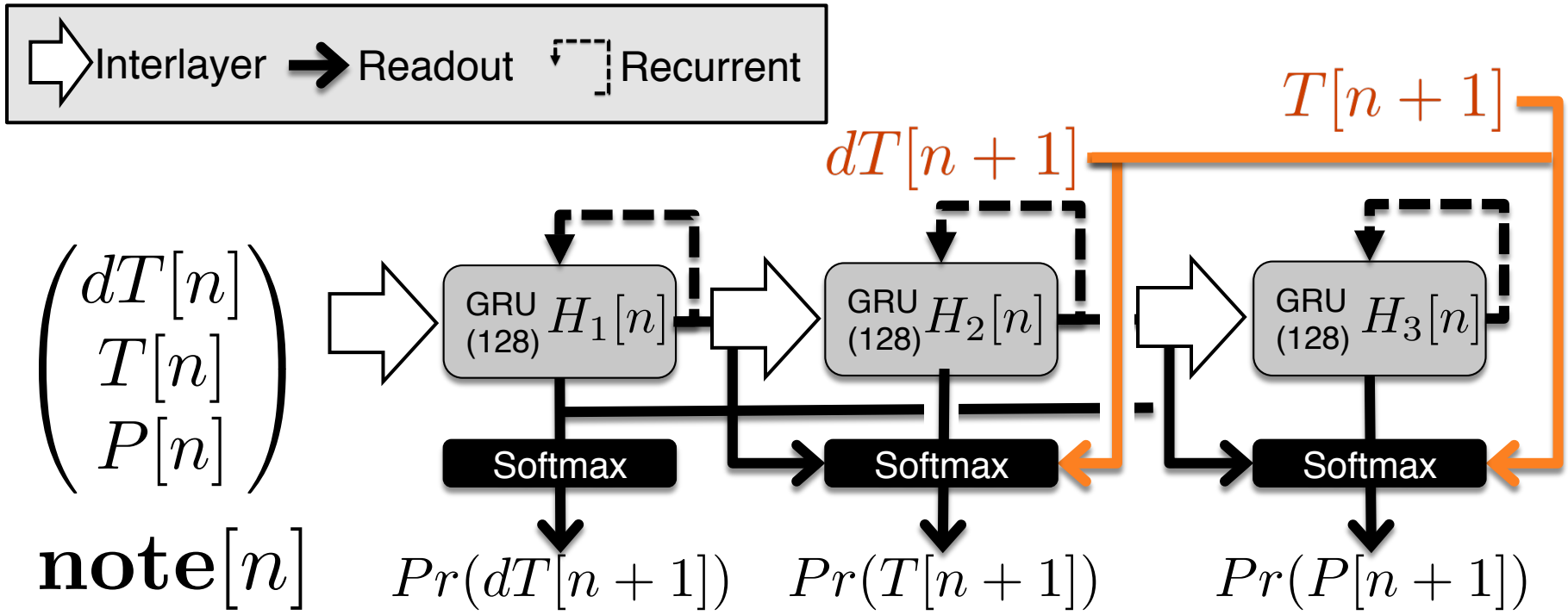
note[n]



Pr(note[n+1])

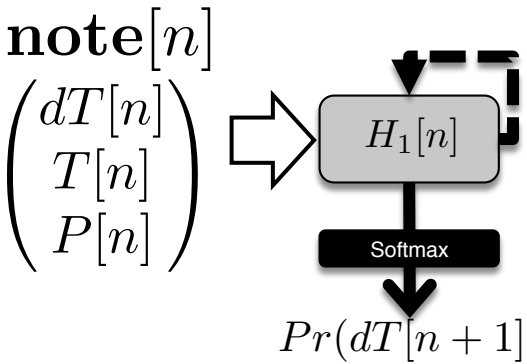


Neural Network Architecture



Probabilistic interpretation

$$Pr(\mathbf{note}[1:N]) = Pr(\mathbf{note}[1]) \prod_{n=1}^{N-1} Pr(\mathbf{note}[n+1]|\mathbf{note}[1:n])$$



$$Pr(\mathbf{note}[n+1]|\mathbf{note}[1:n]) =$$

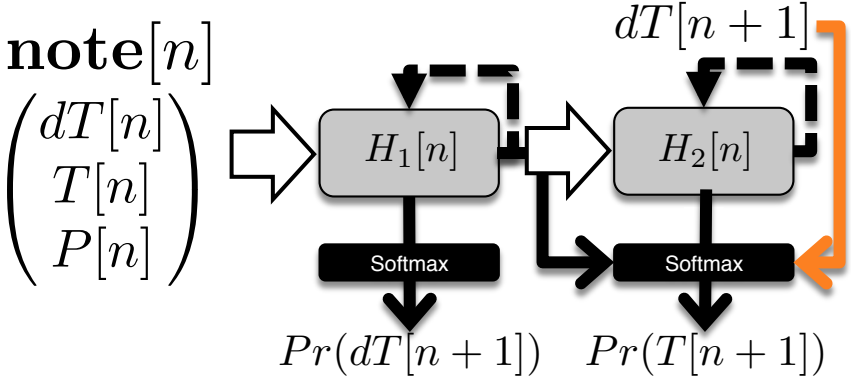
$$Pr(dT[n+1]|\mathbf{note}[1:n]) \times \approx Pr(dT[n+1]|H_{dT}[n])$$

$$Pr(T[n+1]|\mathbf{note}[1:n], dT[n+1]) \times \approx Pr(T[n+1]|H_T[n], dT[n+1])$$

$$Pr(P[n+1]|\mathbf{note}[1:n], dT[n+1], T[n+1]) \approx Pr(P[n+1]|H_P[n], dT[n+1], T[n+1])$$

Probabilistic interpretation

$$Pr(\mathbf{note}[1:N]) = Pr(\mathbf{note}[1]) \prod_{n=1}^{N-1} Pr(\mathbf{note}[n+1]|\mathbf{note}[1:n])$$



$$Pr(\mathbf{note}[n+1]|\mathbf{note}[1:n]) =$$

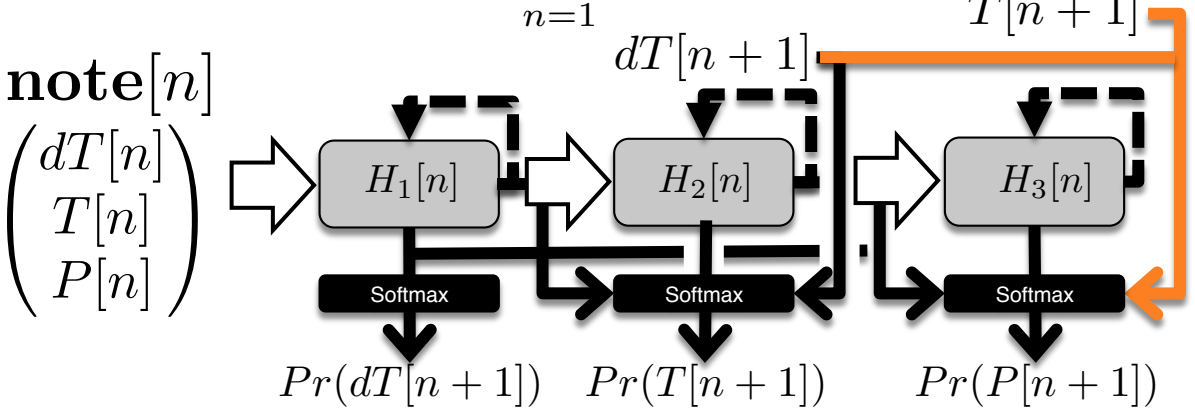
$$Pr(dT[n+1]|\mathbf{note}[1:n]) \times \approx Pr(dT[n+1]|H_{dT}[n])$$

$$Pr(T[n+1]|\mathbf{note}[1:n], dT[n+1]) \times \approx Pr(T[n+1]|H_T[n], dT[n+1])$$

$$Pr(P[n+1]|\mathbf{note}[1:n], dT[n+1], T[n+1]) \approx Pr(P[n+1]|H_P[n], dT[n+1], T[n+1])$$

Probabilistic interpretation

$$Pr(\mathbf{note}[1:N]) = Pr(\mathbf{note}[1]) \prod_{n=1}^{N-1} Pr(\mathbf{note}[n+1]|\mathbf{note}[1:n])$$



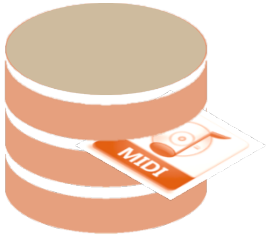
$$Pr(\mathbf{note}[n+1]|\mathbf{note}[1:n]) =$$

$$Pr(dT[n+1]|\mathbf{note}[1:n]) \times \approx Pr(dT[n+1]|H_{dT}[n])$$

$$Pr(T[n+1]|\mathbf{note}[1:n], dT[n+1]) \times \approx Pr(T[n+1]|H_T[n], dT[n+1])$$

$$Pr(P[n+1]|\mathbf{note}[1:n], dT[n+1], T[n+1]) \approx Pr(P[n+1]|H_P[n], dT[n+1], T[n+1])$$

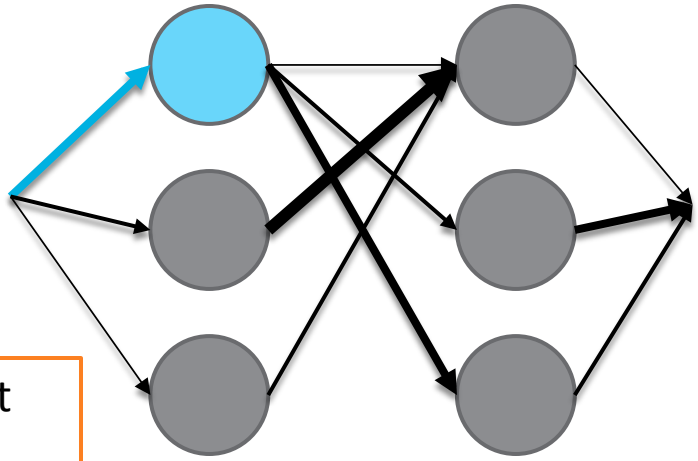
Training | Forward pass



10%

90%

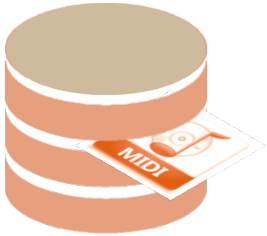
note[n]



Pr(note[n+1])

Randomly present scores from the training set (90% of all scores)

