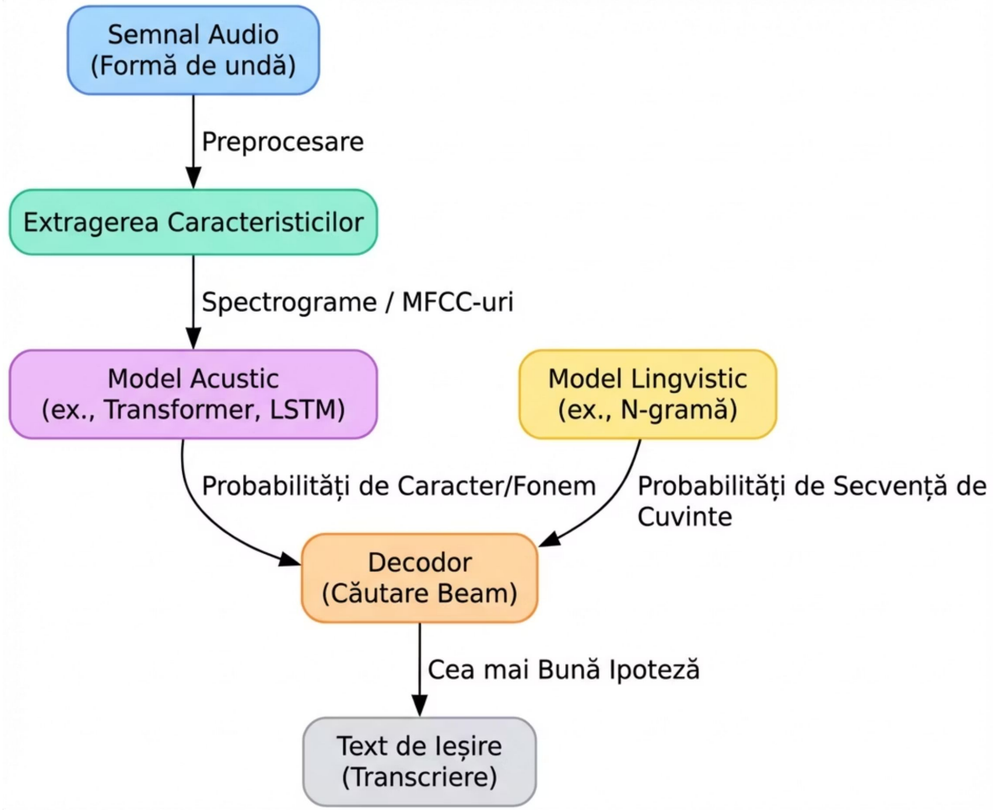
Introducere în Sistemele de Recunoaștere Automată a Vorbirii

ASR (Recunoașterea Automată a Vorbirii) este o tehnologie care transformă vorbirea umană în text. E folosită zilnic în asistenți vocali, dispozitive inteligente și aplicații de transcriere. Un sistem ASR convertește semnalul audio brut într-o transcriere cât mai exactă, printr-un „pipeline” de procesare format din mai multe componente specializate, iar înțelegerea acestui flux e importantă pentru a proiecta și optimiza sistemul.



Extragerea caracteristicilor

Procesul incepe cu o formă de undă audio brută, un semnal continuu reprezentat matematic prin funcţia *x(t)*. Acest semnal este digitalizat şi transformat într-o secvenţă discretă de valori, *x[n]*.

Cu toate acestea, secvența numerică brută nu reprezintă o intrare eficientă pentru un model de învățare automată (*Machine Learning*).

Transformarea semnalului

Componenta de extragere a caracteristicilor transformă acest semnal audio brut într-o reprezentare mai compactă şi mai bogată în informaţii. Această etapă implică tehnici care generează vectori de trăsături, precum Coeficienţii Cepstrali în Scala Mel (MFCC) sau, mult mai frecvent în sistemele moderne, Spectrogramele Log-Mel.

