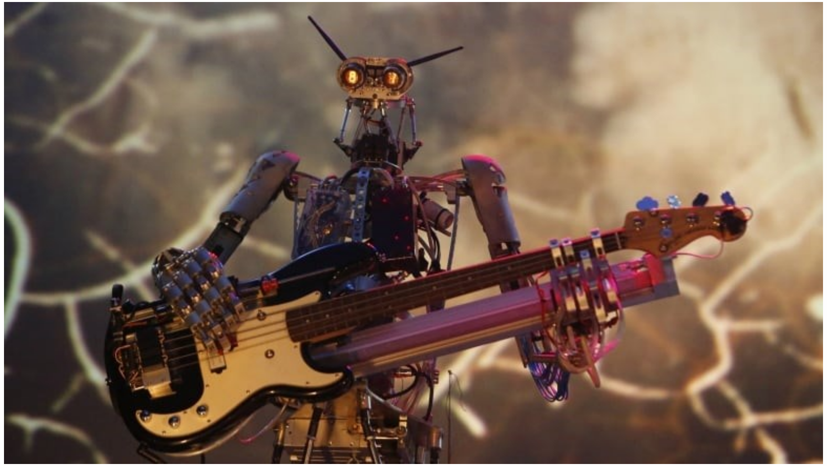
Generative Music with RNN-LSTM



AI & Music

Rocco Zaccagnino
rzaccagnino@unisa.it

Creative AI

Creative AI
euro news.
BEETHOVEN'S 250TH ANNIVERSARY
AI is writing the end of unfinished symphony
BEETHOVEN
X THE AI PROJECT
BEETHOVEN ORCHESTRA BONN
CAMERON CARPENTER
DIRK KAFTAN

Generative Music

Forma di musica in cui un brano musicale si crea da un insieme iniziale di elementi musicali, comportamenti e regole definiti dal compositore e/o da un sistema (naturale o **artificiale**).

"Approach to music creation concerning itself with neither improvisation nor explicit composition, but rather with framing an indeterminate system from which music can emerge" (Priestley, 2014)

"Generative music is like trying to create a seed, as opposed to classical composition which is like trying to engineer a tree" (Toop, 2004)

Generative Music

Non è un genere musicale o uno stile, ma una pratica compositiva in cui il compositore è più interessato a creare un sistema fisico o virtuale che poi genererà la musica autonomamente

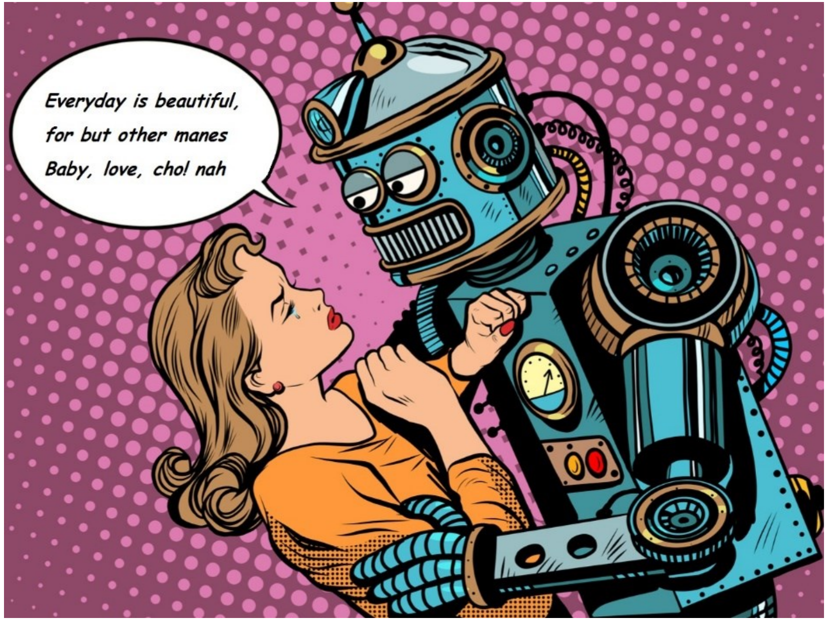
Music & NN

Obiettivo: sviluppare una RNN-LSTM per generare **melodie** automaticamente

- Build e train di una **LSTM** in Keras
- Gestire **time-series** data
- Comprensione **rappresentazione musicale simbolica**
- Concetti musicali di base (e.g., pitch, duration, key)
- Pre-processing di musica simbolica

Pre-requisiti & Tools

- Intermediate Python
- Familiarità con **Tensorflow** & **Keras**
- **Music21** (MIT music library)
- **Musescore** notation platform
- Ubuntu operating system (v. 22.04)



"Artificial intelligence just wrote the best love song of all time (and it's terrible)" ***Verdict***

Melody

Successione di suoni animata dal ritmo e regolata da leggi strofiche così da acquistare contorni, e senso propri ... parte di composizione musicale che costituisce il canto, il motivo fondamentale, come successione lineare di suoni che, accompagnata da altri suoni secondo regole determinate, è la base dell'armonia.

Treccani, *https://www.treccani.it/vocabolario/melodia/*

Da un punto di vista applicativo … sequenza di note e pause

Melody: pitch

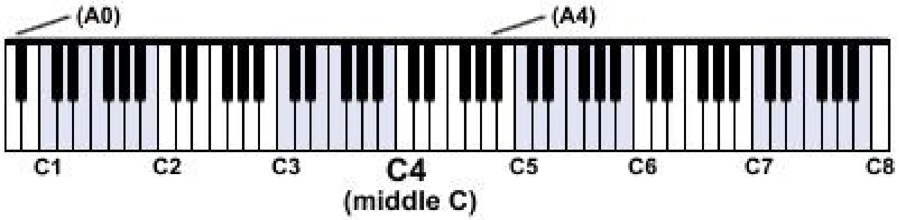
Indica quanto “alta” o “bassa” è la nota. Detta anche ***frequenza.***

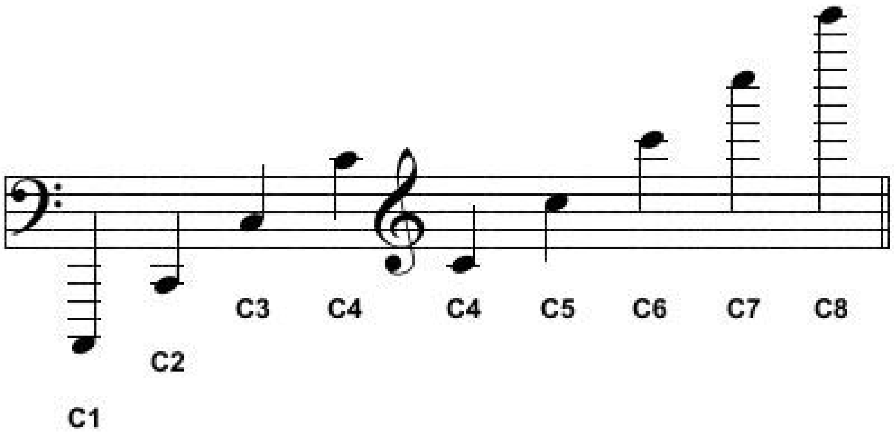
Melody: pitch

Indica quanto “alta” o “bassa” è la nota. Detta anche ***frequenza.***

Scientific pitch notation

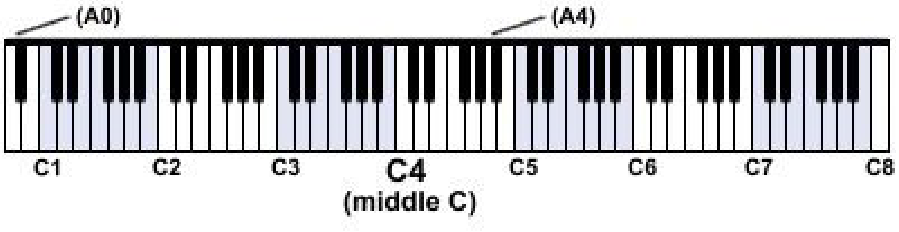
Note name + octave (e.g., C3, D5,..)

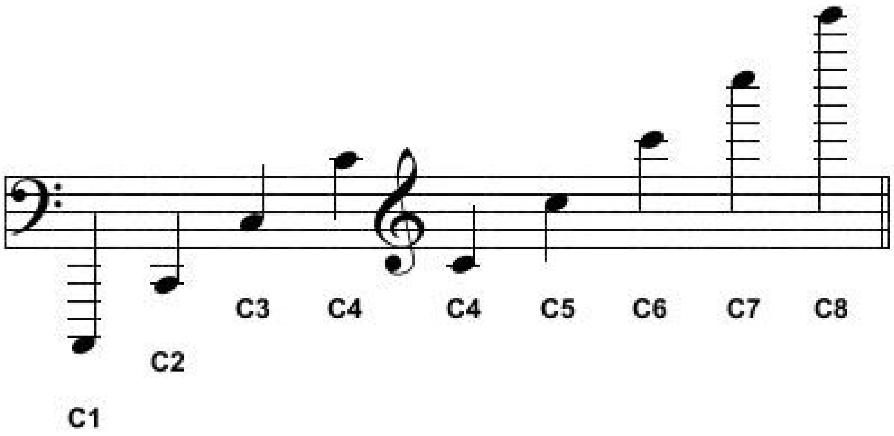




Scientific pitch notation

Note name + octave (e.g., C3, D5,..)





