

Riconoscimento vocale

Autori:

Matteo Ricci 254429

Manuel Di Pietro 254375

Lorenzo Pichilli 255671

Il Riconoscimento vocale è il processo mediante il quale il linguaggio “umano” viene riconosciuto da un computer. Il software è stato implementato in **python3** e utilizza il framework **sklearn** per creare e gestire la rete neurale utilizzata. Il software mette a disposizione due tipi di riconoscimento. Il primo tramite la predizione di una persona in base alla voce registrata o utilizzando un query set, il secondo la predizione delle parole “si” e “no” (è possibile aggiungere anche altre parole). In entrambi i casi, è stato registrato un file audio di estensione “wav” e processato mediante “la trasformazione di fourier”. La trasformata di Fourier consente di dividere un’onda, anche molto complessa(in questo caso,la voce!), in più sotto-componenti, semplificando l’analisi di essa. Per lavorare con tante sinusoidi a diversa frequenza e ampiezza, si utilizzano i numeri complessi per compiere i calcoli, che facilita le operazioni sull’onda iniziale (es: la convoluzione di più segnali nel tempo diventa un semplice prodotto nella dominio della frequenza). Con l’utilizzo dei numeri complessi è stato creato il nostro dataset, che comprende quindi il training set e il test set.

La rete neurale è stata testata con due activation function, la “relu” e la “logistic”, infine si è deciso di utilizzare la “relu” perchè si è rilevata più efficiente. La rete neurale istanziata, ad ogni proprio allenamento, viene salvata in un file denominato “neural.pkl” per evitare di doverla riallenare a ogni avvio del software e permette anche il suo riutilizzo in altri software.

Qui di seguito riporto un esempio di Dataset (per essere interpretati dalla rete neurale sono stati trasformati in numeri reali):

- **Riconoscimento Voce**

Matteo,-120617.164062,46201.5121436,-16932.7397134,58068.3186846,-
6912.86554962,-669.495278803,-57662.4582414,-12467.1191969,-
5881.54114644,-10411.8739913,-14720.1067035,-26481.144855,-
22716.748946,-27965.2826192,-29645.0961672,-10044.9306895,-
42915.9912608,-25977.992171,-30020.3581828,-
56606.1214113,66173.1460204,128602.247274,10586.1415389,39134.137470
2,36358.245187,38737.9798904,30583.9953726,29776.5728994,3780.466298
72,29651.425462,943.024795452,28292.8412827,2237.81935476,30690.1975
528,43722.5830236,6814.01097231,13469.1228122,3200.67210139,5321.214
10203,21391.8537547, ...

- **Riconoscimento Parole**

si,113605.203125,64113.2955758,9003.93155184,9409.13581742,60884.7049
53,7403.36072266,38096.2476943,40926.6841399,17291.479011,13344.1311
376,44516.9011979,51535.5349225,8952.97899841,18432.2599818,1312.590
19554,23615.5837603,25315.6143159,28174.9216016,15448.9774028,16612.
0668316,75514.7809139,7098.82186677,57768.8073181,31775.6198312,2219
0.3687002,4993.49697469,40530.4335416,21688.4743289,41822.1562366,13
718.7765469,25866.2176008,30919.8340087,8032.0489106,29101.690063,18
543.8031381,14998.2494624,5556.45600634,17837.9346185,32980.4960737,
...

- **Esempio di Query_Set utilizzato per indovinare una persona:**

28191.5378802,34165.2998327,8448.63777375,18162.9035588,6534.3215613
,4509.56250345,36780.0430615,17618.7761047,11383.3684595,25287.19589
42,12877.9659102,25751.2556282,32490.1735561,32636.2522596,33073.182
0351,20607.2626203,26733.5725002,24069.2782423,515.710374957,43079.6
023867,31559.4217445,6118.20745873,24459.5643756,23186.6428087,2904.
0219197,12378.3487819,11256.4685756,12548.3247012,6752.07559633,1014
3.3452622,3555.65609308,7284.29846178,1737.27329396,1875.18230272,34
25.89349258,7420.41637944,23890.3219126,16606.7741771,2514.2654101,2
0814.4592648,12837.6157507,3716.6766438,20290.4309808, ...