Beneficii

- Reducerea dimensionalităţii datelor audio
- Eliminarea informaţiilor redundante
- Evidenţierea proprietăţilor fonetice
- Facilitarea procesării ulterioare

> Aceste caracteristici evidenţiază proprietăţile fonetice ale semnalului vocal, facilitând astfel sarcinile de procesare ulterioare ale modelului.

Modelul acustic

Modelul Acustic reprezintă nucleul sistemului ASR. Sarcina sa este de a pune în corespondență caracteristicile audio extrase cu unitățile fundamentale ale vorbirii, cum ar fi fonemele (cele mai mici unități sonore, de exemplu /c/ în „casă") sau, în mod direct, cu caracterele alfabetului.

Intrare

Caracteristici audio extrase (MFCC, Log-Mel)

Procesare

Rețele neurale profunde (RNN, LSTM, Transformer)

Ieșire

Distribuție de probabilitate peste caractere/foneme

Arhitecturi moderne

Modelele acustice moderne folosesc rețele neuronale profunde (*Deep Neural Networks*), cum ar fi Rețelele Neuronale Recurente (RNN), rețelele de tip LSTM sau arhitecturile Transformer. Acestea sunt antrenate pe mii de ore de conținut audio transcris pentru a asimila relația complexă dintre caracteristicile acustice și unitățile lingvistice corespunzătoare.

Modelul lingvistic

Deși modelul acustic poate prezice secvențe de sunete, acesta nu deține cunoștințe legate de gramatică sau de probabilitatea de succesiune a cuvintelor. De exemplu, modelul ar putea considera sintagmele englezești „wreck a nice beach" (a distruge o plajă frumoasă) și „recognize speech" (a recunoaște vorbirea) ca fiind aproape identice din punct de vedere acustic. Aici intervine un model lingvistic.

Funcții

- Modelul de limbaj operează la nivel textual, evaluând probabilitatea de apariție a unei anumite secvențe de cuvinte. Acesta asistă sistemul în alegerea celei mai probabile propoziții dintr-un set de candidați cu similarități acustice.
- Prin integrarea unui model de limbaj, sistemul poate corecta erorile, generând transcrieri care sunt nu doar plauzibile acustic, ci și coerente din perspectivă lingvistică.

Coerență lingvistică

Asigură că transcrierea respectă regulile gramaticale și contextul semantic

Corectare automată

Identifică și corectează erorile bazate pe probabilități lingvistice

Dezambiguizare

Distinge între secvențe acustic similare folosind context

Decodorul

Decodorul este elementul decizional final. Acesta preia ieșirile probabilistice furnizate de modelul acustic și le combină cu scorurile generate de modelul de limbaj pentru a determina cea mai probabilă secvență de cuvinte.

Strategii de decodare

Căutare lacomă (Greedy)

- Simplă și rapidă
- Rezultate suboptime
- Nu explorează alternative

Beam Search (recomandat)

- Explorare multiplă
- Optimizare globală
- Acuratețe superioară

Proprietățile vorbirii umane

Pentru un sistem ASR, o propoziție rostită este, la nivel primar, o formă de undă audio complexă, un flux continuu de date. Procesul de transcriere în text necesită însă identificarea tiparelor lingvistice din cadrul acestui semnal.

Semnal audio brut

Vorba este inițial o formă de undă complexă și nestructurată, o serie de vibrații ale aerului.

Structura lingvistică

Fonetica și fonologia oferă cadrul esențial pentru a înțelege și extrage componentele fundamentale ale vorbirii umane, cum ar fi ***fonemele*** și ***alofonele***.

Acest cadru teoretic permite ca analiza să se concentreze pe unitățile discrete ale vorbirii, transformând fluxul continuu într-o secvență interpretabilă de sunete. Această abordare este crucială pentru a permite sistemelor ASR să recunoască și să proceseze vorbirea eficient.

Fonemele

Cea mai mică unitate sonoră distinctivă dintr-o limbă, al cărei rol este de a diferenția sensul cuvintelor, se numește **fonem**. Fonemele sunt, în esență, elementele atomice fundamentale ale limbajului vorbit. Schimbarea unui singur fonem într-un cuvânt poate altera complet sensul acestuia.

