

UNIVERSITÀ DEGLI STUDI DI TRENTO

CORSO DI LAUREA IN INFORMATICA

---

## Music RNN

---

Dellera ANDREA      158365

August 27, 2015

# 1 Introduzione

TODO

## 2 Codifica

Le note vengono estratte da partiture scritte in formato MusicXML e le features considerate sono cinque:

- nome della nota;
- alterazione;
- durata;
- punto di valore;
- ottava.

Di queste cinque il nome e l'alterazione sono unite in un'unica codifica visto che sono strettamente correlate tra loro così come la durata e punto di valore.

### 2.1 Codifica delle note

Le note nel sistema europeo sono chiamate *la, si, do, re, mi, fa, sol* che nel sistema americano corrispondono a *A, B, C, D, E, F, G*; in questo scritto verrà utilizzato il secondo, perchè più compatto e sintetico.

Per codificare tutte le note servono almeno quattro bit, questo perchè oltre alle sette naturali riportate sopra abbiamo anche quelle alterate dai  $\sharp$  e dai  $\flat$ . Va ricordato però che introducendo entrambe le alterazioni nella codifica si hanno note ridondanti dal punto di vista sonoro. Infatti se  $G \sharp$  e  $A \flat$  indicano due note diverse, perchè cambia la tonalità in cui vengono usate, il suono che viene prodotto quando sono suonate è però lo stesso. Ecco perchè nella codifica utilizzeremo solo il diesis.

**Codifica**

A	0000
A $\sharp$	0001
B	0010
C	0011
C $\sharp$	0100
D	0101
D $\sharp$	0110
E	0111
F	1000
F $\sharp$	1001
G	1010
G $\sharp$	1011

### 2.2 Codifica della durata

Per la durata della nota il ragionamento è analogo. La durata massima di una nota è  $\frac{4}{4}$  ( $\ominus$ ). Si trovano tutte nella forma  $\frac{1}{2^n}$  dove  $0 \leq n \leq +\infty$ . Convenzionalmente però le prime tre della sequenza sono descritte come  $\frac{4}{4}$  ( $\ominus$ ),  $\frac{2}{4}$  ( $\textcircled{J}$ ),  $\frac{1}{4}$  ( $\bullet$ ). Le durate che verranno codificate arriveranno fino ad  $\frac{1}{64}$ .

Come fatto precedentemente verrà assegnata una sequenza di bit ad ogni durata.

Un'altra variabile che entra quando si parla di durata è il *punto di valore* ( $\cdot$ ). Questo strumento aumenta la durata della nota della sua metà e per codificarlo useremo un bit che sarà 0 se non c'è e 1 in caso contrario.

### 2.3 Codifica dell'ottava

In ciò che va a costituire una nota una parte importante è l'*ottava*. L'ottava costituisce l'altezza della nota, un'offset rispetto a quella più bassa. Si può pensarla come una somma di  $n * 12$  *semitoni* rispetto alla nota più bassa. Anche qui la codifica che verrà seguita è binaria. Essendoci undici ottave,  $C_{-1}; \dots; C_9$ , dovrebbero essere usati 4 bit per la rappresentazione ma, visto che le ottave  $C_{-1}, C_0$  e  $C_9$  non vengono praticamente mai utilizzate verranno usati tre bit per rappresentare le otto rimanenti.

### Codifica

$\frac{4}{4}$	000
$\frac{2}{4}$	001
$\frac{1}{4}$	010
$\frac{1}{8}$	011
$\frac{1}{16}$	100
$\frac{1}{32}$	101
$\frac{1}{64}$	110

### Codifica

$C_1$	000
$C_2$	001
$C_3$	010
$C_4$	011
$C_5$	100
$C_6$	101
$C_7$	110
$C_8$	111

## 2.4 Codifica delle pause

Le pause sono uno strumento musicale molto comune, vengono utilizzate per dire ad uno strumento di non suonare. L'unico dato che portano è quello della durata, visto che non hanno una ottava di riferimento. Verranno codificate con *IIII* nel campo di codifica della nota, in quanto denotano una nota inesistente

## 2.5 Esempi di codifica

Do di $\frac{4}{4}$ , della terza ottava:	01100100000
Do di $\frac{1}{4}$ , della terza ottava:	01100100100

## 3 Rete neurale ricorrente

### 3.1 Cenni teorici

TODO

### 3.2 Numero di input

È stato dichiarato nell'incipit che il tipo di rete utilizzato per il machine learning è una rete neurale ricorrente. Questo è vero ma, per approfondire meglio l'argomento, sono state utilizzate anche reti neurali semplici. La differenza tra le due è che nella prima tra i neuroni dell'hidden layer sono presenti dei percorsi con cicli mentre nella seconda no. Teoricamente ad una RNN basterebbe una sola nota in input per imparare una canzone intera, visto che ha memoria di quello che ha visto ed in che ordine lo ha visto, ma si è reso necessario ampliare l'input ad otto note (come si vede in [1]) perchè una sola non risultava sufficiente per limiti di hardware. Con otto note in input la probabilità che la rete indovini il target aumenta perchè rendiamo esplicita un pezzo di storia della canzone (le  $N$  note precedenti al target). Durante la fase di scrittura del codice è stata riscontrata una relazione tra il numero di note in input e la correttezza dell'output che conferma quanto detto; ovvero se come input si dava una sequenza composta solo da una sola nota questa riusciva a predirne correttamente solo un'altra come successiva. Se quindi ci si trovava nella situazione di una doppia scelta ecco che la rete non riusciva più a distinguere le due note. Sempre facendo riferimento allo scritto citato sopra ([1]) si legge che la rete creata era caratterizzata dall'aver otto note come input. Non c'è un modo per decidere in arbitrariamente il numero di note in input ma bisogna trovare un compromesso tra la quantità di sequenze che si possono riconoscere e la complessità del sistema.

### 3.3 Tipo di rete utilizzata

La rete neurale ricorrente è così composta:

- 1 layer di input con  $N_{note} * 11$  neuroni;
- 1 hidden layer con 10 neuroni;
- 1 layer di output con 11 neuroni.

C'è una full connection tra il layer di input e l'hidden layer, tra l'hidden layer e il layer di output e tra l'hidden layer e se stesso.<sup>1</sup> La rete neurale non ricorrente è identica solo non presenta la connessione tra l'hidden layer e se stesso.

## 4 Train della rete

Per allenare la RNN è stato scelto l'algoritmo di BackPropagation, associato a Gradient Descent. L'aggiornamento dei pesi viene fatto in batchlearning, ovvero alla fine di tutte le epoch, per la rete neurale ricorrente e on-line, cioè alla fine di ogni epoch, per la rete neurale non ricorrente. Sono stati utilizzati diversi valori di learning rate e di momentum, due parametri che influiscono su come la rete è allenata e, dopo diverse prove, sono stati utilizzati valori molto piccoli di entrambi. È stata valutata l'ipotesi di usare l'algoritmo di BackPropagation Through Time, versione particolare del precedente adattata alle RNN, ma la complessità dell'implementazione l'ha resa infattibile.

L'errore funzione di errore da minimizzare durante il la fase di train è quello dei minimi quadrati.

---

<sup>1</sup>I pesi che portano dal nodo  $A$  al nodo  $B$  possono essere diversi di quelli che portano dal nodo  $B$  al nodo  $A$

## References

- [1] Peter M Todd. A connectionist approach to algorithmic composition. *Computer Music Journal*, pages 27–43, 1989.