MIDI note notation

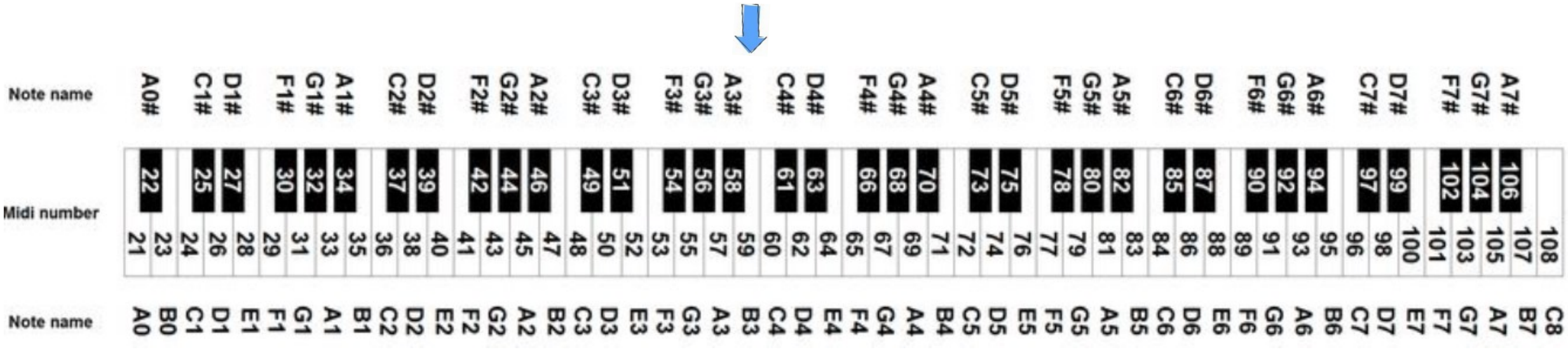
MIDI è uno standard per suonare, editare e registrare musica

Fornisce un ***mapping*** di nomi di note a interi (**C4 : 60**)

MIDI note notation

MIDI è uno standard per suonare, editare e registrare musica

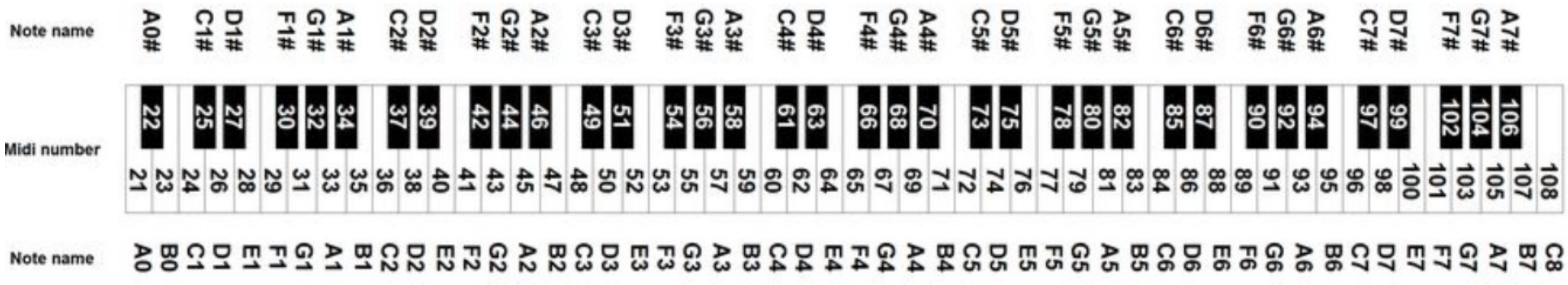
Fornisce un ***mapping*** di nomi di note a interi (**C4 : 60**)



MIDI note notation

MIDI è uno standard per suonare, editare e registrare musica

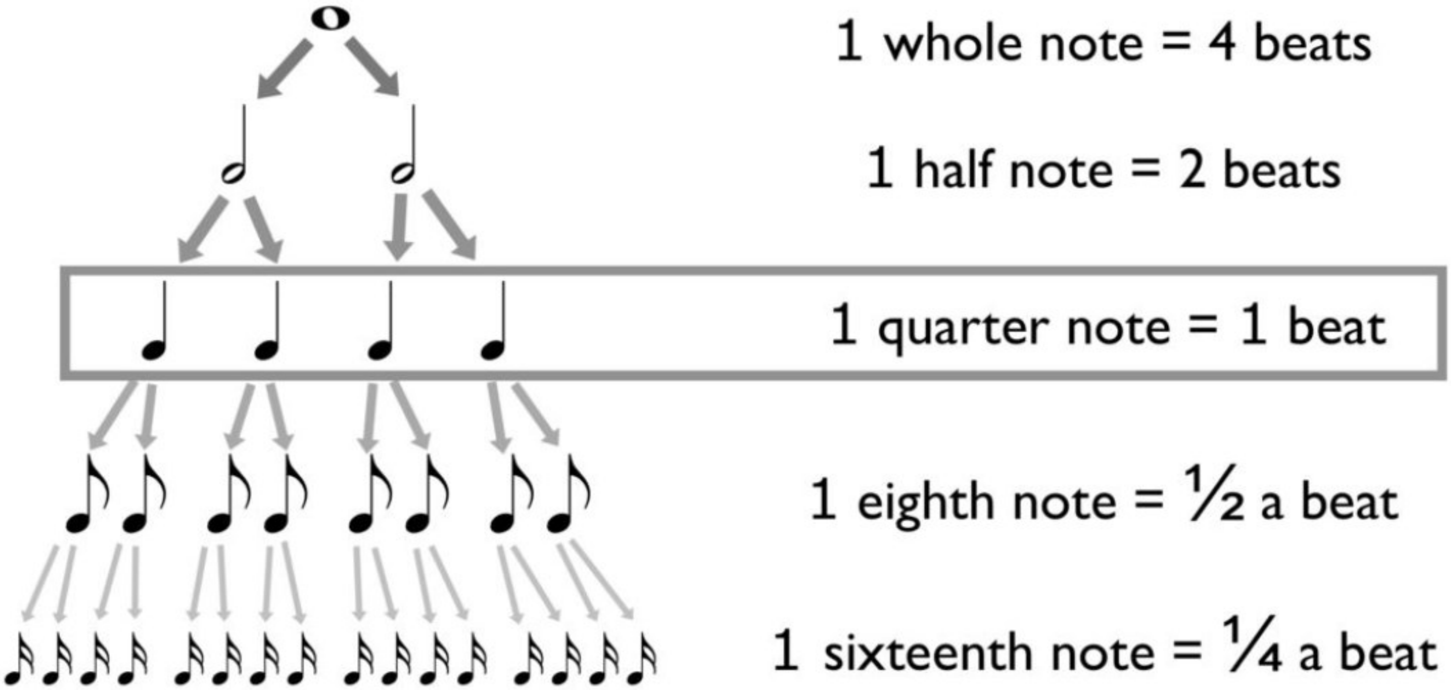
Fornisce un ***mapping*** di nomi di note a interi (**C4 : 60**)



C4 D4 E4 E4 F4 D4 G4 E4 D4 C4

60 62 64 64 65 62 67 64 62 60

Note durations

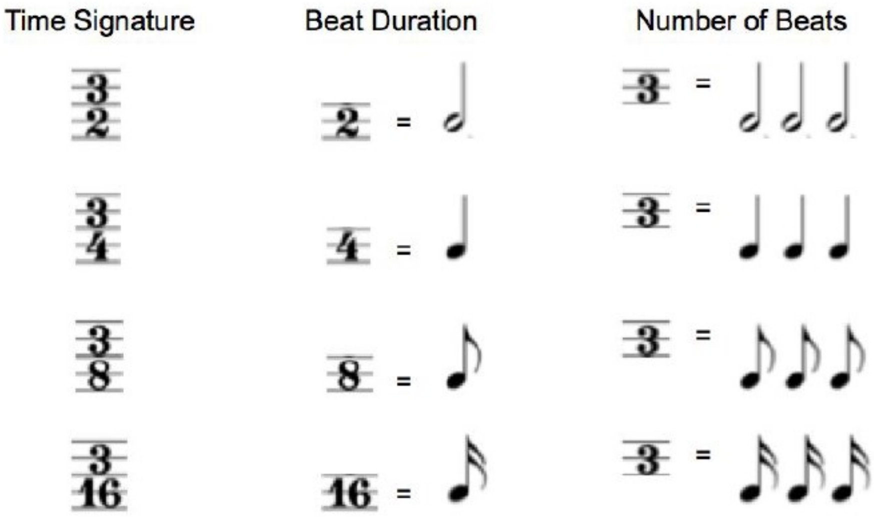


Note durations

1 1 0.5 0.5 0.5 0.5 1 0.5 0.5 1 1

Duration in # beats: 4 4

Time signature



Music representation

Music representation 1

Sequence

(pitch, duration) per ogni nota

E.g., [(C4,1),(D4,1),(E4,.5),…)]

Music representation 2

Time series

Campionamento della melodia ad ogni 16th note

Step: 16th note

Log MIDI note quando occorre la nota

Usiamo il simbolo "_" per indicare la **nota tenuta**

Usiamo il simbolo "*r*" per indicare una **pausa**

Music representation 2 (example)

- 4/4 time signature
- 16 samples per bar
- 4 samples per quarter note

Music representation 2 (example)

[“60”, “_”, “_”, “_”,

Music representation 2 (example)

