Deep Learning Techniques for Music Generation (5)

Jean-Pierre Briot

Jean-Pierre.Briot@lip6.fr

Laboratoire d'Informatique de Paris 6 (LIP6)

Sorbonne Université – CNRS

Programa de Pós-Graduação em Informática (PPGI)

UNIRIO

Recurrent

#1 Limitation – Generation and #2 Limitation – Fixed Length

- Works OK

But:

- Fixed input (and output) length

#1 Limitation – Generation and #2 Limitation – Fixed Length
Solution: Recurrent Network (RNN)

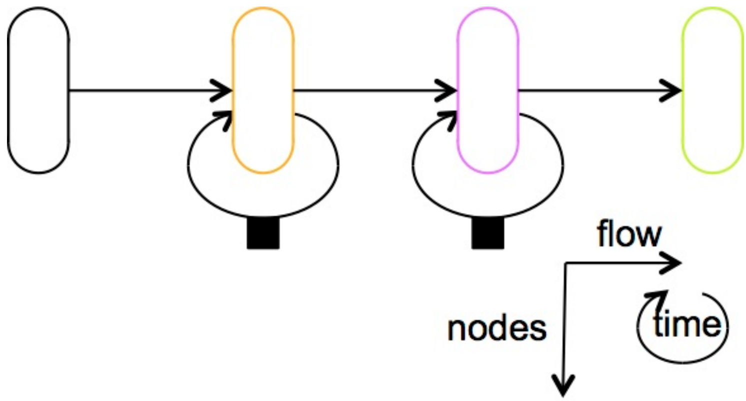
- Works OK

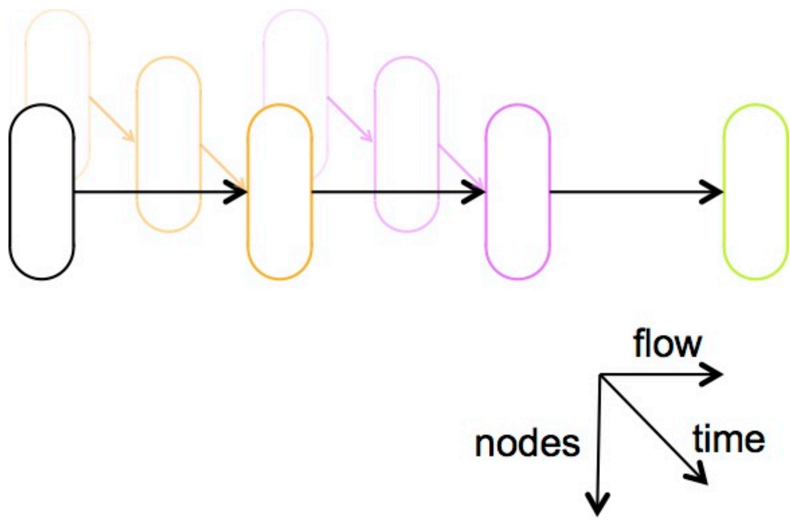
But:

- Fixed input (and output) length

Solution:

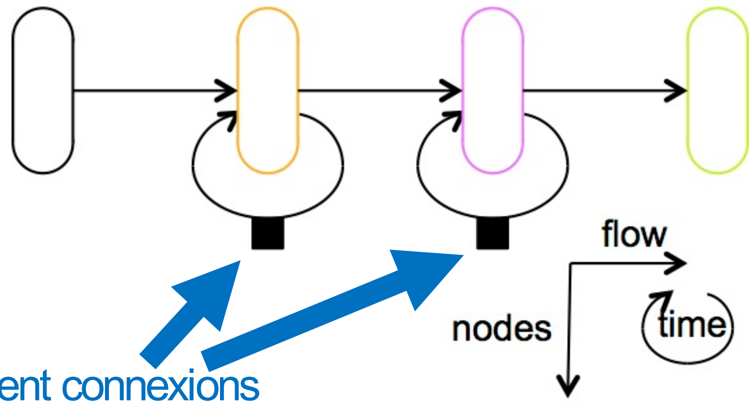
- Recurrent Network (RNN)
- Variable length
- Memorizes previous steps
- Predicts next step



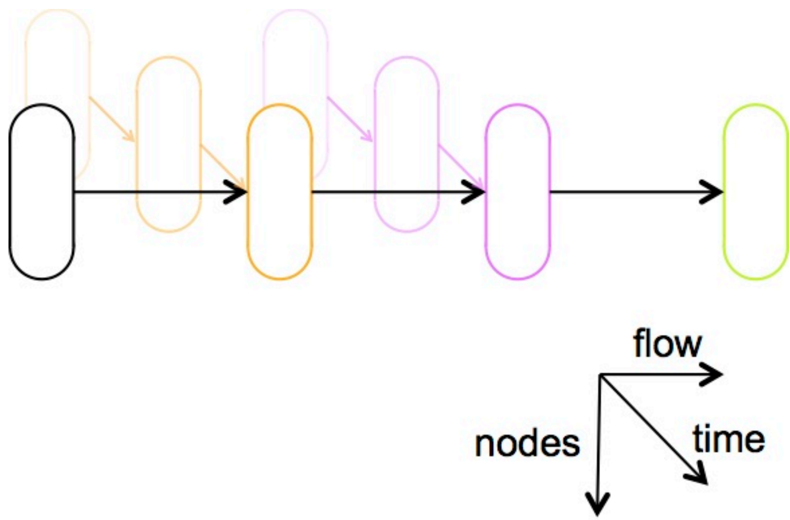


Recurrent Network (RNN)

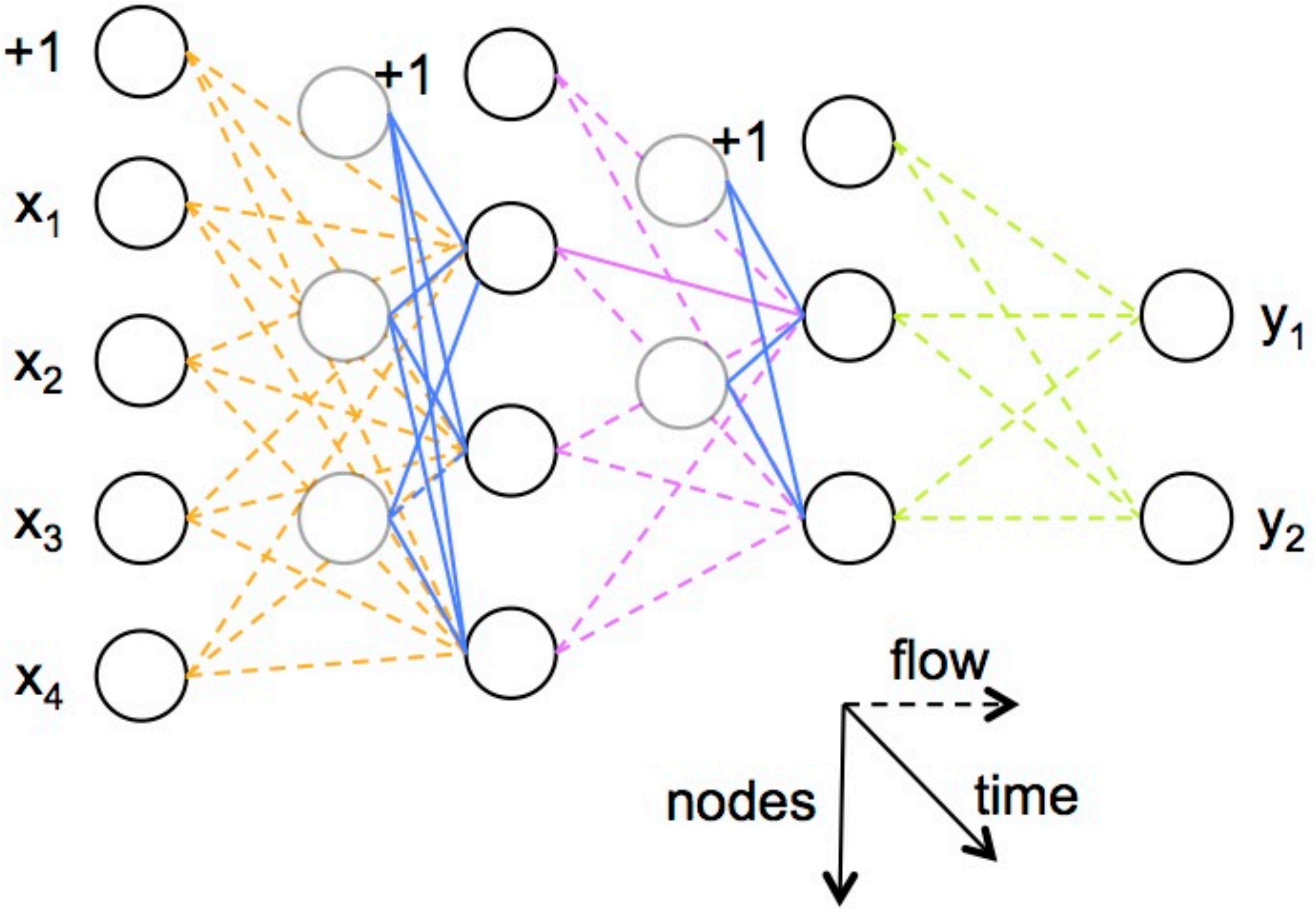
- Memorizes previous steps
- Can learn from previous step
- Predicts next step
- Can learn sequences