All'inizio dell'esecuzione del software, all'utente verrà richiesto in input la registrazione della propria voce e il software mostrerà il seguente:

“R” per registrare una nuova voce:

In questo caso attraverso la funzione `record()` viene registrata per 2 secondi (tempo configurabile nel codice cambiando il valore della variabile passato alla funzione “`Function.record()`”) e successivamente elaborato mediante la funzione `convert_fourier()`, che ha come parametri in input: il nome del file audio “`voice.wav`”, il nome della persona che sta parlando (es. “`Matteo`”) ed in fine il nome del file per il salvataggio del dataset sul quale salvare i dati.

“T” per indovinare la voce di una persona, “P” per predire il nostro TestSet e “D” per aggiungere una parola al dizionario. Alla fine di ogni funzione verrà richiesto all'utente di pronunciare “`si`” nel caso voglia terminare il programma o “`no`” nel caso in cui voglia continuare a utilizzarlo.

Esempi di esecuzione:

1.

```
MacBook-Pro-3:ML3 lorenzo$ python3 main.py
Allenamento in corso...
Iteration 1, loss = 13.54754591
Iteration 2, loss = 14.56345778
Iteration 3, loss = 11.85463396
Iteration 4, loss = 0.00320910
Iteration 5, loss = 0.00333154
Iteration 6, loss = 0.34206969
Iteration 7, loss = 0.34217911
Training loss did not improve more than tol=0.000100 for two consecutive epochs. Stopping.
Fatto
Cosa vuoi fare ? (scrivi R per aggiungere una nuova voce, P per predirre il querySet, I per far indovinare chi e' o D per aggiungere una parola al dizionario): p
Predico...
Accuratezza: 71.4285714286%
Vuoi terminare?: (pronuncia si o no):
Registro...
Registrazione effettuata
no
Cosa vuoi fare ? (scrivi R per aggiungere una nuova voce, P per predirre il querySet, I per far indovinare chi e' o D per aggiungere una parola al dizionario): █
```

2.

```
Cosa vuoi fare ? (scrivi R per aggiungere una nuova voce, P per predirre il querySet, I per far indovinare chi e' o D per aggiungere una parola al dizionario): I
Registro...
Registrazione effettuata
Creo il query set...
Query set creato
provo ad indovinare!!!
Penso sia 'Lorenzo'
Ho indovinato?: (scrivi si o no): si
Aggiornamento dataset...
Iteration 1, loss = 14.90200746
Iteration 2, loss = 14.22484206
Iteration 3, loss = 13.54771387
Iteration 4, loss = 7.11413701
Iteration 5, loss = 7.11425879
Iteration 6, loss = 8.80745754
Iteration 7, loss = 6.43726012
Iteration 8, loss = 2.71260011
Iteration 9, loss = 2.60680607
Iteration 10, loss = 4.40588006
/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-packages/sklearn/neural_network/multilayer_perceptron.py:564: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
Dataset aggiornato!
```

3.

```
Cosa vuoi fare ? (scrivi R per aggiungere una nuova voce, P per predirre il querySet, I per far indovinare chi e' o D per aggiungere una parola al dizionario): r
Inserisci il nome della persona che sta parlando: Lorenzo
Registro...
Registrazione effettuata
Preparing trainingSet and testSet...
Fatto
Allenamento in corso...
Iteration 1, loss = 16.02086969
Iteration 2, loss = 14.68610242
Iteration 3, loss = 7.01091977
Iteration 4, loss = 7.01103383
Iteration 5, loss = 7.01115079
Iteration 6, loss = 7.67868296
Training loss did not improve more than tol=0.000100 for two consecutive epochs. Stopping.
Fatto
Vuoi terminare?: (pronuncia si o no):
Registro...
Registrazione effettuata
si
MacBook-Pro-3:ML3 lorenzo$ █
```

Valutazioni Test

Il nostro dataset contiene 75 registrazioni.

Learning rate	# di layers e nodi	Accuratezza della predizione	TrainingSet	TestSet
0.001	1 layer con # 1000 nodi	100%	90% del dataset.csv	10% del dataset rimanente
0.001	3 layers: <ul style="list-style-type: none">1. 1000 nodi per il 1° layer;2. 1000 nodi per il 2° layer;3. 1000 nodi per il 3° layer;	71.43%	90% del dataset.csv	10% del dataset rimanente
0.001	1 layer con # 2000 nodi	28.57%	90% del dataset.csv	10% del dataset rimanente

Il DataSet, TrainingSet e TestSet utilizzati per le valutazioni dei test sono stati allegati al progetto.