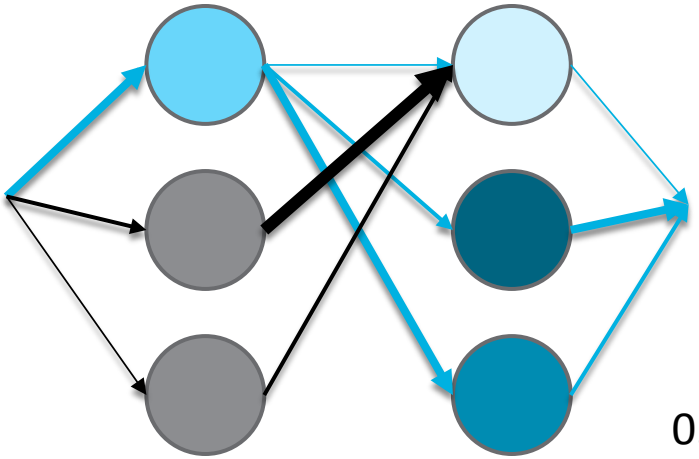
Training | Forward pass



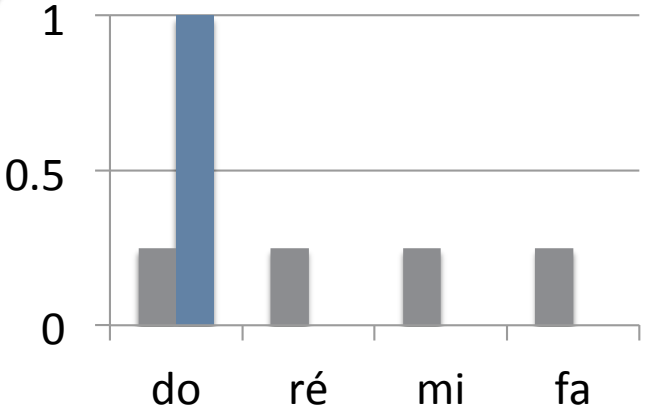
10%

90%

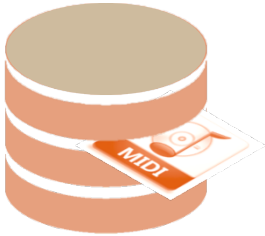
note[n]



Pr(note[n+1])



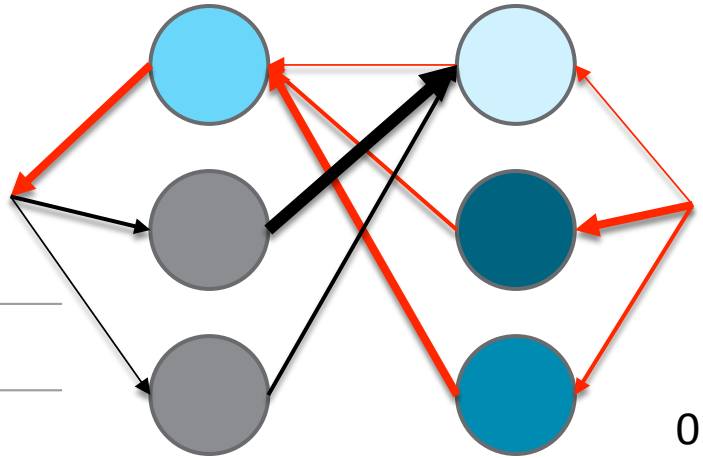
Training | BackProp through time



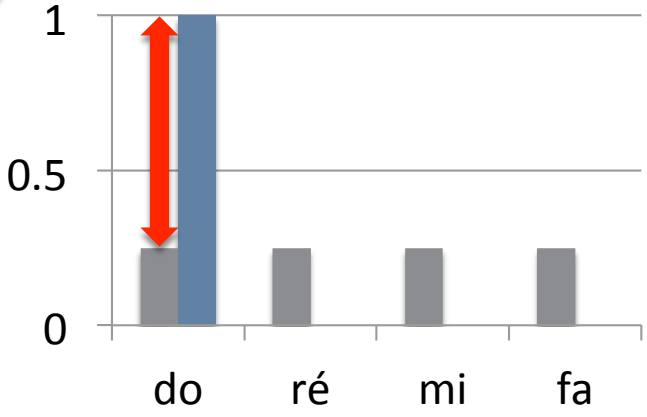
10%

90%

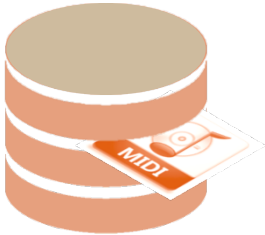
note[n]



Pr(note[n+1])



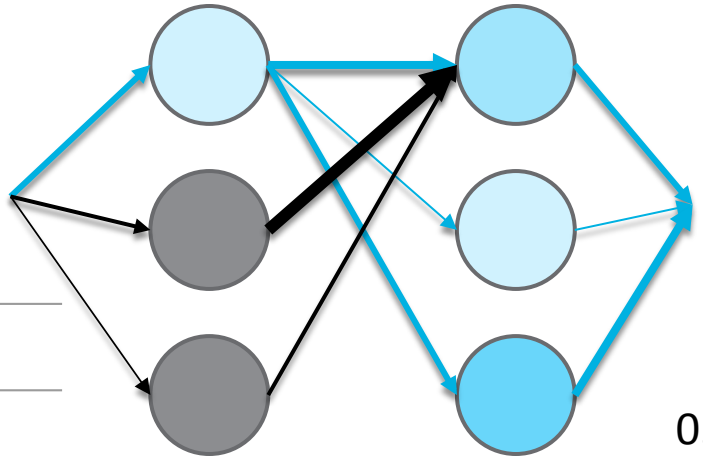
Training | Update parameters



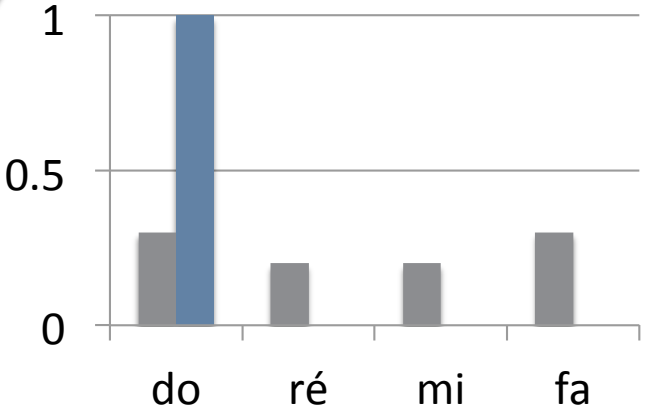
10%

90%

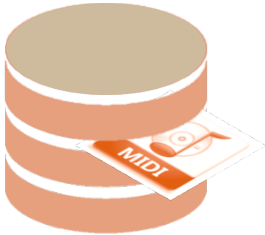
note[n]



Pr(note[n+1])



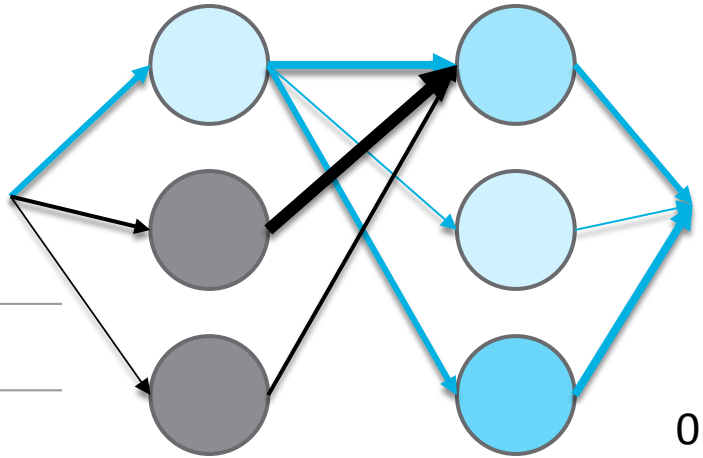
Training | Update parameters



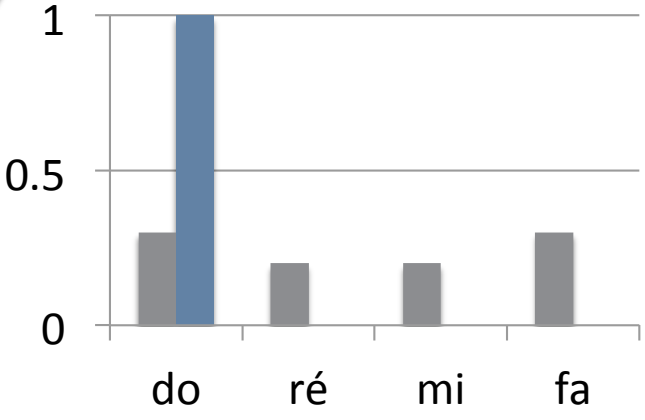
10%

90%

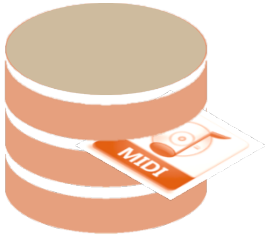
note[n]



Pr(note[n+1])



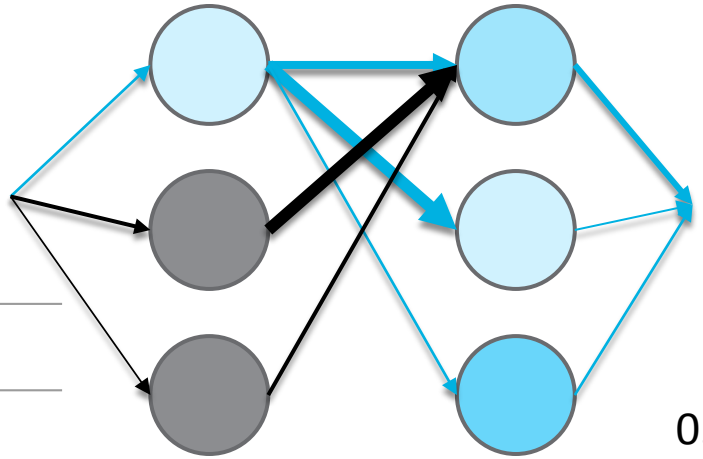
Training | Update parameters



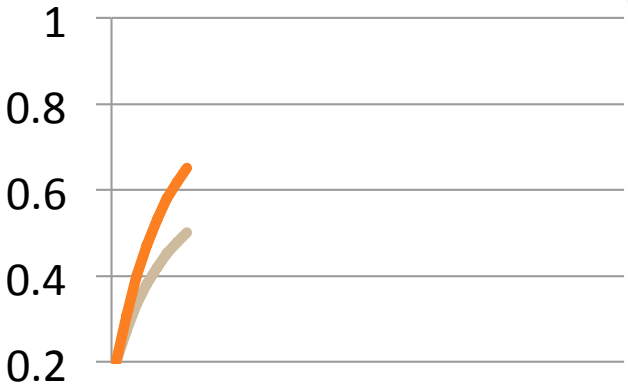
10%

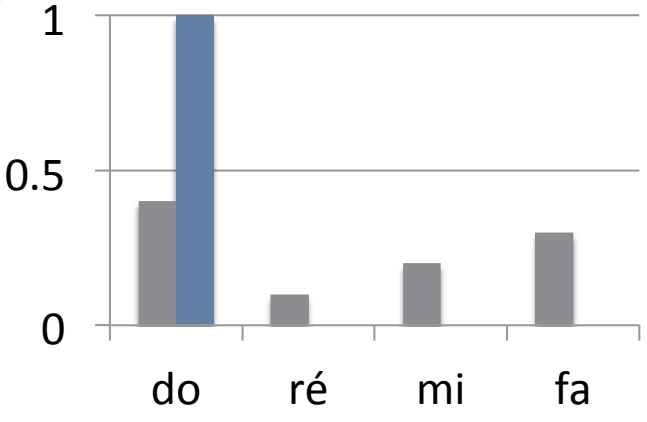
90%

note[n]