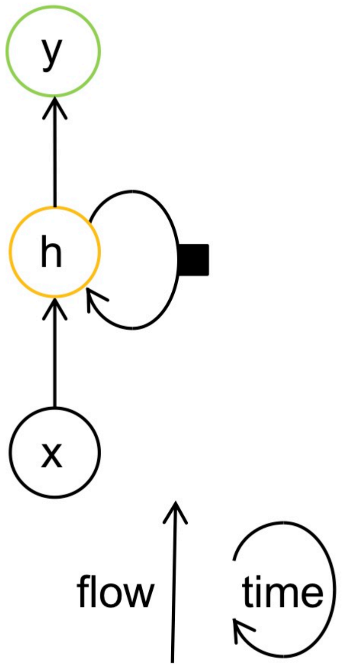
Recurrent connexions

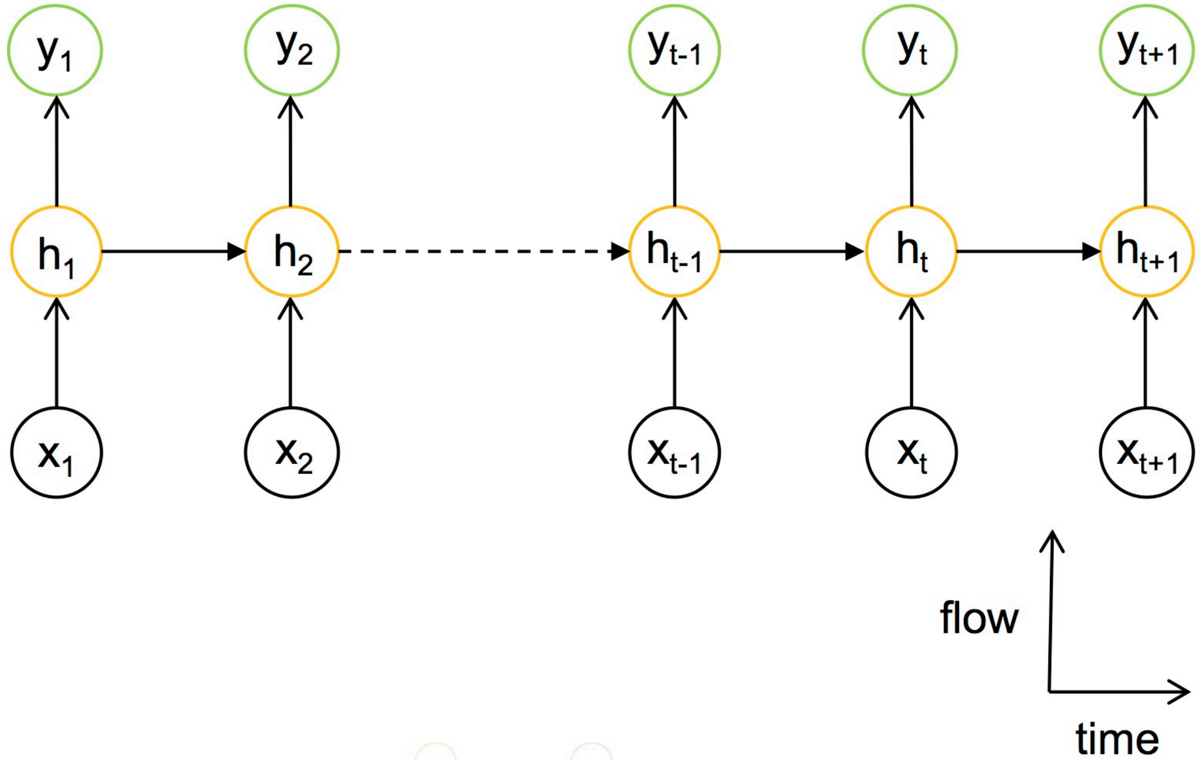


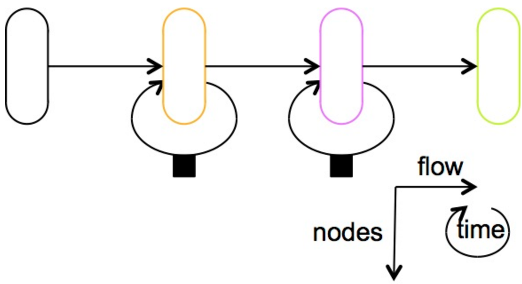
Recurrent Connexions

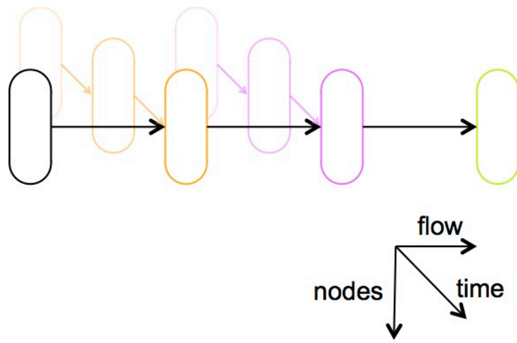


Alternative (More Common) Notation

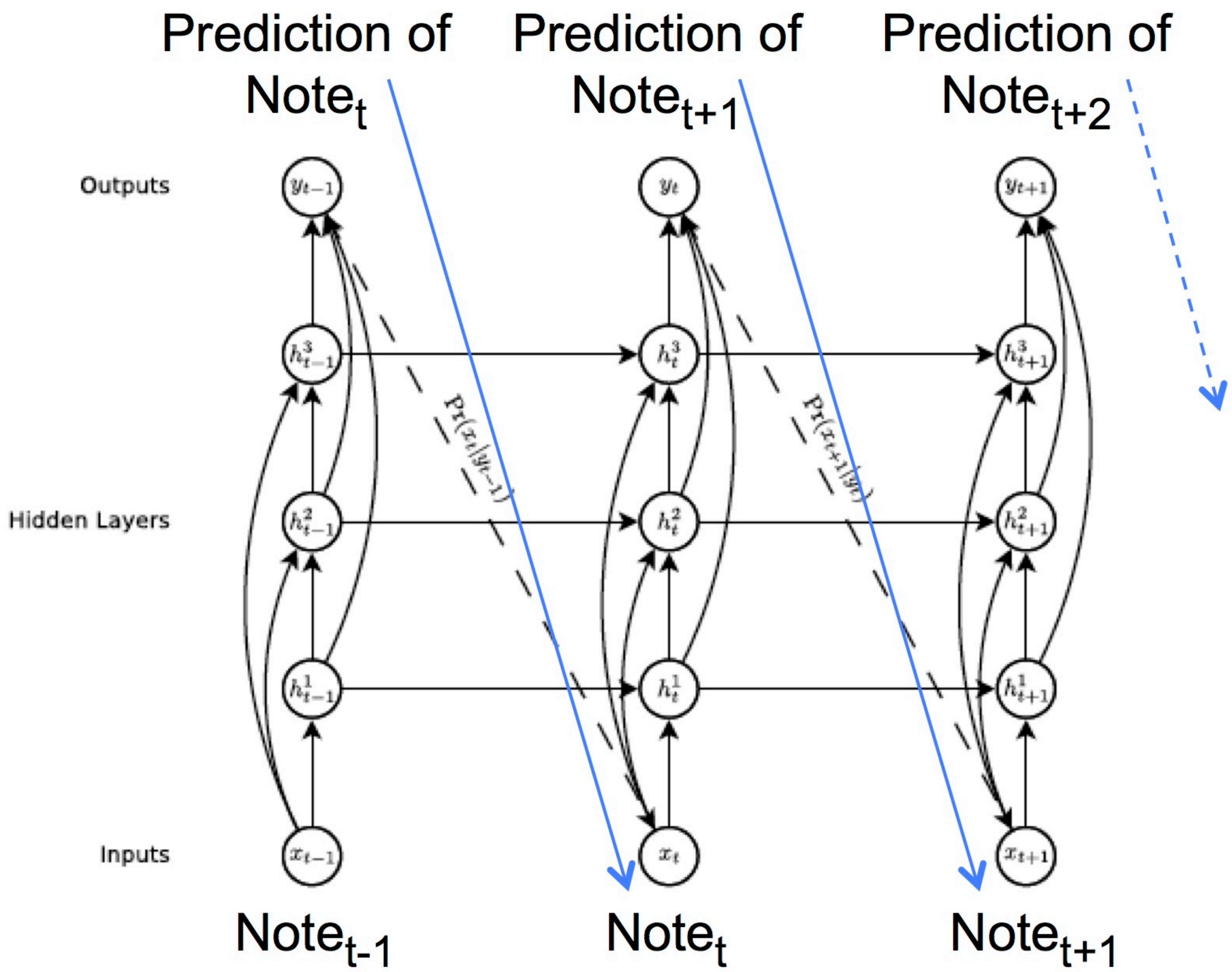






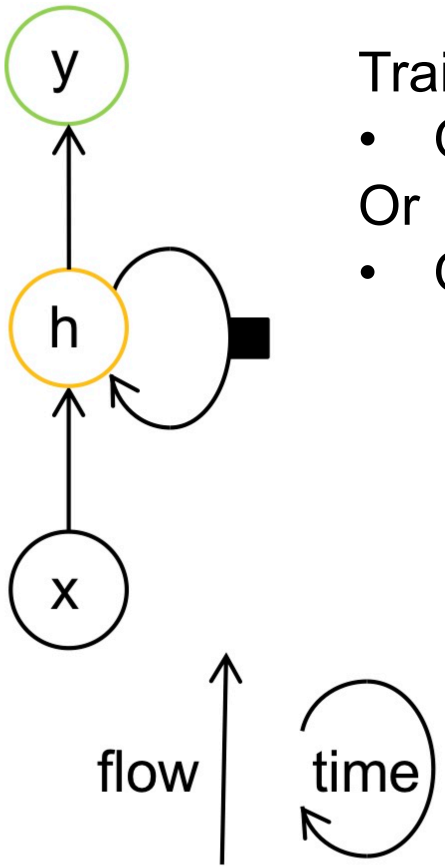


RNN Prediction



Training a RNN

y = Expected next note(x)

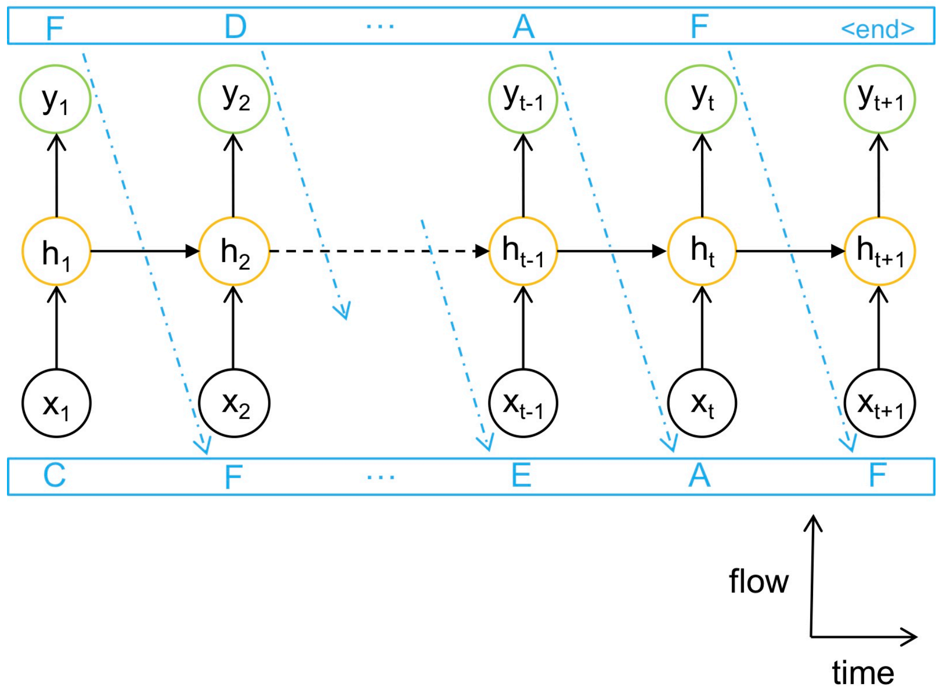


Training with