> De exemplu, în limba engleză, cuvintele „pat” (a mângâia), „bat” (liliac/bâtă) și „cat” (pisică) se disting exclusiv prin sunetul lor inițial. Sunetele reprezentate fonetic prin `/p/`, `/b/` și `/k/` sunt foneme distincte, deoarece substituirea unuia cu altul creează un cuvânt cu sens nou. Notația cu bare oblice, cum ar fi `/p/`, indică o unitate sonoră abstractă, diferită de literele alfabetului (***grafemele***).

Fiecare limbă posedă un inventar finit de foneme. Engleza americană, de pildă, utilizează aproximativ 44 de foneme, incluzând consoane, vocale scurte, vocale lungi și diftongi (sunete vocalice care se transformă dintr-o vocală în alta, precum `/aɪ/` din „buy”).

Alofonele

Deși fonemele reprezintă unitățile abstracte, sunetele pe care le articulăm în mod concret se numesc **fonuri** (sau sunete propriu-zise). Un singur fonem poate fi pronunțat fizic în moduri ușor diferite, în funcție de contextul său fonetic din interiorul unui cuvânt. Aceste variații previzibile ale unui singur fonem poartă denumirea de **alofone**.

Un exemplu clasic în limba engleză este fonemul `/p/`:

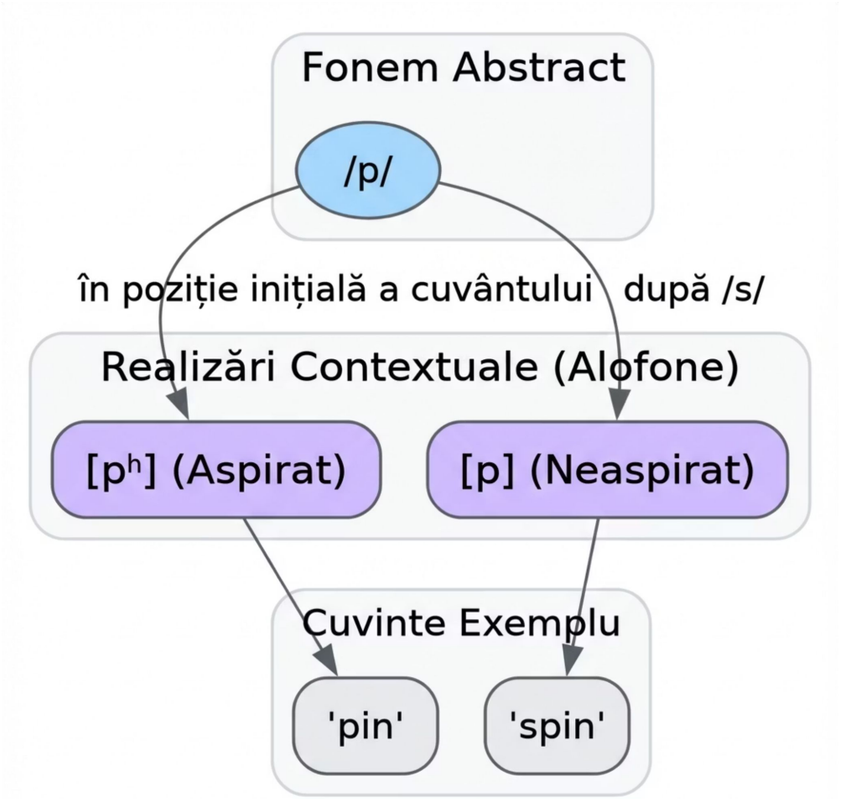
Aspirat [pʰ] în „pin”

În cuvântul „**pin**” (ac), sunetul `/p/` este aspirat, adică este urmat de o ușoară emisie de aer. Notația fonetică pentru acest alofon este `[pʰ]`.

Neaspirat [p] în „spin”

În cuvântul „**spin**” (a se roti), sunetul `/p/` este neaspirat, fără emisia de aer. Notația pentru acest alofon este `[p]`.

