
This is the **published version** of the bachelor thesis:

Torrescusa Rubio, Lucia. *Disseny i implementació d'un sistema de transcripció automàtica de partitures de bateria a partir d'àudio*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334383>

under the terms of the  license.

Disseny i implementació d'un sistema de transcripció automàtica de partitures de bateria a partir d'àudio

Lucía Torrescusa-Rubio

28 de juny de 2026

Resum— La classificació d'esdeveniments d'àudio se sol abordar amb xarxes neuronals convolucionals (CNN) amb *backpropagation*, que pot ser computacionalment costós per a senyals temporals. E-prop [1] és una alternativa neuromòrfica *online* per entrenar xarxes neuronals de polsos (SNN). En aquest treball, hem investigat si aquest principi d'aprenentatge es pot aplicar a la detecció *multi-label* d'onsets de bateria. Hem implementat una SNN recurrent i l'hem entrenat amb e-prop per classificar 14 classes de bateria a partir de representacions temps-freqüència d'àudio. La mateixa tasca també s'ha abordat amb una CNN amb *backpropagation*, que proporciona una base convencional. Els resultats suggereixen que les SNN amb e-prop poden aproximar el rendiment global de les CNN amb *backpropagation*. En experiments amb poques dades, el model e-prop supera la CNN. Aquests resultats motiven l'investigació sobre aprenentatge *online* en sistemes neuromòrfics.

Paraules clau— Classificació d'àudio; detecció d'onsets de bateria; e-prop; multi-label; sistemes neuromòrfics; xarxes neuronals convolucionals; xarxes neuronals de polsos

Abstract— Audio event classification is commonly addressed with Convolutional Neural Networks (CNNs) trained by backpropagation, yet such approaches can be computationally demanding for temporal signals. E-prop [1] offers a neuromorphic online alternative for training recurrent Spiking Neural Networks (SNNs). In this work, we investigated whether this learning principle can be applied to multi-label drum onset detection. We implemented a recurrent SNN and train it using e-prop to classify 14 drum-event classes from time-frequency audio representations. The same task was also addressed with a CNN trained with backpropagation, providing a conventional baseline. Results suggest that SNNs trained with e-prop can approximate the overall performance of CNNs trained with backpropagation. In low-data experiments, the e-prop model outperforms the CNN. These findings motivate further work on online learning in neuromorphic systems.

Keywords— Audio classification; Convolutional Neural Networks; drum onset detection; e-prop; multi-label; neuromorphic systems; Spiking Neural Networks



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

LA detecció d'esdeveniments, o *onsets*, de bateria és un problema de classificació d'àudio que presenta la dificultat afegida que diversos esdeveniments poden succeir alhora. En el cas de la bateria, aquesta tasca consisteix a identificar en quin instant es produeix cada cop i a quina part o instrument correspon. Quan parlem d'una

bateria, cada tambor, timbal o plat que la forma (així com les diferents formes de tocar-los, que poden modificar-ne el so) es pot considerar una part independent. Una bateria pot estar formada per moltes parts, però típicament en té al voltant de deu o quinze.

Amb un model capaç de dur a terme aquesta tasca, es pot assolir un sistema de transcripció automàtica d'àudio de bateria a partitures o representacions simbòliques. Aquest problema s'ha tractat prèviament en articles de la literatura amb CNNs entrenades amb *backpropagation* [2], de tal manera que s'entrenava una xarxa independent per detectar cada part de la bateria. Tot i això, una part important de la literatura s'ha limitat a tres classes principals: *snare*, *kick* i

• E-mail de contacte: l.torrescusa.r@gmail.com

• Menció: Enginyeria de Computació

• Treball tutoritzat per: Xavier Otazu (Departament de Ciències de la Computació)

• Curs 2025/26

l'agrupació dels diferents tipus de *hi-hat*, fet que no permetria obtenir una transcripció completa de la bateria. També es pot trobar literatura sobre abordaments de la classificació d'*onsets* per altres instruments, sempre ressaltant la dificultat afegida de la tasca en música polifònica [3].

En aquest treball es planteja una detecció d'*onsets* multi-classe per a 14 instruments de bateria amb SNN, amb l'objectiu d'obtenir una transcripció temporal més detallada. A més, es vol comparar un enfocament convencional basat en CNNs amb una aproximació a sistemes neuromòrfics. Aquests sistemes proposen millorar l'eficiència computacional i oferir un funcionament més proper al de les xarxes neuronals biològiques, com en el cas de les SNNs entrenades amb e-prop [1].

1.1 Diferències entre CNN i SNN

Les CNNs i SNNs parteixen de principis diferents. Les CNNs són xarxes neuronals artificials de tipus *feed-forward* que processen la informació mitjançant capes successives, habitualment formades per convolucions, funcions d'activació, operacions de *pooling* i capes finals de classificació [4]. En aquest tipus de xarxes, la informació es propaga com activacions numèriques contínues entre capes, i els filtres convolucionals permeten extreure patrons locals de l'entrada. En aquestes xarxes, cada neurona calcula una activació numèrica a partir de les entrades rebudes, i aquesta activació es propaga cap a les capes següents. Per tant, la informació es representa mitjançant magnituds contínues, com poden ser les activacions internes de les capes o les probabilitats de sortida.

En una CNN estàndard, la resposta de la xarxa depèn principalment de l'entrada que es presenta en aquell moment. Tot i que una CNN pot processar dades amb estructura temporal si aquestes es representen com una matriu, no manté necessàriament un estat intern que recordi estímuls anteriors, excepte en variants específiques com les CNN recurrents o altres arquitectures amb mecanismes temporals.

En canvi, les SNNs intenten aproximar-se més al funcionament de les neurones biològiques. En una neurona biològica, la informació arriba a través de sinapsis, modifica l'estat elèctric de la neurona i, si el potencial de membrana supera un cert llindar, es genera un *spike*. Aquest *spike* es transmet a altres neurones a través de les connexions sinàptiques [5]. Així, la resposta d'una SNN no depèn només de l'entrada actual, sinó també dels estímuls anteriors i de l'estat intern acumulat per la xarxa.

Aquesta diferència és important en senyals temporals com l'àudio. En una CNN, tota la finestra d'entrada es processa com una representació bidimensional completa. En una SNN, en canvi, la informació pot introduir-se com una seqüència de *frames*, de manera que l'estat intern de les neurones evoluciona al llarg del temps. Això permet que la xarxa acumuli informació temporal abans de produir una sortida.

Des del punt de vista de l'aprenentatge, les CNNs s'entrenen habitualment amb *backpropagation*, que calcula gradients propagant l'error cap enrere a través de les capes de la xarxa. Aquest mètode és molt efectiu, però no és directament equivalent a un mecanisme biològic. A més, en xarxes recurrents o temporals pot requerir propagar informació a través de seqüències llargues, el que resulta computacional-

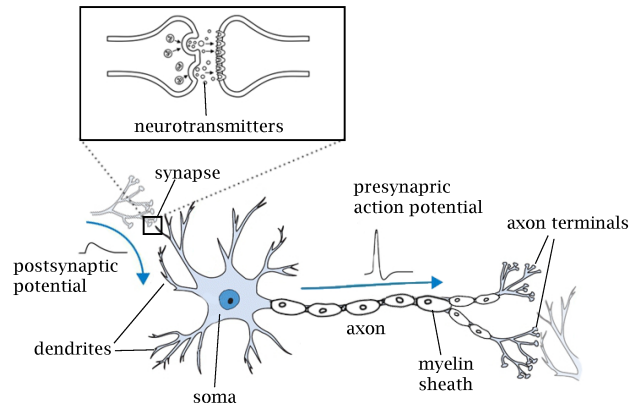


Fig. 1: Estructura típica d'una neurona i una sinapsi [6].

ment costós. Les SNNs presenten una dificultat addicional durant l'entrenament. En una xarxa neuronal convencional, les activacions són valors continus i això permet calcular gradients de manera directa. En canvi, en una SNN, una neurona genera un *spike* només quan el seu potencial de membrana supera un llindar. Aquesta sortida és discreta, ja que passa de no disparar a disparar, i això dificulta aplicar directament els mètodes habituals de càlcul de gradients.