- One example (x, y) = one note

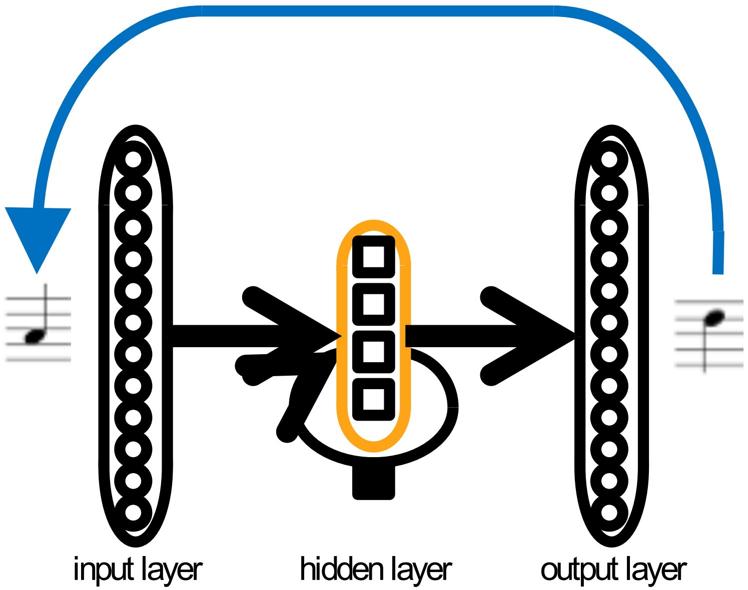
Or

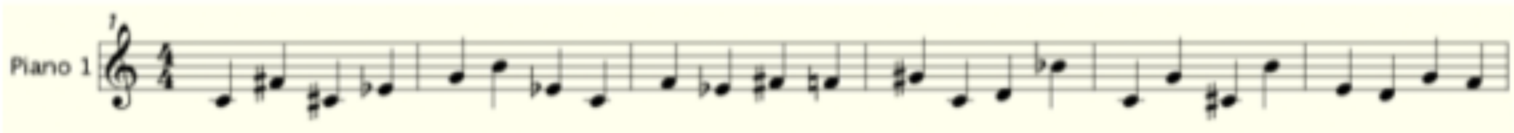
- One example (x, y) = one melody
  with x = y translated 1 step back



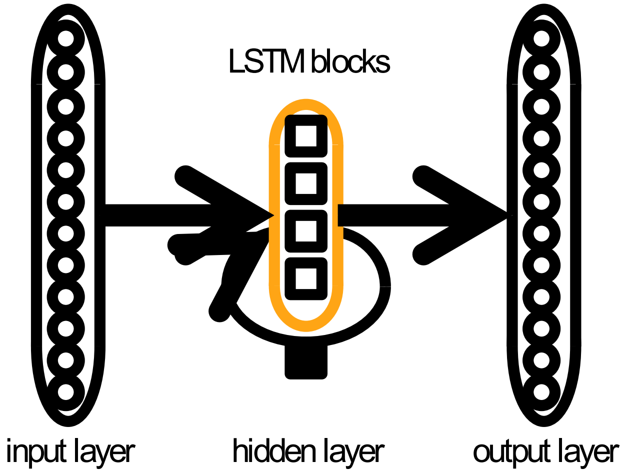
RNN Generation

- Iterated generation
  - Note by Note
  - Reinject Next Note to Produce Next Next Note
  - Arbitrary Length





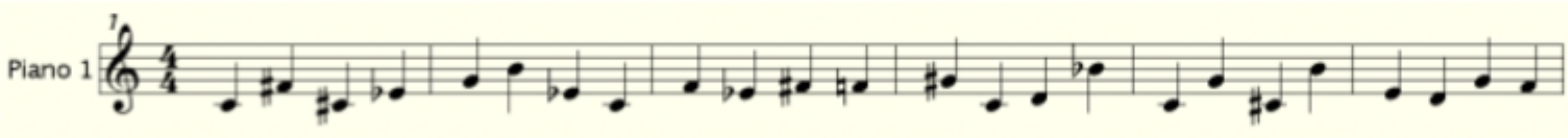
RNN – Iterative Feedforward – #1 Example