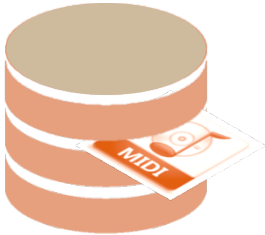


Pr(note[n+1])





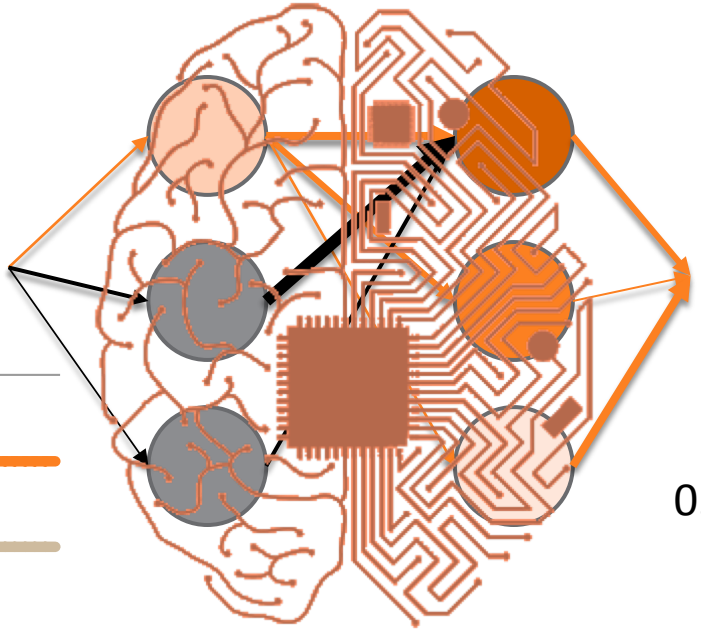
Training | End



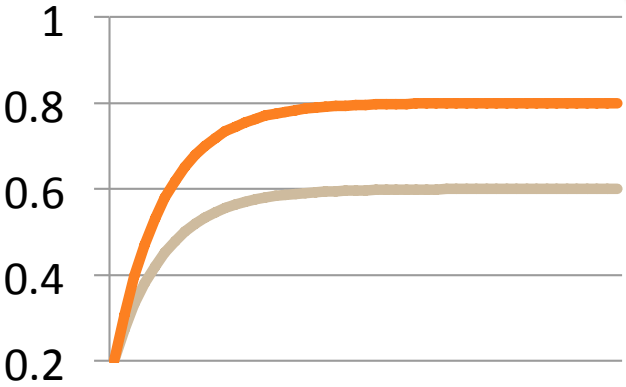
10%

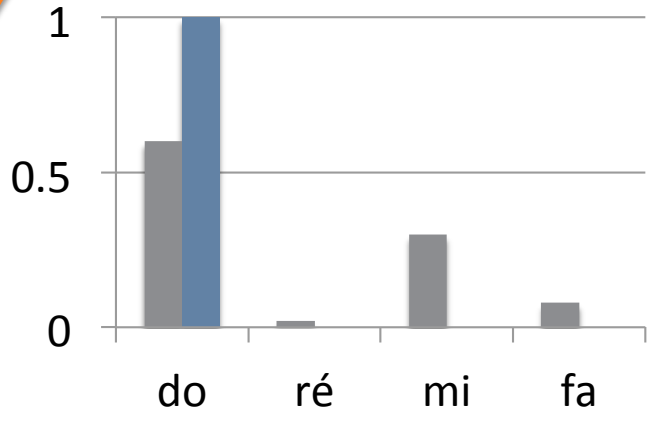
90%

note[n]

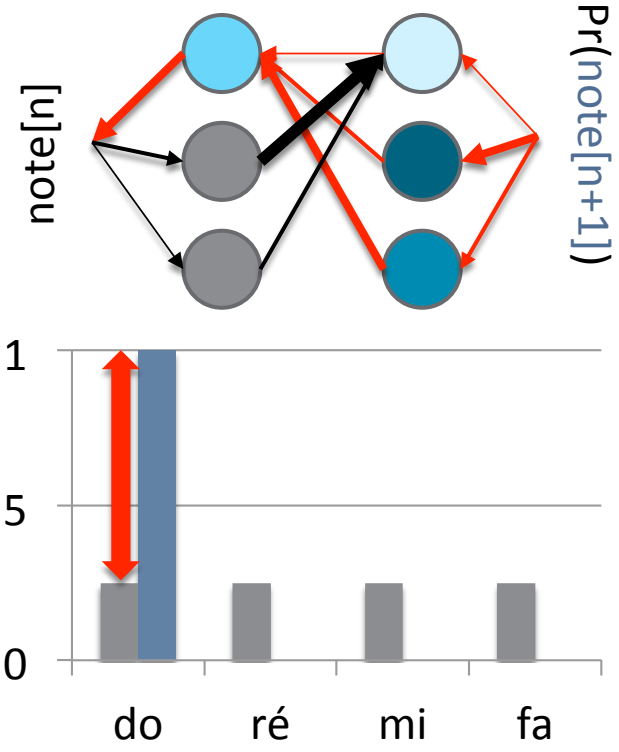


Pr(note[n+1])



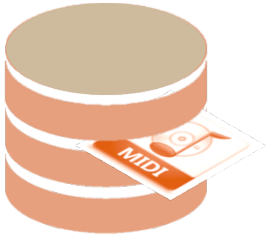


Training



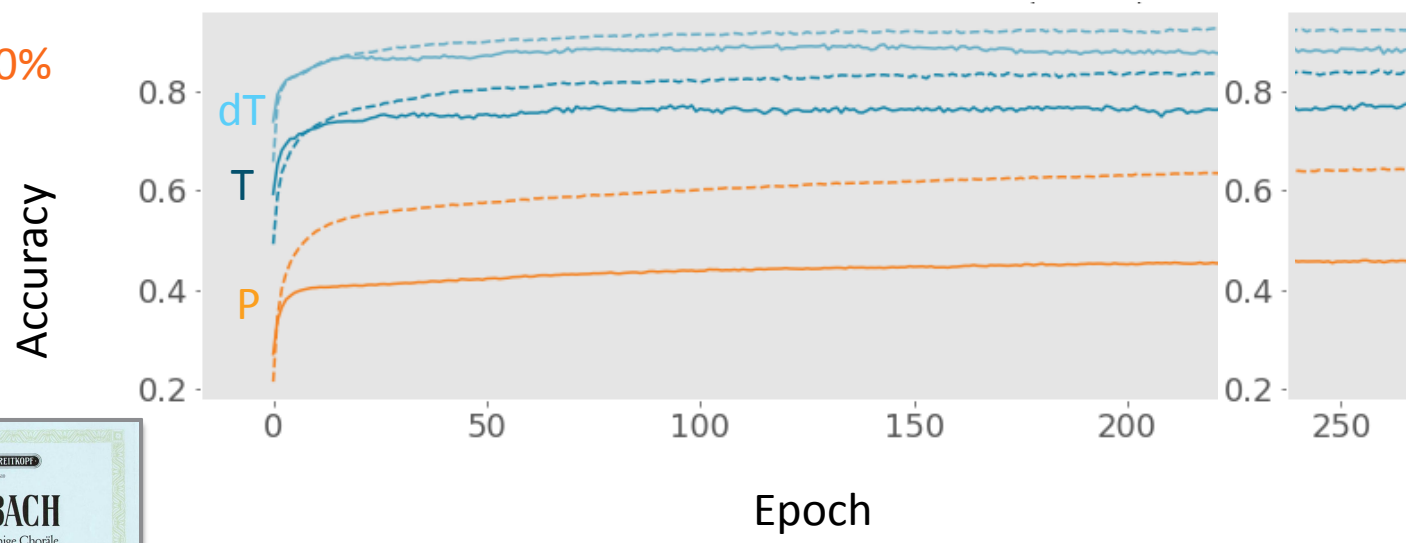
- TBPTT with 128 notes truncation
- Recurrent cells are stateful
  - Hidden state is kept after truncation
- Adam optimizer
  - Weighted (dT: 0.1, T: 0.3, P: 0.6) cross-entropy loss functions

