INFORMATICA MUSICALE

UNIVERSITA' DEGLI STUDI DI CATANIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA
LAUREA TRIENNALE IN INFORMATICA
A.A. 2020/21
Prof. Filippo L.M. Milotta

ID PROGETTO: 0F

TITOLO PROGETTO: "Ok Google, facciamo due chiacchiere?"

AUTORE 1: Cigna Gaia

AUTORE 2: Di Mauro Davide

AUTORE 3: Falcone Chiara

Indice

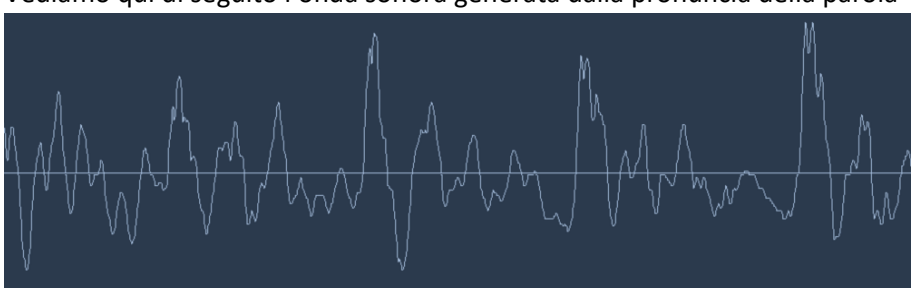
1. Obiettivi del progetto

Elaborazione del linguaggio naturale e NLP

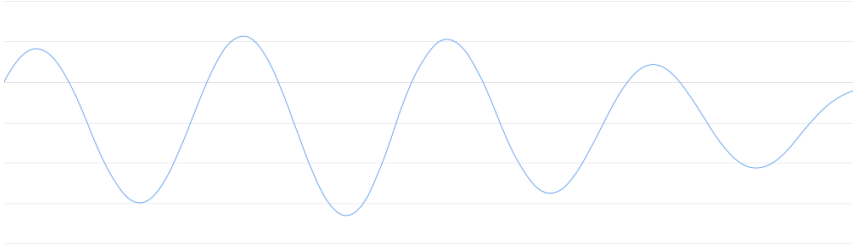
Per capire il funzionamento del NLP, dobbiamo distinguere le sue due componenti: la comprensione e la generazione.
Quando parliamo di comprensione del linguaggio naturale, parliamo della capacità dell'elaboratore di comprendere il nostro linguaggio tramite sistemi di riconoscimento vocale che convertono il linguaggio naturale in quello artificiale. La maggior parte di questi sistemi sono basati sui modelli Hidden Markov, modelli che utilizzano la statistica e la matematica per determinare ciò che viene pronunciato. In pratica, suddividono la voce pronunciata in piccole unità (di solito 10-20 millisecondi) e la confrontano con il discorso preregistrato per determinare il fonema che hai pronunciato in ogni unità del tuo discorso (un fonema è l'unità più piccola di parola che c'è). Guardando la serie di fonemi, il modello determina statisticamente sotto forma di testo le parole e le frasi che hai detto.
Nella parte di comprensione reale, il computer deve capire cosa rappresenta ogni parola. Per farlo si avvale di un processo chiamato Part-of-Speech (POS), che consiste nell'etichettare ogni parola secondo una determinata categoria come nome, verbo, avverbio, pronome, e così via. Gli algoritmi NLP moderni utilizzano l'apprendimento automatico delle macchine per applicare queste regole al linguaggio naturale e determinare il significato più probabile dietro ciò che è stato detto.
La generazione del linguaggio naturale (NLG, o Natural Language Generation), invece, risulta molto più semplice da realizzare. Questa applicazione traduce il linguaggio artificiale di un computer in testo e può anche fare un passo in avanti traducendo quel testo in un discorso udibile con sintesi vocale (è il processo inverso a quello visto poco fa). Inizialmente, il sistema NLP determina quali informazioni tradurre in testo. Successivamente organizza la struttura di come te lo dirà, utilizzando un lessico e un set di regole grammaticali, per formare frasi complete.
Infine, utilizzando un database vocale (che contiene registrazioni da un doppiatore), il motore riunisce tutti i fonemi registrati per formare una sequenza di parole coerente.