[“60”, “_”, “_”, “_”,

“62”, “_”, “_”, “_”,

Music representation 2 (example)

[“60”, “_”, “_”, “_”,

“62”, “_”, “_”, “_”,

“64”, “_”, “64”, “_”,

Music representation 2 (example)

[“60”, “_”, “_”, “_”,

“62”, “_”, “_”, “_”,

“64”, “_”, “64”, “_”,

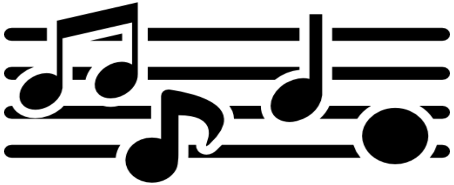
“65”, “_”, “62”, “_”,

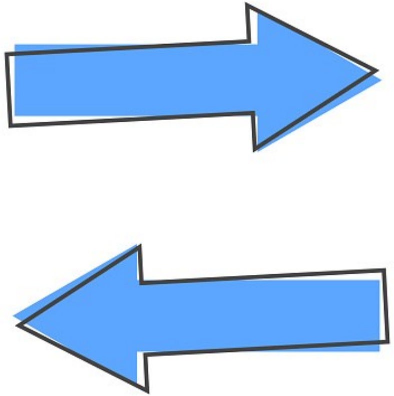
…]

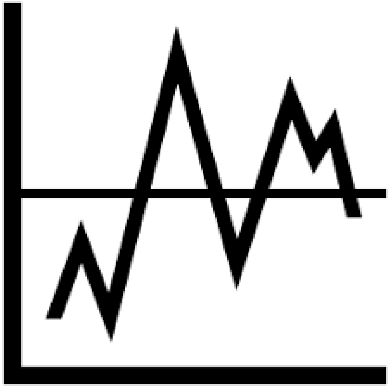
Melody generation problem

Considerare la melodia come **“time-series data”**

Melody generation problem



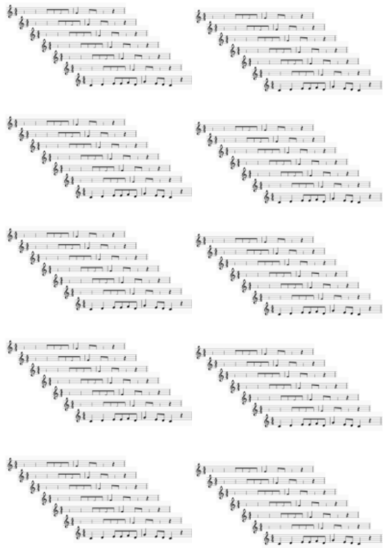




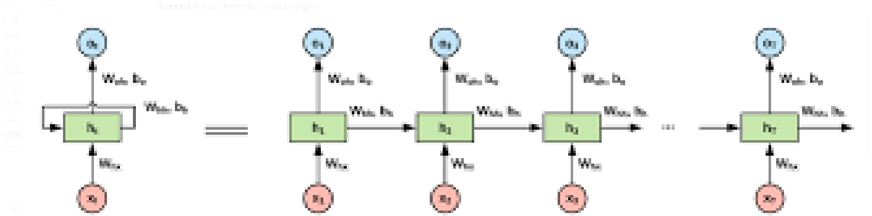
Time-series prediction problem

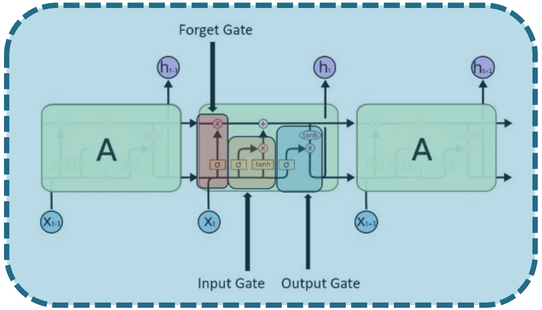
The melody generator (training)

Melodies dataset



RNN-LSTM





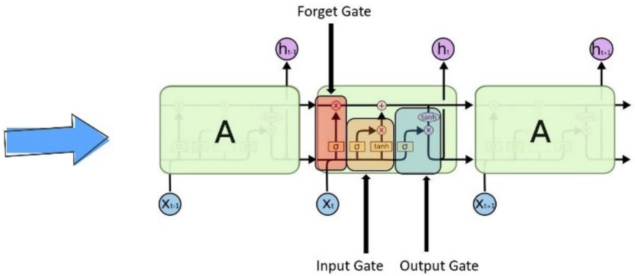
Next note prediction

The melody generator (inference)

1

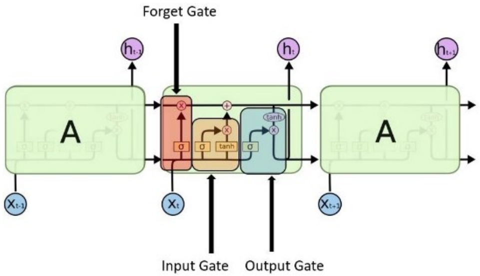
The melody generator (inference)

2



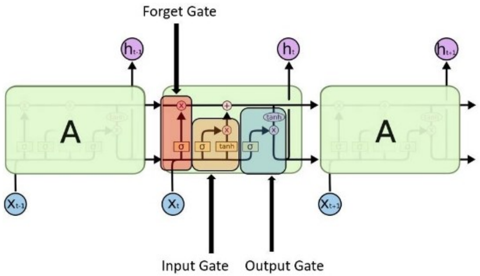
The melody generator (inference)

3



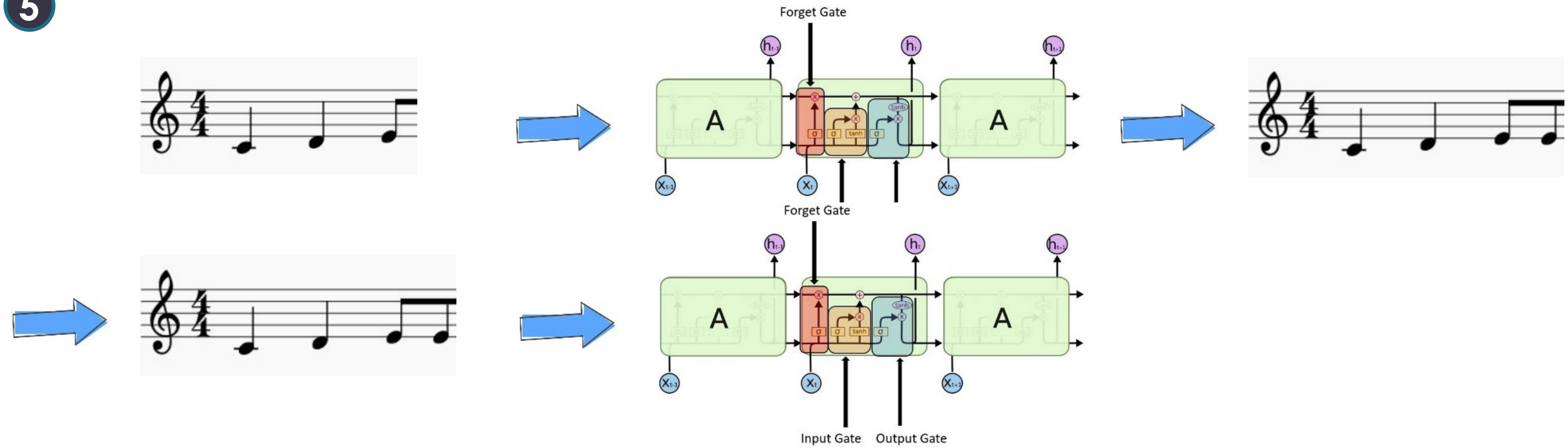
The melody generator (inference)

4

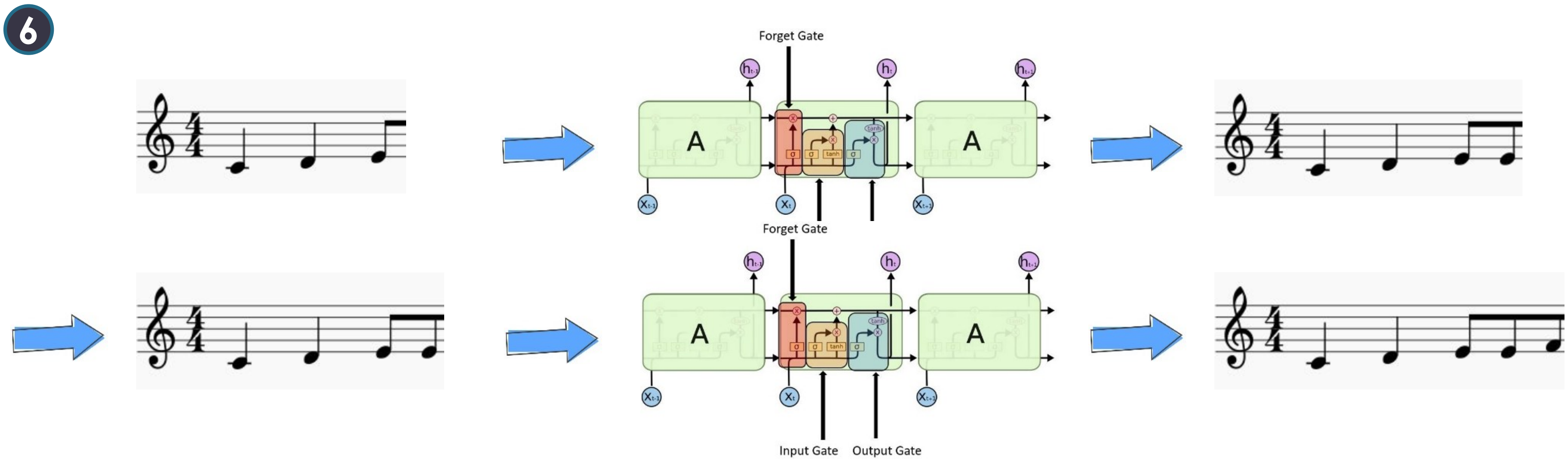


The melody generator (inference)