Pentru un vorbitor nativ de limba engleză, [p^h] și [p] par a fi același sunet „p”, deoarece creierul uman le grupează automat în cadrul aceluiași fonem, /p/. Cu toate acestea, pentru un sistem de calcul care analizează o formă de undă, aceste două sunete sunt distincte din punct de vedere acustic, cu o diferență măsurabilă în semnalul audio.



> Această relație ilustrează o provocare fundamentală în recunoașterea vorbirii: sistemul ASR trebuie să învețe că alofone acustic distincte (e.g., [p^h] și [p]) sunt asociate aceluiași fonem abstract /p/ și, în cele din urmă, aceleiași litere „p” în transcrierea finală.

Coarticularea

Variația sunetelor vorbirii este complicată suplimentar de fenomenul de **coarticulare**. Acesta descrie modul în care pronunția unui sunet este influențată de sunetele învecinate. Vorbirea umană nu este o secvență de fonuri discrete, ci un flux continuu unde sunetele se întrepătrund și se contopesc.

Un exemplu elocvent este poziția limbii la pronunțarea sunetului `/n/` în cuvintele englezești „`ten`” și „`tenth`”:

În „`ten`” (zece)

Limba atinge creasta alveolară, tipic pentru `/n/`.

În „`tenth`” (al zecelea)

Limba se deplasează anticipativ, modificând `/n/` înaintea `/θ/`.

Această mișcare anticipativă modifică proprietățile acustice ale sunetului `/n/`. Coarticularea implică faptul că același fonem poate avea o semnătură acustică diferită la aproape fiecare rostire, ceea ce reprezintă o provocare semnificativă pentru sistemele ASR.

Relevanța pentru sistemele ASR

Înțelegerea fonemelor, a alofonelor și a coarticulării nu constituie doar un exercițiu teoretic, ci subliniază dificultatea principală în recunoașterea vorbirii: variabilitatea imensă a semnalului vocal.

Variația alofonică

Pronunții fizice distincte ale aceluiași fonem, fără a schimba sensul.

Efectele de coarticulare

Influența sunetelor adiacente asupra pronunției, creând un flux continuu.

Diferențe dintre vorbitori

Variații în ton, accent, ritm și particularitățile individuale de articulare.

Modelele moderne de învățare profundă (Deep Learning) excelează în gestionarea acestei variabilități. Prin antrenarea pe seturi masive de date, ele învață să asocieze diverse tipare acustice cu unitățile lingvistice corecte, fie că sunt foneme sau, mai frecvent în sistemele end-to-end, caractere ori cuvinte.

Acest lucru este posibil datorită tehnicilor avansate de extragere a caracteristicilor, pe care le vom explora în detaliu în capitolul următor, care transformă semnalul audio într-o reprezentare mult mai robustă la aceste variații.

Semnale audio digitale

Sunetele pe care le auzim, de la vorbire la muzică, există sub forma unor **unde acustice continue**, reprezentând variații ale presiunii aerului. Calculatoarele, în schimb, operează pe date numerice discrete. Pentru a realiza această tranziție fundamentală, trebuie să convertim semnalul audio analogic continuu, reprezentat matematic ca o funcție de timp $x(t)$, într-o secvență discretă de numere, $x[n]$.

Acest proces de conversie este esențial pentru întregul domeniu al procesării audio digitale și implică trei etape principale:

Eșantionarea (Sampling)

Transformarea semnalului continuu într-o serie de valori discrete la intervale regulate de timp, definind **frecvența de eșantionare**.

Cuantificarea (Quantization)

Atribuirea unei valori numerice fiecărui eșantion, folosind un set finit de niveluri discrete și definind **rezoluția în biți**.

Codarea (Encoding)

Reprezentarea eșantioanelor cuantificate în format binar pentru stocare sau transmisie, adesea utilizând algoritmi de compresie.

