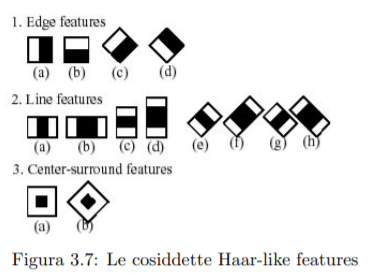
Approfondimenti

1. Haar & cascade classifiers & immagine integrale: L’obiettivo è quello di ottenere il cosiddetto “classificatore a cascata per volti frontali” da poter utilizzare per localizzare volti all’interno di immagini. L’algoritmo è in grado di rilevare volti in posa approssimativamente frontale, ovunque essi siano presenti nell’immagine e l’output `e costituito da una serie di regioni rettangolari, ognuna centrata su un volto e i cui limiti racchiudono il volto stesso. Tale metodo basato su una tecnica di apprendimento statistico supervisionato chiamata Adaboost. Il sistema classifica le patch dell’immagine come volto o “non-volto” in base alla risposta a dei semplici template rettangolari; la feature così ottenuta prende il nome di Haar-like feature



Per ogni template, il corrispondente valore della feature dato dalla somma dell’intensità dei pixel pb(i) collocati all’interno dell’area in nero,

meno la somma dell’intensità dei pixel pw(i) nell’area bianca:



Le features caratterizzano così il contrasto locale delle zone salienti dell’immagine. L’utilizzo di features dei valori di intensità dei singoli pixel dell’immagine, `e giustificato principalmente da due motivi:

1. Le features sono in grado di codificare la conoscenza di un particolare dominio; un vasto e generico insieme di Haar-like freature, unitamente ad un processo di selezione delle features, può aumentare la capacità di apprendimento dell’algoritmo.

2. I sistemi di features operano più velocemente; uno dei principali punti di forza del metodo `e proprio dato dal fatto che la localizzazione del volto viene eseguita in tempo reale, facendo scorrere sull’immagine una finestra di ricerca, di dimensione fissa a varie scale.

Le feature rettangolari possono essere calcolate molto rapidamente adottando una rappresentazione intermedia dell’immagine detta integral image. Il pixel p(x, y) dell’integral image, indicata con ii(x, y), è dato dalla somma di tutti i pixel dell’immagine I(x, y) che stanno a sinistra e sopra (x, y):



La integral image può essere calcolata in un solo passo attraverso l’espressione ricorsiva:

s(x, y) = s(x, y − 1) + i(x, y)

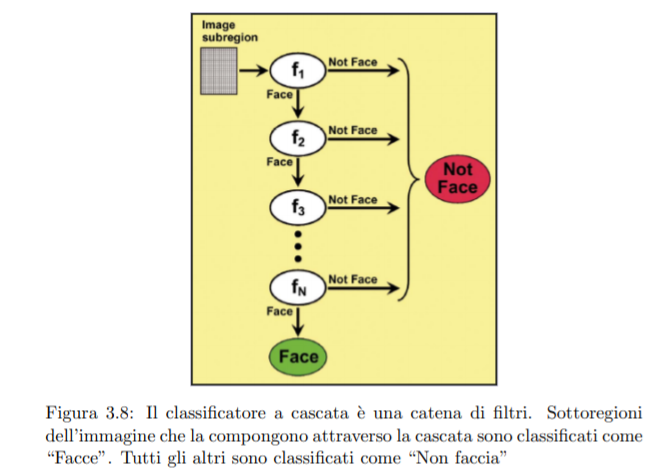
ii(x, y) = ii(x − 1, y) + s(x, y)