Riconoscimento vocale

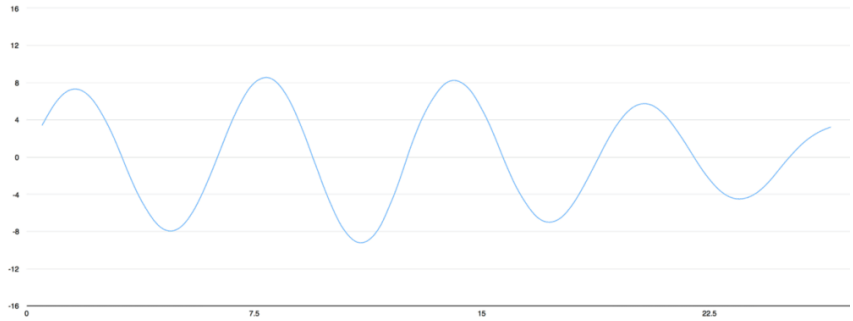
Per il riconoscimento vocale si utilizza il deep learning, il quale si basa sul concetto di rete neurale artificiale, ossia un modello matematico ispirato, dal punto di vista funzionale, ai sistemi neurali biologici del cervello umano. Una rete neurale artificiale profonda è composta da una serie di neuroni che sono disposti su più livelli collegati fra loro. Tali reti sono in grado di elaborare, però, come input, solo dati numerici e non stringhe testuali. Ad oggi, il deep learning, attraverso la combinazione di word embeddings e reti convolutive e ricorrenti, rappresenta l'approccio maggiormente adottato per affrontare problematiche relative all'elaborazione e comprensione del linguaggio naturale.
Ma come avviene il riconoscimento vocale?
Poiché la nostra voce può produrre una stessa parola, o frase, con diverse velocità, è importante che la macchina comprenda ciò che "le si dice" a prescindere da questo fattore. Dunque, abbiamo diversi step da considerare per questa procedura.
Anzitutto, poiché il suono viene trasmesso come onda sonora, è necessario che questa onda venga trasformata in numeri. Ma come avviene questa trasformazione?
Vediamo qui di seguito l'onda sonora generata dalla pronuncia della parola "ciao".



Zoommiamo su una piccola parte delle onde sonore per vedere meglio:



Per trasformare questa onda sonora in numeri è sufficiente registrare l'altezza dell'onda per punti equidistanti: questo processo è chiamato campionamento.



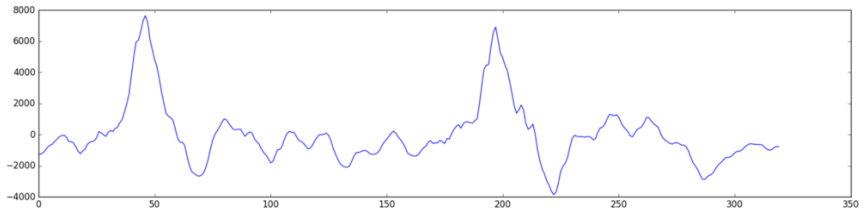
Stiamo prendendo una lettura migliaia di volte al secondo e registrando il numero che rappresenta l'altezza dell'onda sonora in quel determinato punto nel tempo.
Per il riconoscimento vocale, una frequenza di campionamento di 16 kHz (16.000 letture al secondo) sarà sufficiente a coprire la gamma di frequenza della voce umana. Campioniamo allora l'onda sonora della parola "ciao" pronunciata 16.000 volte al secondo. Ecco i primi 100 numeri:

```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41, -169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448
, -397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451, 1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461,
 4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499, -488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148, -
1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325, 350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544]
```

Ora abbiamo una serie di numeri, ognuno rappresentante l'ampiezza dell'onda sonora a 1 / 16000i di secondo di intervallo. Adesso, andremo a pre-processare questi numeri prima di inserirli nella rete neurale.
Cominciamo raggruppando i numeri in gruppi che rappresentino 20-millisecondi di audio ciascuno. Ecco i nostri primi 20 millisecondi di audio (vale a dire i nostri primi 320 campioni):

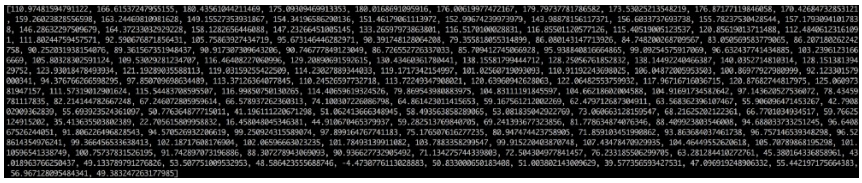
```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41, -169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448
, -397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451, 1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461,
 4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499, -488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148, -
1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325, 350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544,
-1815, -1725, -1341, -971, -959, -723, -261, 51, 210, 142, 152, -92, -345, -439, -529, -710, -907, -887, -693, -403, -180, -14, -12, 29, 89, -47, -
398, -896, -1262, -1610, -1862, -2021, -2077, -2105, -2023, -1697, -1360, -1150, -1148, -1091, -1013, -1018, -1126, -1255, -1270, -1266, -1174, -10
03, -707, -468, -300, -116, 92, 224, 72, -150, -336, -541, -820, -1178, -1289, -1345, -1385, -1365, -1223, -1004, -839, -734, -481, -396, -580, -52
7, -531, -376, -458, -581, -254, -277, 50, 331, 531, 641, 416, 697, 810, 812, 759, 739, 888, 1008, 1977, 3145, 4219, 4454, 4521, 5691, 6563, 6909,
6117, 5244, 4951, 4462, 4124, 3435, 2671, 1847, 1370, 1591, 1900, 1586, 713, 341, 462, 673, 60, -938, -1664, -2185, -2527, -2967, -3253, -3636, -38
59, -3723, -3134, -2380, -2032, -1831, -1457, -804, -241, -51, -113, -136, -122, -158, -147, -114, -181, -338, -266, 131, 418, 471, 651, 994, 1295,
 1267, 1197, 1291, 1110, 793, 514, 370, 174, -90, -139, 104, 334, 407, 524, 771, 1106, 1087, 878, 703, 591, 471, 91, -199, -357, -454, -561, -605,
-552, -512, -575, -669, -672, -763, -1022, -1435, -1791, -1999, -2242, -2563, -2853, -2893, -2740, -2625, -2556, -2385, -2138, -1936, -1803, -1649,
 -1495, -1460, -1446, -1345, -1177, -1088, -1072, -1003, -856, -719, -621, -585, -613, -634, -638, -636, -683, -819, -946, -1012, -964, -836, -762,
 -788]
```