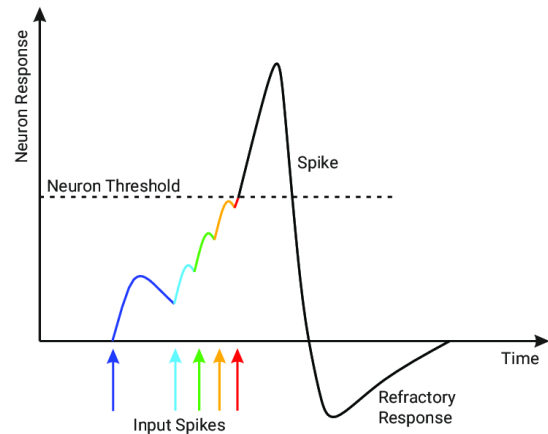


Fig. 2: Exemple de l'activitat dinàmica d'una neurona de polsos. El potencial de membrana evoluciona amb les entrades rebudes i, quan supera el llindar, es genera un *spike* [7].

En aquest treball, la CNN s'utilitza com a línia base convencional, ja que és una arquitectura àmpliament utilitzada en classificació d'àudio. La SNN, en canvi, s'estudia com una alternativa neuromòrfica per comprovar si un mecanisme d'aprenentatge basat en e-prop pot adaptar-se a la detecció *multi-label* d'*onsets* de bateria. La comparació entre ambdues arquitectures permet analitzar no només el rendiment final, sinó també com canvia el comportament dels models en escenaris amb moltes dades i amb dades limitades.

1.1.1 E-prop

El mètode e-prop proposa una alternativa per entrenar xarxes SNN recurrents evitant una retropropagació completa de l'error a través del temps [1]. La idea principal és aproximar el gradient de cada pes recurrent com la combinació

de dos termes: una traça d'elegibilitat i un senyal d'aprenentatge. De manera simplificada, l'actualització d'un pes recurrent W_{ji} es pot expressar com:

$$\frac{dE}{dW_{ji}} = \sum_t L_j^t e_{ji}^t \quad (1)$$

on E és la funció d'error, L_j^t és el senyal d'aprenentatge associat a la neurona postsinàptica j en l'instant t , i e_{ji}^t és la traça d'elegibilitat de la connexió entre la neurona presinàptica i i la neurona postsinàptica j .

La traça d'elegibilitat es pot entendre com una memòria local de cada connexió. En una sinapsi biològica, no totes les connexions haurien de modificar-se igual després d'un error: té més sentit modificar aquelles que han estat actives recentment i que, per tant, podrien haver contribuït a la sortida produïda. En e-prop, la traça d'elegibilitat indica fins a quin punt l'activitat recent d'una connexió la fa candidata a ser actualitzada.

Mentre que la traça d'elegibilitat només conté informació local de la connexió, el senyal d'aprenentatge aporta informació sobre l'error de la xarxa. Així, la traça indica quines connexions han estat rellevants, i el senyal d'aprenentatge indica en quina direcció s'haurien de modificar per reduir l'error.

D'aquesta manera, e-prop permet entrenar pesos recurrents sense haver de propagar explícitament l'error a través de tota la seqüència temporal. Tot i ser una aproximació, aquest mecanisme és especialment interessant en el context de les SNNs perquè manté una relació més directa amb la dinàmica temporal de les neurones de polsos i amb una interpretació neuromòrfica de l'aprenentatge.

1.2 Objectius

Així doncs, es va definir el següent objectiu principal:

- Dissenyar i implementar un sistema de transcripció automàtica d'àudio de bateria amb una CNN entrenada amb *backpropagation* i una SNN entrenada amb e-prop.

Per assolir aquest objectiu, es van definir els següents subobjectius:

- Obtenir un model CNN funcional per a la detecció d'*onsets* de bateria per a 14 instruments definits i establir-lo com a línia base de comparació.
- Obtenir un model SNN funcional per a la mateixa tasca i en condicions comparables al model CNN.
- Aconseguir que el sistema sigui aplicable amb resultats satisfactoris a música polifònica.

2 METODOLOGIA

L'objectiu principal del treball es va dividir en dos grans blocs: el desenvolupament del model CNN i el del model SNN. Tots dos models es van construir sobre un mateix *pipeline* de dades, amb el mateix conjunt de classes, el mateix format d'etiquetes i un protocol d'avaluació comú. Aquesta decisió va permetre que la comparació entre arquitectures fos el més justa possible, ja que les diferències observades

als resultats es podien atribuir principalment al model i no a canvis en el preprocessament o en les dades d'entrada. Tot i que en aquest apartat els dos models es descriuen per separat, el desenvolupament va tenir un solapament important: alguns experiments fets amb la CNN van orientar decisions posteriors de la SNN, i a la inversa.

2.1 Obtenció dels datasets

En aquest projecte es va treballar amb dues bases de dades tant per entrenar com per avaluar els models, diferenciant entre l'escenari de "només bateria" i l'escenari "polifònic". Per l'escenari d'àudio únicament de bateria, es va fer servir The Expanded Groove MIDI Dataset [8], que compta amb més de 400 hores d'àudio de bateria amb la seva transcripció en MIDI.

Inicialment, es va considerar MedleyDB 2.0 [10] per l'escenari de música polifònica, ja que semblava un conjunt de dades actual i prometedor. No obstant això, es va descartar perquè no proporcionava anotacions de bateria prou detallades per components, sinó només una etiqueta general que indicava si sonava alguna part de la bateria.

Finalment, es va fer servir RWC [9], format per cançons polifòniques i les seves transcripcions MIDI. Un dels principals problemes d'aquesta tasca en l'escenari polifònic va ser la manca d'un bon dataset de música polifònica amb suficients transcripcions de bateria. RWC, tot i ser un dels més extensos en aquest escenari, no presentava totes les classes que es volien detectar ni tenia presència de bateria en totes les seves cançons, cosa que es va haver de tenir en compte a l'hora d'avaluar l'escenari polifònic.

2.2 Definició de classes

Es van definir catorze classes de bateria a detectar en tots dos models: bombo (*bass drum*), *x-stick*, caixa (*snare*), pedal de xarles (*hi-hat pedal*), *hi-hat* tancat, *hi-hat* obert, tom, tom de base (*floor tom*), *vibraslap*, *crash*, plat del *ride* (*ride bow*), campana del *ride* (*ride bell*), plat xinès i plat *splash*.

Aquesta agrupació es va establir a partir de tres criteris principals. En primer lloc, es van analitzar les notes de percussió del protocol MIDI [11] i la seva correspondència amb els instruments de bateria. En segon lloc, es van estudiar formes d'ona i espectrogrames de diferents cops de bateria per identificar similituds acústiques entre instruments. Finalment, es va tenir en compte com es traduïrien aquestes classes a una representació de partitura, ja que l'objectiu final del sistema no era només detectar esdeveniments, sinó generar una transcripció musical interpretable.

Algunes notes MIDI es van agrupar dins d'una mateixa classe quan tenien una funció musical o una resposta acústica semblant. Per exemple, diferents toms es van agrupar sota la classe *tom*, i el mateix va succeir amb diferents *crashes*. Aquesta decisió va permetre mantenir un vocabulari prou detallat per representar una bateria completa, però sense arribar a una fragmentació excessiva que dificultés l'entrenament.