s(x, −1) = 0, ii(−1, y) = 0

Il passo successivo consiste nel processo di apprendimento del classificatore. Dato un insieme di feature ed un training set costituito da immagini positive e negative, un qualsiasi metodo di apprendimento può essere usato per addestrare una funzione di classificazione; il sistema adotta una variante di Adaboost. Si noti che per rilevare facce indipendentemente dalla posa, `e necessario addestrare due versioni del feature detector, una per i volti frontali (o quasi frontali) e una per quelli di profilo. L’insieme di feature ottenuto col procedimento appena descritto ed estremamente ampio; se ad esempio la risoluzione del detector è di 24 × 24 pixel, l’insieme di Haar-like feature è pari ad oltre 180.000 elementi. Il metodo di apprendimento, chiamato weaklearning, `e in grado di selezionare da tale insieme un set limitato di feature combinandolo con un opportuno weak-classifier. Questo avviene selezionando la feature rettangolare che meglio separa gli esempi positivi da quelli negativi; per ogni feature, il weak-learner determina la funzione di classificazione a soglia ottimale. Un weak-classifier hj (x) associato alla feature x, consiste in una feature fj , una soglia Θj e una polarità pj:

hj (x) = 1 sse pjfj (x) < pjΘj

L’ultimo passaggio consiste nella ricostruzione di una cascata di weakclassifier, ovvero un classificatore a stadi che sia in grado di respingere molte sotto-finestre negative e, contemporaneamente, di rilevare la quasi totalità delle istanze positive. La cascata viene realizzata posizionando nelle prime posizioni i classificatori più semplici, in modo da eliminare rapidamente buona parte delle finestre negative; in un secondo tempo entrano in gioco i classificatori più complessi in grado di scartare i rimanenti falsi-positivi



Vantaggi Haar Cascades:

• è l’algoritmo più usato per la face detection in real time;

• estremamente accurato grazie alla presenza di classificatori;

• bassa possibilità di falsi positivi;

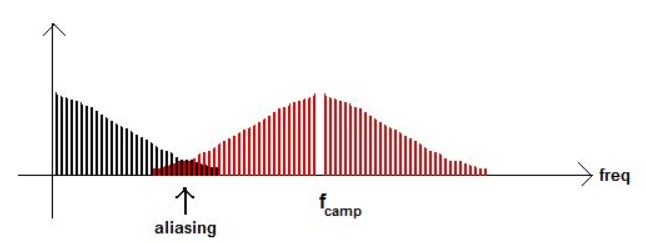
Svantaggi Haar Cascade:

• occorre molto tempo per il training

1. Frequenza di campionamento (sampling rate): La frequenza di campionamento è la misura espressa in [hertz](https://it.wikipedia.org/wiki/Hertz) del numero di volte al secondo in cui un [segnale](https://it.wikipedia.org/wiki/Segnale_(fisica)) [analogico](https://it.wikipedia.org/wiki/Analogico) viene misurato e memorizzato in forma [digitale](https://it.wikipedia.org/wiki/Digitale_(informatica)) (da "digit" che in inglese significa "cifra"; in questo caso, nel mondo informatico e elettronico, il significato varia in "cifra binaria").

In altre parole la frequenza di campionamento è il parametro che si utilizza quando si "traduce" un fenomeno naturale - comprensibile per l'essere umano - in una rappresentazione numerica - "comprensibile" cioè utilizzabile per un computer e per quelle macchine il cui funzionamento è basato sul [bit](https://it.wikipedia.org/wiki/Bit_(informatica)).

1. Aliasing: il campionamento di un segnale produce un segnale campionato il cui spettro è composto da diverse repliche del segnale originale, repliche collocate a frequenze multiple della frequenza di campionamento. Se il segnale da campionare non avesse una banda limitata, le frequenze più alte delle diverse repliche dello spettro si sovrapporrebbero, interferendo tra loro. In questi casi la ricostruzione corretta del segnale a partire dai campioni non sarebbe più possibile. Tale fenomeno prende il nome di aliasing



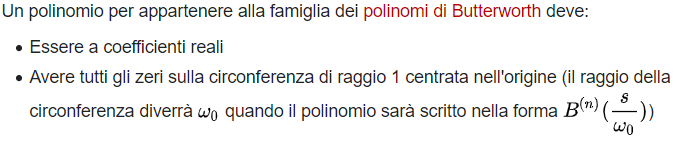
Il fenomeno dell'aliasing si verifica sempre quando si campiona un segnale con banda non limitata. Per evitare l'aliasing, la soluzione è limitare la banda del segnale usando un filtro passa basso che elimini le frequenze più elevate. Tale filtro viene detto appunto anti-aliasing.