Tracciare quei numeri come un semplice grafico a linee ci dà una stima approssimativa dell'onda sonora originale per ogni blocco di 20 millisecondi di tempo:

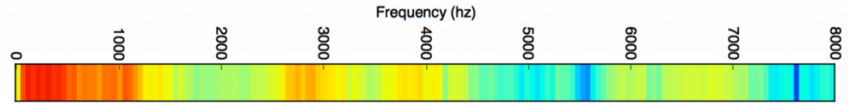


Pur essendo questa registrazione lunga soltanto un cinquantesimo di secondo, essa è un complesso miscuglio di diverse frequenze del suono, infatti troviamo suoni più bassi e suoni più alti insieme. Per semplificare l'elaborazione di questi dati da parte di una Rete Neurale, spezzetteremo quest'onda sonora in parti più piccole. Da un lato le basse frequenze, poi le medio basse frequenze e così via. In questo modo, quando sommiamo l'energia contenuta in ciascun frammento siamo in grado di creare un'impronta univoca per quella sequenza sonora. Lo facciamo con una operazione matematica chiamata trasformata di Fourier.

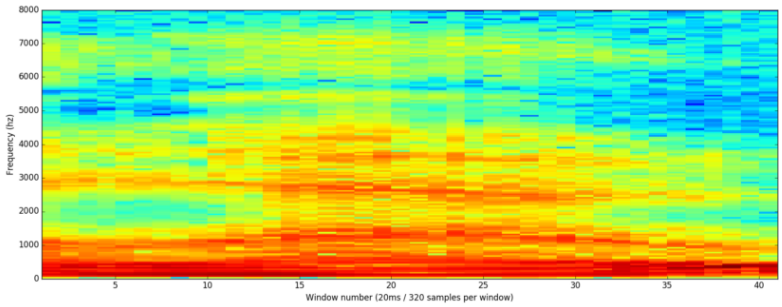
Il risultato finale è un punteggio di quanto sia importante ogni gamma di frequenza, dai toni bassi (cioè note basse) a quelli più alti. Ogni numero che segue rappresenta la quantità di energia che è contenuta in ogni banda 50Hz della nostra clip audio di 20 millisecondi:



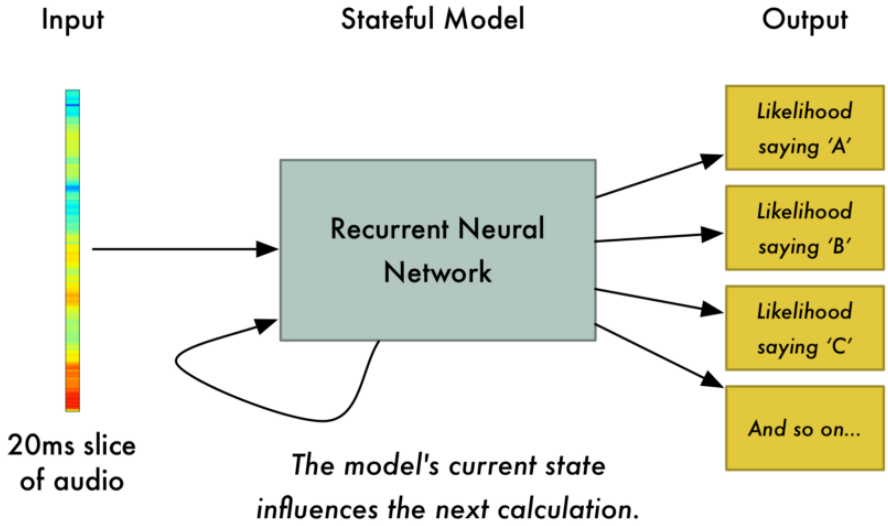
Ma è molto più facile da capire quando lo rappresentiamo in modo grafico:



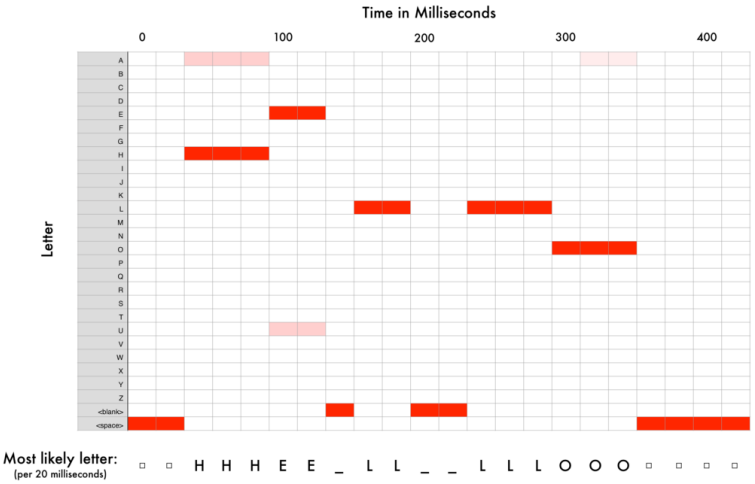
Se ripetiamo l'operazione su tutti i blocchi da 20 millisecondi del nostro file audio, otteniamo uno spettrogramma (ogni colonna da sinistra a destra è un blocco di 20ms):



Ora che abbiamo il nostro audio in un formato che è facile da processare, dovremo inserirlo in un Deep Neural Network. L'input della Rete Neurale saranno gli spettrogrammi dei blocchi audio di 20 millisecondi identificati in precedenza. Per ogni fetta audio, si cercherà di capire la lettera corrispondente al suono emesso.