Training | Learning Curves

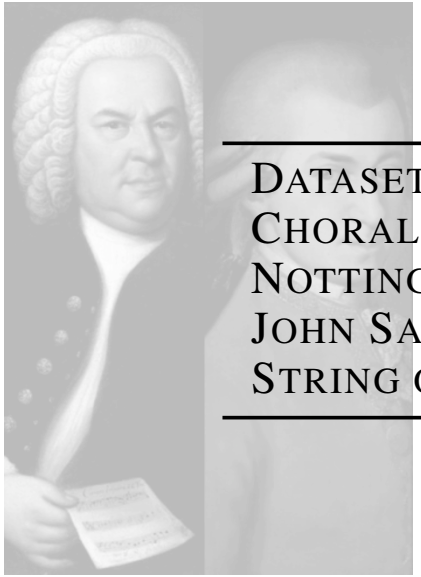


10%

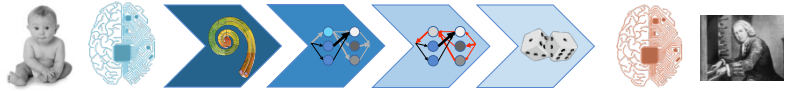
90%



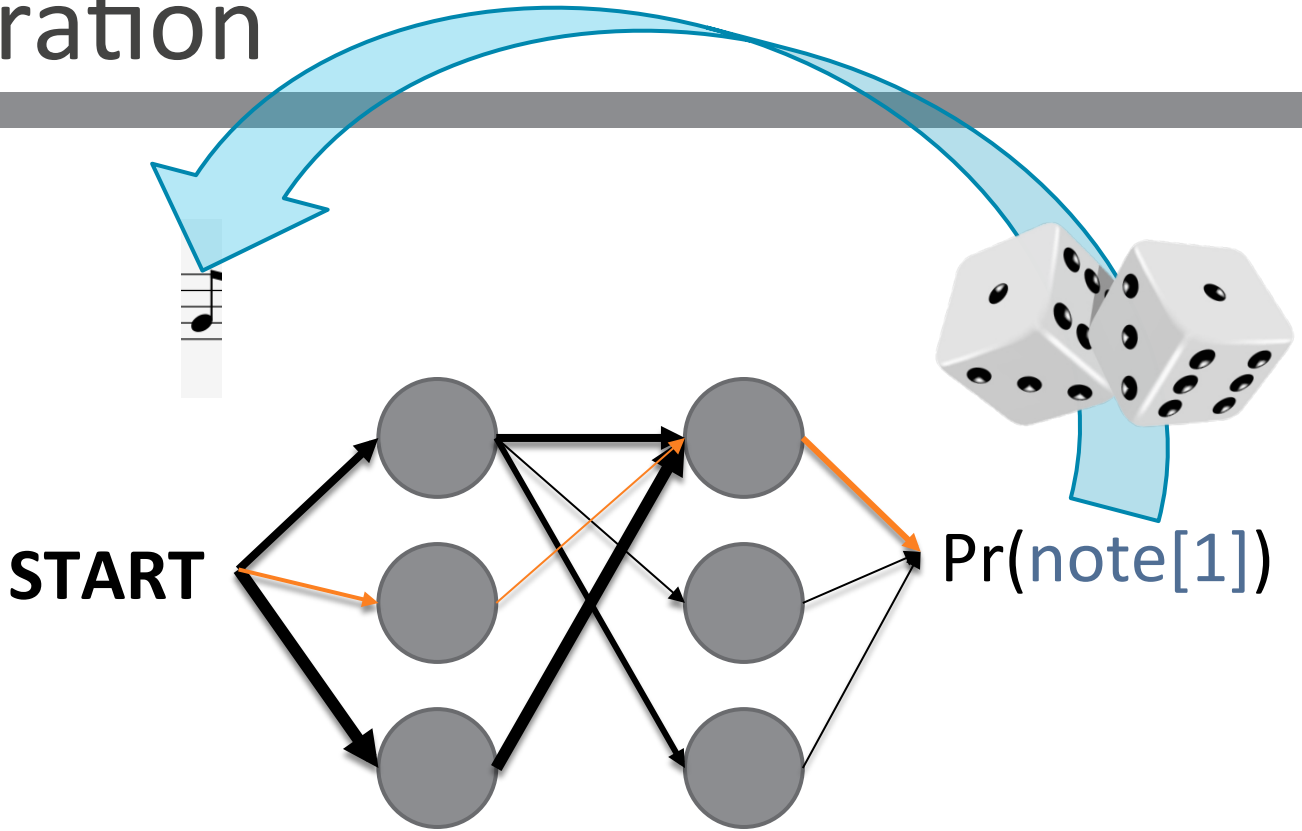
Training| Other Datasets

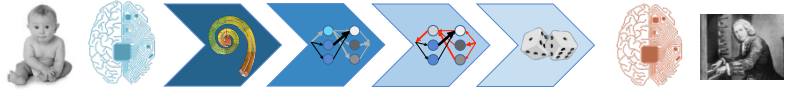


Dataset	NLL	dT	T	P	Size [score]	Size [note]
Chorales	0.419	0.97	0.91	0.77	357	95’337
Nottingham	0.587	0.98	0.89	0.70	1037	313’975
John Sankey	1.002	0.89	0.77	0.45	135	358’211
String quartets	0.936	0.88	0.83	0.49	215	738’739

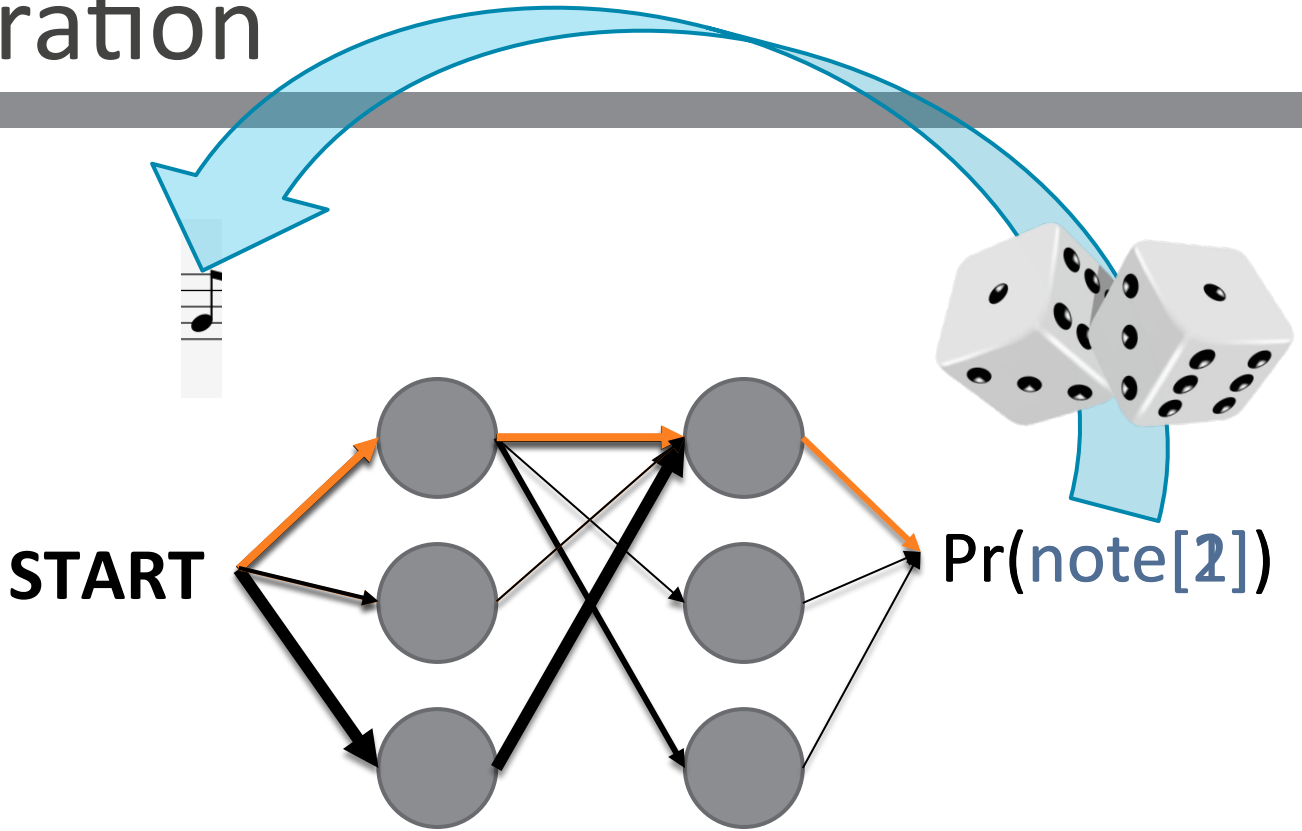


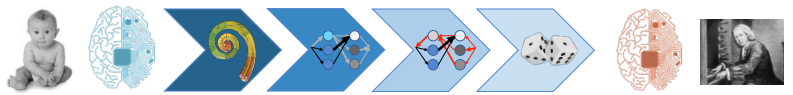
Generation



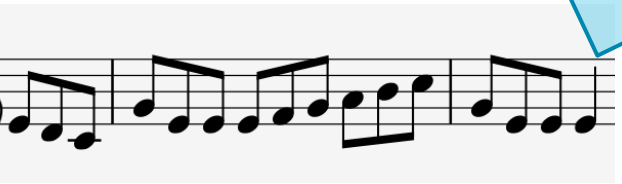


Generation

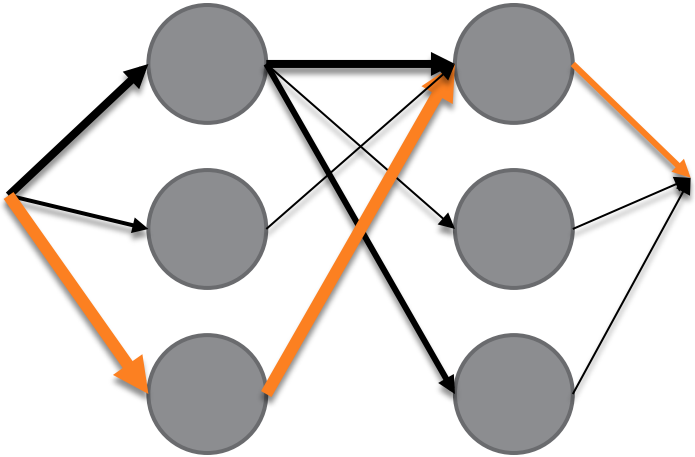




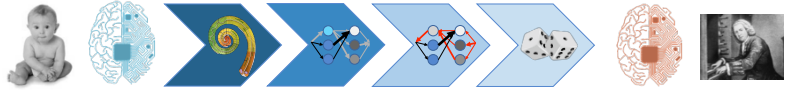
Generation



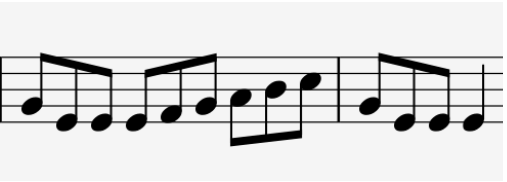
note[n]



Pr(note[n+1])

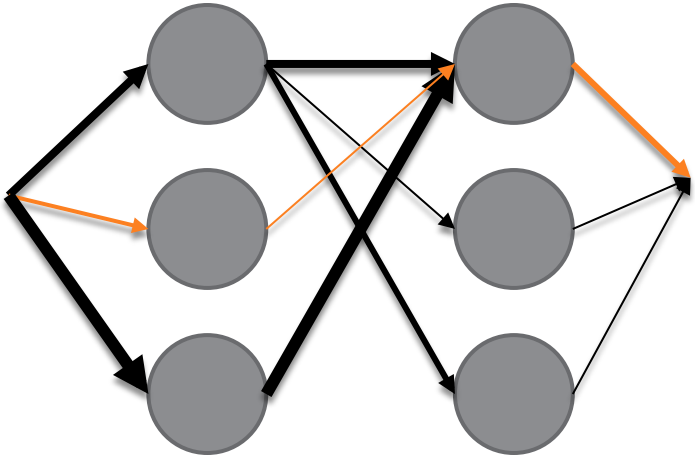


Generation

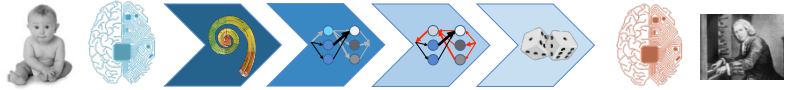


END

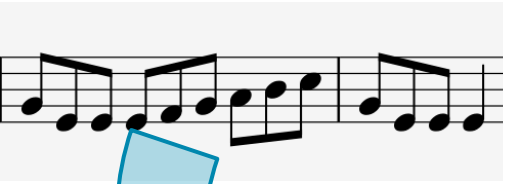
note[n]



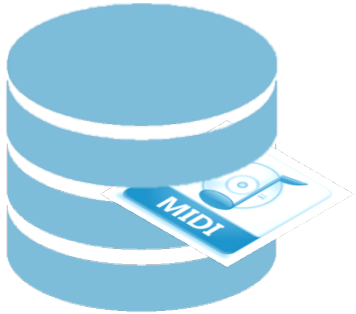
Pr(note[n+1])



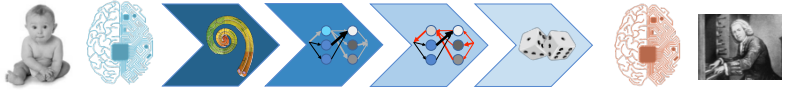
Generation | Generated corpus



END



sites.google.com/view/bachprop/



Generation | Temperature Sampling

- Temperature of sampling can be adapted to the confidence we give to the model predictions
- A model trained with a corpus that exhibits many repetition of patterns, will generate scores with more examples of these repetitions for lower sampling temperatures
  - Lower temperature will reduce the probability to select an undesired note that is not part of the pattern to be repeated

Evaluation | Turing Test

Quel est votre niveau de musique? ○ Novice ○ Intermédiaire ● Expert

---

Extraits musicaux Ecoutez les extraits et répondez aux deux questions.