5

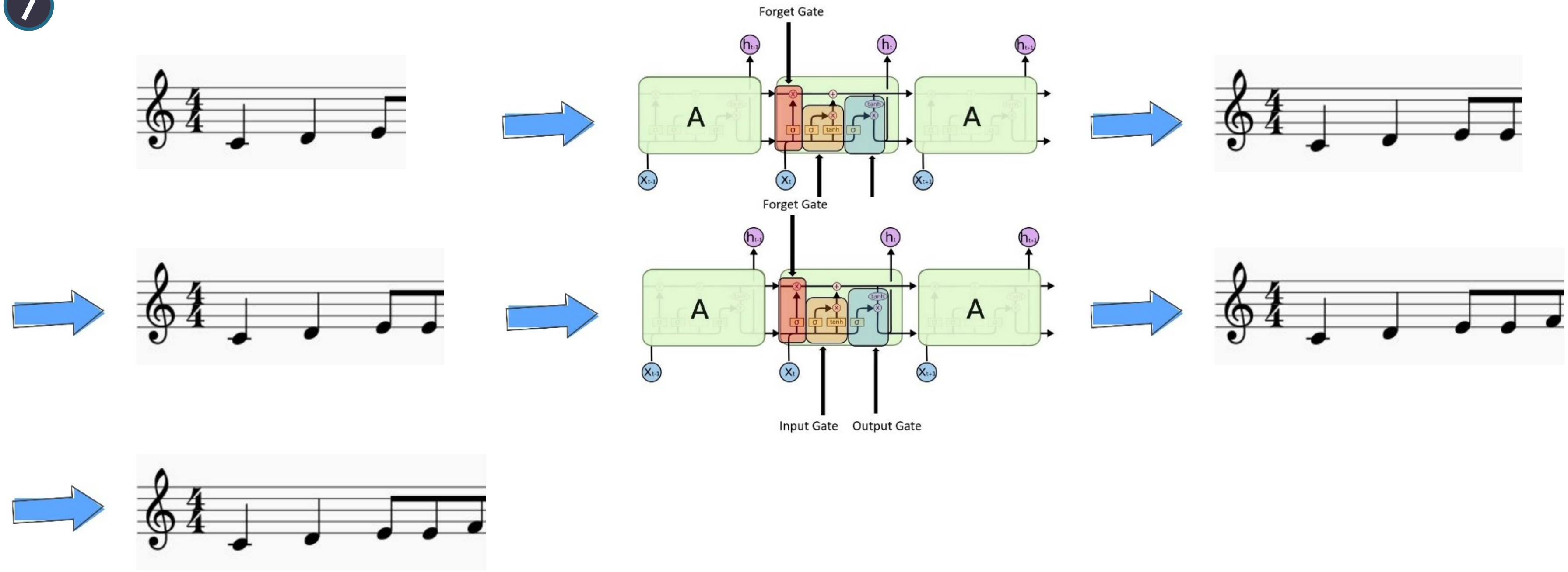


The melody generator (inference)



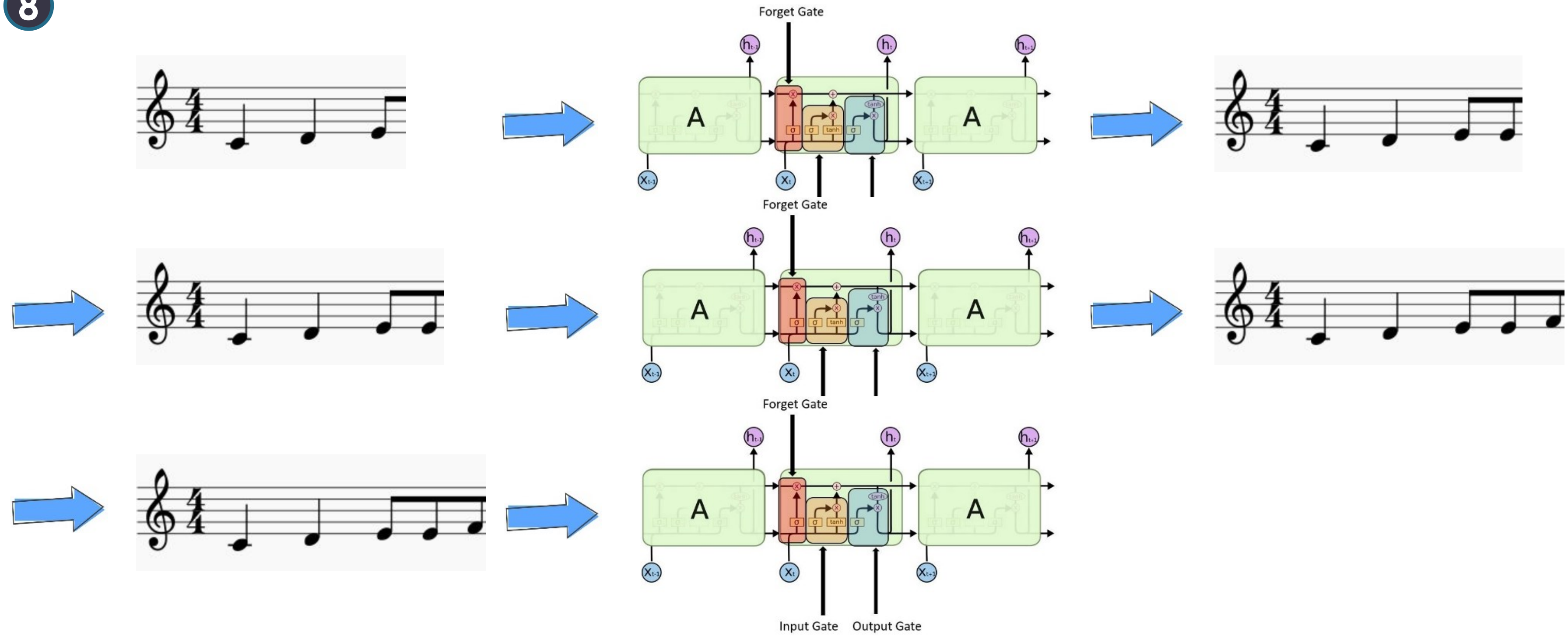
The melody generator (inference)

7



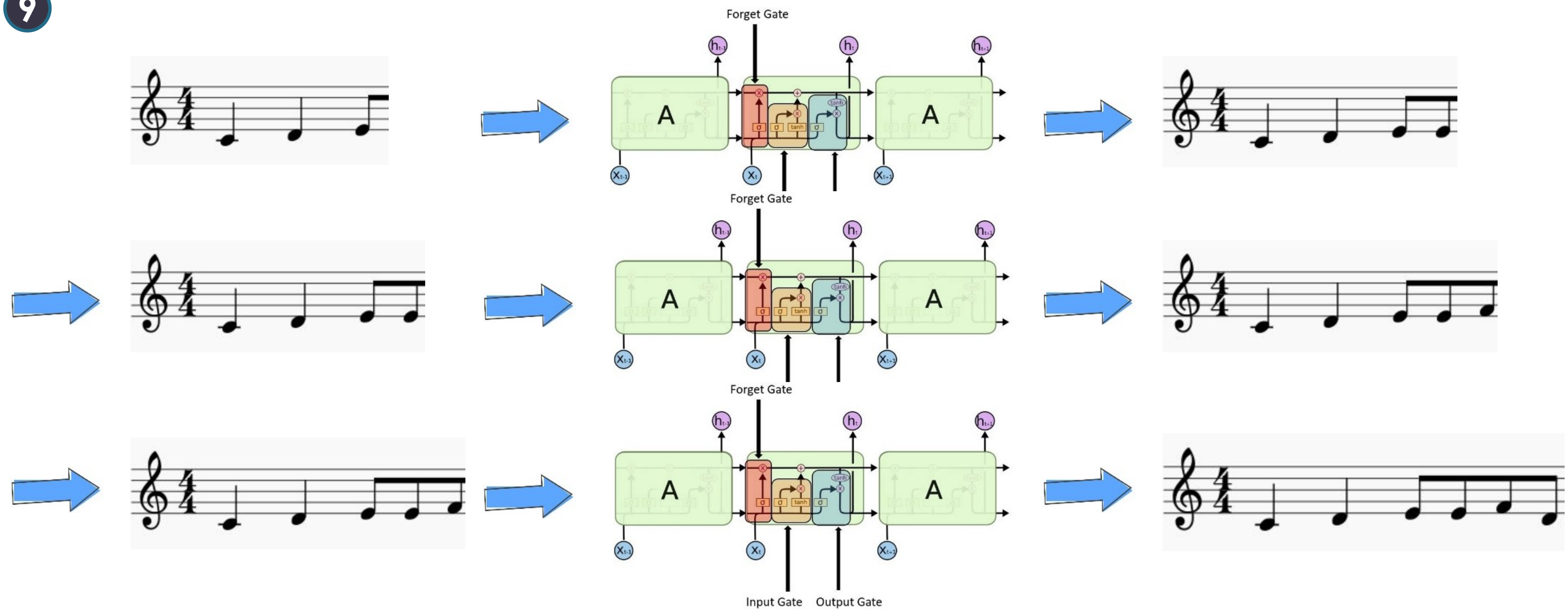
The melody generator (inference)