2.3 Construcció de les etiquetes

Els fitxers MIDI es van transformar en un format intermedi JSONL, amb una entrada per pista. En aquest fitxer es van

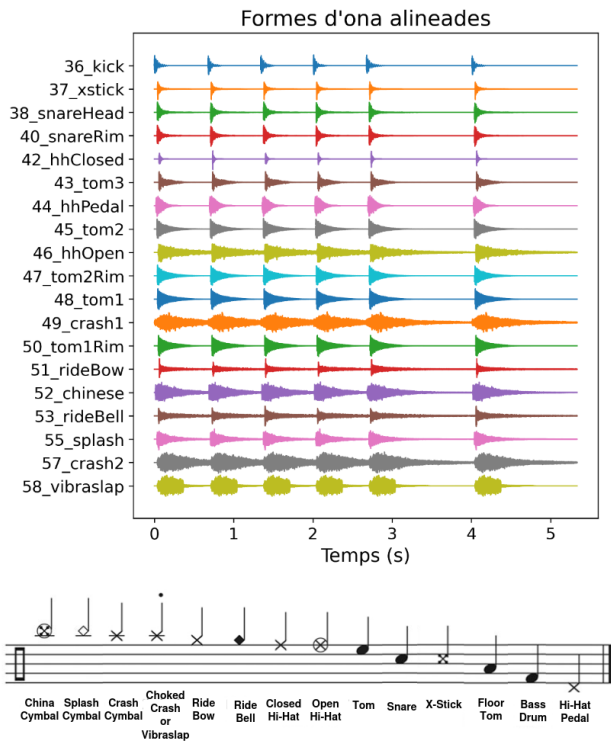


Fig. 3: Classificació de les classes. A dalt, representació de formes d'ona de la selecció d'instruments, amb una pista individual per classe i sis cops alineats al mateix instant. Indicada a cada etiqueta també es troba el seu corresponent MIDI. A sota, representació en partitura de les catorze classes seleccionades.

guardar els instants temporals d'activació de cada classe i, quan estava disponible, la velocitat MIDI associada a cada *onset*. Aquesta informació permetia conservar tant l'instant d'activació de cada cop, com també una indicació aproximada de la seva intensitat.

A partir d'aquestes anotacions es van generar les etiquetes utilitzades durant l'entrenament. El problema es va formular com una classificació *multi-label*, ja que una mateixa finestra temporal podia contenir diversos instruments simultàniament. Per exemple, és habitual que un cop de bombo aparegui al mateix temps que un *hi-hat*, o que una caixa coincideixi amb un plat. Per tant, cada mostra d'entrenament no tenia una única etiqueta, sinó un vector binari de 14 posicions, una per cada classe.

2.4 Representació temps-freqüència de l'àudio

Les pistes d'àudio es van convertir a representacions temps-freqüència amb la llibreria Librosa. Es van provar diferents alternatives d'entrada: espectrogrames mel, espectrogrames log-mel, espectrogrames logarítmics sense escala mel i Coeficients Cepstrals de Freqüència Mel (MFCC). Les representacions basades en l'escala mel són molt utilitzades actualment en tasques de classificació d'àudio amb xarxes neuronals, ja que adapten l'eix freqüencial de l'àudio a una escala més propera a la percepció de l'oïda humana. Per aquest motiu, es van considerar una opció adequada com a punt de partida per als primers experiments. Els paràmetres base van ser una freqüència de mostreig de

22050 Hz, una FFT de 1024 mostres i un *hop length* de 256 mostres. Aquesta configuració proporcionava un compromís entre resolució temporal i cost computacional, amb aproximadament 11,6 ms entre frames consecutius.

En aquest treball, el terme *frame* fa referència a cada columna de la representació temps-freqüència, és a dir, a cada instant temporal discretitzat després d'aplicar la transformada.

Els espectrogrames i MFCC es van precomputar i emmagatzemar en fitxers *.npy* per estalviar temps de processament.

2.5 Generació de finestres temporals

Els models no es van entrenar directament sobre pistes completes, sinó sobre finestres temporals d'un segon. Per a cada finestra, l'objectiu era determinar quines classes presentaven un *onset* al voltant del centre temporal.

Per reduir l'efecte de petits desajustos temporals entre l'anotació MIDI i la discretització en frames, es va aplicar una tolerància al voltant del centre de la finestra. D'aquesta manera, si un *onset* apareixia molt proper al centre, també es considerava positiu. Aquesta tolerància era necessària perquè els esdeveniments musicals no sempre queden perfectament alineats amb el frame central després del preprocesament.

2.6 Desbalanceig de classes

El conjunt de dades presentava dos tipus de desbalanceig: una distribució desigual entre instruments i una gran proporció de finestres sense cap *onset*. Per evitar que els models aprenguessin a predir absència d'esdeveniment, es va aplicar una estratègia combinada: sobre mostreig de finestres positives, mostreig ponderat segons la freqüència de cada classe i incorporació de *hard negatives* en regions properes a *onsets*.

TAULA 1: DISTRIBUCIÓ DEL NOMBRE D'ESDEVENIMENTS PER CLASSE EN EL CONJUNT DE DADES GROOVE.

Parelles wav/midi	45537
Classe	Nombre d'esdeveniments
snare	4141989
kick	2842803
hihat_closed	2030222
hihat_pedal	1520239
ride_bow	1449573
tom	613784
floor_tom	424582
hihat_open	363400
xstick	327318
ride_bell	215860
splash_cymbal	89096
chinese_cymbal	42054
vibraslap	36387
crash	28251

2.7 Model CNN amb backpropagation

La CNN implementada va ser una xarxa convolucional formada per tres blocs convolucionals. Cada bloc incloïa con-

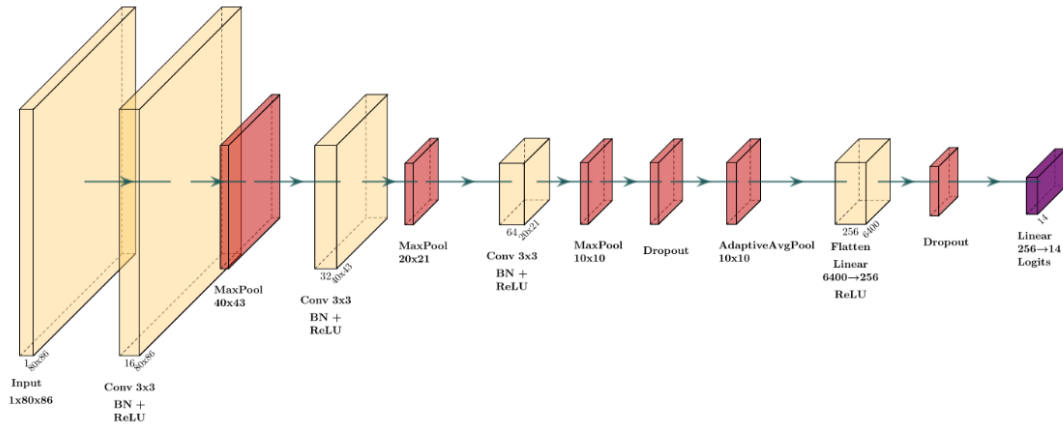


Fig. 4: Arquitectura general de la CNN definida per a la classificació *multi-label* de les 14 classes de bateria.

volució, normalització, activació ReLU i *max pooling*. El disseny inicial de l'arquitectura es va basar en treballs previs de l'estat de l'art sobre transcripció automàtica de bateria [2, 12] i en tècniques recents de classificació de so [13]. A partir d'aquestes referències, es va desenvolupar una implementació pròpia adaptada als objectius del projecte, modificant l'arquitectura, el format d'entrada i la sortida del model per abordar una tasca de classificació *multi-label* amb 14 classes de bateria.

La sortida convolucional es projectava amb *average pooling* adaptatiu i un classificador final produïa 14 probabilitats, una per classe.

L'entrenament es va fer amb *BCEWithLogitsLoss*, una *loss function* adequada per a classificació *multi-label*, i amb l'optimitzador AdamW per adaptar els pesos. Primer es van provar diferents representacions espectrals i, posteriorment, MFCC, que finalment es va adoptar com a entrada principal per a la comparació amb la SNN. Aquesta decisió va ser progressiva pels resultats obtinguts. Inicialment es van utilitzar espectrogrames mel i *log-mel*, ja que diverses referències de classificació d'àudio amb CNN feien servir representacions basades en l'escala mel [2, 3, 12, 13]. No obstant això, aquests primers experiments no van assolir resultats prou satisfactoris, fet que va motivar la prova de representacions menys comprimides, com el *log-spectrogram*.

Els resultats amb *log-spectrogram* van ser satisfactoris, però durant l'estudi del codi de referència d'e-prop es va observar que l'exemple fonètic treballava amb MFCC [1]. Per aquest motiu, es va incorporar també aquesta representació als experiments amb la CNN. Els MFCC van obtenir finalment els millors resultats i, a més, permetien comparar la CNN i la SNN amb una entrada comuna.