1. Quel extrait a été composé par un humain (vs machine)? 1 ● 2 ○

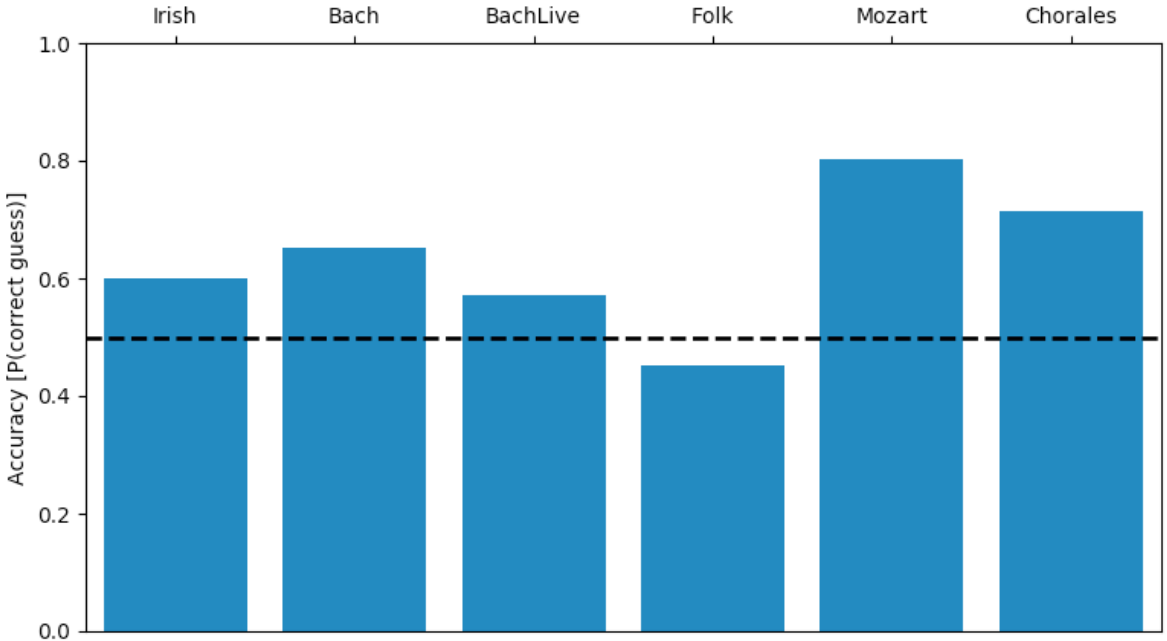
(Pour 2 réponses correctes à la suite, vous recevrez une récompense)

2. Quel extrait préferez-vous? 3 ● 4 ○

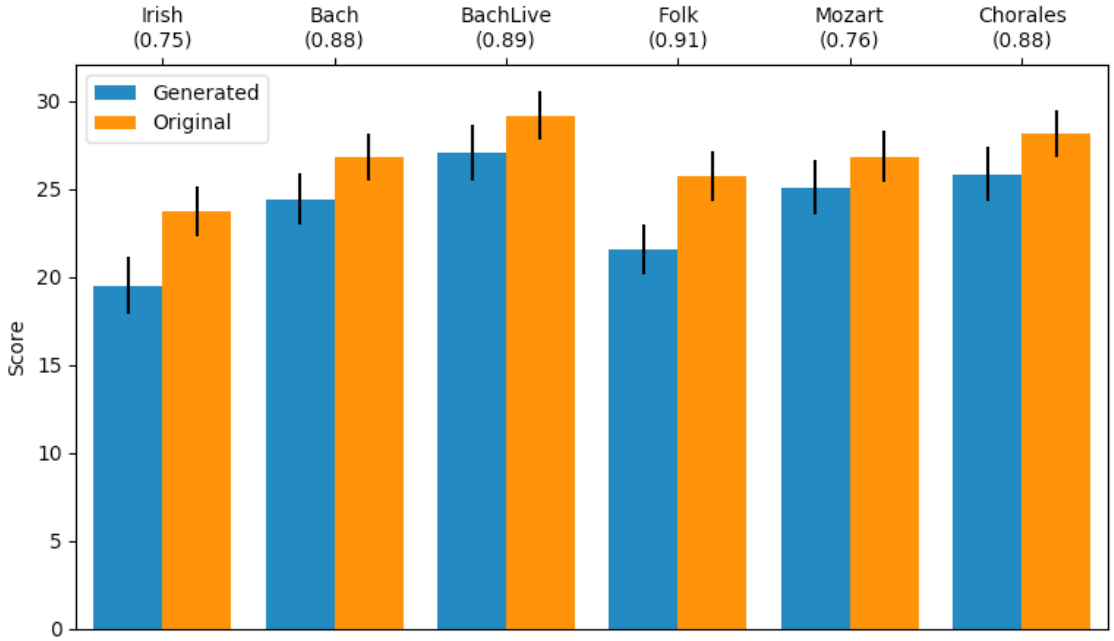
Appuyez sur **Submit** lorsque vous avez répondu aux deux questions.

Résultats 224 votes

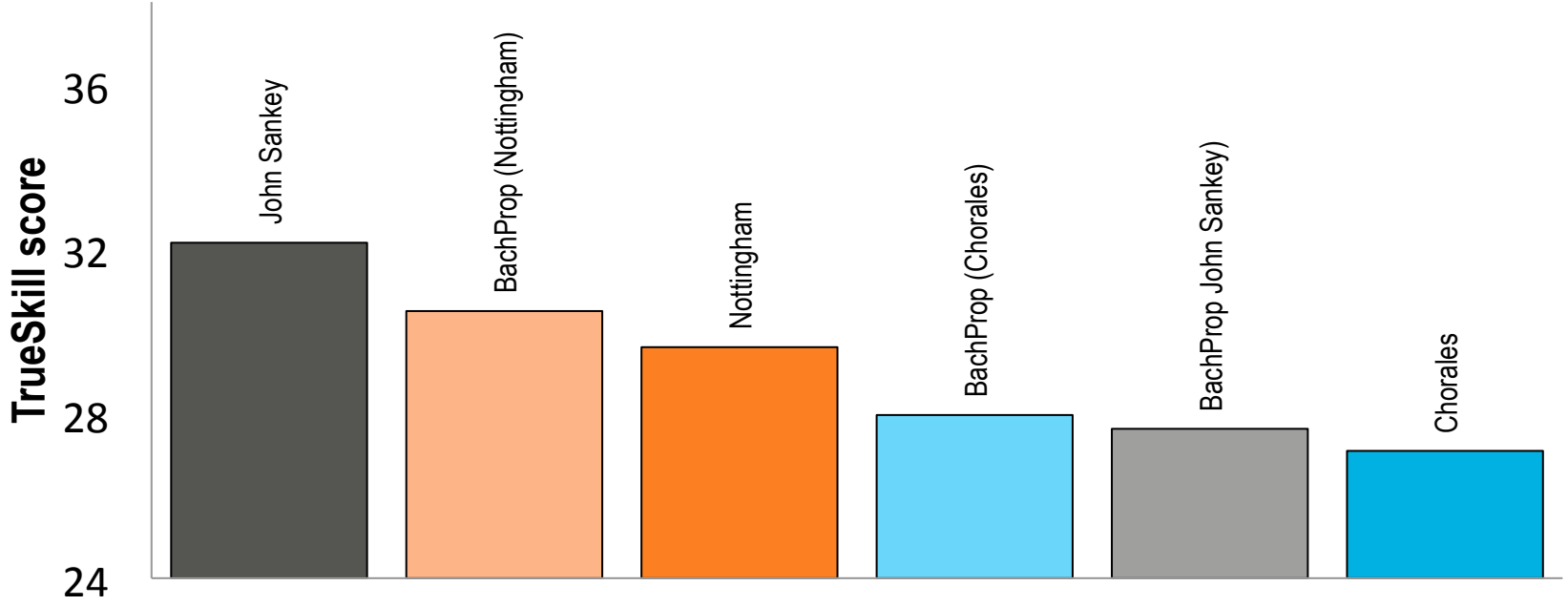
Evaluation | Turing Test



Evaluation | Scoring corpora



Evaluation | Subjective



https://www.crowdai.org/challenges/ai-generated-music-challenge

crowdAI

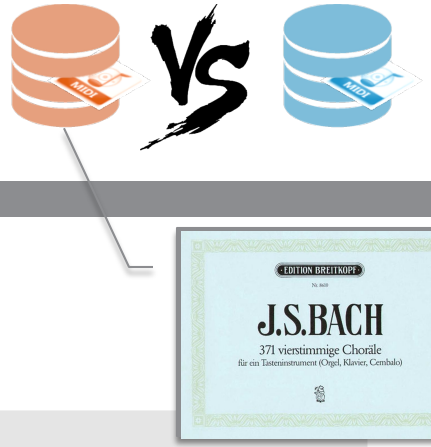
Other models | Controls

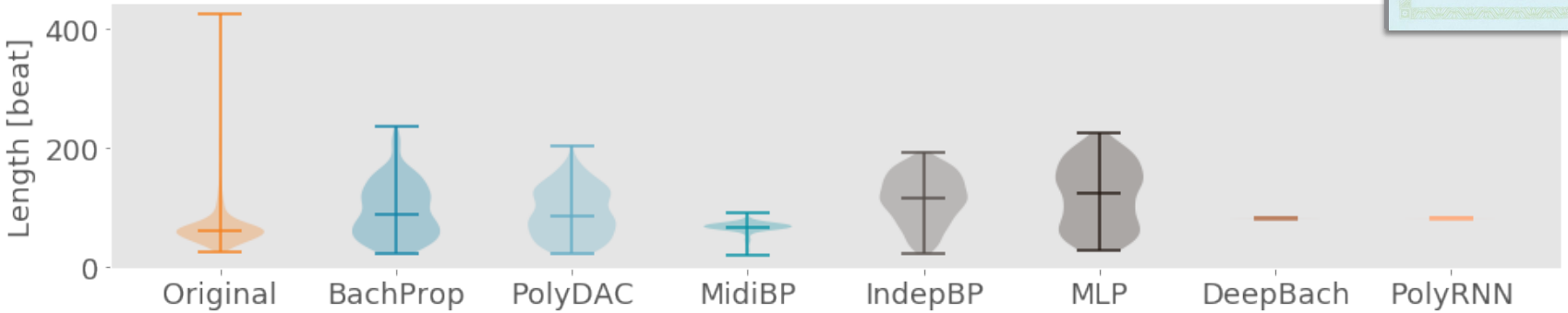
- **MLP**
  - 3 ReLU layers; History of 5 notes
- **PolyDAC**
  - 3 separate networks
- **IndepBP**
  - Independence between note features
- **MidiBP**
  - Uses MIDI data structure
- **DeepBach**
  - Temporal discretization; 4 voices; Designed for the Bach Chorales dataset
- **PolyRNN**
  - Temporal discretization; NEW_NOTE(PITCH), CONTINUED_NOTE(PITCH) and STEP_END events