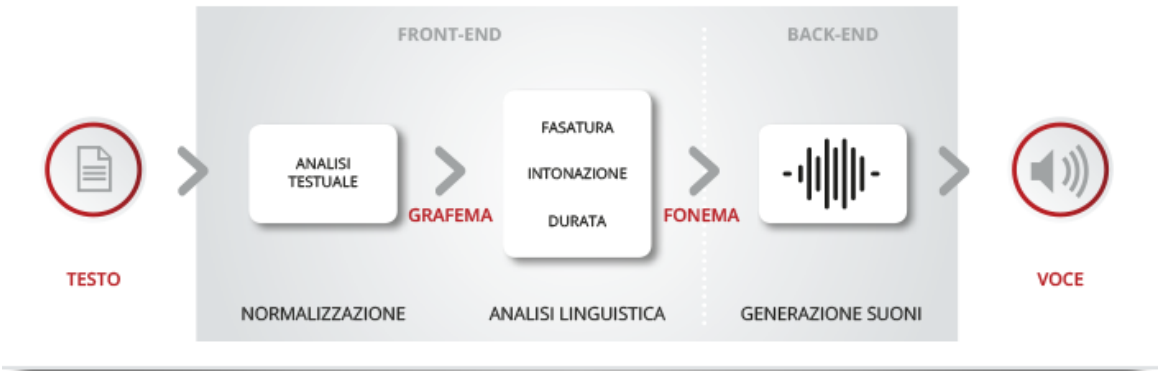
Useremo una Rete Neurale Ricorrente, cioè una Rete Neurale i cui risultati sono influenzati dai risultati precedenti. Questo perché ogni lettera individuata influisce sulle possibilità che una specifica lettera venga dopo. Ad esempio, se abbiamo detto "CIA", è molto probabile che dopo diremo "O", ovvero "CIAO" e non qualcosa di completamente insensato come "QGSOPR". Quindi, avere in memoria le lettere precedenti permette alla Rete Neurale di capire con maggiore accuratezza cosa diremo dopo. Una volta processate tutte le clip audio di 20 millisecondi nella Rete Neurale (un pezzo alla volta), otterremo una mappatura di ogni blocco audio alle lettere che più probabilmente avremo pronunciato nella registrazione audio originale. Ecco l'aspetto della mappatura della pronuncia della parola "Ciao":



A questo punto, abbiamo alcuni passaggi da fare per pulire i dati in uscita. In primo luogo, sostituiremo tutte le lettere ripetute eccessivamente con lettere singole (dove sensato). Poi andremo a rimuovere eventuali spazi vuoti. Questo ci lascia con alcune possibili trascrizioni che possono tutte somigliare alla parola pronunciata. Ma la macchina deve capire quale delle possibili trascrizioni è quella che abbiamo effettivamente pronunciato. Prevedendo un carattere alla volta la Rete Neurale produce la trascrizione più sensata a livello della singola lettera e non della parola intera.
Per risolvere questo problema, è sufficiente combinare queste previsioni di pronuncia basate sulla somma delle previsioni delle singole lettere, con un vasto dataset di parole corrette (libri, articoli, etc..) con l'obiettivo di scartare le trascrizioni che hanno una rilevanza statistica bassa.
Delle nostre eventuali trascrizioni, la macchina capirà quale sia la parola pronunciata poiché essa apparirà più frequentemente in un database di testi (per non parlare di nostri dati audio di formazione basati su originale) e quindi è probabilmente la trascrizione più corretta. Così sceglieremo "Hello" come la nostra trascrizione finale.

Sintesi vocale

Le macchine di oggi, oltre ad essere in grado di svolgere il processo di riconoscimento vocale, sono in grado di svolgere anche la sintesi vocale. La sintesi vocale non è altro che la riproduzione artificiale della voce umana.
Questo fenomeno avviene grazie all'interazione tra componenti software e hardware. Per quanto riguarda il lato software, una componente molto importante è detta Text to Speech System; esso è in grado di convertire un testo ricevuto in input in una sua rappresentazione linguistica e/o fonetica.



La sintesi vocale è composta fondamentalmente da due parti: front end e back end. La prima si occupa di normalizzare il testo e del processo dell'assegnazione delle trascrizioni fonetiche alle parole, mentre la seconda si occupa di convertire la rappresentazione linguistica in suono.