Per a la música polifònica es va aplicar *fine-tuning* a partir del model entrenat amb només bateria. El model base havia estat entrenat amb aproximadament 140.000 finestres d'àudio de Groove i MFCC com a representació d'entrada. Posteriorment, els pesos es van ajustar amb el dataset RWC preprocessat amb la mateixa estructura d'*onsets* per classe.

2.8 Model SNN amb e-prop

La SNN que es va desenvolupar es va basar en l'exemple de classificació fonètica amb e-prop proposat per Bellec et al. [1]. Aquest exemple resolvia una tasca de classificació *frame-wise* de fonemes a partir de característiques MFCC,

i presentava paral·lelismes clars amb el problema d'aquest treball: tots dos casos parteixen d'una representació seqüencial d'àudio i requereixen obtenir prediccions al llarg del temps. Tot i això, l'exemple original no resolvia una tasca de detecció d'*onsets* ni un problema *multi-label*, de manera que no es podia utilitzar directament. Per aquest motiu, una part important del desenvolupament va consistir a entendre, reproduir i adaptar aquest exemple al context de la transcripció automàtica de bateria.

En primer lloc, es va analitzar el funcionament del codi original i la formulació matemàtica de l'e-prop. L'objectiu era separar quines parts corresponien al model neuronal, quines parts implementaven el mecanisme d'aprenentatge local i quines parts eren específiques de la tasca fonètica original. Aquesta distinció era necessària perquè el model final havia de conservar la idea central de l'e-prop, però modificant el format d'entrada, la sortida i la funció d'error per adaptar-se a una classificació *multi-label* de catorze classes de bateria.

El codi original de l'e-prop estava implementat en TensorFlow, mentre que aquest projecte s'havia desenvolupat en PyTorch. Per mantenir un entorn comú i facilitar la comparació amb la CNN desenvolupada, es va reimplementar l'exemple en PyTorch, per més tard poder adaptar-lo a la tasca. Aquesta reimplementació va incloure la funció de *spike* amb pseudoderivada, utilitzada com a gradient substitutiu, el càlcul de les traces d'elegibilitat, el senyal d'aprenentatge local i la neurona ALIF (*Adaptive Leaky Integrate-and-Fire*) [14]. En aquesta neurona, l'estat intern inclou el potencial de membrana, la variable d'adaptació, la sortida binària de *spike* i el comptador del període refractari. Aquest estat permet que la resposta de cada neurona depengui no només de l'entrada actual, sinó també de la seva activitat anterior.

A partir d'aquest bloc neuronal es va construir una LSNN (*Long Short-Term Spiking Neural Network*) amb una capa recurrent [15]. En la configuració original de classificació fonètica, la xarxa treballava amb 13 coeficients MFCC com a entrada, 300 neurones regulars, 100 neurones adaptatives i 61 sortides, una per cada classe fonètica. Aquesta estructura va servir com a punt de partida per les similituds entre les dues tasques, però es va haver de modificar per adaptar-la a la detecció d'*onsets* de bateria, que es va formular com un problema *multi-label*. En el model proposat en aquest pro-

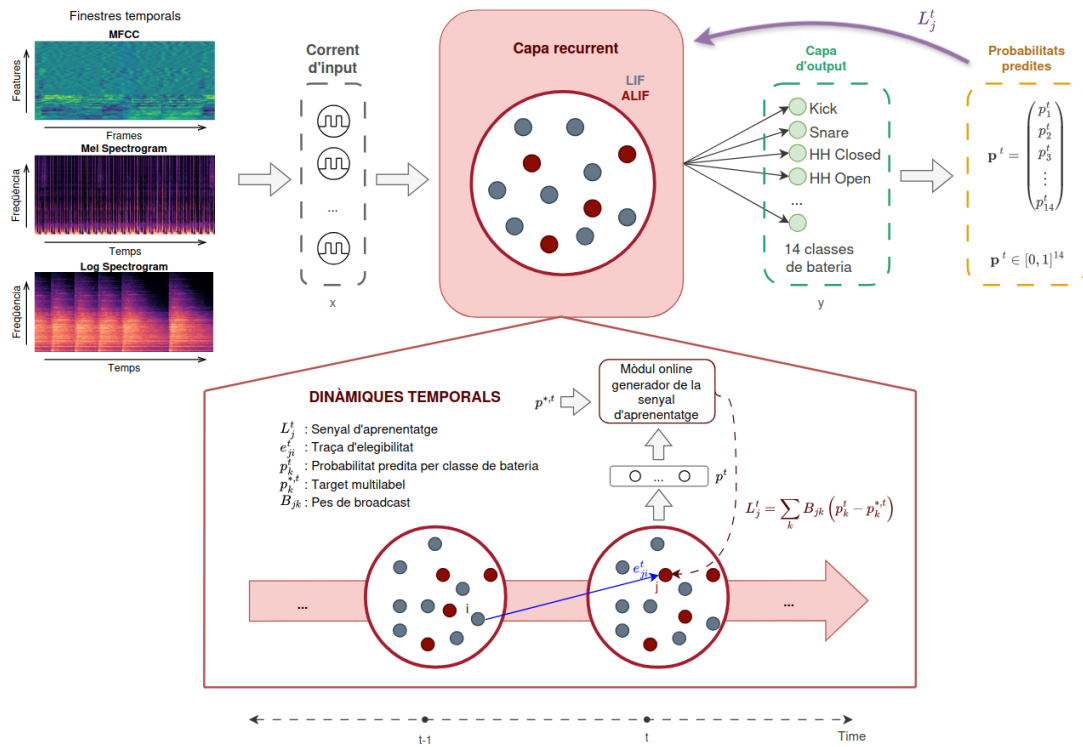


Fig. 5: Arquitectura de la SNN recurrent entrenada amb e-prop adaptatiu.

jecte, l'entrada estava formada per 120 coeficients MFCC, i la capa recurrent comptava amb 192 neurones regulars i 54 neurones adaptatives. La sortida es va reduir a 14 neurones, una per cada classe de bateria definida.

A diferència de la CNN, que interpreta cada finestra com una imatge temps-freqüència, la SNN processa cada *frame* de la representació acústica com un pas temporal. Això permet que l'estat intern de les neurones evolucioni al llarg de la finestra i acumuli informació temporal abans de produir la predicció final.

També es va estudiar una particularitat de l'exemple original: la presència de dues xarxes SNN amb e-prop, una *forward* i una *backward*. Es van fer proves per comprovar l'efecte d'eliminar la xarxa *backward*. Aquesta simplificació empitjorava el rendiment en l'exemple fonètic, però reduïa la complexitat de la implementació i permetia validar de manera més controlada la reproducció del mètode. Per aquest motiu, es va optar per una sola xarxa, prioritzant una implementació funcional, interpretable i biològicament plausible abans d'afegir més complexitat al sistema.

L'adaptació a la transcripció de bateria va requerir modificar la capa de sortida del model. L'exemple fonètic original utilitzava una sortida *softmax*, adequada per a classificació multiclasse, on cada *frame* pertany a una única classe. En canvi, en la bateria poden aparèixer diversos instruments simultàniament dins d'una mateixa finestra, com per exemple *kick* i *hi-hat*, o *snare* i *crash*. Per això, la sortida es va adaptar a 14 neurones independents, una per cada classe definida, i es va interpretar mitjançant una activació sigmoid. D'aquesta manera, cada classe podia activar-se o no de manera independent.

Aquesta modificació també va afectar el senyal d'aprenentatge local de l'e-prop. En l'exemple original, aquest senyal es calculava a partir de la diferència entre una distribució *softmax* i una etiqueta *one-hot*. En aquest treball, en

canvi, es va calcular a partir de l'error independent de cada classe respecte a l'etiqueta *multi-label*. Això va permetre mantenir el principi d'aprenentatge local de l'e-prop, basat en la combinació de traces d'elegibilitat i senyals d'aprenentatge, però aplicat a una tasca en què diversos esdeveniments poden aparèixer simultàniament.