Bach Chorales| Comparison of Architectures

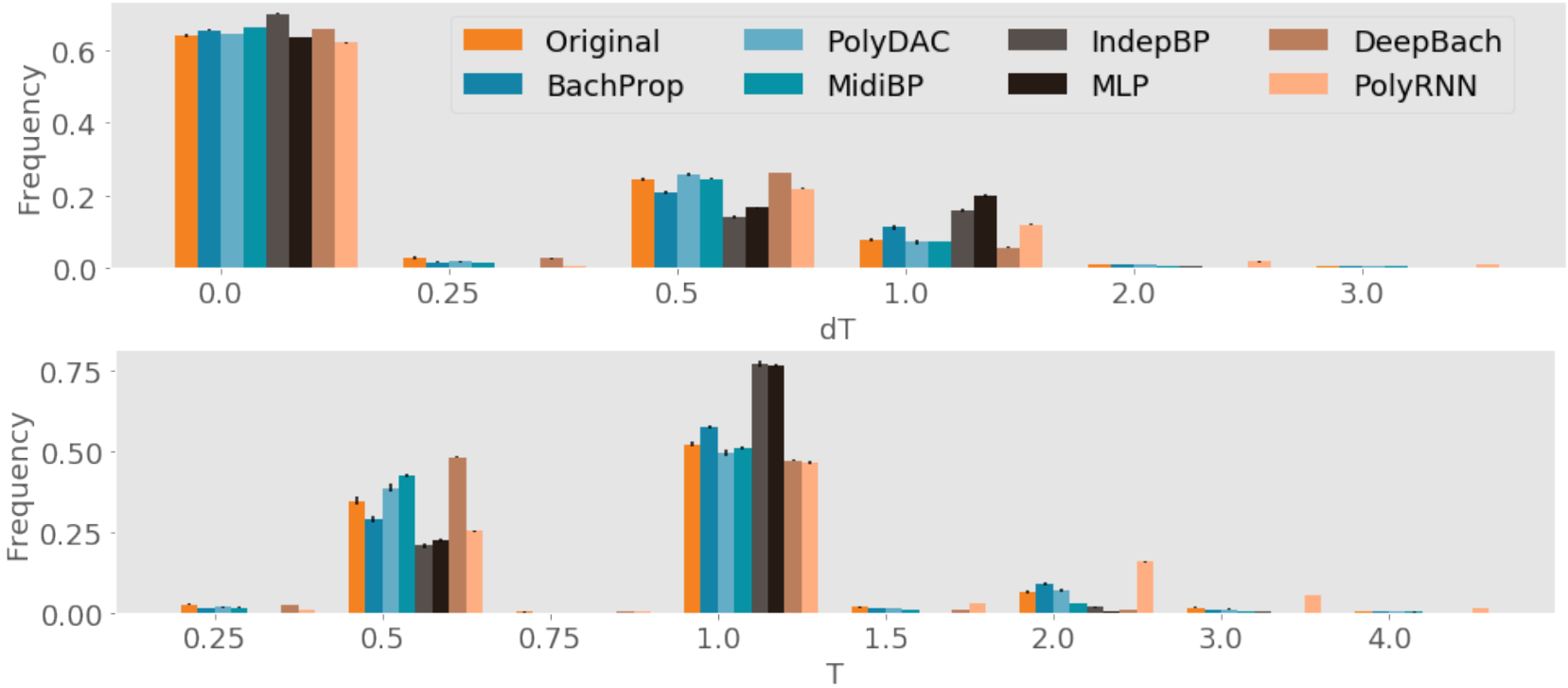
MODEL	NLL	dT	T	P
BACHPROP	0.419	0.97	0.91	0.77
POLYDAC	0.647	0.97	0.94	0.69
INDEPBP	0.647	0.97	0.75	0.63
MLP	0.796	0.95	0.76	0.49