Il processo di normalizzazione del testo è la fase in cui un testo grezzo, contenente parole, numeri, viene convertito in parole scritte. Alla base di questa fase devono essere affrontate diverse problematiche, ovvero: a livello di sistema di scrittura, per esempio, bisogna tener conto i cambi di direzione. Nella lingua araba si legge da destra a sinistra, cosa che non avviene se si seguono numeri e caratteri latini; a livello prosodico e di intonazione, a seconda della lingua, è necessario saper gestire il tono di voce, la lunghezza delle vocali, l'accento, le pause. Per quanto riguarda i numeri è importante riuscire a leggere le cifre in tutti i modi possibili (a due a due, a tre a tre, etc); altra questione da affrontare è quella di essere in grado di distinguere le abbreviazioni, le quali potrebbero assumere più significati in base al contesto in cui vengono utilizzate.
Come abbiamo già detto, il back end si occupa di convertire la rappresentazione linguistica in suono. Anche qui sorgono delle problematiche, in questo caso inerenti alla pronuncia delle parole. La soluzione a ciò la troviamo in due tecniche distinte: il dizionario e le regole. Il dizionario contiene le parole e le corrispettive pronunce corrette. Questo approccio sicuramente ha il grande vantaggio di essere rapido e preciso però, com'è possibile dedurre, sorgono degli inconvenienti nel momento in cui viene data in input una parola non presente all'interno del dizionario. Inoltre, anche volendo, integrando una nuova parola all'interno del dizionario, dobbiamo tenere conto che la memoria, in cui vengono memorizzate, è limitata, quindi questo è un altro grosso svantaggio. L'approccio basato sulle regole permette di capire la pronuncia delle parole partendo dall'ortografia. A differenza dell'approccio appena descritto, esso non presenta problemi in termini di spazio di memoria però, a causa di ortografie meno in uso, la complessità di questo processo aumenta. Per esempio, in inglese, nella parola "of" è l'unica parola in cui la lettera f viene letta [v]. Per alcune lingue, in particolare quelle con un'ortografia fonemica (un po' come l'italiano in cui la maggior parte delle volte una parola viene letta così com'è scritta), viene utilizzata la combinazione di entrambi gli approcci.
La sintesi vocale, quindi, è la rappresentazione artificiale della voce umana, ma in che modo ciò avviene? Grazie a diverse tecnologie, tra cui la sintesi concatenante, la sintesi formante, la sintesi basata sull'apprendimento profondo.
La sintesi concatenante è una tecnica utilizzata per sintetizzare dei suoni e ciò avviene attraverso la concatenazione di unità precedentemente registrate. Essa viene utilizzata nella sintesi vocale per generare sequenze di suoni specificate dall'utente da un database costruito dalle registrazioni di altre sequenze. La sintesi formante, a differenza di quella concatenante, non utilizza parole pre-registrate, ma consiste nella generazione di forme d'onda di cui si modulano alcuni parametri acustici come la frequenza fondamentale, i toni e i livelli di rumore. La sintesi basata sull'apprendimento profondo avviene attraverso l'utilizzo delle reti neurali. Ad utilizzare quest'ultimo tipo di sintesi è WaveNet.

<u>Analisi di un caso: WaveNet</u>

WaveNet è una tecnologia di sintesi vocale, sviluppata da Google Deep Mind. Nel suo nucleo troviamo un tipo particolare di rete neurale. Essa prende in input lo spettrogramma fornito dal sistema di text-to-speech (TTS) e restituisce come output dei campioni sonori, uno per volta, con una frequenza massima di 16Khz, più che sufficiente per generare voci umane fedeli e sperimentare anche con qualche traccia musicale.
Prima di entrare nel dettaglio bisogna aver chiari alcuni concetti:

A) Convoluzioni Causali Dilatate

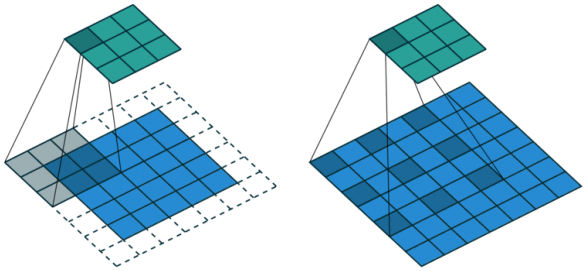
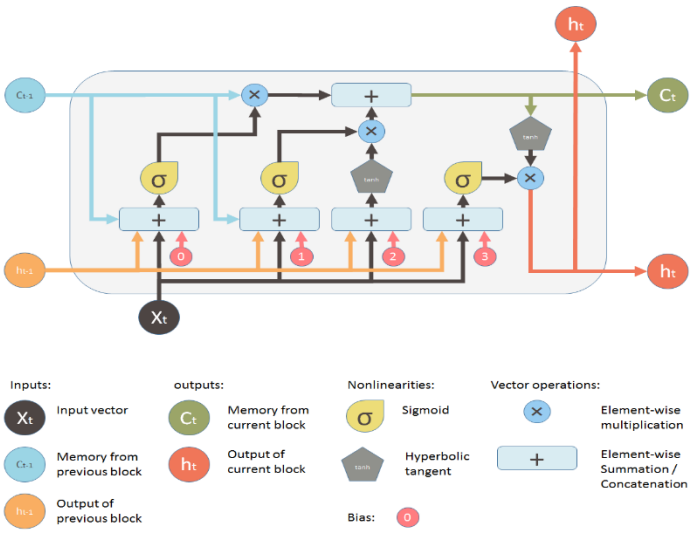


Figura 1: Convoluzione e Convoluzione dilatata

Come si può vedere in Figura 1, a differenza di una convoluzione (a sinistra), una convoluzione dilatata (a destra) prende in input valori equidistanti e non adiacenti tra loro, in pratica si applica una convoluzione con un kernel più grande, ma con degli 0 al posto dei valori da escludere nella dilatazione. In particolar modo

WaveNet utilizza un caso particolare di convoluzioni dilatate dette Causali poiché esse permettono di non violare la legge per cui valori predetti a tempo T non siano dipendenti da valori predetti a tempi futuri T+1, T+2, etc…, ma solo da valori precedenti.