8



The melody generator (inference)

9

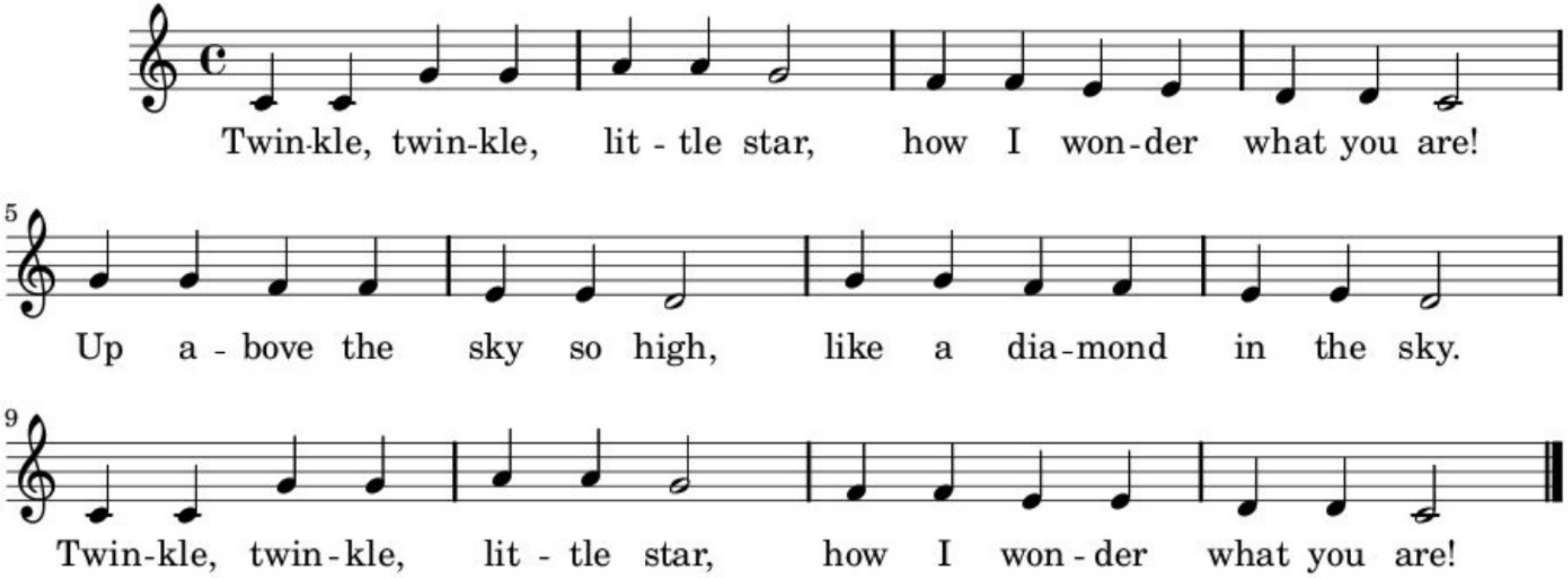


Perché utilizzare RNN-LSTM?

Le melodie hanno ***long-term structural patterns***

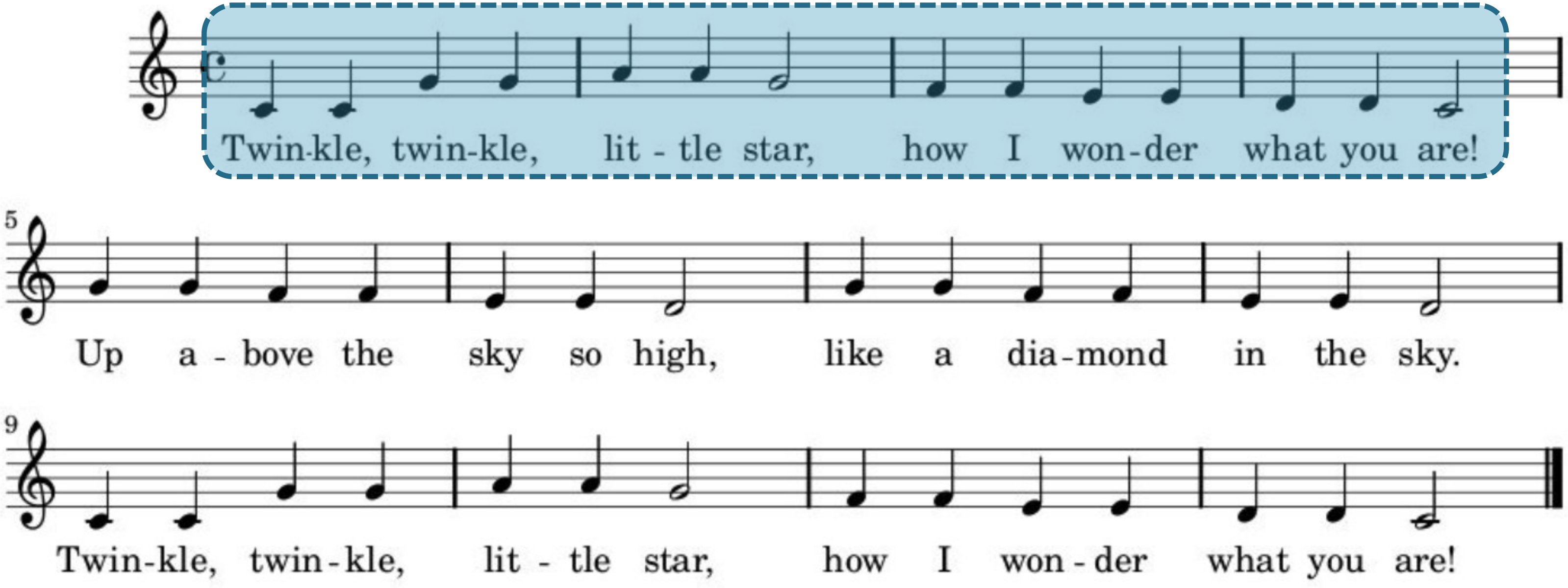
Perché utilizzare RNN-LSTM?

Le melodie hanno ***long-term structural patterns***



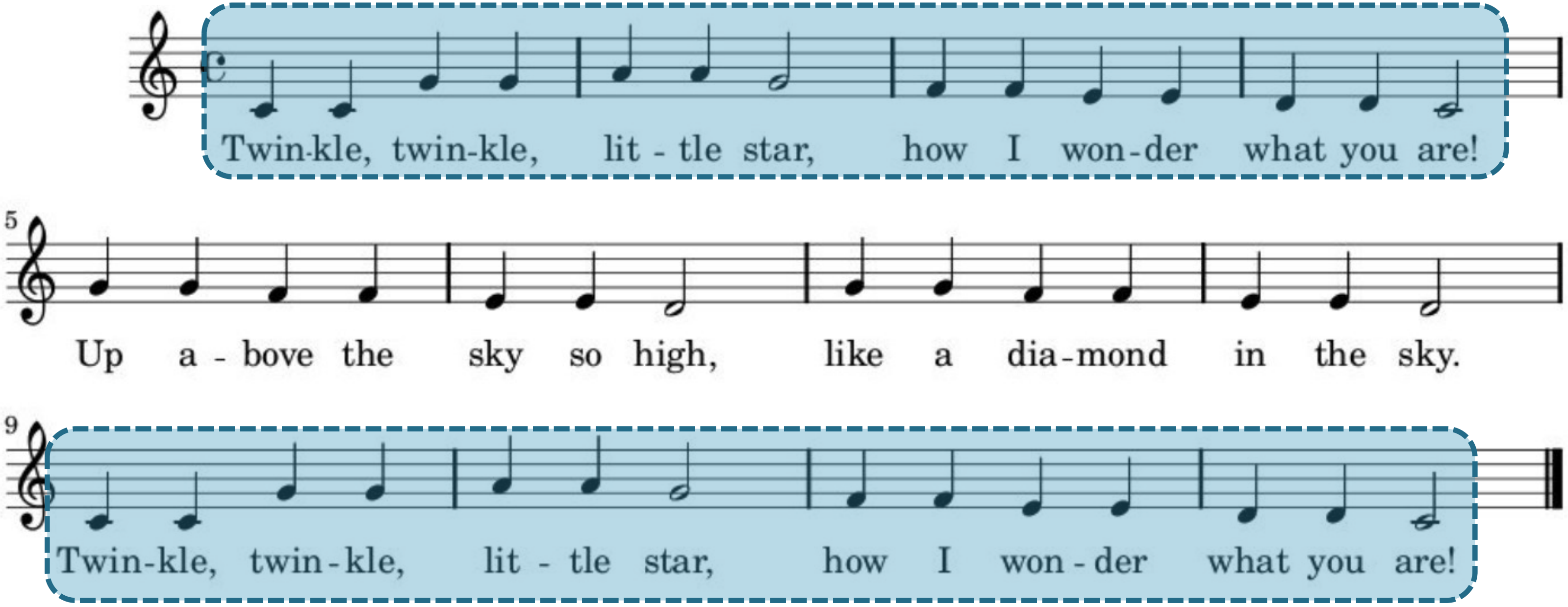
Perché utilizzare RNN-LSTM?