Finalment, es va substituir el conjunt de dades fonètic pel dataset de bateria preprocessat. Per mantenir la comparació amb la CNN, es va reutilitzar el mateix esquema de finestres temporals centrades en *onsets*, les mateixes catorze classes i les mateixes mètriques d'avaluació. A partir d'aquesta adaptació, el model SNN quedava preparat per entrenar-se tant amb àudio de només bateria com amb música polifònica, mantenint el mateix format de sortida i el mateix protocol d'avaluació que el model CNN.

2.9 Protocol d'avaluació

Els models es van avaluar amb un protocol comú basat en mètriques globals i mètriques per classe. Les mètriques principals van ser la micro-F1 i la macro-F1. La micro-F1 resumeix el rendiment global tenint en compte totes les prediccions, mentre que la macro-F1 calcula el rendiment mitjà entre classes i és més sensible a les classes minoritàries. També es van registrar precisió i recall, tant globals com per classe.

Aquesta combinació de mètriques era necessària perquè el problema presentava un fort desbalanceig. Un model podia obtenir una micro-F1 raonable si encertava bé les classes més freqüents, però continuar fallant en classes poc representades. Per això, la macro-F1 i els resultats per classe van ser especialment importants per interpretar la qualitat real de la transcripció.

3 RESULTATS

3.1 CNN amb només bateria

Els primers experiments amb CNN van comparar espectrogrames *log-mel* amb diferents nombres de bandes mel. La Taula 2 resumeix els millors resultats obtinguts en aquests entrenaments segons la mètrica *val_macroF1*.

TAULA 2: MILLORS RESULTATS DELS ENTRENAMENTS AMB ESPECTROGRAMES LOG-MEL SEGONS *val_macroF1*.

N. bandes	micro-F1	macro-F1	precision	recall
80	0.6282	0.5126	0.6022	0.6566
128	0.6004	0.4920	0.5610	0.6457
256	0.6155	0.5042	0.5906	0.6425

Els resultats no van ser prou satisfactoris, i això va motivar una comparació més àmplia entre altres tipus de representacions d'entrada. En aquest segon conjunt d'experiments es van comparar espectrogrames mel, espectrogrames *log-mel* i espectrogrames logarítmics sense escala mel. Els resultats es mostren a la Taula 3.

TAULA 3: MILLORS RESULTATS SEGONS *val_macroF1* ALS ENTRENAMENTS AMB DIFERENTS TIPUS D'ESPECTROGRAMES COM A ENTRADA.

Experiment	micro-F1	macro-F1	precision	recall
Mel 80	0.3952	0.2399	0.3565	0.4435
Log-mel 80	0.6256	0.5108	0.6024	0.6506
Log-spectrogram	0.6748	0.5420	0.6623	0.6878

En aquesta comparació, els espectrogrames logarítmics sense escala mel van superar els mel i *log-mel*, fet que suggeria que la compressió mel podia eliminar informació útil per diferenciar alguns instruments de bateria. No obstant això, la incorporació posterior dels MFCC va modificar aquesta conclusió. Els MFCC van superar els resultats obtinguts amb les representacions espectrals, com es pot veure a la Fig. 6, i van oferir una entrada més compacta i adequada per als experiments finals. Per aquest motiu, es van adoptar com a representació comuna en la comparació entre CNN i SNN.

En comparar les classes comunes amb la literatura [2], resumides a la Taula 4, el model va presentar un comportament especialment positiu en *kick* (BD) i *snare* (SD). En canvi, el *hi-hat* (HH) va continuar sent una de les classes més difícils de classificar. Aquesta dificultat també va aparèixer en altres classes de tipus plat, com *crash*, *ride bell* o *vibraslap*, ja que comparteixen components freqüencials similars i poden aparèixer amb intensitats, durades i formes d'atac molt variables.

La comparació s'ha de fer amb precaució, ja que la literatura utilitza un vocabulari de tres classes principals: *bass drum*, *snare drum* i *hi-hat*, i no diferencia entre *hi-hat* tancat, obert i pedal. En canvi, en aquest projecte aquestes variants es tracten com a classes separades dins d'un model *multi-label* de catorze sortides. Per a la comparació del model CNN multiclasse s'ha fet servir la classe de *hi-hat* tancat, ja que és la que té més presència al conjunt de dades emprat [8].

En conjunt, els resultats per classe van mostrar que les classes amb atacs més clars i més representació al conjunt de dades, com *kick* i *snare*, eren més fàcils d'aprendre. En

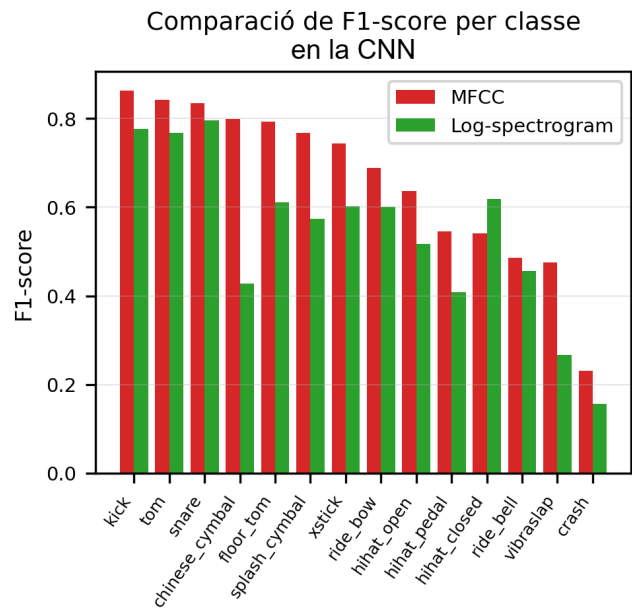


Fig. 6: Resultats d'F1-score per classe entre log-spectrograms i MFCC al model de CNN.

canvi, les variants de *hi-hat* i diversos plats van presentar més dificultats, probablement per la seva similitud tímbrica i per la variabilitat de la seva durada i intensitat.

3.2 SNN amb només bateria

En el model SNN es van comparar MFCC i log-spectrograms. Els log-spectrograms van donar valors molt baixos de F1, mentre que els MFCC van permetre un aprenentatge més estable. El millor experiment amb només bateria va obtenir una micro-F1 de 0.57 i una macro-F1 de 0.44. Aquest resultat indica que el model SNN amb un entrenament *online* i biològicament plausible va aprendre la tasca, tot i quedar per sota de la CNN amb *backpropagation* en l'escenari amb més dades.

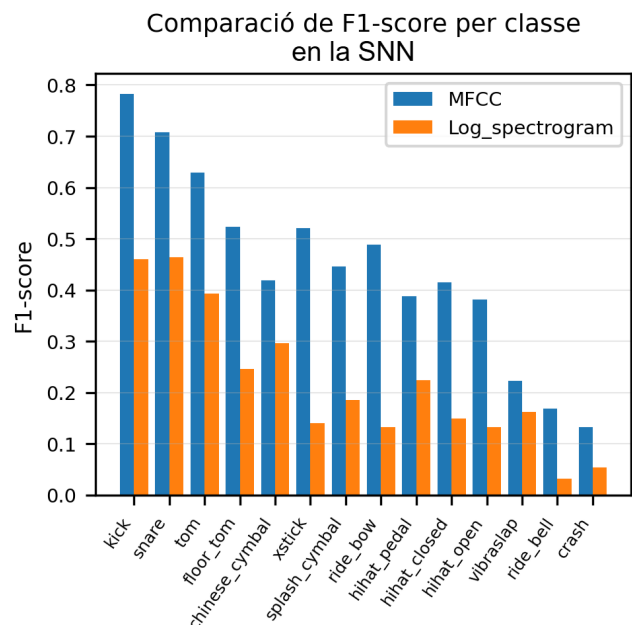


Fig. 7: Resultats d'F1-score per classe entre log-spectrograms i MFCC al model de SNN.

TAULA 4: COMPARACIÓ DE RESULTATS PER INSTRUMENT AMB EL MODEL CNN BASE.