A) Long-Short Term Memory



Le LSTM sono una particolare architettura di rete neurale. Esse, a differenza delle semplici reti neurali, riescono a mantenere in una memoria temporanea ciò che può essere importante per predizioni successive ed usa tali informazioni per affinare le predizioni successive. Ciò è cruciale nella generazione di audio poiché, se si vuole ottenere un audio quanto più fedele alla forma d'onda del suono originale, ogni singolo campione deve dipendere fortemente dai precedenti per evitare discontinuità brusche tra un campione ed il suo successivo.

Figura 2: LSTM

A) Come funziona WaveNet?

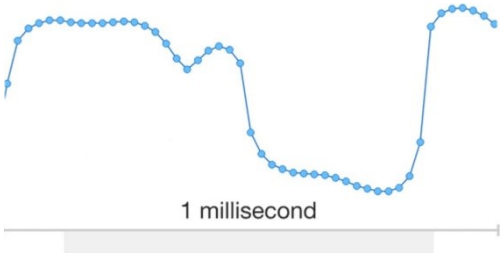


Figura 3: Campioni generati da WaveNet

Per iniziare è importante capire quale sia l'obiettivo. Dopo tutte le operazioni compiute WaveNet ci restituirà un campione del nostro audio ad un tempo t, uno alla volta, in modo consecutivo: generato un campione, questo finirà nell'input del prossimo campione da generare e così via. Si devono generare 16.000 campioni al secondo e se si conta che l'audio è solitamente codificato con una risoluzione minima di 16 bit si ha che ogni singolo campione può assumere 65.536 valori diversi. Ciò non è efficiente a livello di costi computazionali. Per questo WaveNet utilizza la codifica μ-law, in questa maniera i possibili valori assunti da un campione vengono ridotti a 256. In fase di training l'input è formato da vere voci umane registrate.

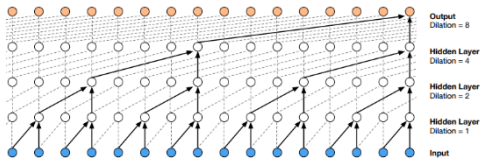


Figura 4: Livelli convoluzionali

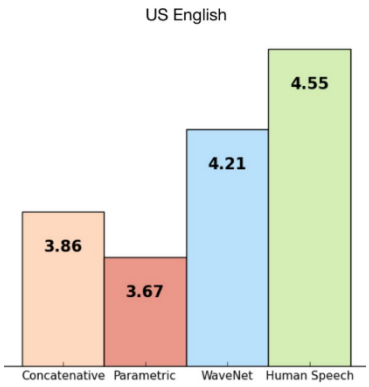
Dopo aver ricevuto in input lo spettrogramma dal TTS, WaveNet utilizza vari livelli di convoluzioni causali dilatate per rispettare l'ordine temporale dei campioni e per poter riuscire a gestire un input molto grande con pochi livelli convoluzionali così da diminuire i tempi di computazione.

Dopo svariati livelli convoluzionali e algoritmi di scrematura dei dati WaveNet sceglie uno tra i 256 valori possibili attraverso una distribuzione probabilistica Softmax.
Seguendo questa formula

$$p(x) = \prod_{t=1}^{T} p(x_t | x_1 \dots \dots . x_{t-1})$$

si riesce a condizionare i campioni successivi ed aggiungendo ulteriori condizioni alla probabilità si possono inserire variabili globali come il tono di voce, l'accento o variabili locali che guidano WaveNet nelle caratteristiche peculiari del parlato, come l'ordine delle parole, suoni lunghi o corti ed intonazioni particolari.

Applicazioni su Google Duplex



Da test sperimentali si è appurato che WaveNet riesce molto più fedelmente ad emulare il parlato umano, per tale motivo è utilizzato in numerosi servizi di Google. Tra questi il più importante è sicuramente Google Assistant; grazie a WaveNet Infatti Assistant inglese e tedesco è molto più "vivo" e fedele, permettendo una maggior facilità e confort nell' utilizzarlo ed interfacciarsi con esso. Proprio grazie alla sua alta fedeltà WaveNet può essere usato per qualcosa di più complesso: da circa 2 anni, infatti gli utenti americani possono utilizzare un servizio, Google Duplex, che permette di far compiere una telefonata a Google Assistant, per prenotare un tavolo al ristorante o per rispondere ai centralini telefonici senza essere disturbati, basterà indicare ad Assistant orario e luogo. Alcune demo hanno mostrato come gli interlocutori dall'altra parte non si siano accorti minimamente della natura robotica di Assistant, risultata così simile ad una voce umana che l'ente per le comunicazioni americano ha costretto Google ad inserire un annuncio che esplicitasse che la telefonata era fatta da Assistant.

