

UNIVERSITÀ DI TRENTO

CORSO DI LAUREA IN INFORMATICA

A Musical Version of Recurrent Neural Network

Dellera ANDREA 158365

April 28, 2015

1 Introduzione

Il mondo della musica sempre stato qualcosa di astratto, di impalpabile. Regge ormai da decenni la figura del rocker che, grazie alla sua chitarra, anima folle sempre più grandi e si oppone alle scelte del governo corrente. Un po' più in disuso è la figura del musicista in voga negli anni d'oro della musica classica, da Mozart lo scrittore maltrattato dalla vita a Beethoven che, nonostante la sordità, riuscì a comporre un capolavoro come la *sonata al chiaro di luna*.

Ma il processo creativo che sta dietro alla stesura di un brano può essere riprodotto da un qualcosa che non pensa come un computer? Può un algoritmo, una tecnica di programmazione, portare una macchina a produrre delle melodie?

Lo scopo di questa tesi è di creare delle melodie utilizzando una tecnica di machine learning per portare la macchina ad acquisire brani già scritti e successivamente creare delle melodie.

2 Codifica

Il primo passo per l'implementazione di questo obiettivo è la codifica delle note. Qui entrano in gioco vari parametri (sono stati presi in considerazione i più essenziali):

- nome della nota;
- durata della nota;
- presenza del 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 europeo corrispondono a *A, B, C, D, E, F, G*; in questo paper verrà utilizzato il secondo, perchè più compatto e sintetico.

Come era già stato analizzato da Todd [1] il problema principale consiste nel rendere più generale possibile una certa melodia, rendendola quindi scollegata dalla chiave in cui è stata scritta¹. Il metodo migliore per perseguire questo obiettivo è rappresentare la distanza tra due note in semitoni, cioè la differenza più piccola tra due note. Ecco che quindi tutto ciò di cui si avrà bisogno sarà la nota di partenza. Ecco quindi che la stessa melodia, diversa a seconda della chiave di lettura, diventa uguale quando si guarda la differenza tra le note.

Notazioni

| | | | | | | | |
|------------|----|----|----|----|----|----|-----|
| Americana: | A | B | C | D | E | F | G |
| Europea: | La | Si | Do | Re | Mi | Fa | Sol |

Intervalli

| | | | |
|---|----|----|----|
| A | C | E | G |
| A | +3 | +4 | +3 |
| C | E | G | Bb |
| C | +3 | +4 | +3 |

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.

Con questa codifica il sistema ad intervalli non viene perso. Per, date due note x_1 e x_2 , se $x_1 \leq x_2$ l'intervallo si ottiene facendo $x_2 - x_1$ mentre se $x_2 \leq x_1$ si farà $x_1 - x_2$.

¹esistono diverse tonalità in cui è possibile scrivere una certa melodia, determinate dalla successione delle note (TODO)

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 $\frac{4}{4}$ (\circ). 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 $\frac{4}{4}$ (\circ), $\frac{2}{4}$ (\flat), $\frac{1}{4}$ (\bullet). Le durate che verranno codificate arriveranno fino ad $\frac{1}{64}$.

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

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 |

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 sar 0 se non c'è e 1 in caso contrario.

2.3 Codifica dell'ottava

In ci che va a costituire una note 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 pi bassa. Anche qui la codifica che verrà seguita 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 usate ne verranno usati 3 per rappresentare le otto rimanenti.

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 *1111* nel campo di codifica della nota, in quanto denotano una nota inesistente

References

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