Facciamo un esempio. Supponiamo di considerare un'onda con frequenza 50 Hz. Supponendo di ritenere significative solo le prime 100 armoniche, possiamo considerare come banda del segnale quella compresa fra 0 e 5 kHz. Di conseguenza, come già visto, la frequenza di campionamento va scelta a 10 kHz o a un valore superiore (Teorema di Nyquist-Shannon).

Tuttavia, campionando a 10 kHz il segnale originale, che contiene frequenze superiori ai 5 kHz, si produrrebbe comunque l'aliasing. In altre parole, le frequenze ritenute trascurabili sono comunque campionate e interferiscono con le frequenze "utili" del segnale.

La soluzione a questo problema è filtrare il segnale prima di campionarlo con un filtro passa-basso con frequenza di taglio 5 kHz. In questo modo si rende il segnale a banda limitata, eliminando effettivamente quelle armoniche che, se campionate, andrebbero a interferire con una corretta ricostruzione del segnale stesso.

1. Filtro butterworth (frequenza di nyquist): il filtro buttherworth è uno dei più semplici filtri elettronici Il suo scopo è mantenere il più piatto possibile il modulo della risposta in frequenza nella banda. Sono filtri buttherworth tutti i filtri la cui funzione di trasferimento ha come denominatore e numeratore dei polinomi di butterworth.



In applicazioni che usano filtri per riformare la frequenza dello spettro di un segnale, la forma o grandezza del roll-off può anche essere chiamata “transition band” e per un semplice filtro di primo ordine potrebbe essere troppo lungo o grande. Così, sono stati creati filtri con più di un ordine. Questi tipi di filtri sono comunque chiamati attraverso un numero ordinale.

Generalmente: la complessità del filtro è definita dal suo ordine e dipende dal numero di componenti reattivi come induttori presenti nel design del filtro stesso. Inoltre, sappiamo che il tasso di roll-off e, di conseguenza, la larghezza della banda di transizione, dipende dal numero del filtro e che per un semplice filtro di primo ordine il tasso sarà di 20db/decade o 6db/ottava.

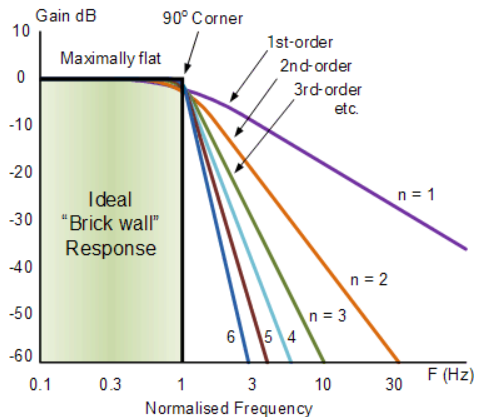
Poi per un filtro di ordine n, avremo 20ndb/decade o 6ndb/ottava

I filtri di ordine alto sono formati mettendo insieme in cascata filtri si primo o secondo ordine

La decade è un incremento o decremento pari a 10 in un segnale

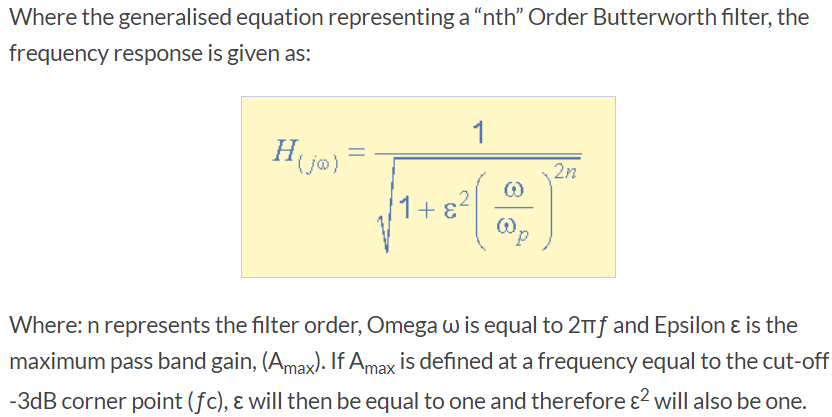
L’ottava sta nel raddoppiare o dimezzare la scala della frequenza

In ogni caso, nel dominio delle frequenze usiamo scale logaritmiche per denotare un valore di frequenza.