Model	Precisió (%)			Recall (%)			F1-score (%)		
	BD	SD	HH	BD	SD	HH	BD	SD	HH
CNN base [2] amb una xarxa per instrument	77.5	57.9	71.0	75.0	67.0	89.7	76.2	62.1	79.3
CNN multiclasse amb log-spectrogram	80.4	82.7	54.6	77.1	78.0	74.2	77.9	79.7	62.2
CNN multiclasse amb MFCC	86.9	83.8	49.7	85.6	83.0	59.2	86.3	83.4	54.0

TAULA 5: RESULTATS DE L'ENTRENAMENT DE LA SNN AMB DIFERENTS REPRESENTACIONS DE DADES SEGONS *val_macroF1*.

Experiment	micro-F1	macro-F1	precision	recall
Log-spectrogram	0.2801	0.2196	0.1787	0.6482
MFCC	0.5676	0.4436	0.5236	0.6197

L'anàlisi per classe (Fig. 7) va mostrar millors resultats en *kick*, *snare* i *tom*. En canvi, *crash*, *ride bell* i *vibraslap* van ser més difícils. Aquest patró coincideix parcialment amb la CNN: les classes freqüents i amb atacs clars són més fàcils, mentre que els instruments menys representats o amb timbres semblants presenten més errors.

3.3 Música polifònica

En l'escenari polifònic, el rendiment dels dos models va baixar de manera clara. Això podria ser degut a una necessitat de preprocessar la música per fer destacar més la bateria, i així donar l'oportunitat al model de poder diferenciar instruments.

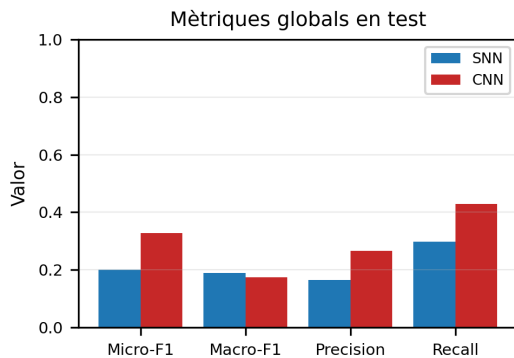


Fig. 8: Mètriques resultants dels entrenaments de cada model amb música polifònica.

Els resultats per classe en música polifònica van ser molt irregulars. Les classes amb millors valors van ser *hi-hat_closed*, *kick*, *crash* i *snare*. En canvi, *vibraslap*, *chinese_cymbal* i *splash_cymbal* van obtenir valors molt baixos o nuls. Aquest comportament s'explica tant per la complexitat de les mescles musicals com pel baix nombre d'exemples disponibles en algunes classes del dataset RWC.

3.4 Comparació entre CNN i SNN

La comparació final es va fer en dos escenaris: *full-data* i *few-shot*. En *full-data*, la CNN va obtenir millors resultats globals i per classe, indicant que aprofita millor una quantitat elevada de dades d'entrenament. En canvi, en *few-shot*, la SNN va superar la CNN en les mètriques globals i en diverses classes. Aquest resultat suggereix que l'aprenentat-

ge amb e-prop pot ser especialment interessant en contextos amb dades limitades.

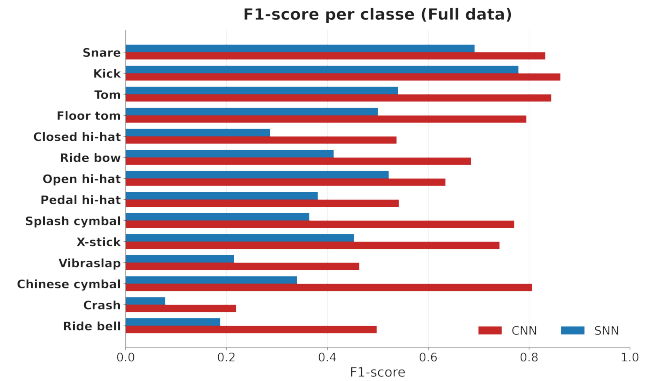


Fig. 9: Mètrica F1 per classe en l'escenari amb més de 140.000 finestres de mostra.

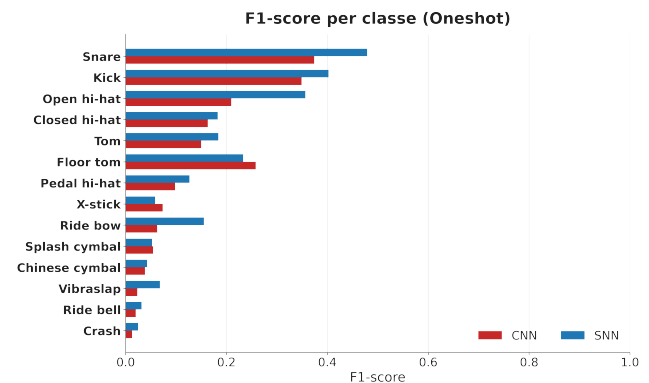


Fig. 10: Mètrica F1 per classe en l'escenari amb un únic exemple per classe.

Així, la diferència entre models no és constant, sinó que depèn de l'escenari experimental. La CNN és més robusta quan hi ha moltes dades disponibles, mentre que la SNN mostra un comportament competitiu quan l'entrenament es fa amb menys exemples. Aquest resultat suggereix que l'aprenentatge amb e-prop podria ser interessant en contextos on la disponibilitat de dades anotades és limitada, ja que en aquest escenari la SNN va conservar millor el rendiment global que la CNN.

3.5 Resultat qualitatiu i sortida simbòlica

A més de les mètriques quantitatives, es va analitzar qualitativament la sortida del sistema per comprovar si les prediccions obtingudes podien interpretar-se com una transcripció musical. Aquest pas és important perquè l'objectiu final del projecte no era només obtenir una classificació correcta de finestres d'àudio, sinó generar una representació temporal

d'esdeveniments que es pogués transformar en una partitura de bateria.

La sortida directa dels models consisteix en una seqüència de prediccions *multi-label*. Per a cada finestra temporal, el model retorna una probabilitat independent per a cadascuna de les catorze classes. Després d'aplicar un llindar de decisió, aquestes probabilitats es converteixen en una llista d'*onsets* detectats, on cada esdeveniment queda definit pel seu instant temporal i per la classe de bateria associada. Aquesta llista és el pas intermediari entre la classificació d'àudio i la representació musical final.

Per convertir aquesta sortida en una representació de partitura, cada classe detectada es va associar amb la seva posició corresponent dins de la notació de bateria. Aquesta correspondència ja s'havia tingut en compte durant la definició de classes, de manera que les agrupacions no es van fer només segons criteris acústics, sinó també segons la seva possible traducció a pentagrama. Per exemple, classes com *kick*, *snare*, *hi-hat* o *ride* tenen una posició específica dins de la notació de bateria, i això permet que la sortida del model sigui interpretable musicalment.

La Fig. 11 mostra un exemple qualitatiu de la sortida generada pel sistema. A la part superior es mostra el *target*, és a dir, la transcripció esperada a partir de les anotacions originals. A la part inferior es mostren les prediccions dels models CNN entrenat amb *backpropagation* i SNN entrenat amb e-prop. En aquest exemple, el model aconsegueix recuperar una part important de l'estructura rítmica, però també deixa sense detectar alguns cops de *hi-hat* tancat, marcats en vermell. Aquest tipus d'error és coherent amb els resultats quantitius obtinguts, ja que les variants de *hi-hat* són de les classes més difícils de separar.

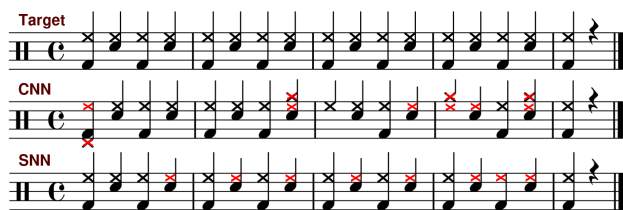


Fig. 11: Exemple qualitatiu del sistema. A dalt, el *target*. Al mig, la sortida del model CNN entrenat amb *backpropagation*. A sota, la sortida del model SNN entrenat amb e-prop. En vermell es ressalten les notes incorrectes, o bé per no detectades o detectades incorrectament, corresponents a *hi-hat* tancat.