Eșantionarea

Prima etapă în conversia semnalului audio analogic este **eșantionarea (sampling)**, un proces prin care se măsoară amplitudinea undei audio la intervale de timp precise și regulate. Gândiți-vă la aceasta ca la o serie rapidă de „instantanee” ale sunetului

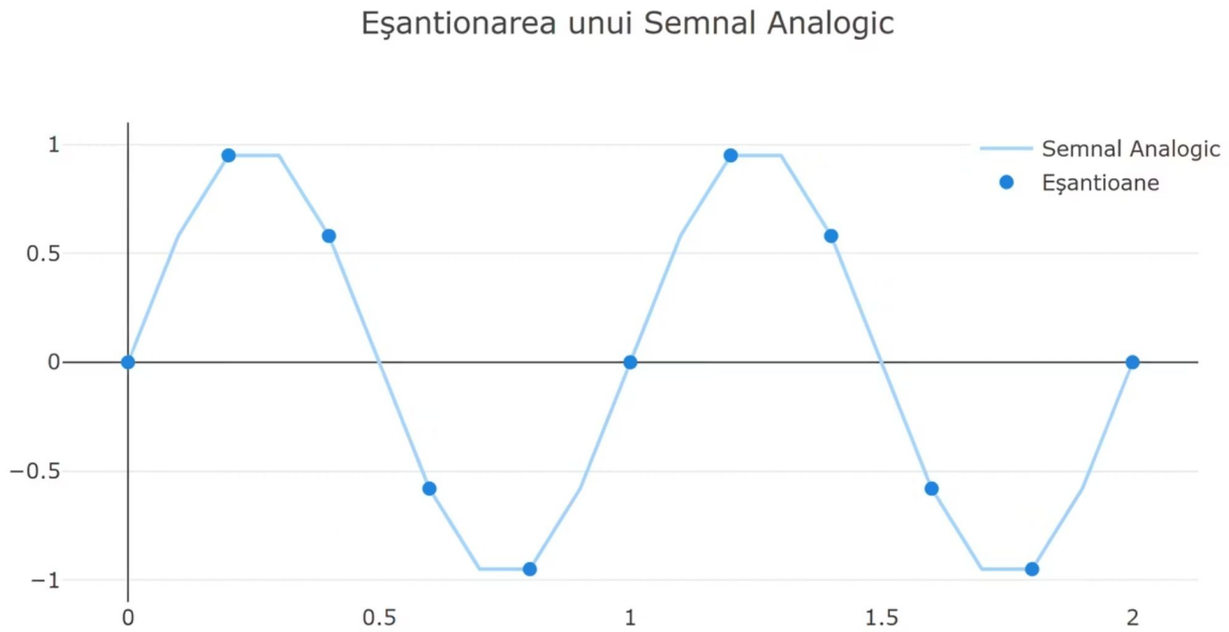
Rata la care sunt prelevate aceste eșantioane se numește **frecvență de eșantionare**, măsurată în Hertz (Hz). Pentru sistemele de recunoaștere a vorbirii (ASR), o frecvență uzuală este de **16.000 Hz (16 kHz)**, însemnând 16.000 de măsurători pe secundă.

Pentru o calitate superioară se folosesc rate de **44.100 Hz (44,1 kHz)**.

Teorema eșantionării Nyquist-Shannon

Justificarea acestor frecvențe vine de la **Teorema Nyquist-Shannon**, care afirmă că o frecvență de eșantionare trebuie să fie de cel puțin două ori mai mare decât cea mai înaltă frecvență prezentă în semnalul original pentru a-l putea reconstrui fidel. Această rată minimă este **rata Nyquist**.

Deoarece cele mai relevante frecvențe din vorbirea umană se găsesc sub 8 kHz, o frecvență de 16 kHz este considerată suficientă. Eșantionarea sub rata Nyquist duce la **aliasing** (replierea spectrului), unde frecvențele înalte sunt eronat percepute ca fiind joase, distorsionând semnalul.