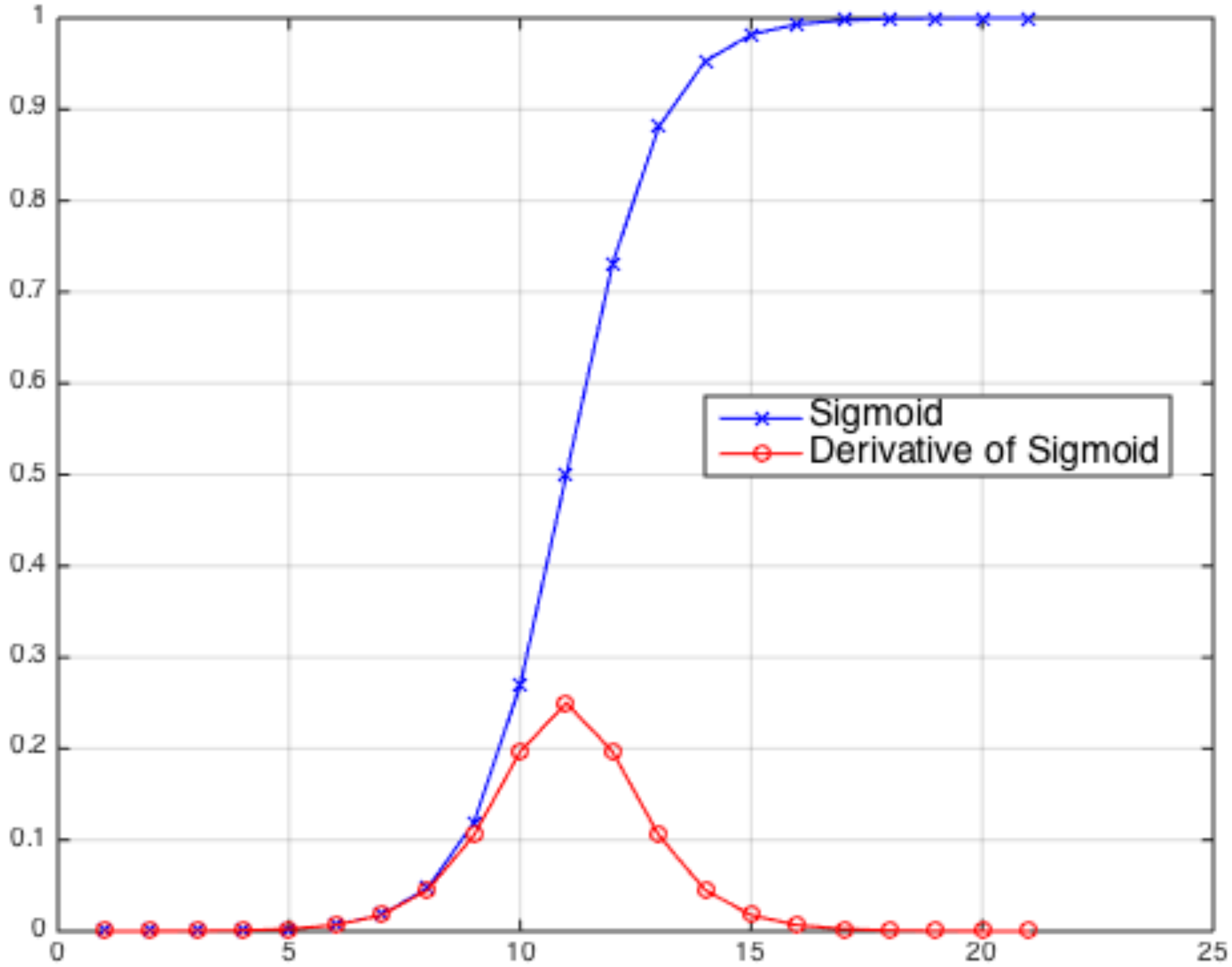

```
input_size = number_notes
lstm_layer_size = 32
output_size = number_notes

model = Sequential()
model.add(LSTM(lstm_layer_size,
              input_shape = (time_steps, input_size)))
model.add(Dense(output_size,
                activation = 'softmax'))
```

Synthetic corpus : arpeggio of C major chord
4 examples

X (note)	*y (next note)*
C	E
E	G
G	B
B	C

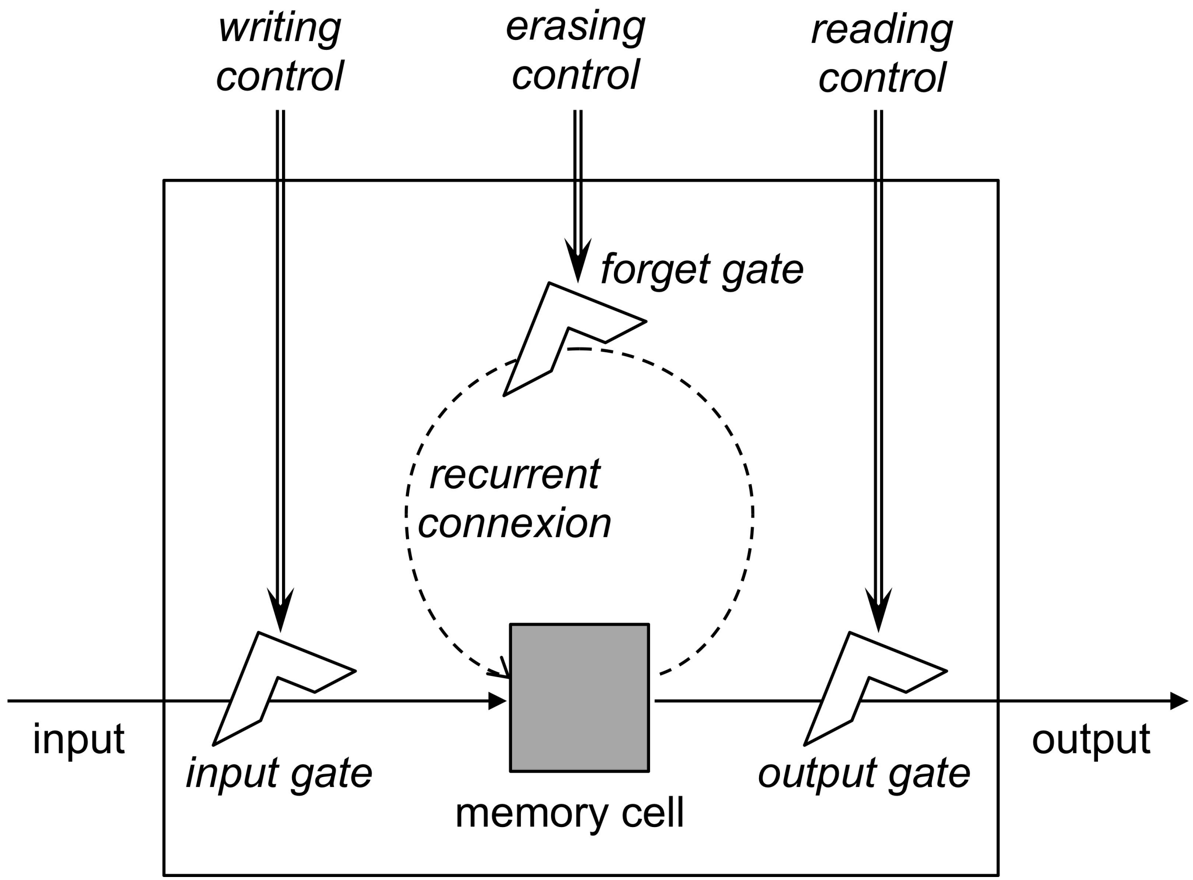


Gradient Vanishing/Explosion



LSTM (Long Short-Term Memory) [Hochreiter and Schmidhuber, 1997]

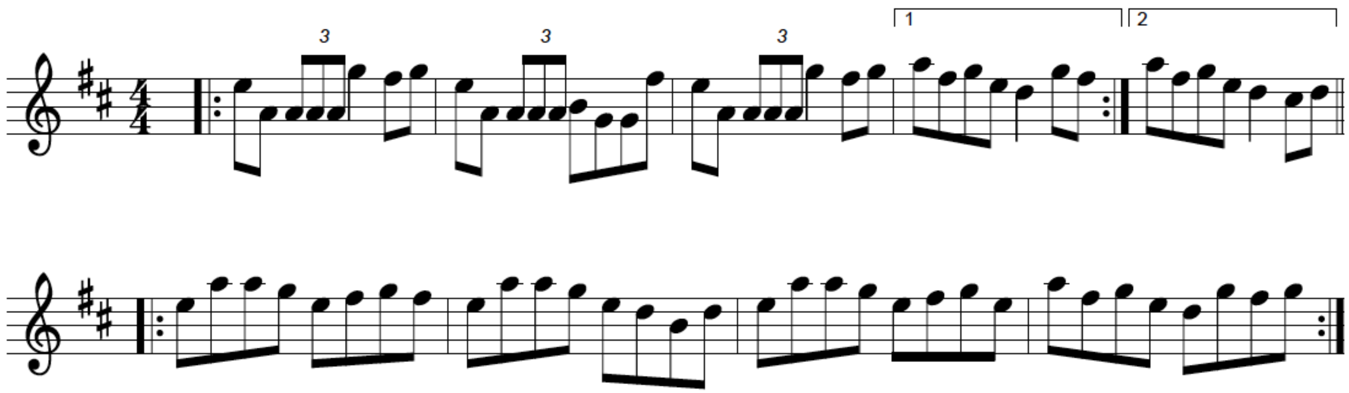
- Protection of Memory by Gates
- Gates are controlled by differentiable functions
- Thus subject to Training
- Training of the Meta-Level (Control)



RNN – Iterative Feedforward – #2 Example

- Ex: Celtic melody generation [Sturm et al., 2016]
- Celtic Folk Music Corpus (Melodies)
- Text Encoding (ABC Notation)

X: 1
T: A Cup Of Tea
R: reel
M: 4/4
L: 1/8
K: Amix
|:eA (3AAA g2 fg|eA (3AAA BGGf|eA (3AAA g2
fg|1afge d2 gf:|2afge d2 cd||
|:eaag efgf|eaag edBd|eaag efge|afge dgfg:|