Aquest resultat qualitatiu ajuda a interpretar millor les mètriques. Un valor d'F1-score no mostra directament si la transcripció manté el patró rítmic general, ni si els errors afecten parts musicalment rellevants. En canvi, la visualització en partitura permet observar si el model conserva l'estructura temporal del fragment i quines classes tendeix a ometre o confondre. En aquest cas, es pot veure que ambdós models són capaços de generar una sortida musicalment coherent, però encara presenten limitacions en esdeveniments repetitius i freqüents com el *hi-hat*.

Aquesta anàlisi també mostra una limitació del plantejament basat en finestres centrades en l'*onset*. El sistema decideix si una classe està activa al voltant del centre de cada finestra, però no modela explícitament la durada completa de la nota ni la seva evolució posterior. Això pot ser

suficient per a instruments amb atacs molt marcats i durades curtes, com *kick* o *snare*, però pot ser més problemàtic per als plats i els *hi-hats*, que tenen cues sonores més llargues i una major variabilitat tímbrica. Per aquest motiu, una línia futura important seria incorporar més context temporal posterior a l'atac o combinar la detecció local d'*onsets* amb informació rítmica.

4 CONCLUSIONS

El treball va permetre construir una CNN funcional per a la detecció *multi-label* d'*onsets* de bateria, adaptar una SNN recurrent entrenada amb e-prop per la mateixa tasca i comparar ambdós models sota un protocol comú. D'aquesta manera, es va assolir l'objectiu principal de dissenyar un sistema de transcripció automàtica d'àudio de bateria amb dues arquitectures diferents.

Els resultats van mostrar que la representació d'entrada és un factor clau. Tot i que els espectrograms logarítmics van ser la millor opció espectral inicial per a la CNN, els MFCC van obtenir els millors resultats finals i van permetre comparar la CNN i la SNN en condicions equivalents. Aquesta observació és especialment rellevant perquè difereix parcialment de treballs de classificació d'àudio on els mel-espectrograms solen obtenir millors resultats a les CNN [16, 17, 18].

La CNN va obtenir el millor rendiment en l'escenari *full-data*, però la SNN entrenada amb e-prop va mostrar un comportament competitiu en experiments amb poques dades. Per tant, encara que la CNN continuï sent la millor opció en condicions generals, la SNN representa una alternativa prometedora per a escenaris amb dades limitades i per a futurs sistemes neuromòrfics d'aprenentatge *online*.

L'objectiu d'aplicar el sistema a música polifònica es va assolir parcialment. Els models es van adaptar a RWC mitjançant *fine-tuning*, però el rendiment va ser limitat a causa de la presència d'altres instruments, la baixa disponibilitat de cançons amb bateria anotada i el fort desbalanceig entre classes. Aquesta és la principal limitació del treball i una de les raons per les quals la transcripció de bateria en música polifònica continua sent una tasca difícil en la literatura.

Com a línies futures, seria convenient incorporar context musical de nivell superior, com tempo, patrons de *groove* i dependències temporals entre esdeveniments. També seria interessant explorar finestres que incloguin més informació posterior a l'*onset*, especialment per als plats, així com arquitectures híbrides que combinin detecció local, context rítmic i separació de fonts.

ÚS DE LA IA GENERATIVA

Al llarg d'aquest projecte s'ha fet servir IA generativa únicament com a suport per a la correcció gramatical, sintàctica i estilística de l'informe. En cap cas s'ha utilitzat per trobar fonts o generar idees, conclusions, interpretacions dels resultats o decisions metodològiques.

AGRAÏMENTS

Vull agrair al meu tutor, Xavier Otazu, la seva guia, disponibilitat i suport al llarg del desenvolupament d'aquest projecte. També vull agrair al grup de recerca *Bridging Research in AI and Neuroscience*, del Centre de Visió per Computador, l'acollida i l'ajuda rebuda durant el treball.

Finalment, agraeixo a la comunitat *Barccsyn* l'oportunitat de presentar aquest projecte en format pòster al seu congrés anual del 2026, així com la retroacció i els comentaris rebuts, que van ajudar a valorar el treball des d'una perspectiva externa.

REFERÈNCIES

- [1] G. Bellec *et al.*, “A solution to the learning dilemma for recurrent networks of spiking neurons”, *Nat Commun*, vol. 11, núm. 1, des. 2020, doi: 10.1038/s41467-020-17236-y.
- [2] C. Jacques i A. Roebel, “Automatic drum transcription with convolutional neural networks”, *21st International Conference on Digital Audio Effects*, set. 2018. [En línia]. Disponible a: <https://hal.science/hal-02018777v1>
- [3] J. Schlüter i S. Böck, “Improved musical onset detection with Convolutional Neural Networks”, *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 6979–6983, doi: 10.1109/ICASSP.2014.6854953.
- [4] S. Indolia, A. K. Goswami, S. P. Mishra i P. Asopa, “Conceptual Understanding of Convolutional Neural Network – A Deep Learning Approach”, *Procedia Computer Science*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.
- [5] D. Goodman i M. Ghosh, *Neuroscience for Machine Learners*. Zenodo, des. 12, 2023, doi: 10.5281/zenodo.10366802.
- [6] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara i N. Le, “Spiking Neural Networks and Their Applications: A Review”, *Brain Sciences*, vol. 12 no. 7, pp. 863, 2022, doi: 10.3390/brainsci12070863
- [7] C. M. Parameshwara, S. Li, C. Fermüller, N. J. Sanket, M. S. Evanusa i Y. Aloimonos, “SpikeMS: Deep Spiking Neural Network for Motion Segmentation”, *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, doi: 10.1109/IROS51168.2021.9636506.
- [8] L. Callender, C. Hawthorne, i J. Engel, “Improving Perceptual Quality of Drum Transcription with the Expanded Groove MIDI Dataset”, des. 2020, [Dataset]. Disponible a: [arXiv:2004.00188v5](https://arxiv.org/abs/2004.00188v5)
- [9] M. Goto, S. Balke i M. Mueller, “RWC Music Database”. Zenodo, feb. 16, 2026. doi: 10.5281/zenodo.18656623.
- [10] R. Bittner, J. Wilkins, H. Yipy, i J. P. Bello, “MedleyDB 2.0 Audio”. Zenodo, ago. 11, 2016, [Dataset], doi: 10.5281/zenodo.1715175.
- [11] J. Rothstein, *MIDI: A Comprehensive Introduction*, 2nd ed. Madison, EUA: A-R Editions, 1995.
- [12] J. Väisänen, “Konvoluutioneuroverkko metallimusiikkia sisältävän polyfonisen ääniraidan rumpiskutapahtumien havaitsemisessa”, Treball de Fi de Màster, 2020. [En línia]. Disponible a: https://jyx.jyu.fi/jyx/Record/jyx_123456789_73276
- [13] B. Yayla, “Urban Sound Classification using 2D CNN with SE Blocks”, repositori de GitHub, juny 15, 2025. Consultat el 24 de febrer de 2026 [En línia]. Disponible a: <https://github.com/BeytullahYayla/Urban-Sound-Classification>
- [14] W. Gerstner, W. M. Kistler, R. Naud i L. Paninski, *Neuronal Dynamics. From single neurons to networks and models of cognition*, Cambridge, U.K.: Cambridge University Press, 2014.
- [15] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein i W. Maass, “Long short-term memory and Learning-to-learn in networks of spiking neurons”, *NeurIPS*, Montréal, Canada, 2018.
- [16] M. Telmem, N. Laaidi i H. Satori, “The impact of MFCC, spectrogram, and Mel-Spectrogram on deep learning models for Amazigh speech recognition system”, *International Journal of Speech Technology*, vol. 28, pp. 299-312, març 2025, doi: 10.1007/s10772-025-10183-3.
- [17] S. Carvalho i E. Ferreira, “Automatic Classification of Bird Sounds: Using MFCC and Mel Spectrogram Features with Deep Learning”, *Vietnam Journal of Computer Science*, vol. 10, no. 1, pp. 39-54, 2023. doi: 10.1142/S2196888822500300.
- [18] P. Rawat, M. Bajaj, S. Vats i V. Sharma, “A comprehensive study based on MFCC and spectrogram for audio classification”, *Journal of Information and Optimization Sciences*, vol. 44, no. 6, pp. 1057-1074, 2023.