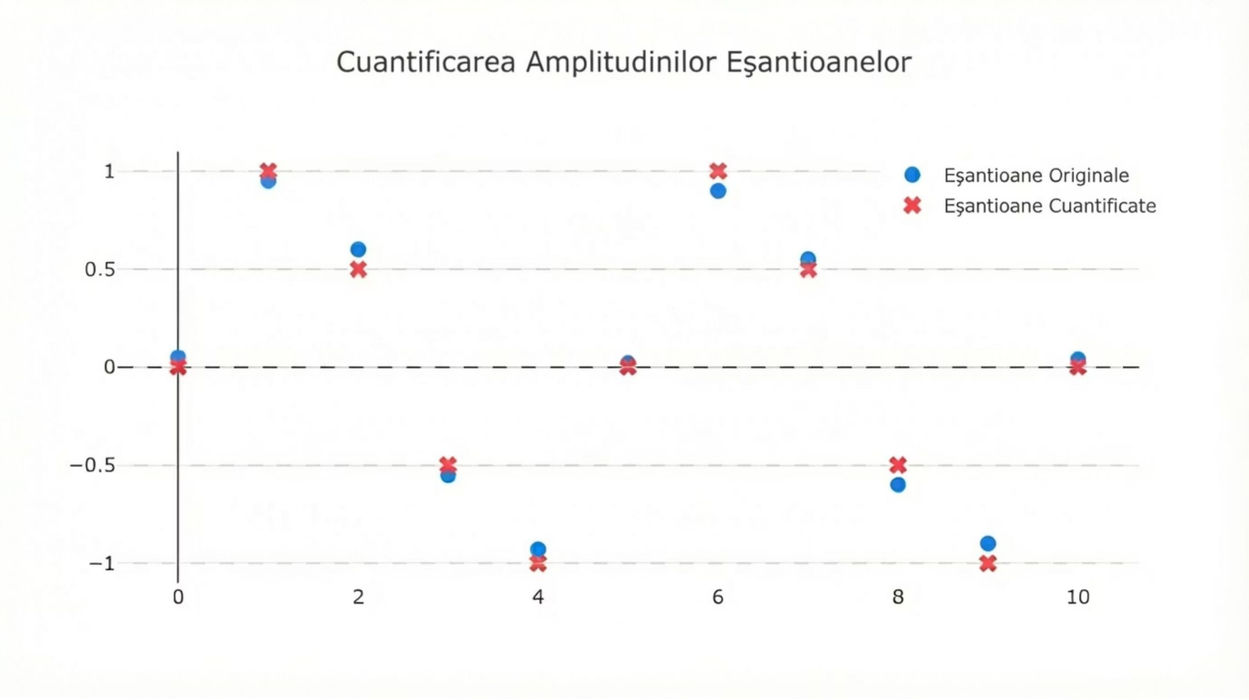
În urma eșantionării, semnalul analogic continuu se transformă într-o secvență discretă de valori. Amplitudinile acestor eșantioane sunt, în această etapă, încă numere reale, putând lua orice valoare în intervalul lor de variație.

Cuantificarea

După eșantionare, cuantificarea transformă amplitudinile continue ale eșantioanelor în valori discrete. Acest proces se bazează pe rezoluția în biți (sau adâncimea de biți), care determină numărul de niveluri disponibile. O rezoluție mai înaltă (mai mulți biți) asigură o reprezentare mai fidelă a sunetului original și reduce zgomotul de cuantificare.

Audio pe 8 biți: *2^8 = 256* niveluri discrete.
Audio pe 16 biți: *2^{16} = 65.536* niveluri discrete. Acesta este standardul pentru majoritatea aplicațiilor ASR și pentru calitatea audio de tip CD.

Cuantificarea poate fi conceptualizată prin suprapunerea une grile peste amplitudinile eșantionate și „alinierea” (sau rotunjirea) valorii fiecărui eșantion la cea mai apropiată linie grilei. O grilă mai fină (o rezoluție în biți mai mare) determină valoare rotunjită mult mai apropiată de cea originală, conservând astfel detalii de finețe.



Amplitudinile eșantioanelor originale (cercurile albastre) sunt asociate celui mai apropiat nivel discret disponibil (liniile întrerupte), rezultând valorile cuantificate (crucile roșii).