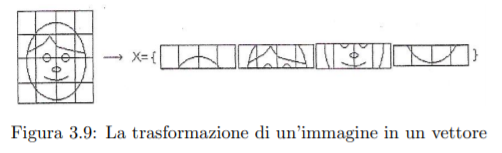
Il filtro è anche detto “maximally flat” perché il pass band è creato per avere una frequenza di risposta che è più piatta possibile da 0db a -3db senza increspature. Le frequenze più alte oltre il punto di cuf off vengono portate a 0.

Uno svantaggio del filtro buttherworth è che usa la grandezza del pass band con il costo di un intero transition band quando il filtro cambia da pass band a stop band.

Più alto è l’ordine, più ripida è la cascata che si genera nel filter design e il filtro più vicino diventa simile ad un muro di mattoni. In poche parole, la frequenza ideale non è ottenibile.



1. Il volto visto come un vettore: L’immagine di un volto sostanzialmente può essere trasformata in un vettore. Se chiamiamo w la larghezza dell’immagine e h la sua altezza, entrambe in pixel, il numero delle componenti del vettore che si vuole ottenere `e dal prodotto w ∗ h, dove ogni pixel dell’immagine iniziale corrisponde ad una componente del vettore. La costruzione di tale vettore quindi può essere effettuata tramite una semplice concatenazione delle righe della matrice dell’immagine iniziale



Il vettore descritto appartiene ad uno spazio vettoriale. Tale spazio è

chiamato “spazio delle immagini” ed `e lo spazio di tutte le immagini

la cui dimensione è di w ∗ h pixels. Tutti i volti presenti in ogni immagine si “assomigliano” fra loro; infatti ognuno `e caratterizzato dalla presenza di due occhi, una bocca, un naso ecc, ovvero tutte caratteristiche localizzate nella stessa area.

Ciò significa che tutti i punti rappresentanti i volti non si spargono

in maniera omogenea nello spazio, bensì tendono a localizzarsi in un

ristretto “cluster” nello spazio immagine. Lo spazio delle immagini `e uno spazio di dimensione w∗h. Ovviamente non tutti i pixels che compongono l’immagine di un volto possono essere considerati “rilevanti” ed in particolare ognuno di questi dipende strettamente dal suo immediato vicino. Tale considerazione ci permette di capire che la dimensione dello spazio dei volti sarà sicuramente diversa rispetto allo spazio delle immagini iniziali, ed in particolare la dimensione di tale spazio sarà sicuramente di una dimensione minore

La frequenza di nyquist è la frequenza di campionamento/2 ed è il numero che rappresenta la minima frequenza recepepibile

1. Normalizzazione L2: serve a normalizzare i valori.
2. Deep convolutional neural network: Le Convolutional Neural Networks (CNN) sono una famiglia di reti neurali largamente impiegate nell’ambito di computer vision e più generalmente con dati che hanno relazioni spaziali.

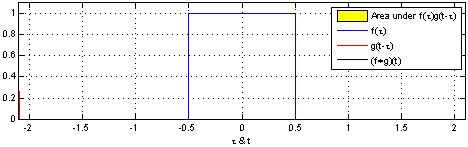
Se dovessi rappresentarle graficamente, quello che ne verrebbe fuori è un grafo con una serie di passi dove, partendo da un input complesso, vengono estratte man mano informazioni, chiamate features, sempre più semplici e facilmente riconoscibili dal computer.

Le CNN seguono quindi un'architettura a livelli, tipicamente non ciclica. I livelli più importanti sono i layer di convoluzione, da cui prende il nome.