Le melodie hanno ***long-term structural patterns***



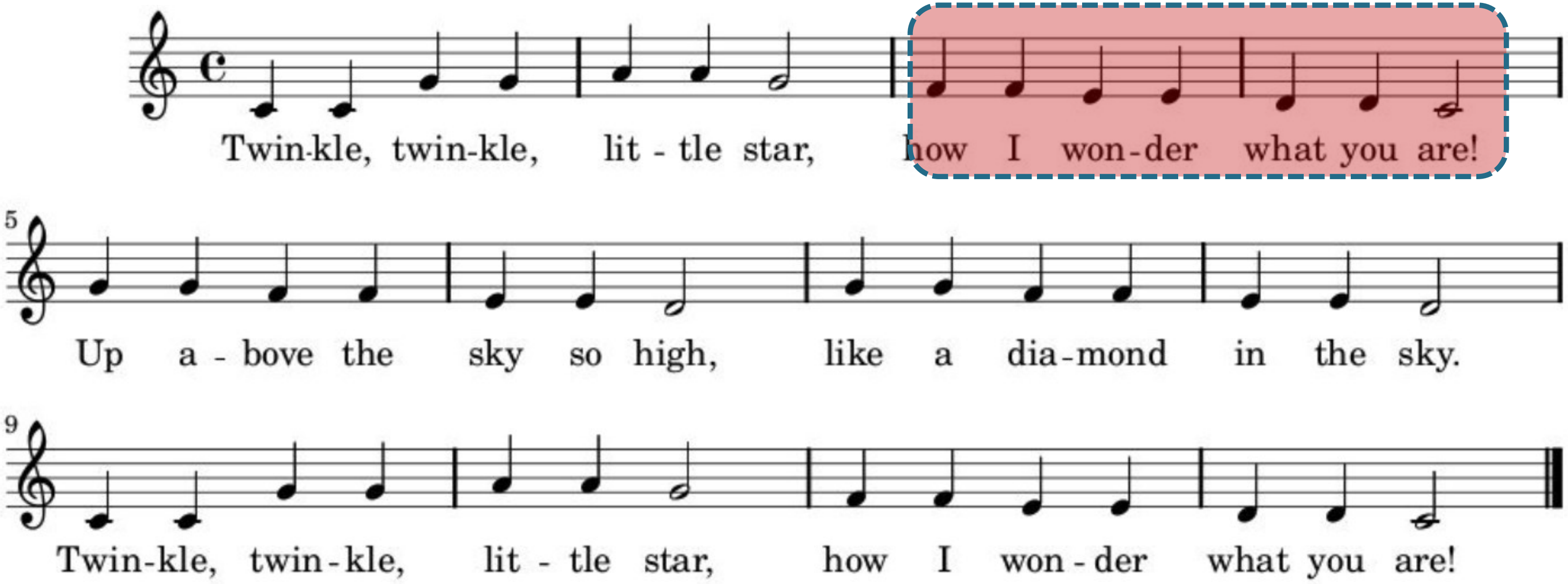
Perché utilizzare RNN-LSTM?

Le melodie hanno ***long-term structural patterns***



Perché utilizzare RNN-LSTM?

Le melodie hanno ***long-term structural patterns***

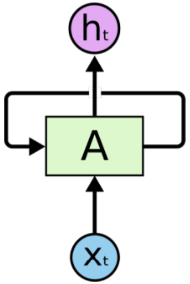


Perché utilizzare RNN-LSTM?

Le melodie hanno ***long-term structural patterns***

Le LSTMs catturano ***long-term structural dependencies***

Meccanismo di ricorrenza: **RNN**



FNN:

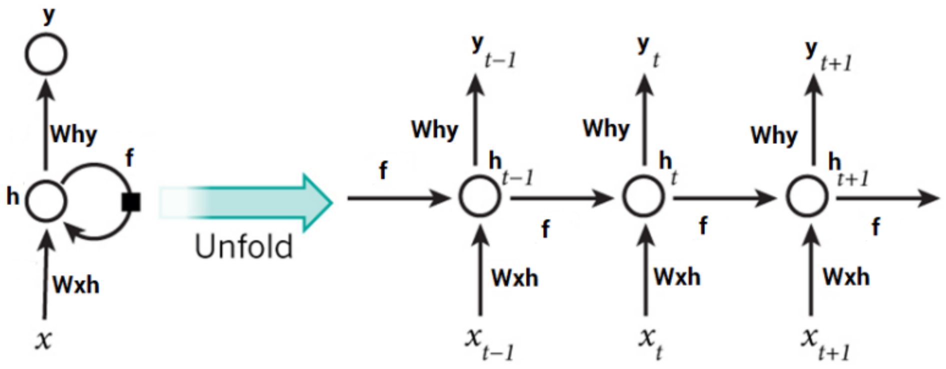
1. **No loops: informazione in una sola direzione**
2. **No memory: decisioni in base all'input corrente**
3. **No series: non gestiscono time-series data**

RNN:

1. **Gestiscono time-series data**
2. **Considerano input corrente e precedente**
3. **Hanno un meccanismo di memoria**

Recurrent Neural Network: RNN

$$h_t = f(W_{xh}\, x_t + W_{hy}\, h_{t-1})$$

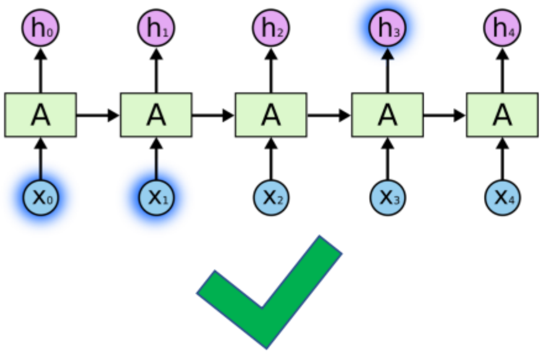


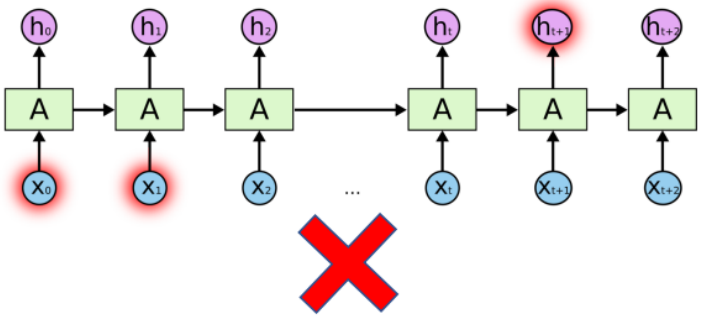
- h_t : hidden state at time step t
- x_t : input at time step t
- W_{xh} and W_{hy} : weight matrices. Filters that determine how much importance to accord to both the present input and the past hidden state.

Recurrent Neural Network: RNN

- A small example where RNN can work perfectly :
  - Prediction of the last word in the sentence : “The clouds are in the sky”

- RNN can’t handle situation where the **gap** between the **relevant information** and the point where it is needed is **very large**.





Long-Short Term Memory Network: LSTM

A livello informale, l’output di una LSTM in un dato momento dipende da 3 cose:

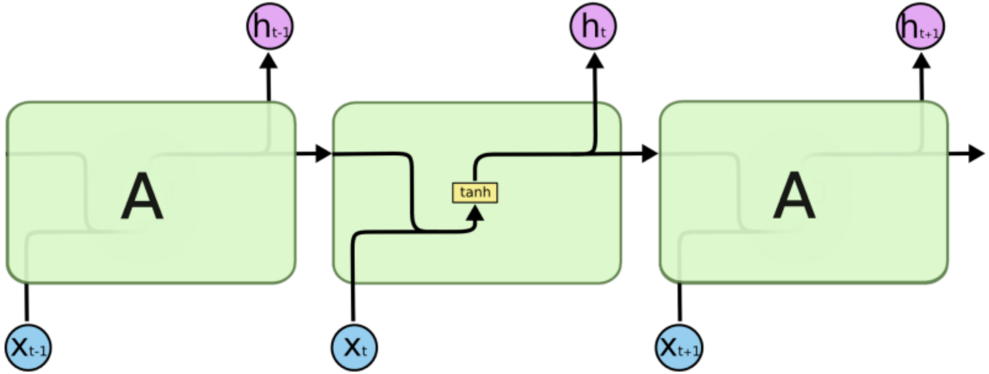
1. Dalla *current long-term memory* della rete, detta **cell state**
2. Dall’output al tempo precedente, noto come **previous hidden layer**
3. Dall’input al tempo corrente, noto come **current time step**

Le LSTMs usano una serie di *gates* che controllano come le informazioni di una sequenza di dati temporali vengono gestiti. Ci sono 3 gates: **forget gate**, **input gate**, e **output gate.**

Long-Short Term Memory Network: LSTM

- **Long Short Term Memory networks** – usually just called "**LSTMs**" – are a special kind of RNN, capable of learning **long-term dependencies.** *Hochreiter & Schmidhuber (1997)*

- All recurrent neural networks have the form of a **chain of repeating modules** of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.