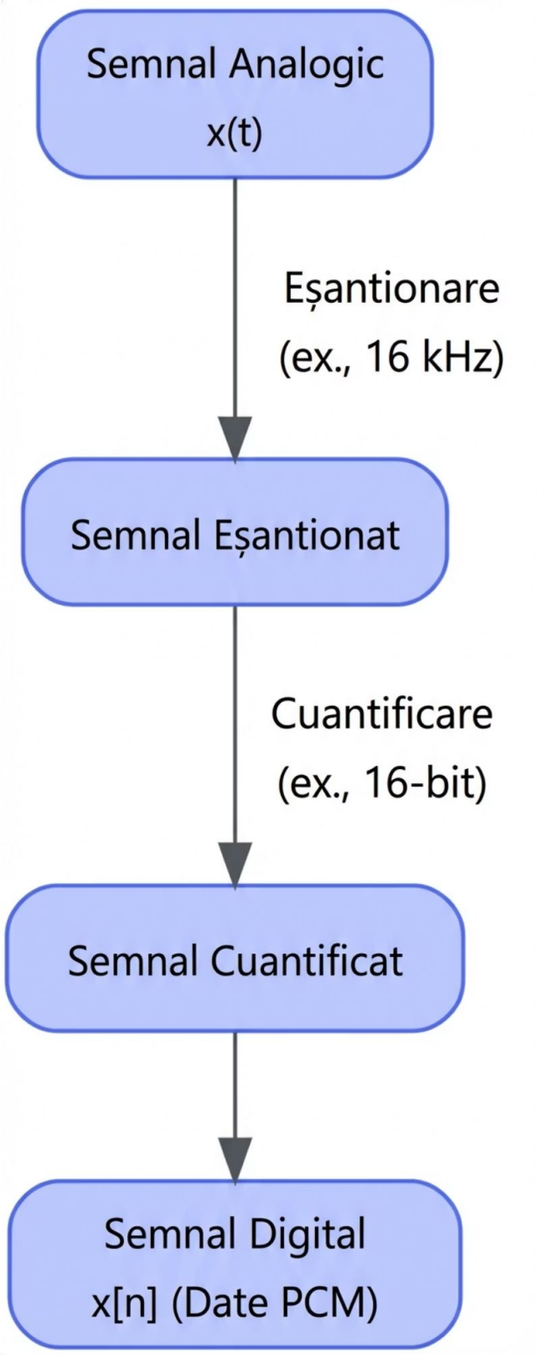
Codarea

Secvența de numere obținută prin eșantionare și cuantificare se numește **date PCM (Pulse-Code Modulation)** și reprezintă forma directă, necomprimată, a unui semnal audio digital.

Pentru stocare se folosesc formate audio, cel mai comun fiind **WAV (.wav)**, care conține un antet cu metadate (frecvență de eșantionare, rezoluție în biți, canale) urmat de datele PCM brute.

În sistemele **ASR**, sunt preferate **WAV necomprimat** sau formate **lossless** precum **FLAC**, deoarece păstrează integral informația acustică. Formatele **lossy** (ex. **MP3**) pierd detalii și pot reduce precizia, mai ales la modele de înaltă fidelitate.

Astfel, unda analogică este transformată într-o secvență numerică **x[n]**, gata pentru modele de învățare profundă. Pasul următor este extragerea de **caracteristici** care scot în evidență proprietățile fonetice relevante pentru recunoaștere.



Lucrul cu date audio în Python

Procesarea semnalelor audio necesită instrumente specializate. Datele audio brute, reprezentând o secvență numerică a amplitudinii semnalului în timp, trebuie încărcate dintr-un fișier într-un format manipulabil, cel mai frecvent un tablou `NumPy`.

Aici intervine **Librosa**, un pachet Python dedicat analizei muzicale și audio. Acesta oferă funcțiile esențiale pentru încărcarea, salvarea și analiza datelor audio, potrivit pentru fluxul nostru de procesare în recunoașterea automată a vorbirii (ASR).