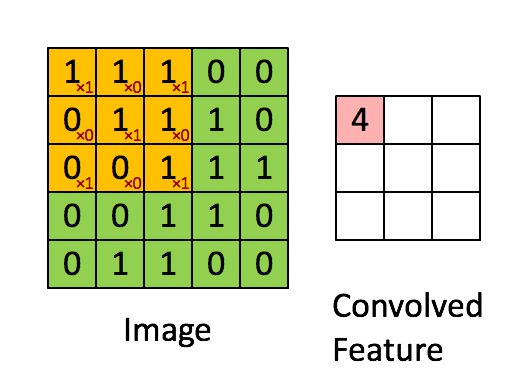
Layer comuni

Convoluzione

La convoluzione è un’operazione matematica che ricorda il funzionamento di uno scanner. Il risultato è funzione della traslazione di una funzione sull’altra, al variare del tempo



Poiché tipicamente le CNN ricevono in input dei dati bidimensionali, come ad esempio le immagini, avremo una o più matrici in input ed una chiamata kernel che verrà traslata al fine di calcolarne il risultato, chiamato feature map. Il passo di traslazione prende il nome di stride.

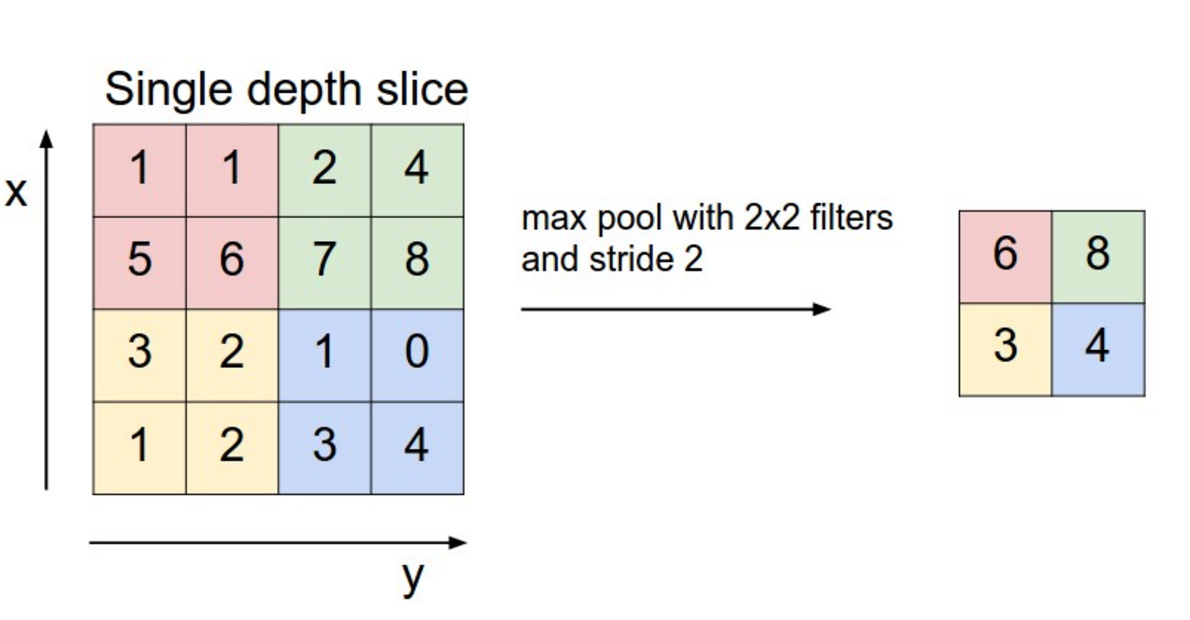


ReLu

L’unità lineare rettificata (ReLu) consente un addestramento più rapido ed efficace mappando i valori negati a zero e mantenendo valori positivi. talvolta si parla di attivazione, perché solo le feature attivate vengono fatte passare al layer successivo.