RNN Celtic Melody Generation

- Iterated generation
  - Note by Note
  - Arbitrary Length
- Ex: Celtic melody generation [Sturm et al., 2016]
- Celtic Folk Music Corpus (Melodies)
- Text Encoding (ABC Notation)
- Ex. of Melody Generated

Played by a human accordeonist

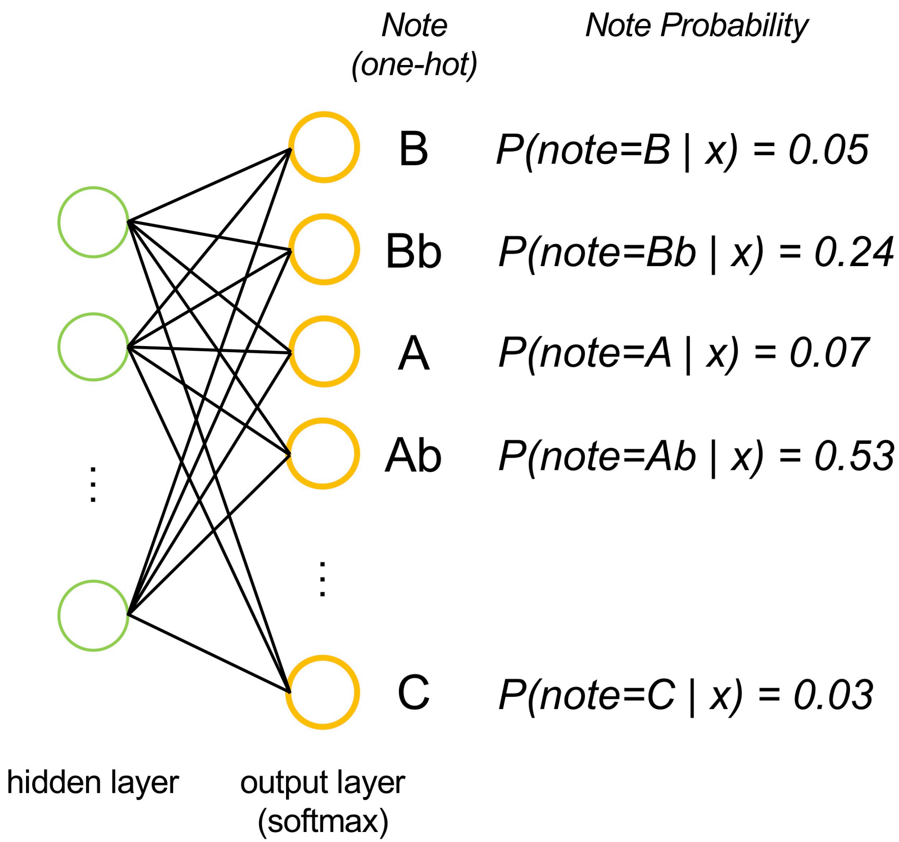
#3 Limitation – Variability

- No Variability in the Generation
- Because Neural Networks are Deterministic
  - Same Input -> Same Output
  - Same First Note -> Same Whole Melody Generated

- Solution:
  - Sampling

#3 Limitation – Variability – Solution: Sampling

- Input Representation: One-Hot Encoding
  - Corresponds to a Piano Roll Representation
- Softmax Ouput Layer
- Classification Task (between possible Notes)

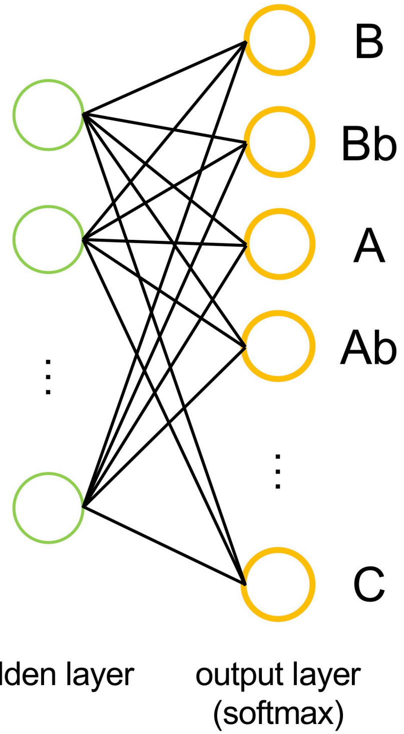


Sampling

- Deterministic Strategy:
  – Choose the Class (Note/Pitch) with the Highest Probability
- Sampling (Variability)
  – Sample within Possible Notes (Classes) (following the Probability Distribution)

Note (one-hot) *Note Probability*

np.random.multinomial(1, note_one_hot_encoding)



P(note=B | x) = 0.05

P(note=Bb | x) = 0.24

P(note=A | x) = 0.07

P(note=Ab | x) = 0.53

P(note=C | x) = 0.03

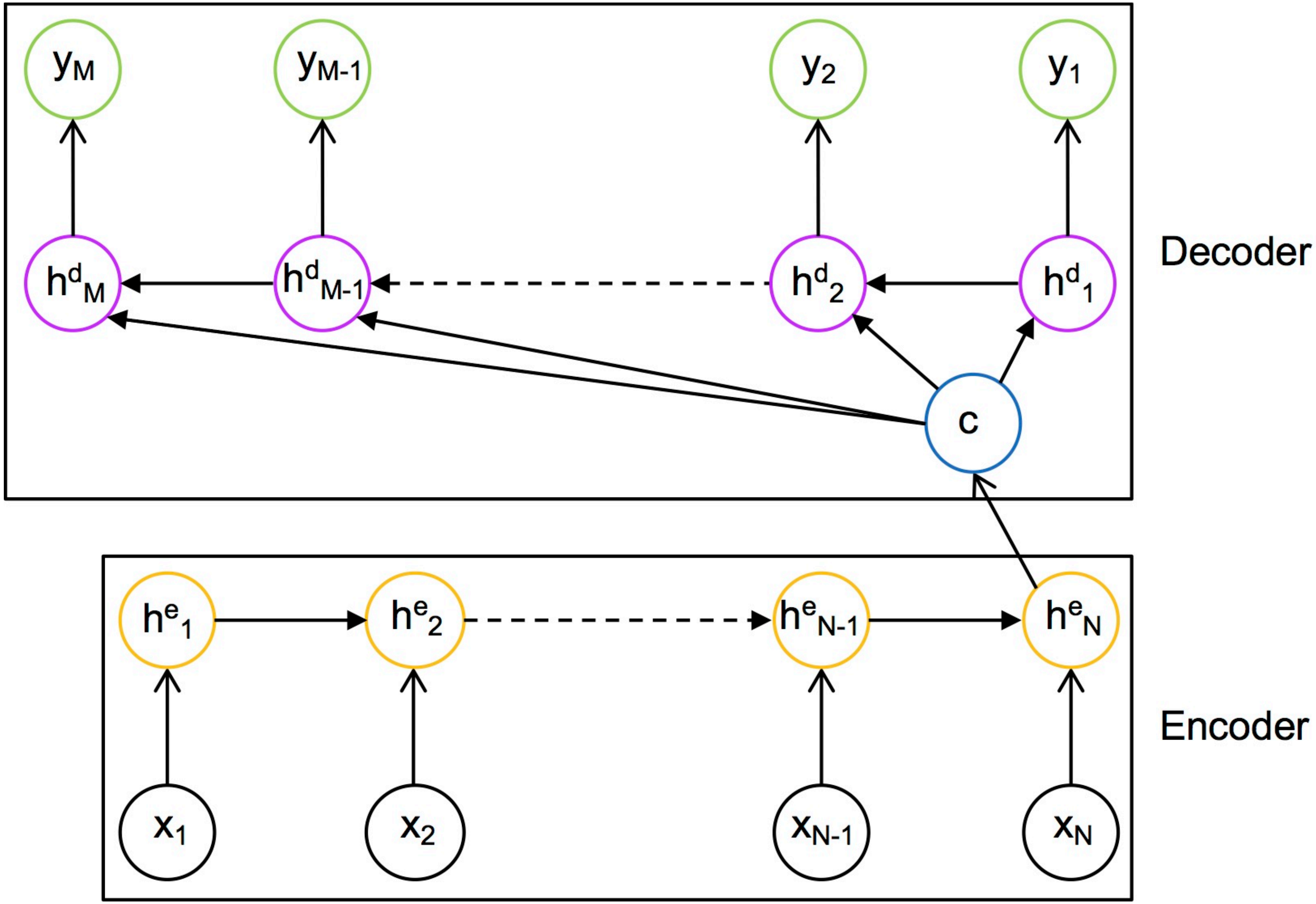
No Sampling

- In fact, Celtic melody generation [Sturm et al., 2016] is using sampling,
- Whereas Blues melody generation [Eck & Schmidhuber, 2002] is not