Instalare Librosa

Prima etapă este instalarea bibliotecii **Librosa**. Deoarece este disponibilă în Python Package Index (PyPI), o puteți instala rapid utilizând managerul de pachete `pip`. Este, de asemenea, recomandat să instalați pachetul `soundfile`, pe care Librosa îl folosește ca motor de bază pentru manipularea fisierelor audio.

```
pip install librosa soundfile
```

Odată instalată, funcția principală pentru a încărca un fișier audio este `librosa.load()`. Aceasta gestionează complexitatea decodării diferitelor formate audio (precum `.wav` sau `.mp3`) și convertește semnalul într-un format numeric standardizat.

Funcția returnează două valori esențiale:

`y` **(serie de timp):** Un tablou NumPy care conține valorile amplitudinii semnalului audio. În mod implicit, datele sunt normalizate în intervalul `[-1.0, 1.0]`.
`sr` **(rata de eșantionare):** Rata la care au fost prelevate eșantioanele audio, măsurată în Hertz (Hz). Librosa reeșantionează automat la `22050 Hz` dacă nu se specifică altfel.

Încărcarea fișierelor audio cu Librosa

```
import librosa

audio_path = 'sample.wav'
y, sr = librosa.load(audio_path)

print(f"Forma seriei de timp audio: {y.shape}")
print(f"Rata de eșantionare: {sr} Hz")
print(f"Durata audio: {len(y) / sr:.2f} secunde")
```

Acest cod va produce un rezultat similar cu:

```
Forma seriei de timp audio: (110250,)
Rata de eșantionare: 22050 Hz
Durata audio: 5.00 secunde
```

Astfel, un semnal audio de 5 secunde, eșantionat la 22050 Hz, va produce un tablou unidimensional cu 110.250 de eșantioane (`22050 * 5 = 110250`).

Controlul ratei de eșantionare

Deși 22.050 Hz este o valoare implicită frecventă în procesarea audio, rata de eșantionare de 16.000 Hz (sau 16 kHz) reprezintă un standard larg adoptat în domeniul ASR. Aceasta este suficientă pentru a capta frecvențele necesare vorbirii umane și reduce costul computațional.

Puteți controla rata de eșantionare utilizând argumentul `sr` din funcția `librosa.load()`:

`sr=16000`: încarcă și reeșantionează fișierul audio la 16.000 Hz.
`sr=None`: încarcă fișierul audio la rata sa de eșantionare originală, nativă.

Încărcarea fișierului cu reeșantionare

```
import librosa

audio_path = 'sample.wav'

# Încărcarea cu rata de eșantionare nativă
y_native, sr_native = librosa.load(audio_path, sr=None)
print(f"Rata de eșantionare nativă: {sr_native} Hz")
print(f"Forma seriei de timp audio (nativă): {y_native.shape}")

# Încărcarea și reeșantionarea la 16 kHz
y_16k, sr_16k = librosa.load(audio_path, sr=16000)
print(f"Noua rată de eșantionare: {sr_16k} Hz")
print(f"Forma seriei de timp audio (16kHz): {y_16k.shape}")
```

Observați cum forma tabloului se modifică atunci când reeșantionăm. O rată de eșantionare mai mică va genera un tablou de dimensiuni reduse, deoarece sunt necesare mai puține eșantioane pentru a reprezenta aceeași durată a materialului audio.

Operațiuni audio de bază

Librosa oferă și o suită de instrumente pentru manipularea fișierelor audio. O etapă de preprocesare frecventă și esențială în ASR constă în eliminarea (tăierea) perioadelor de liniște de la începutul și de la sfârșitul unei rostiri.

Această procedură este utilă deoarece elimină porțiunile de semnal care nu conțin vorbire.
Funcția `librosa.effects.trim()` realizează exact acest lucru, identificând și decupând automat segmentele de liniște.

```
import librosa.effects

# Eliminarea perioadelor de liniște de la început și sfârșit
y_trimmed, _ = librosa.effects.trim(y)

print(f"Lungimea originală a seriei de timp: {len(y)} eșantioane")
print(f"Lungimea seriei de timp după tăiere: {len(y_trimmed)} eșantioane")
print(f"Durata originală: {len(y) / sr:.2f} secunde")
print(f"Durata după tăiere: {len(y_trimmed) / sr:.2f} secunde")
```