Evaluation | Lengths

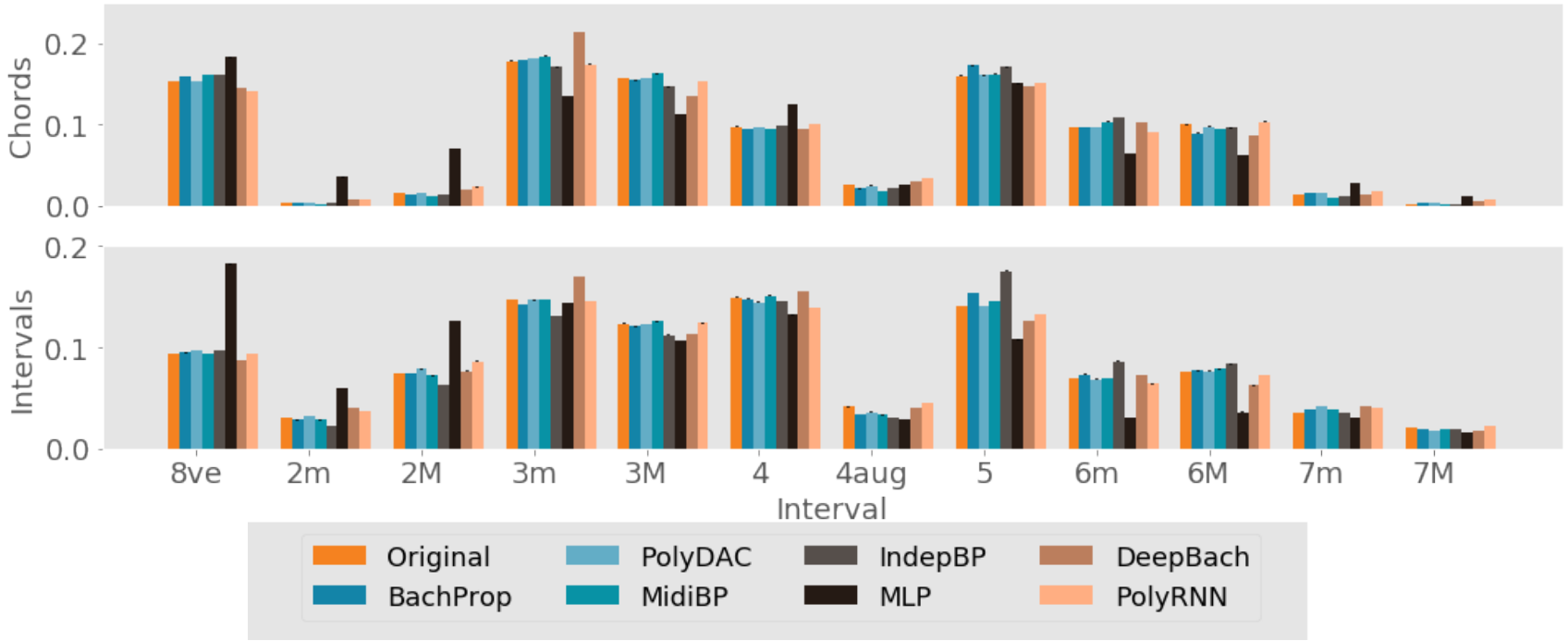




Evaluation | Rhythms

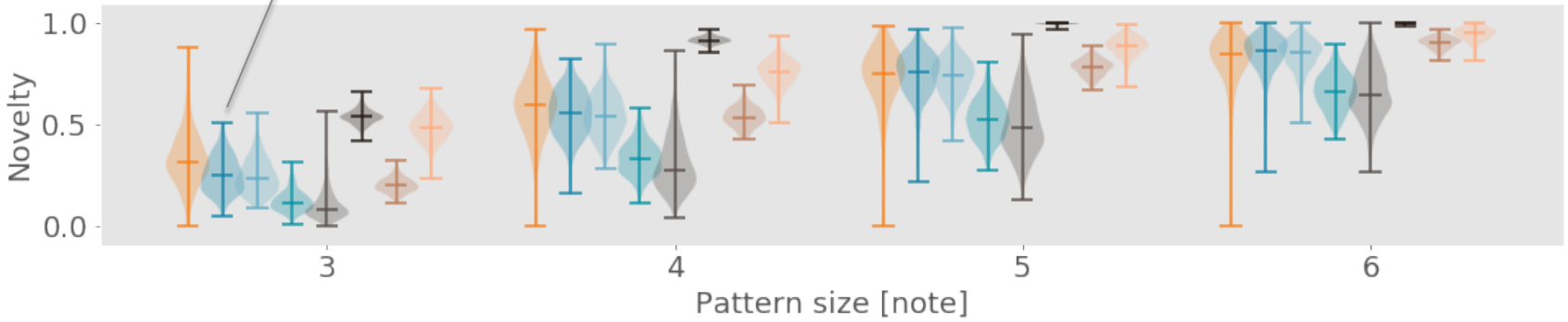


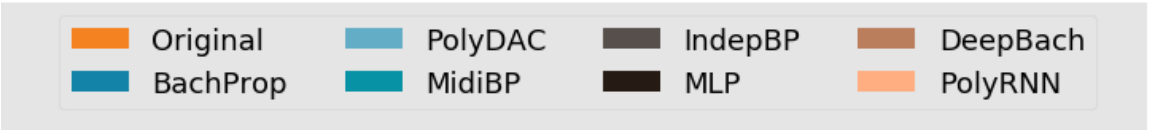
Evaluation | Intervals



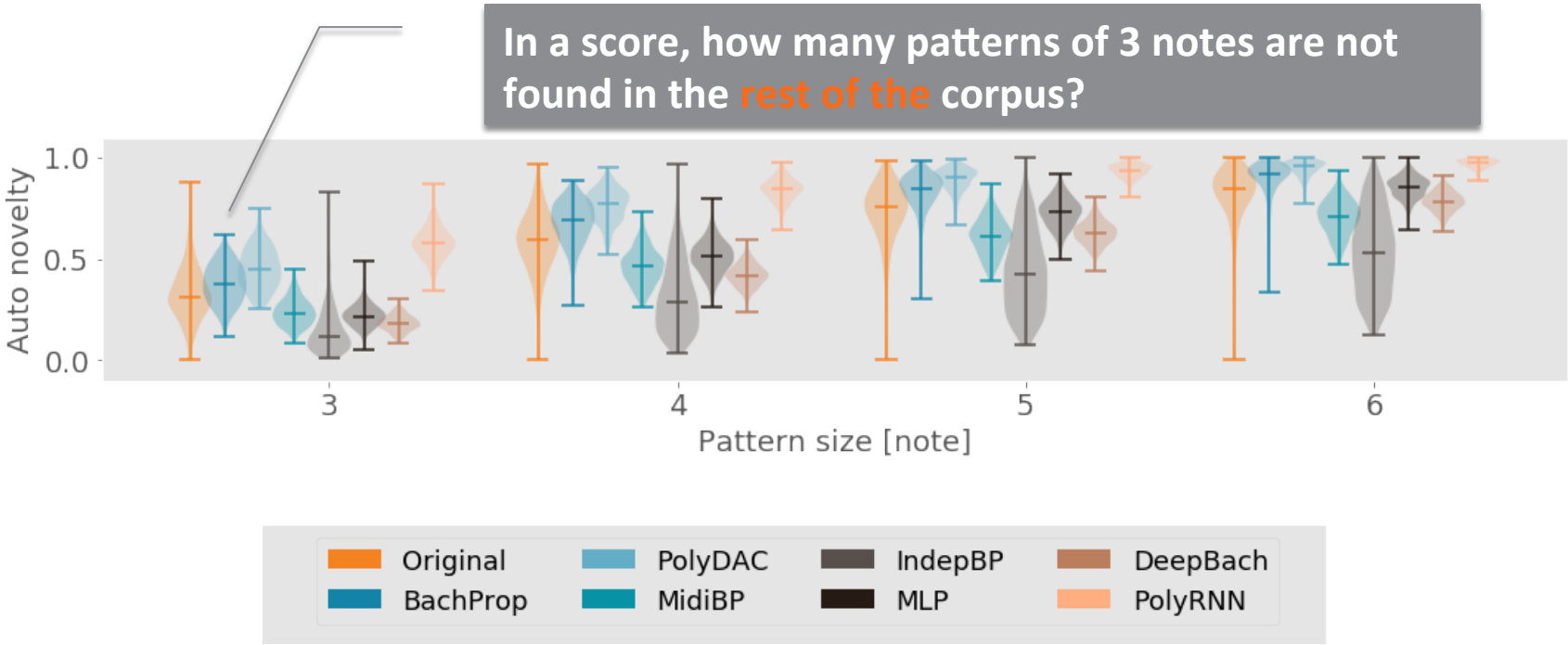
Evaluation | Originality

In a score, how many patterns of 3 notes are not found in the original corpus?





Evaluation | Diversity



Evaluation | Take Home Message

- Plagiarism-free music (*creativity*)
  - Novelty profiles
- Predefined style or mood
  - Training on a *homogeneous* dataset
  - Comparative statistics
- Pleasant to listen to
  - Subjective evaluation
- Diversity in generated pieces
  - Auto-novelty

Contributions

1. Note-sequence based representation of music with minimal distortion of the rhythm for training neural network models
2. A network architecture that learns to generate pleasant music in this representation
3. A set of metrics to compare generative models that operate on different representations of music

Live interpretation | Ada String Quartet

sites.google.com/view/bachprop/

Youtube Florian Colombo

What's next?

- Repetitions are intrinsic to music
  - Probabilistic models are by design bad at generating repetitions
    - Structured generation
    - Higher order grammar
    - Attention mechanisms
- Improving the model with future direction information
  - Condition on harmony progression

Aims of Grammar-ANNs Interaction

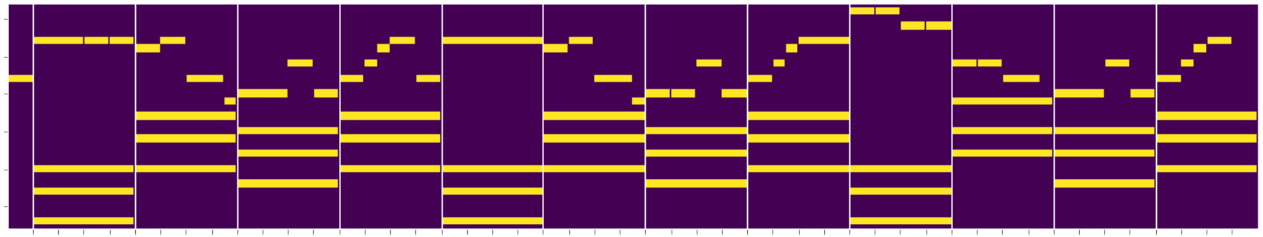
- Improve quality/structure of generated music by constraining the rules of generation
- thereby keeping the activity and dynamics of the networks in *comfortable* states
- And preventing mistakes that could bring the networks in *uncomfortable* states

Structured Generation | Algorithm

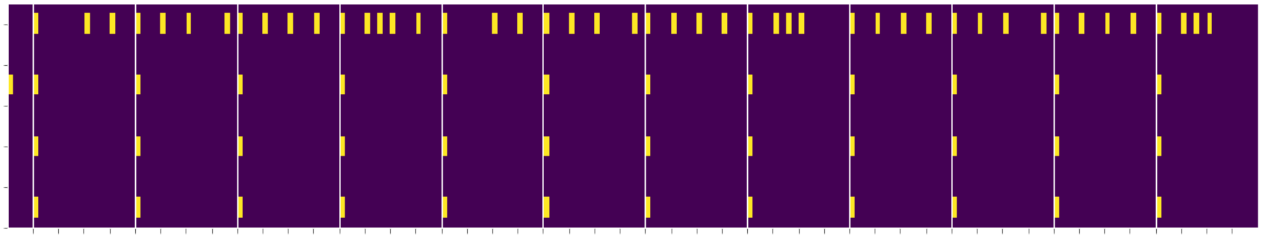
1. Compute reference repetition profiles
2. Impose the reference dT repetition profile to generate a new dT sequence from the dT network
3. Given this new dT sequence, impose the reference T repetition profile to generate a new T sequence from the T network
4. Given these new dT and T sequences, impose the reference P repetition profile to generate a new P sequence from the P network

Repetition Profile | dT Example

- Automatic upbeat and metric extraction



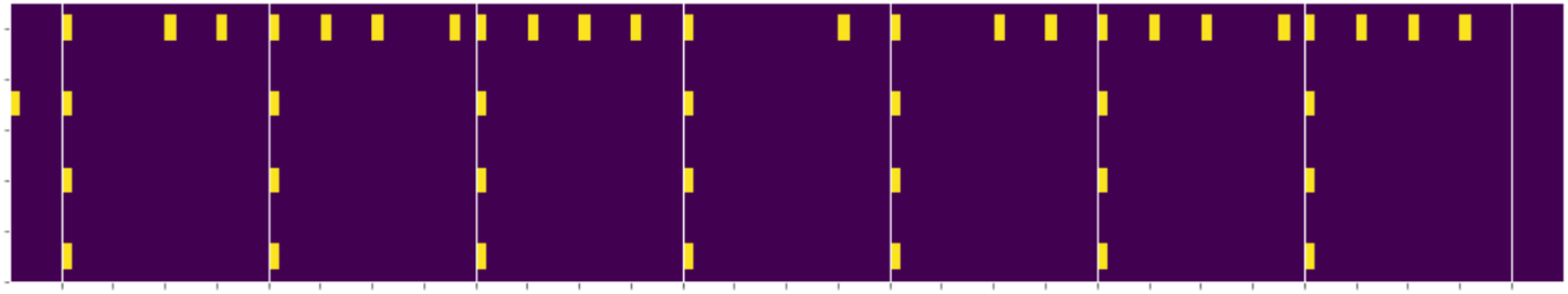
- Repetition profile is extracted by comparing bars



[0 0 0 0 1 2 3 0 3 2 3 8]

Structured Generation | dT

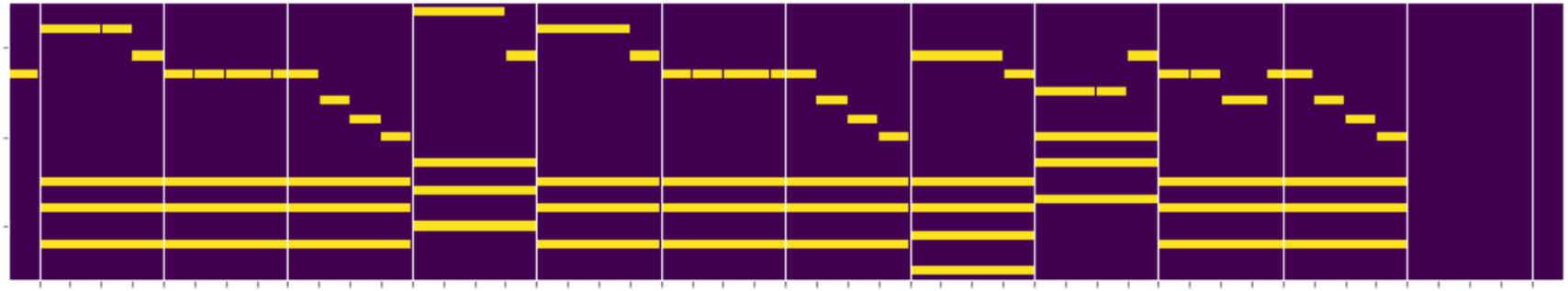
```
[0, 0, 0, 0, 1, 2, 3]
[0, 3, 2, 3, 8]
```



```
Updating hidden state: 100%|██████████| 46/46 [00:04<00:00, 10.40it/s]
Generating continuation: 100%|██████████| 64/64 [00:06<00:00,  9.81it/s]
```