Long-Short Term Memory Network: LSTM

- **LSTM** have the same chain like structure except for the repeating module.

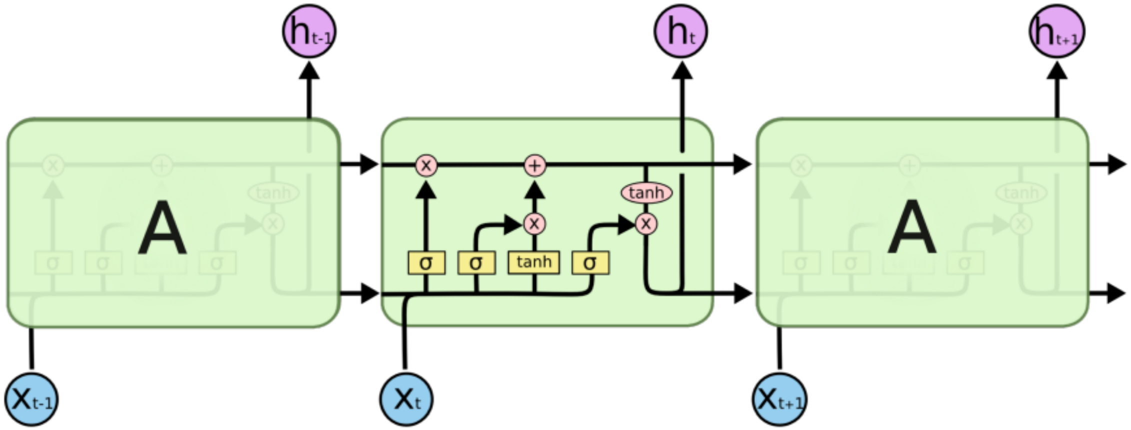
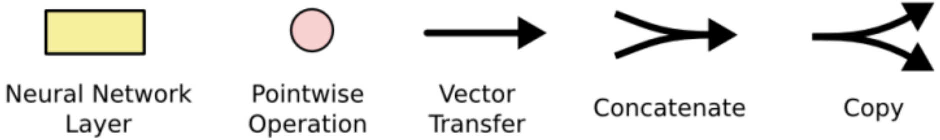
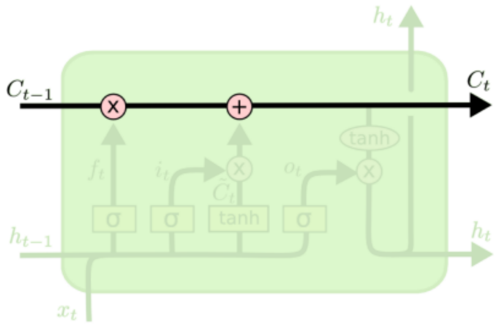


Fig6: The repeating module in a standard RNN contains a single layer [4]

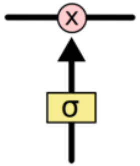


Long-Short Term Memory Network: LSTM

- The core idea behind LSTMs is the **cell state**.



- The LSTM has the ability to **remove** or **add** information to the cell state : thanks to **gates**

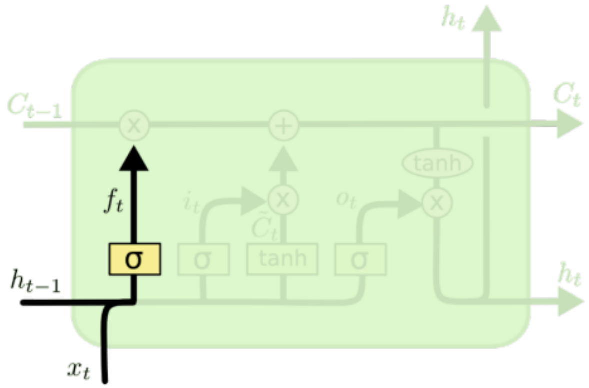


- Gates are composed out of a sigmoid neural net layer and a pointwise multiplication operation

Long-Short Term Memory Network: LSTM

- Step-by-Step LSTM Walk Through

  - **Step 1:** Decide what information to **throw away** from the cell state, **forget layer.**



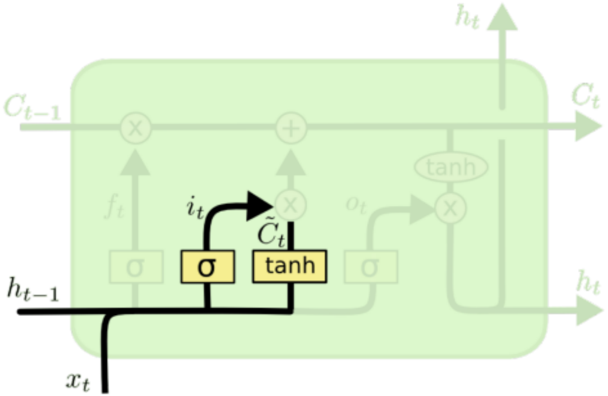
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

- **1** represents "completely keep this"
- **0** represents "completely get rid of this."

Long-Short Term Memory Network: LSTM

- Step-by-Step LSTM Walk Through

  - **Step 2**: Decide what new information we're going to store in the cell state



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

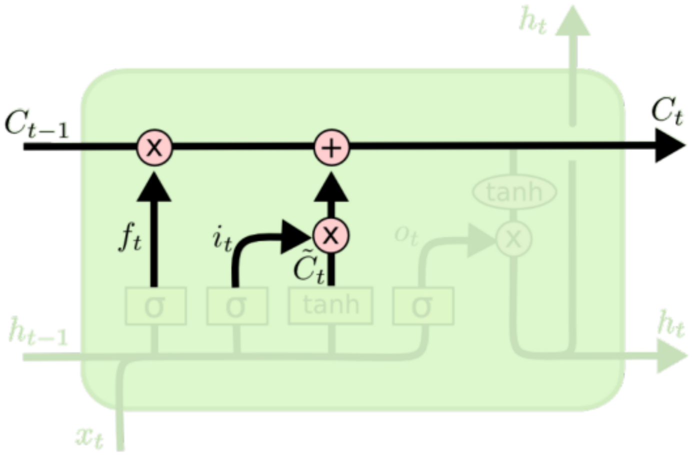
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

  - **Input gate layer** : decides which values we will update
  - **Tanh layer** : creates a vector of new candidate values

Long-Short Term Memory Network: LSTM

- Step-by-Step LSTM Walk Through

  - **Step 3**: Update the cell state

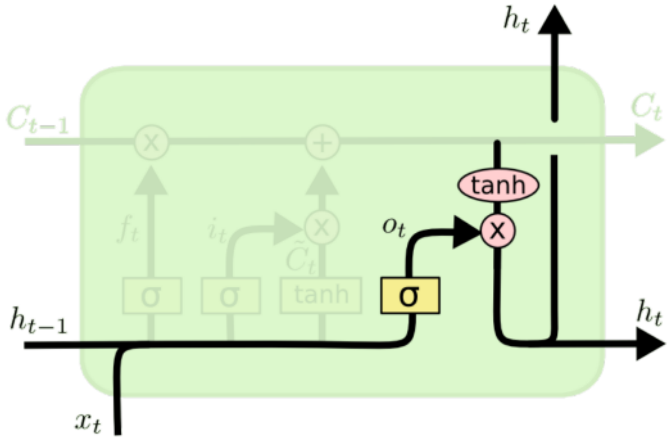


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- **Example** : "I grew up in France… I speak fluent *French*."

Long-Short Term Memory Network: LSTM

- Step-by-Step LSTM Walk Through
  - **Step 4**: Decide what is the output



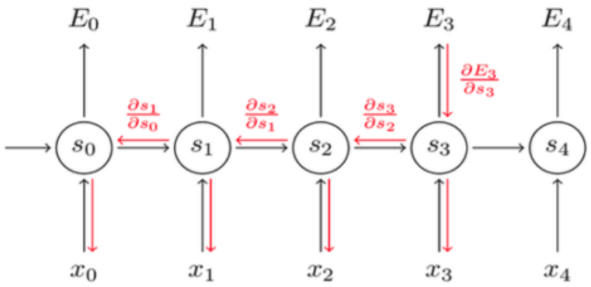
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

- **Example** : "I grew up in France… I speak fluent *French*."

LSTM: Backpropagation Through Time

- **Backpropagation**: Uses partial derivatives and the chain rule to calculate the change for each weight efficiently. Starts with the derivative of the loss function and propagates the calculations backward.

- **Backpropagation Through Time**, or BPTT, is the training algorithm used to update weights in recurrent neural networks like LSTMs.