Sottocampionamento

A seguito di un layer di convoluzione è comune trovare un layer di sottocampionamento che riduca le dimensioni delle matrici in output. Le operazioni più comuni sono:



* MaxPooling: ogni cella della matrice risultante contiene il valore massimo della sottomatrice di partenza.
* AvarangePooling: il valore di ciascuna cella è la media dei valori della sottomatrice di partenza.

Full connected

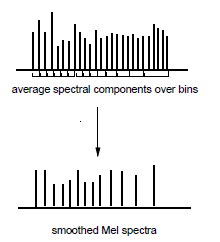
I layer full connected possono essere visti come dei vettori di neuroni. L’output di ciascun neurone è inviato a tutti i neuroni del layer successivo, con diversi valori di peso.

Cosa è Keras

Keras è un framework ad alto livello che lavora on-top di engine quali TensorFlow offrendo un livello di astrazione maggiore, semplificando la creazione dei nostri modelli di rete neurale, il training e la valutazione finale.

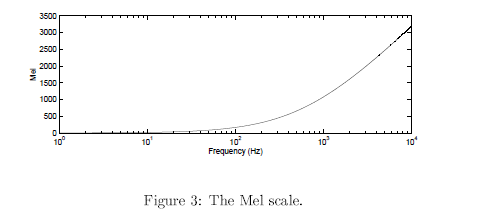
1. Triplet loss: basta quello che sta sulla doc
2. Rete Siamese: basta quello che sta sulla doc
3. MFCC: MFCC sono stati la feature dominante usata per lo speech recognition per molto tempo. Il loro successo è dovuto alla loro abilità di rappresentare lo spettro di amplitude del discorso in una forma compatta. Ogni step nel processo di creare features MFCC è motivato da considerazioni computazionali. Passi di creazione di un MFCC:

* Il primo passo è di dividere il segnale del discorso in frames, di solito applicando una funzione di windowing a intervalli fissi. L’obiettivo qui è di modellare piccole sezioni (intorno ai 20 ms teoricamente) che siano statisticamente stazionarie. La funzione window, tipicamente una finestra di Hamming, rimuove gli effetti di bordo (edge effects). Generiamo un vettore di feature cepstral per ogni frame
* Il prossimo step sta nel compiere la DFT per ogni frame.
* Lo step successivo si occupa di attuare uno smooth dello spettro per enfatizzare le frequenze percepite con del significato: il vero e proprio parlato. Questo è raggiunto raccogliendo 256 componenti spectral in 40 frequenze binarie come mostrato in figura



* Sebbene uno si aspetterebbe che questi binari siano equamente distribuiti in frequenza, è emerso come per il discorso, le frequenze minori siano percettivamente più importanti delle alte frequenza. Perciò, lo spacing dei bin segue la scala di frequenza chiamata “Mel”

La scala Mel è basata sul mapping tra la frequenza attuale e l’intonazione percepita poiché apparentemente il sistema uditivo umano non percepisce l’intonazione in maniera lineare. Il mapping è approssimativamente lineare attorno a 1kHz e logaritmicamente oltre.



* Poi conserviamo solamente il logaritmo dello spettro di amplitude perché la rumorosità percepita del segnale è approssimativamente su scala logaritmica
* I componenti dei vettori spectral-Mel calcolati per ogni frame sono strettamente correlati. Le feature dello speech sono tipicamente modellate da densità di tipo GMM. Inoltre, per ridurre il numero di parametri nel sistema, l’ultimo step sta nell’applicare una trasformata ai vettori mel-spectral con lo scopo di de-correlare i loro componenti. Il metodo più usato è la Trasformata Discreta Coseno. Usando questa trasformata, abbiamo circa 133 features ottenute per ogni frame

Delta: il feature vector MFCC descrive solo l’involucro di potere spettrale di un singolo frame, ma sembra che il discorso abbia anche informazioni di tipo dinamico come la traiettoria degli mfcc nel tempo. Esse si mostrano calcolando la derivata e ponendole nell’vettore delle feature originale.