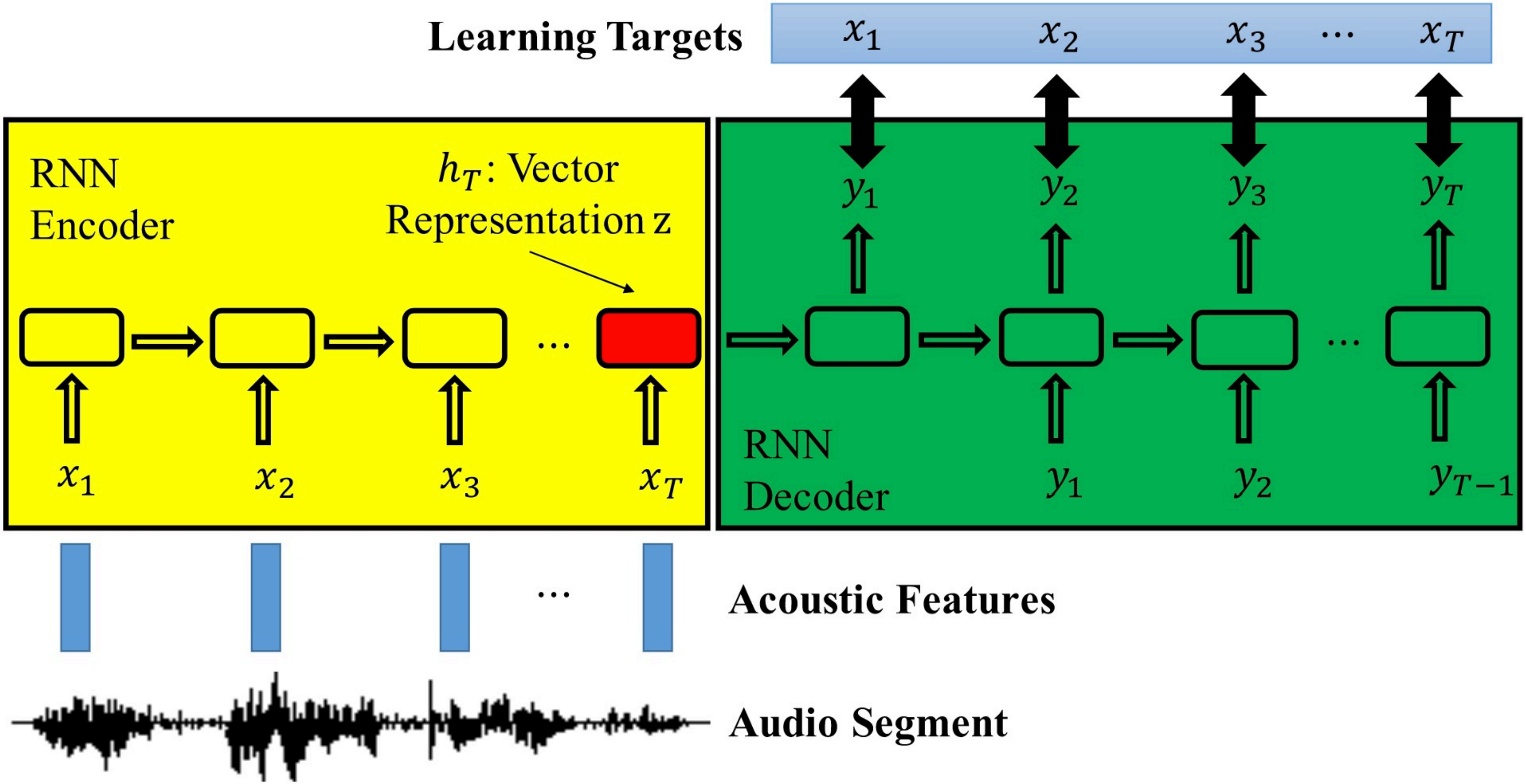
http://www.iro.umontreal.ca/~eckdoug/blues/lstm_0224_1510.mp3

RNN Encoder-Decoder

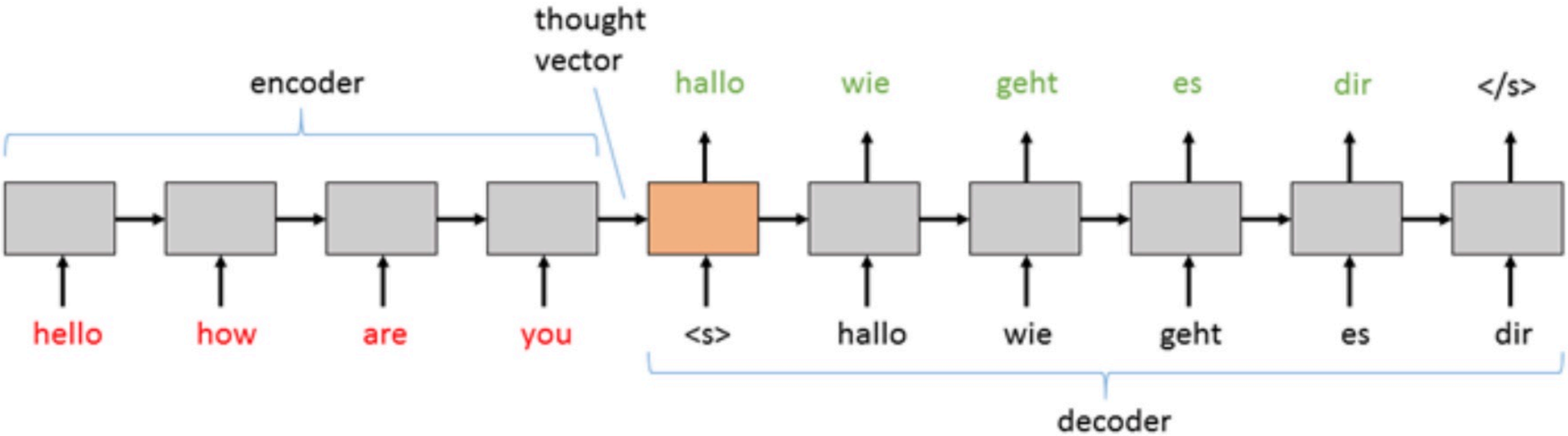
RNN Autoencoder : RNN Encoder-Decoder



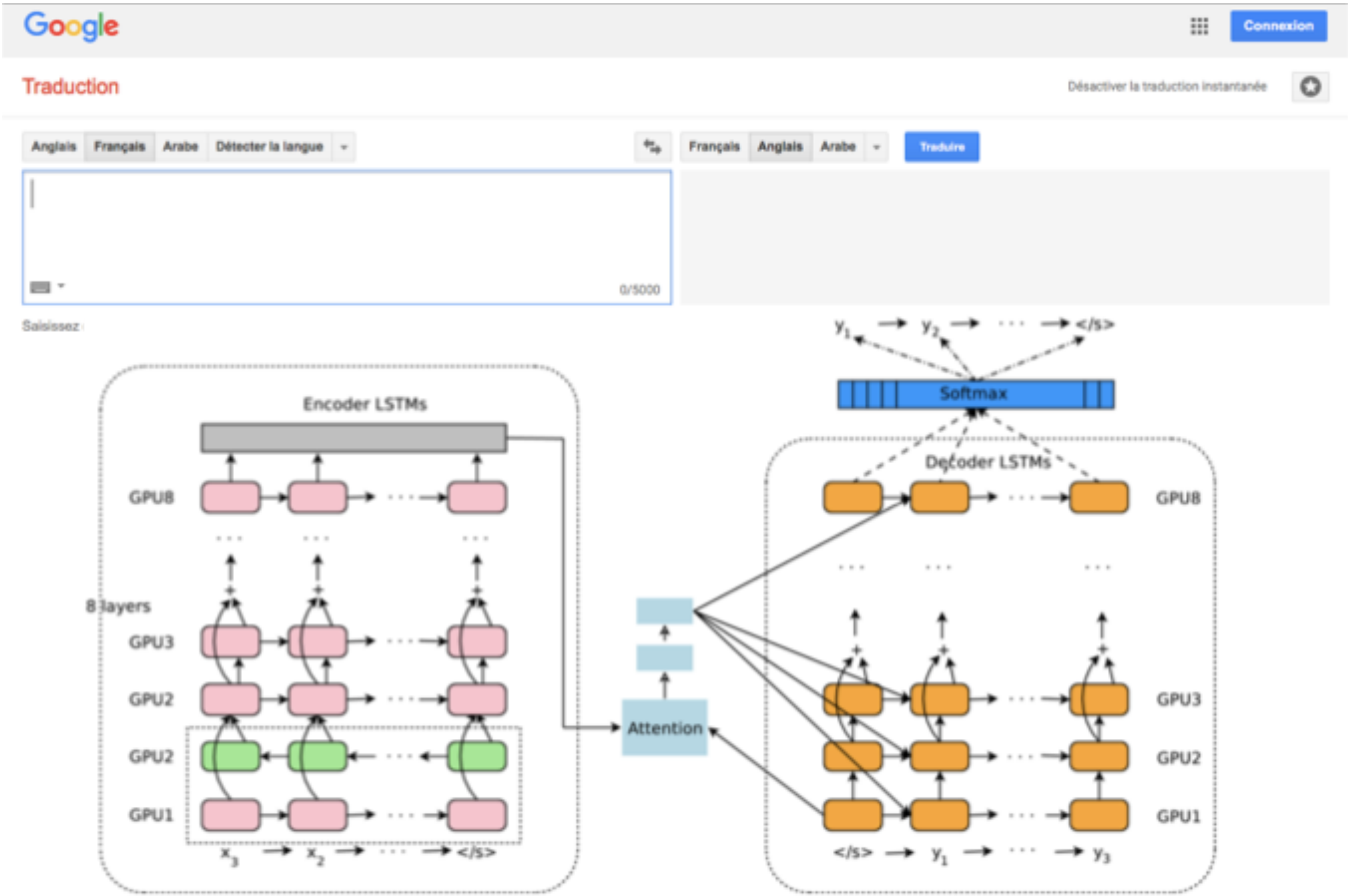
From Speech to Text [Chung et al., 2016]