2. Riferimenti Bibliografici

https://lorenzogovoni.com/elaborazione-del-linguaggio-naturale/ : questo documento fornisce una chiara idea sul natural language processing e sul suo funzionamento generale.

https://www.agendadigitale.eu/cultura-digitale/linguaggio-naturale-e-intelligenza-artificiale-a-che-punto-siamo/ : nel sito qui riportato, ritroviamo un'ottima descrizione del deep learning applicato al riconoscimento vocale.

https://medium.com/botsupply/il-machine-learning-%C3%A8-divertente-parte-6-86cd682ff71a : in questo articolo ritroviamo nel dettaglio la procedura del riconoscimento vocale passo dopo passo, con molti esempi e, soprattutto, immagini molto chiare e utili.

https://www.celi.it/blog/2017/01/sintesi-vocale-come-dare-una-voce-alle-macchine/ : questo documento definisce sinteticamente cos'è la sintesi vocale e quali sono le sue fasi.

https://it.qaz.wiki/wiki/Speech_synthesis : questo documento fornisce una chiara descrizione delle fasi della sintesi vocale e analizza le diverse tecnologie che ne fanno utilizzo.

https://deepmind.com/blog/article/wavenet-generative-model-raw-audio : contiene un articolo scientifico in allegato compilato da ricercatori del progetto esplicativo dei dettagli della tecnologia.

https://medium.com/@evinpinar/wavenet-implementation-and-experiments-2d2ee57105d5 : esempio di implementazione grossolana di WaveNet facilitandone la comprensione del funzionamento.

https://www.quora.com/How-does-Google-Duplex-work : spiegazione sintetica degli step e meccanismi nascosti dietro l'applicazione più famosa di WaveNet.

3. Argomenti Teorici Trattati

Trasformata di Fourier

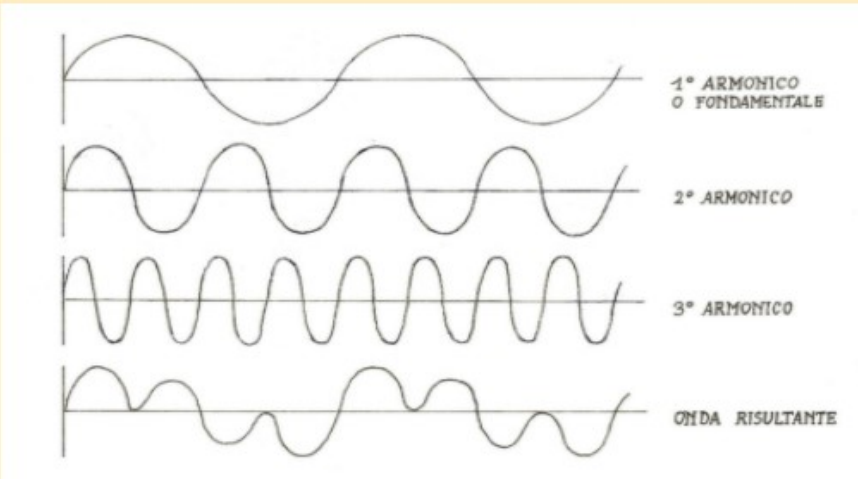
Poiché la pronuncia di una parola nel riconoscimento vocale è un'onda sonora aperiodica, ci serviamo della trasformata di Fourier (piuttosto che della serie), che vediamo qui di seguito:

$$y(t) = \int_{-\infty}^{+\infty} C(n) e^{i\omega n t} dn \qquad C(n) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} y(t) e^{-i\omega n t} dt$$

Nella trasformata di Fourier le frequenze delle onde elementari non apparterranno all'insieme discreto dei multipli della frequenza fondamentale, ma varieranno in un insieme continuo. Essa ci permette di individuare le componenti di frequenza di un segnale.

Frequenza fondamentale

Il suono è un segnale complesso costituito da una serie di componenti dette armoniche, ognuna a frequenza multipla rispetto alle precedenti. Tra queste armoniche, la prima è detta frequenza fondamentale che corrisponde, dal punto di vista psicoacustico, all'altezza o tono del suono prodotto.



Spettrogramma

Lo spettrogramma è la rappresentazione grafica del suono. Ciò è permesso grazie alla relazione di tre variabili che caratterizzano qualsiasi suono: la frequenza (asse delle ordinate), il tempo (asse delle ascisse), l'intensità (scala di grigi o colore). La relazione tra l'intensità del suono e la scala di grigi può essere lineare o logaritmica.

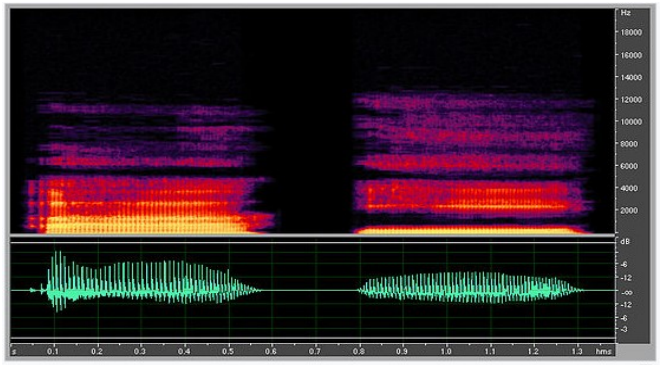


Figura 3 - Spettrogrammi dei suoni vocalici "a" ed "i" pronunciati da un madrelingua italiano e relative forme d'onda

Spesso a questa rappresentazione viene associata la rappresentazione della forma d'onda per rendere più veloce la comparazione tra la forma d'onda e le sue componenti spettrali.