Il caso di studio: ESAC dataset

ESAC: *Essen Associate Code and Folksong Database (http://www.esac-data.org)*

Sviluppato come raccolta di one-part music da European folksong databases

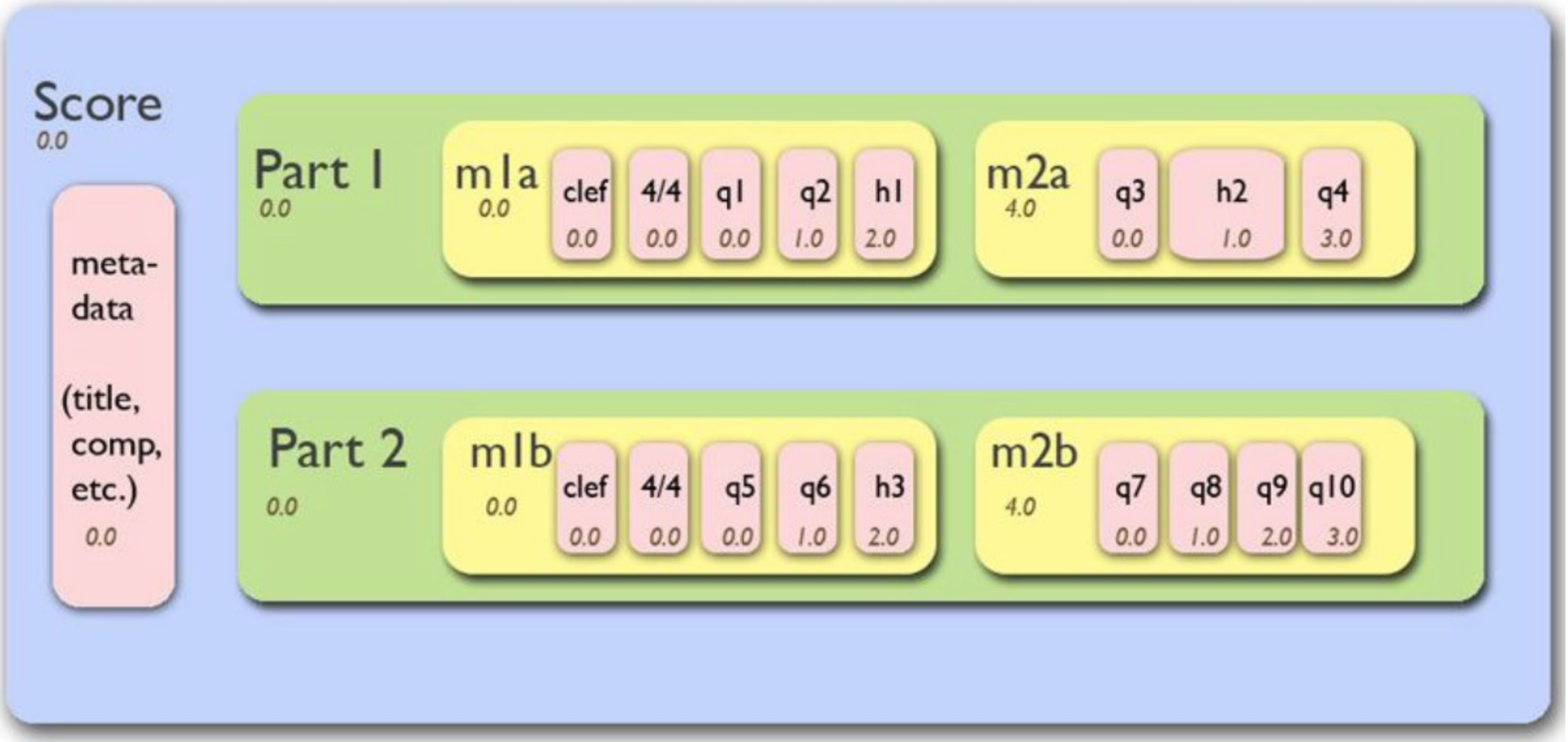
Più di 20000 melodie strumentali soprattutto dalla Germania e Polonia

KERNSCORES Music Collection - *https://kern.humdrum.org/cgi-bin/browse?l=/*

KernScores

1. si concentra sulle informazioni musicali funzionali, non ortografiche
2. progettato per facilitare le analisi, non la generazione di suoni
3. *https://www.humdrum.org/guide/ch02/*

Music21 score

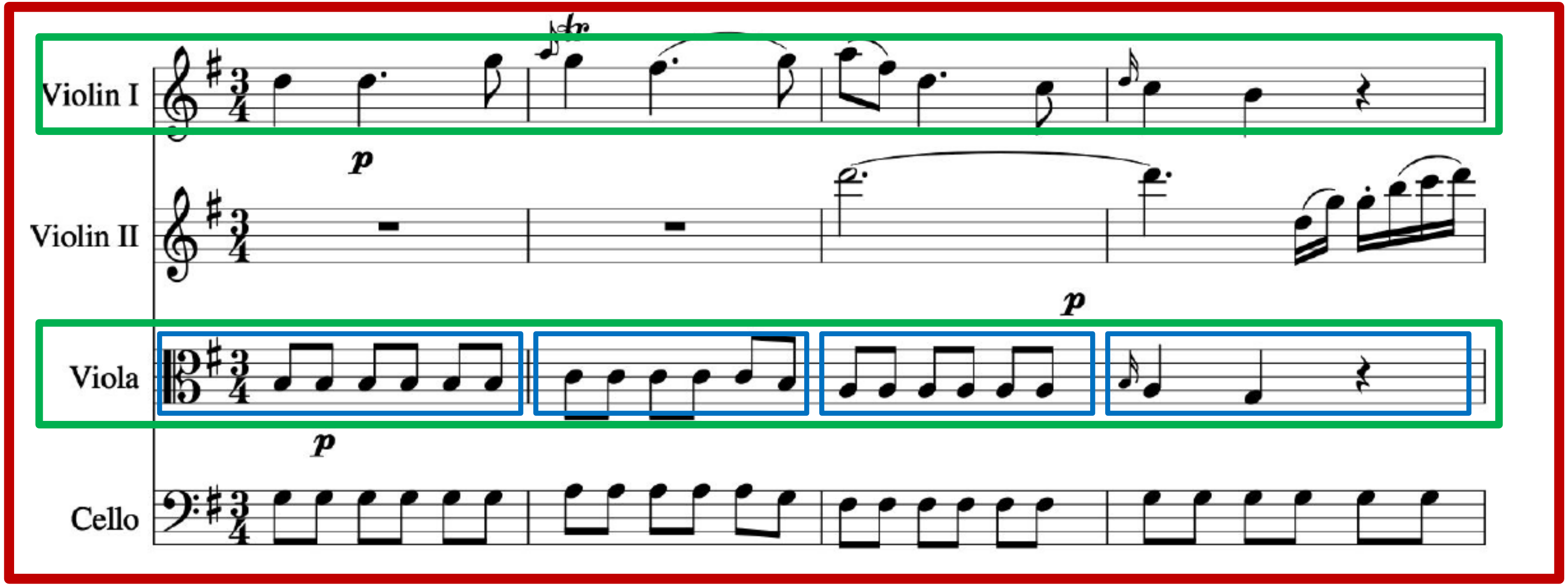


Music21 score: String quartet

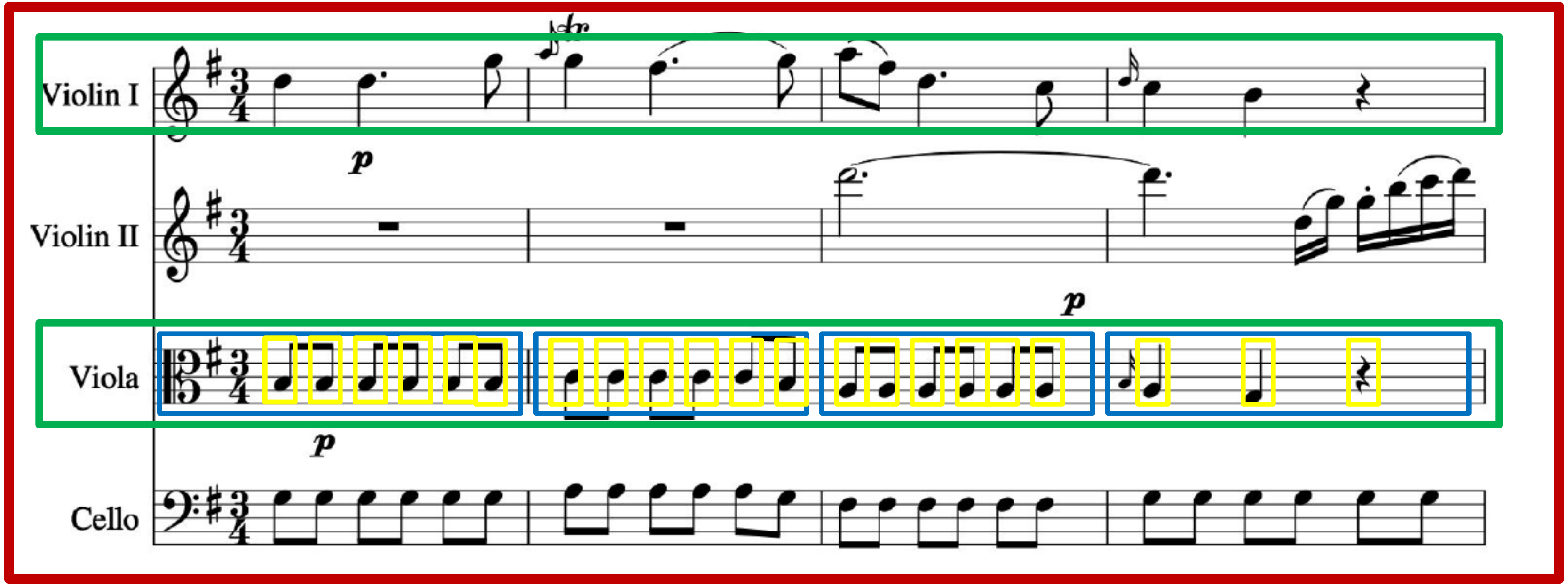
Score

Score + Parts + Measures

Score + Parts + Measures



Score + Parts + Measures + Notes



Grazie