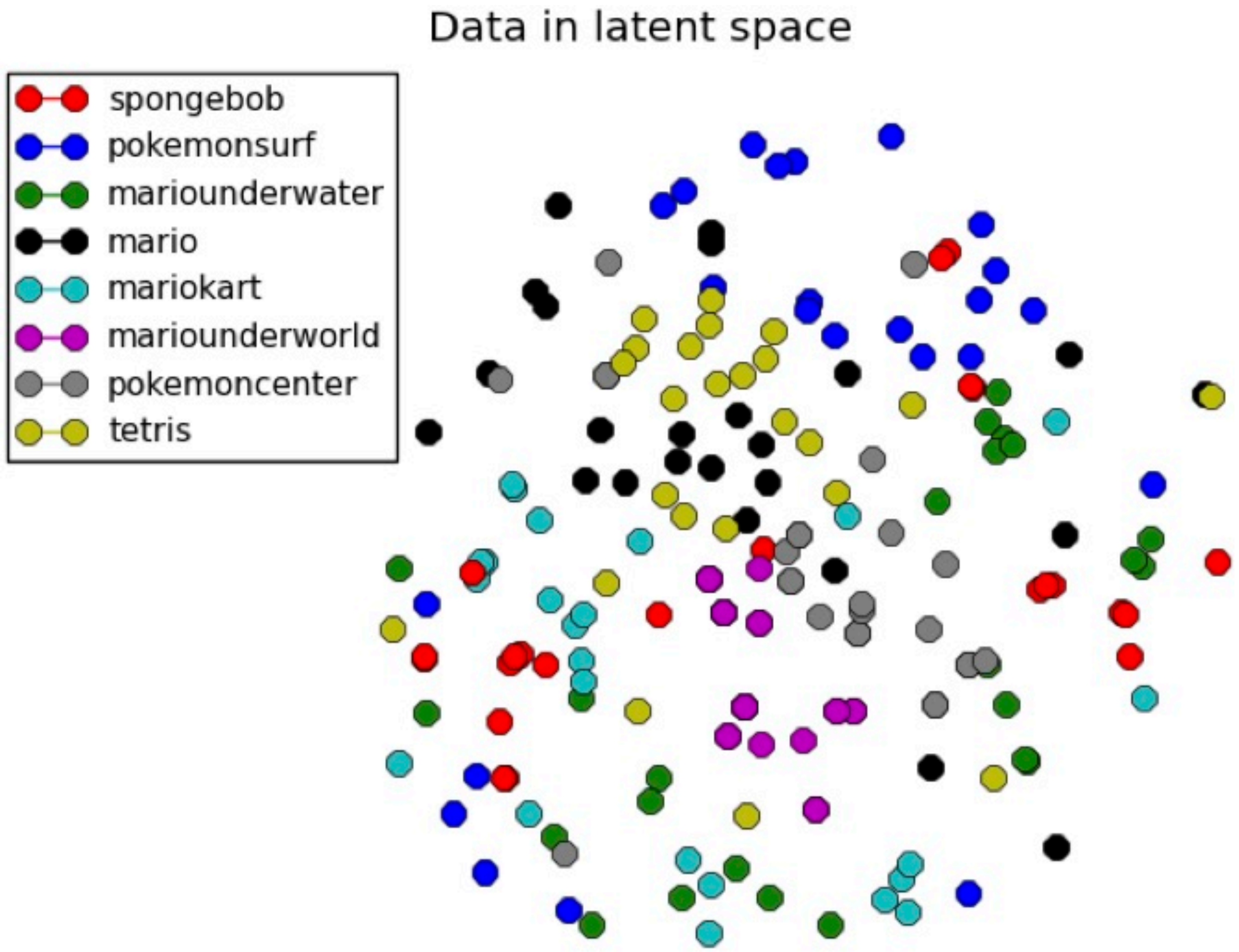
Translation
Sequence to Sequence



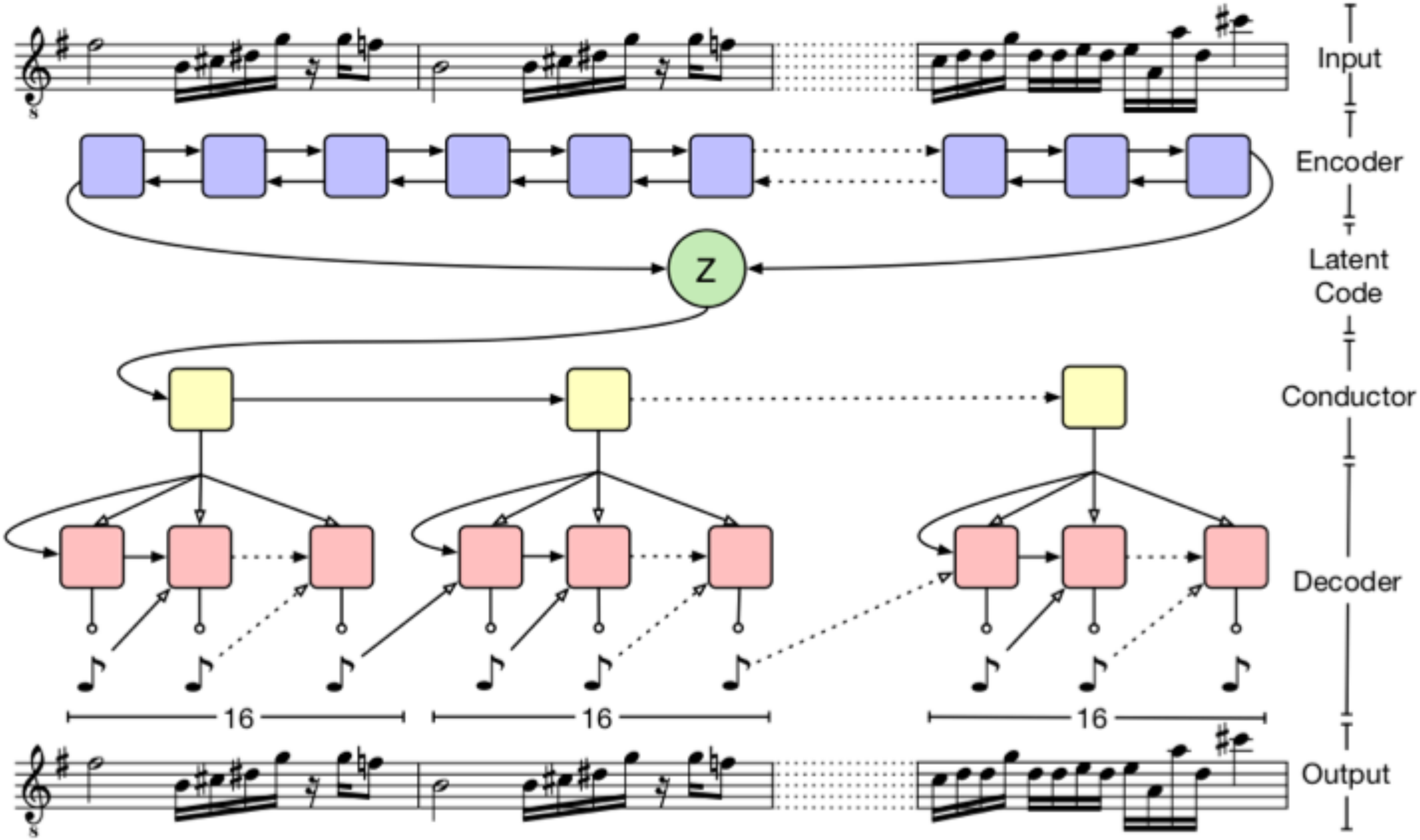
Translation



Variational RNN Encoder-Decoder
VRAE [Fabius and van Amersfoort, 2015]