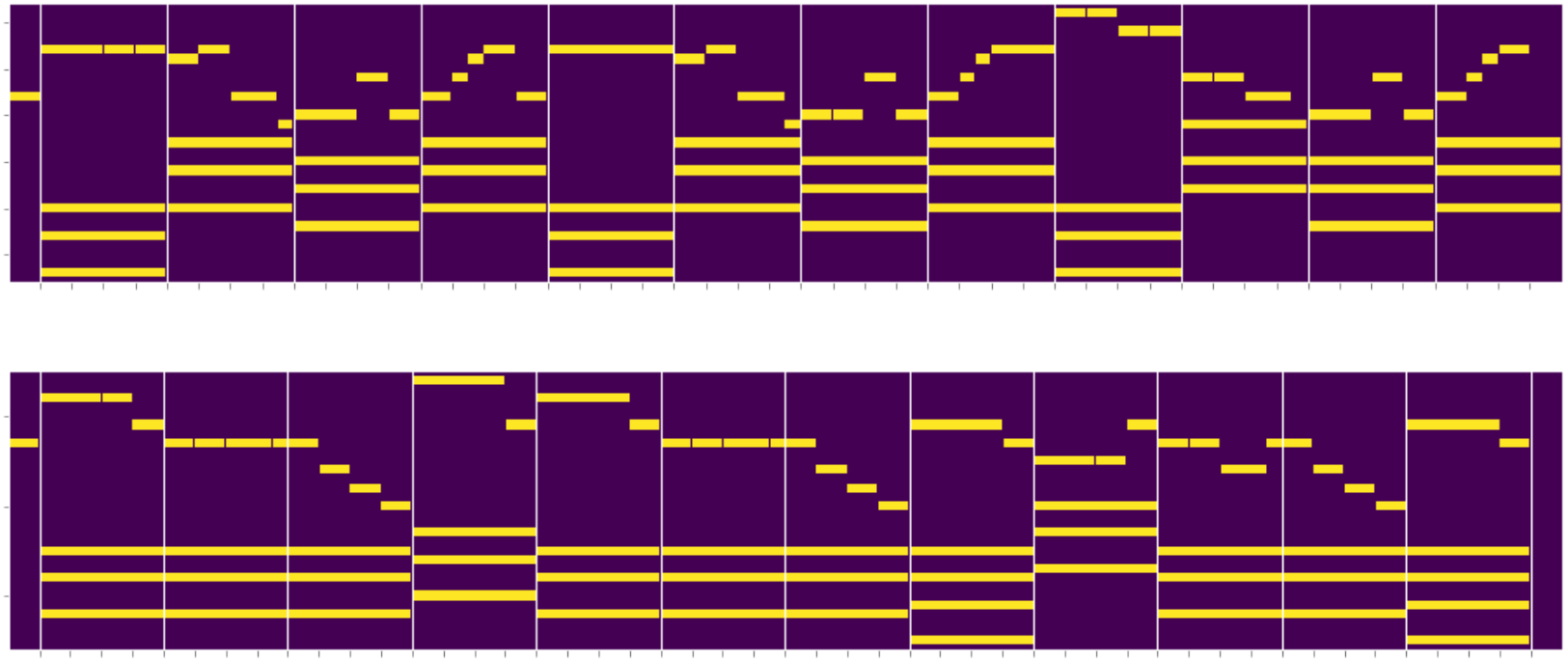
Structured Generation | P

```
[0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 3]
[8]
```

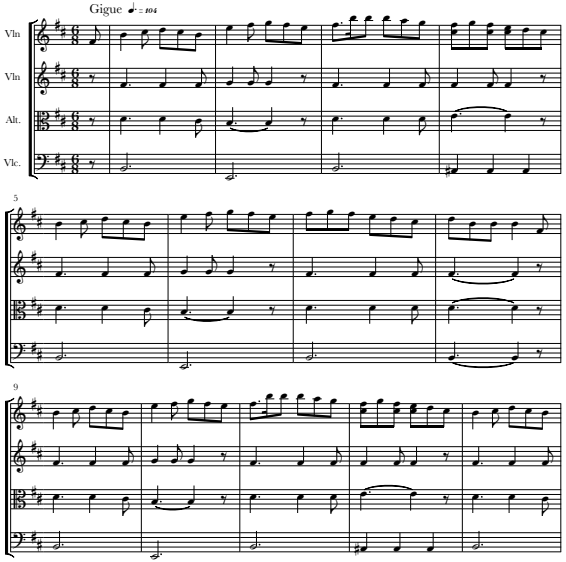


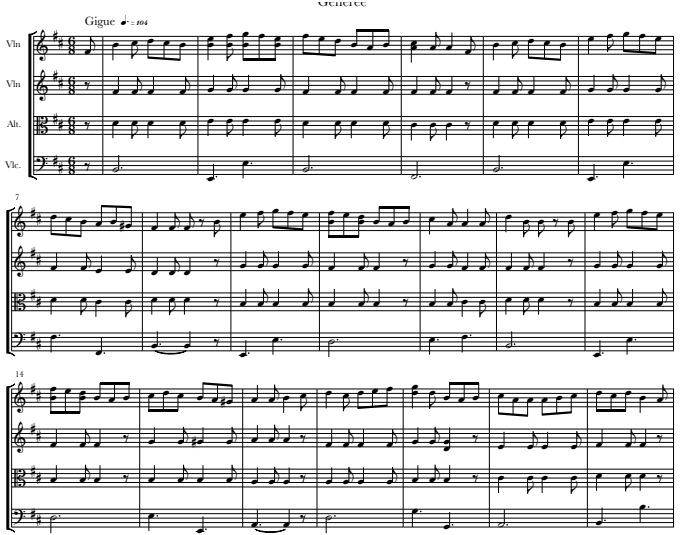
```
Generating a P continuation | dT: 100%|██████████| 80/80 [00:14<00:00,  5.44it/s]
```

Reference vs Generated

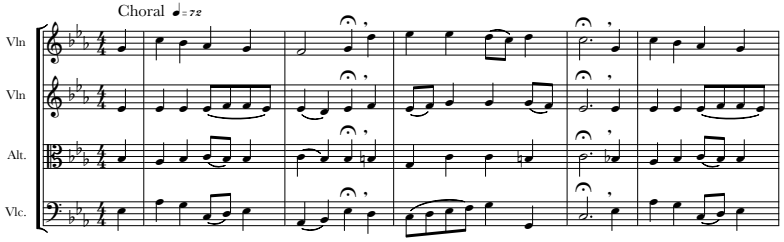


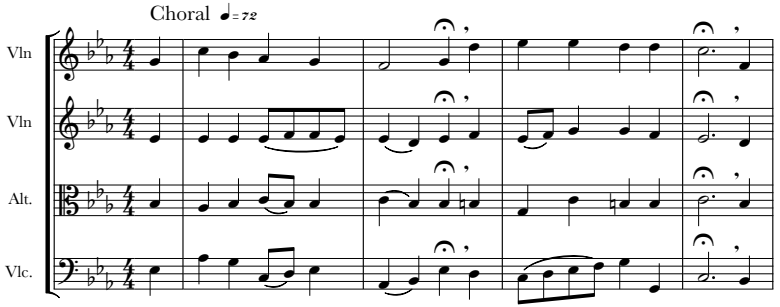
Structured generation | Examples

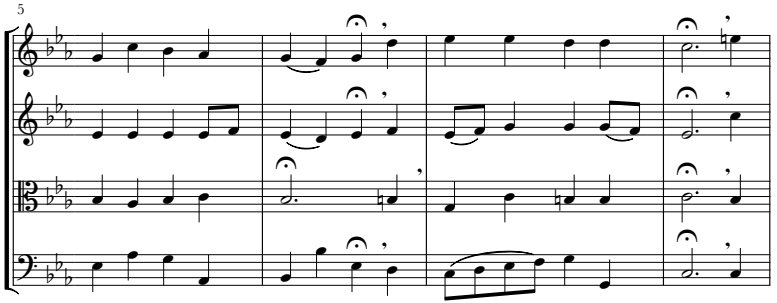




Structured generation | Examples



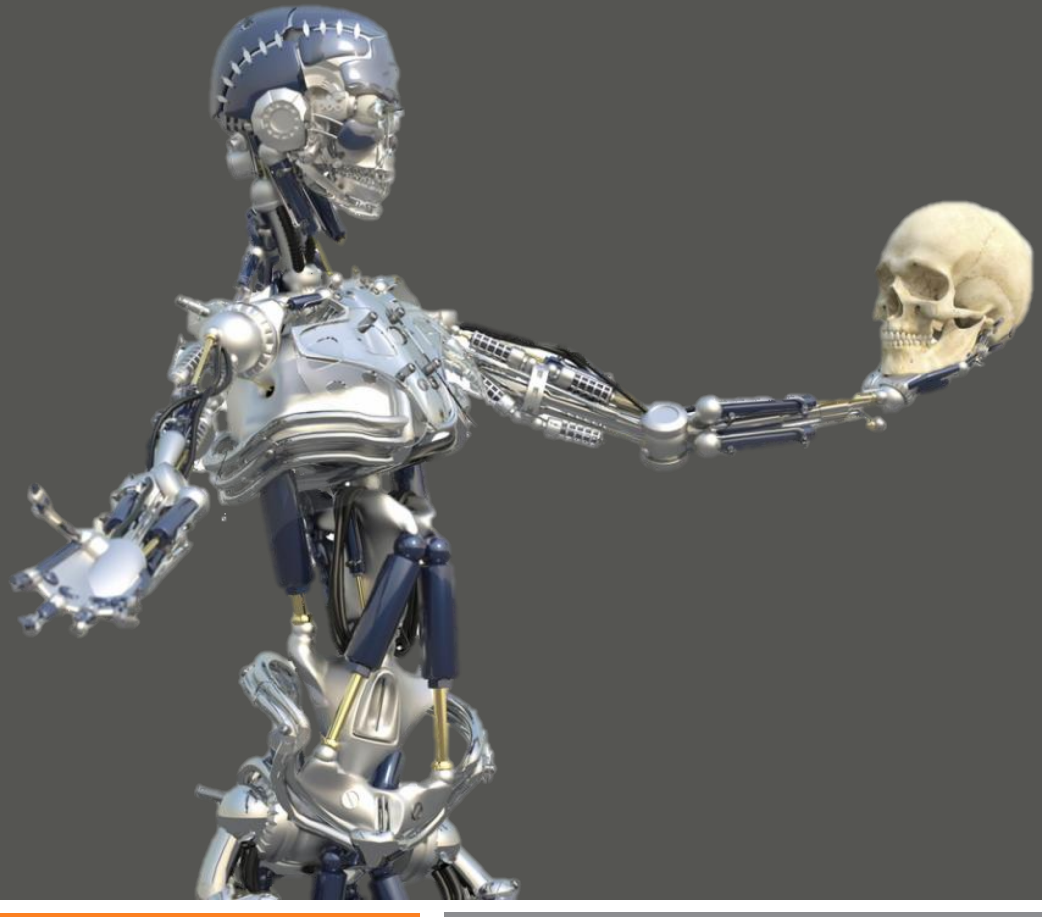




Thanks for your attention!

Wulfram Gerstner

Johanni Brea
Samuel P. Muscinelli



QUESTIONS?

Link to exercises and slides

https://github.com/FlorianColombo/MagliasoMusic2019