Lo spettrogramma viene ampiamente impiegato negli studi riguardanti la voce umana. Infatti, grazie ad esso, è possibile individuare i foni di una parola permettendo la ricostruzione di quest'ultima, cosa non possibile se conoscessimo esclusivamente la forma d'onda.

Campionamento

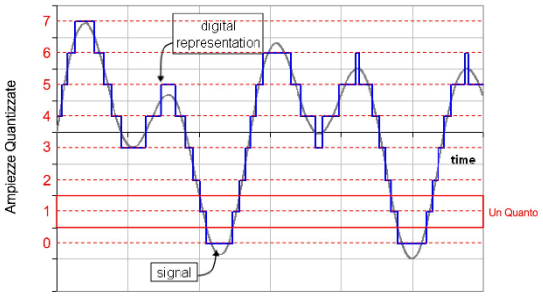
Poiché in questo progetto trattiamo il riconoscimento vocale, la nostra onda sonora sotto esame è la voce umana che è un'onda aperiodica.

Nel riconoscimento vocale abbiamo visto che per trasformare l'onda sonora (in questo caso la pronuncia della parola "ciao") in numeri, registrando l'altezza dell'onda per punti equidistanti, ci serviamo del campionamento. Infatti, proprio come nell'esempio sopra riportato, descritto chiaramente nell'immagine, consideriamo solo alcuni valori del segnale. La frequenza con cui scegliamo i vari campioni è detta "tasso di campionamento". Per segnali non periodici viene utilizzata la trasformata di Fourier, che vedremo tra poco.

Per una qualità del segnale finale più elevata, è sicuramente preferibile una frequenza di campionamento alta. Però, bisogna considerare che, oltre una certa soglia, prendere troppi campioni può essere dispendioso in termini di memoria.

Quantizzazione

La quantizzazione è un processo fondamentale nella digitalizzazione del suono. Essa permette di discretizzare i valori assunti dai campioni che descrivono l'onda sonora.



In poche parole, si dividono i possibili valori d'ampiezza che può assumere un'onda sonora in fasce, dove tutti i valori all'interno della stessa fascia saranno rappresentati da un solo valore (quanto) nella versione digitalizzata. Tale suddivisione in fasce può essere di due tipi: uniforme e non uniforme. Nel primo caso ogni fascia contiene un numero uguale di valori da mappare e quindi ogni quanto rappresenta un range in numero uguale di valori, nel secondo caso, invece, in base alle applicazioni si hanno fasce che ricoprono più valori di altre, così da avere maggiore precisione in aree più importanti come, ad esempio, i suoni con ampiezze piccole, dove variazioni di ampiezza vengono percepite più facilmente dall'orecchio umano. Tale riduzione dei possibili valori ovviamente causa errori, che danno vita a distorsioni, dette appunto "rumore di quantizzazione", simile al rumore bianco. Nel caso della quantizzazione uniforme l'errore massimo è pari a $\frac{VFS}{2Q}$ dove VFS (valore di fondo scala) è il range di valori che si vuole rappresentare e Q è il numero di livelli di quantizzazione. Il numero di livelli di quantizzazione dipende dal numero di bit che si scelgono per rappresentarli con Q=2^N con N=numero di bit. Tali errori danno origine a distorsione che è misurata tramite l'SQNR (Signal to Quantization Noise Ratio), un indice che calcola il rapporto tra la potenza del segnale utile e quella del rumore. E' descritto dalla formula: $SQNR = 2^N * \sqrt{\frac{3}{2}}$ con N il numero di bit usati. Da tale formula è facilmente deducibile che un SQNR alto indichi una qualità maggiore, poiché un numero più alto è dato da numero maggiore di bit. Può essere anche calcolato in decibel con la formula: $SQNR_{db} = 10log_{10}(SQNR^2)$ (Nel caso dei CD a 16 bit si ha SQNR=96db).

La codifica μ-law

La codifica μ-law nasce per essere utilizzata nei servizi di telefoni ISDN che permettevano un traffico di 64Kbps che con un campionamento a 8Khz per campionare la voce umana (fino a 3-4Khz), il quale lasciava solo 8 bit per campione. La codifica μ-law permette, attraverso una quantizzazione non uniforme (logaritmica), di ottenere una qualità simile ad una quantizzazione uniforme a 13-14 bit usandone solo 8. Tale risultato si ottiene operando una spaziatura sulla gamma dinamica che è più grande per i valori di ampiezza larga e più piccola (maggiore precisione) nei valori di ampiezza debole, quindi le ampiezze alte saranno rappresentate da meno valori, mentre quelle basse da più valori. Il restringimento dei valori dalla profondità in bit originale a 8 bit è dato dalla formula:

$$y = \begin{cases} 128 + \dfrac{127}{\ln(1+\mu)} \times \ln(1+\mu|x|) & x \geq 0 \\ 127 - \dfrac{127}{\ln(1+\mu)} \times \ln(1+\mu|x|) & x < 0 \end{cases}$$

Dove μ=255 ed x è il valore dell'input normalizzato tra -1 ed 1.