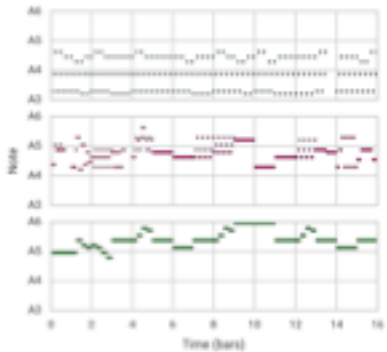


MusicVAE [Roberts et al., 2018]

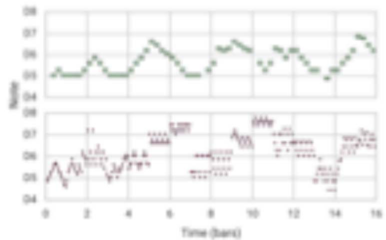


MusicVAE [Roberts et al., 2018]

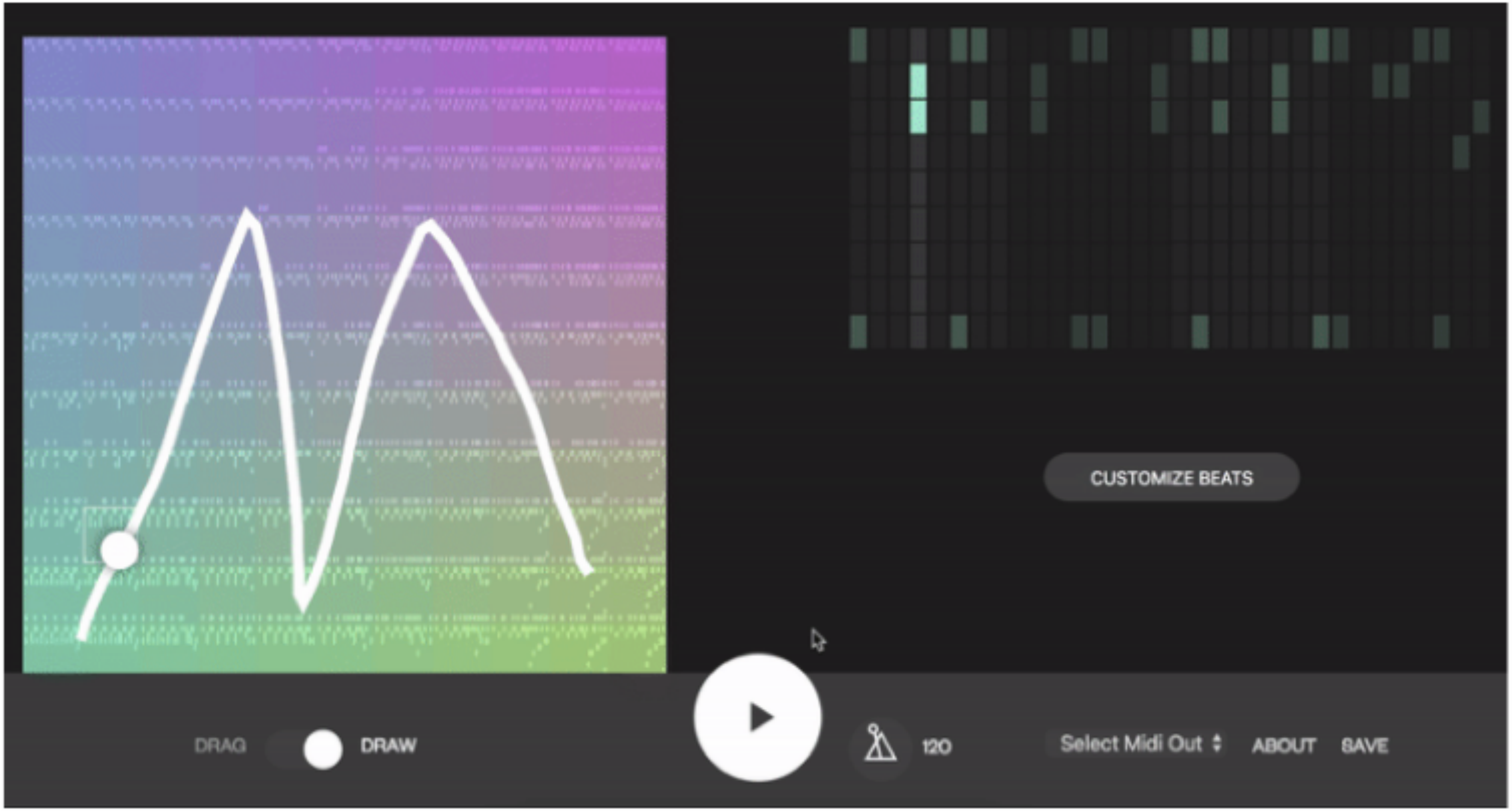
- Hierarchical
  - Conductor RNN
  - Bottom RNN
- Longer term generation
- Structure



- Translation
- Interpolation (morphing)
- Averaging of some points
- Addition or subtraction of an attribute vector capturing a given characteristic
  - This attribute vector is computed as the average latent vector for a collection of examples sharing that attribute (characteristic)

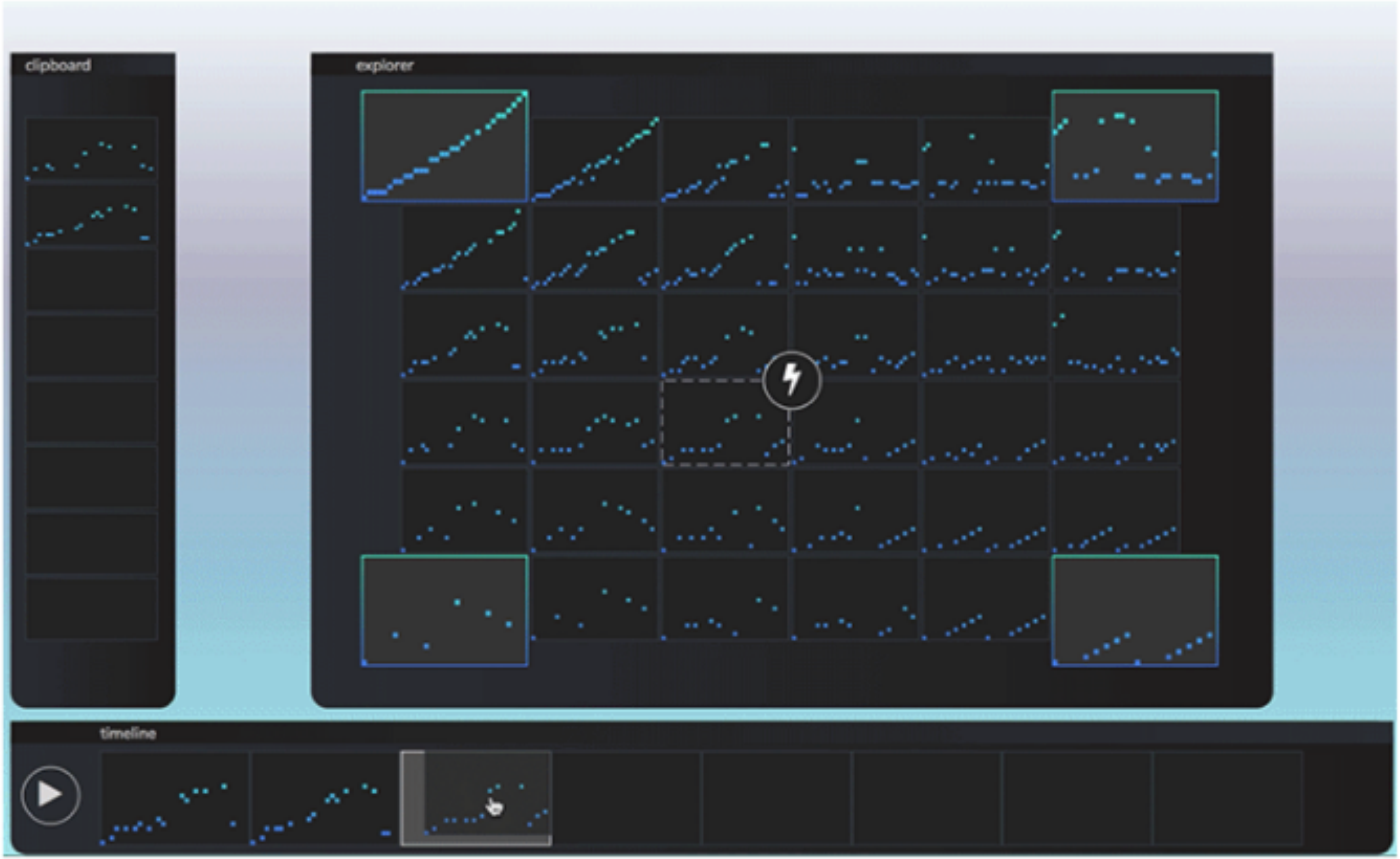


BeatBlender in TensorFlow.js
MusicVAE [Roberts et al., 2018]



https://experiments.withgoogle.com/ai/beat-blender/view/

LatentLoops in TensorFlow.js
MusicVAE [Roberts et al., 2018]



https://teampieshop.github.io/latent-loops/