La forma del tratto vocale si manifesta nell’involucro dello spettro di potere a short-time, e il lavoro degli MFCC è di rappresentare accuratamente questo involucro

La scala Mel collega frequenze percepite di un tono puro alla sua frequenza attuale misurata. Gli uomini sono bravu nel discernere piccoli cambiamenti nell’intonazione a basse frequenze che alle alte frequenze. Incorporando questa scala rendiamo il match delle nostre feature molto vicino a ciò che gli umani sentono.

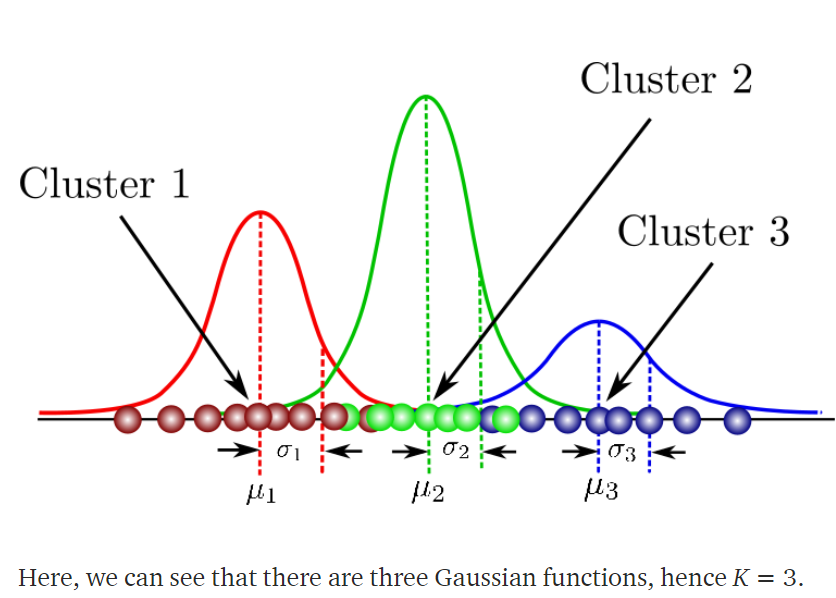
Sito 2: Un segnale audio è costantemente in cambiamento, così per semplificare le cose assumiano che su una scala di tempo piccola il segnale non cambio molto. Questo perché dividiamo il segnale in finestre di 20/40ms. Se il frame fosse molto più corto noi non abbiamo abbastanza campioni per ottenere una stima dello spettro affidabile.

Lo step successimo sta nel calcolare il potere dello spettro di ogni frame. Questo è motivato da dalla coclea (un organo dell’orecchio) che vibra in base alle frequenza che riceviamo. La nostra stima del periodogramma attua un lavoro simile per noi, identificando quelle frequenze che sono presenti nel frame. La stima del periodogramma contiene un sacco di informazioni inutili. In particolare, la coclea non divide la differenza tra due frequenza spaziate vicine. L’effetto diventa più pronunciato all’aumentare delle frequenze. Per questo prendiamo solo pezzi dei bin del periodogramma e li sommiamo per avere un idea di quanta energia ci sia nelle regioni. Questo è possibile grazie al mel filterbank: componente che ci da indicazioni su quanta energia esista adattandosi dinamicamente.

Una volta che abbiamo filtrato le energie, facciamo il logaritmo. Questo è motivato dall’orecchio umano: noi non ascoltiamo la rumorosità su una scala lineare. Generalmente il doppio è in realtà la moltiplicazione per 8 (ottave spiegate nel filtro). Ma perché un log e non un cubo o un quadrato? Il log ci permette di usare la mean cepstral subtraction che è una tecnica di normalizzazione del canale

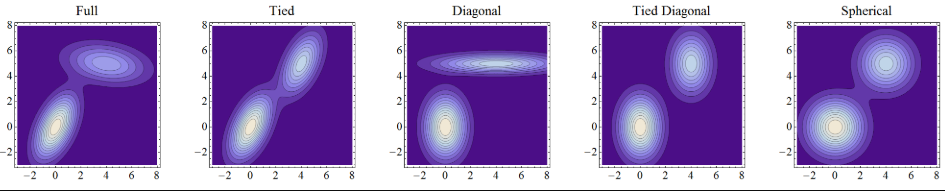
Lo step finale è nel calcolare la DCT. Viene fatto per due ragioni: i nostri filterbanks sono sovrapposti e le energie dei filterbank sono correlate tra loro. La DCT decorrela le energia che implicano matrici di covarianza diagonale che sono molto usate negli Hidden Markov Models.

1. Gaussian Mixture Model: una GMM è una funzione che è compresa di gaussiane, ognuna identificata da k che è il numero di clusters del nostro data set. Ogni gaussiana ha una media, una covarianza e una probabilità mista



Ricordiamo che le probabilità miste sono probabilità e quindi la loro somma deve dare 1. Dobbiamo stare attenti affinchè ogni gaussiana faccia rientrare in sé stessa tutti i data point di un determinato cluster. Questo è precisamente ciò che fa la massima somiglianza

Una distribuzione gaussiana è determinata dalla sua matrice di covarianza che determina le direzioni e la lunghezza degli assi dei contorni di densità che hanno forma di ellissi. Nel codice noi potevamo scegliere tra una covarianza full, tied, diagonal e spherical. Sono tutti tipi di modelli spiegabili nel caso bidimensionale



Full: le componenti possono adottare indipendentemente ogni posizione e forma

Tied: le componenti hanno la stessa forma ma la forma di base non è fissata

Diagonal: gli assi di contorno sono orientati sulle assi di coordinate, ma comunque le eccentricità potrebbero variare tra le componenti

Spherical: una situazione diagonal con contorni circolari

1. Expectation Maximization: Ci sono molte tecniche per stimare i parametri di un Gaussian Mixture Model, una stima per massima somiglianza è il più comune. Nel caso in cui il nostro dataset sia compreso tra molti punti generati da diversi processi. Questi punti hanno una distribuzione di probabilità gaussiana ma il dato è combinato e le distribuzioni sono abbastanza simili che non è ovvio ci sia corrispondenza tra le distribuzioni in un determinato punto. I processi sono soliti generare un data point che rappresenti una variabile latente. Questo influenza il dato ma non è osservabile. Per questo l’algoritmo EM è un approccio appropriato da usare per stimare i parametri della distribuzione. Nell’algoritmo EM, lo step E stimerebbe il valore per la variabile latente per ogni data point e lo step M ottimizzerebbe i parametri della distribuzione di probabilità per ottenere la migliore densità dei dati. Il processo è ripetuto finché un buon set di valori latenti e una massima somiglianza sono ottenuti.

Un comune problema di modelling riguarda come stimare la distribuzione di probabilità di un dataset. Ci sono molte tecniche per risolvere questo problema e uno dei più comuni è quello di massima somiglianza che riguarda il trattare il problema come un ottimizzazione di un problema già cercato dove cerchiamo un set di parametri che risulti il best fit per i nostri dati campione.

Una limitazione della massima somiglianza è che assume che il dataset sia completo o completamente osservato. Questo non significa che il modello ha accesso a tutti i dati; invece, assume che tutte le variabili che siano rilevanti per il problema siano presenti. Ma questa cosa non è sempre vera. Generalmente ci riferiamo a queste variabili nascoste o non osservate come “latenti” ed in presenza di queste la stima per massima somiglianza convenzionale non lavora bene. L’expectation maximization cerca i parametri di modello appropriati in presenza di variabili latenti, quindi risolve questo problema.

L’EM nasce come approccio iterativo che cicla su due passaggi. Il primo tenta di stimare variabili lantenti. Il secondo tenta di ottimizzare i parametri del modello per spiegare al meglio i dati. Questo algoritmo è applicato largamente ed è molto usato in machine learning per problemi di apprendimento non supervisionato come stima o clustering.

1. Normalizzazione minmax e score standard: la minmax è 0 al più piccolo, 1 al più grande e il resto in proporzione. Lo score standard fa in modo di avere media 0 e varianza 1