APÈNDIX

A.1 Anàlisi prèvia de les classes de bateria

Abans d'entrenar els models, es va dur a terme una anàlisi prèvia de les diferents parts de la bateria amb l'objectiu de definir un conjunt de classes coherent tant des del punt de vista acústic com musical. Aquesta fase era necessària perquè el protocol MIDI inclou un nombre elevat de notes de percussió, però no totes apareixen amb la mateixa freqüència al conjunt de dades ni totes tenen una correspondència directa amb una classe independent útil per a la transcripció final.

En primer lloc, es van revisar les notes de percussió definides al protocol MIDI [11] i es van comparar amb les notes presents al dataset Groove [8]. Aquesta revisió va permetre identificar quins instruments apareixien realment a les dades i quines notes podien agrupar-se sense perdre informació rellevant per a la partitura. Per exemple, algunes variants d'un mateix instrument es van agrupar en una sola

Acoustic Bass Drum (35)	B1	
Electric Bass Drum (36)	C2	
Acoustic Snare (38)	D2	C#2 (37) Side Stick
Electric Snare (40)	E2	D#2 (39) Hand Clap
Low Floor Tom (41)	F2	
High Floor Tom (43)	G2	F#2 (42) Closed Hi-Hat
Low Tom (45)	A2	G#2 (44) Pedal Hi-Hat
Low-Mid Tom (47)	B2	A#2 (46) Open Hi-Hat
Hi-Mid Tom (48)	C3	
High Tom (50)	D3	C#3 (49) Crash Cymbal 1
Chinese Cymbal (52)	E3	D#3 (51) Ride Cymbal 1
Ride Bell (53)	F3	
Splash Cymbal (55)	G3	F#3 (54) Tambourine
Crash Cymbal 2 (57)	A3	G#3 (56) Cowbell
Ride Cymbal 2 (59)	B3	A#3 (58) Vibraslap
Hi Bongo (60)	C4	<middle C>
Mute Hi Conga (62)	D4	C#4 (61) Low Bongo
Low Conga (64)	E4	D#4 (63) Open Hi Conga
High Timbale (65)	F4	
High Agogo (67)	G4	F#4 (66) Low Timbale
Cabasa (69)	A4	G#4 (68) Low Agogo
Short Whistle (71)	B4	A#4 (70) Maracas
Long Whistle (72)	C5	
Long Guiro (74)	D5	C#5 (73) Short Guiro
Hi Wood Block (76)	E5	D#5 (75) Claves
Low Wood Block (77)	F5	
Open Cuica (79)	G5	F#5 (78) Mute Cuica
Open Triangle (81)	A5	G#5 (80) Mute Triangle

Fig. 12: Notes pertanyents a l'escala de percussió en MIDI. Els que tenen presència al dataset Groove estan ressaltats en groc.

classe quan tenien una funció musical similar o quan la seva distinció no era imprescindible per a l'objectiu del sistema.

Per estudiar aquestes agrupacions, es van generar mostres individuals de diferents elements de bateria utilitzant una bateria electrònica i un *sampler* de bateria. Aquestes mostres es van enregistrar en un context controlat, amb cops aïllats i patrons rítmics senzills, per poder observar millor les diferències entre instruments. A partir d'aquestes gravacions es van analitzar formes d'ona (Fig. 3), espectrogrames i descriptors espectrals amb l'objectiu d'identificar similituds i diferències acústiques entre classes.

Les formes d'ona van permetre observar diferències en la durada i en la forma de l'atac dels instruments. Els instruments com *kick* i *snare* presentaven atacs més marcats i concentrats temporalment, mentre que els plats i els *hi-hats* mostraven cues sonores més llargues i una variabilitat més gran. Aquesta diferència és important perquè el sistema treballa amb finestres centrades en l'*onset*; per tant, els instruments amb atacs curts i ben definits tendeixen a ser més fàcils de detectar que aquells en què part de la informació distintiva apareix després de l'atac inicial.

Els espectrogrames també van ser útils per justificar algunes agrupacions. En el cas dels toms, per exemple, diferents notes MIDI presentaven patrons espectrals semblants i una funció musical propera, de manera que es van agrupar sota la classe *tom*. De manera similar, algunes variants de *crash* o de *snare* es van tractar com una mateixa classe quan la seva separació no aportava una millora clara per a la transcripció final. En canvi, altres elements es van mantenir separats perquè tenien una funció musical o una resposta acústica diferenciada, com *ride bow* i *ride bell*.

Aquesta anàlisi també va mostrar que els plats i les variants de *hi-hat* podien ser especialment problemàtics. Tot i tenir diferències musicals clares, comparteixen components freqüencials i poden variar molt segons la intensitat, la durada i la forma d'execució. Aquesta observació va ajudar

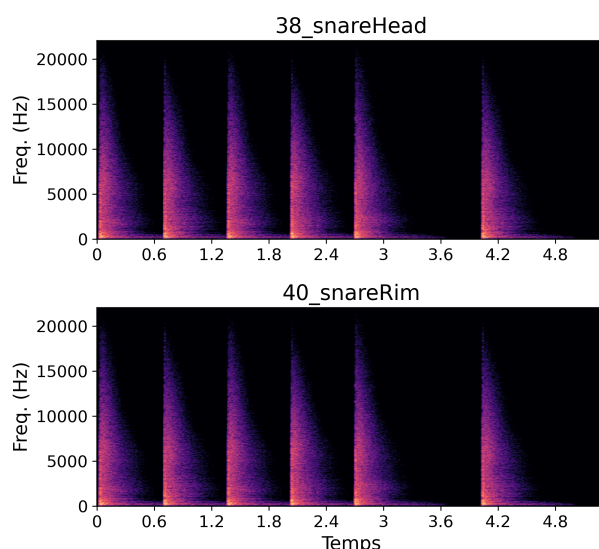


Fig. 13: Forma espectral d'un cop al centre de la caixa (*snare*) i a la vora (*snare rim*).

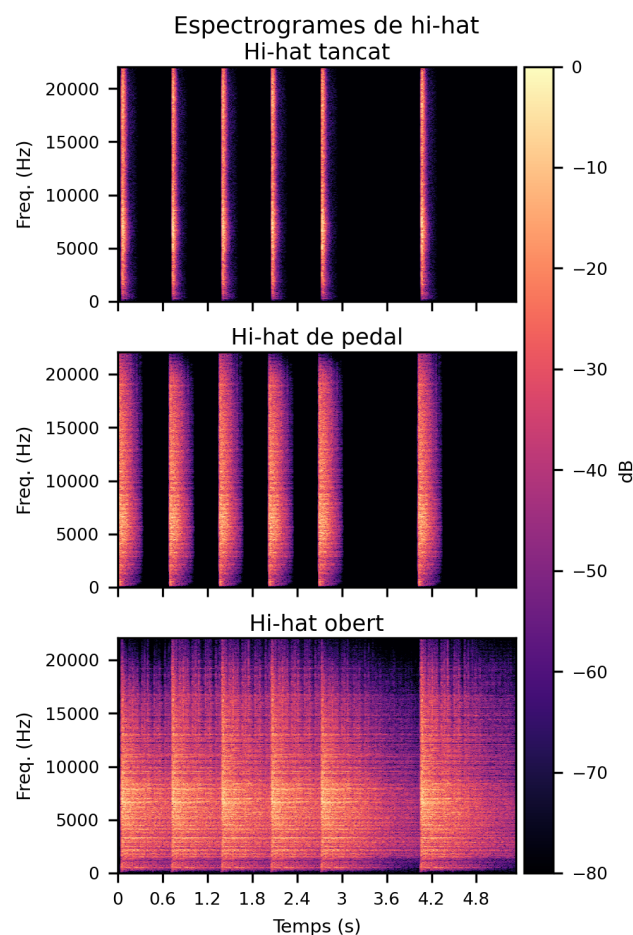


Fig. 14: Espectrogrames de les tres maneres diferents de tocar un hi-hat. Es pot observar que tots coincideixen en la forma espectral a l'inici del cop, i que la variació es troba a la cua sonora, el que pot dificultar la seva classificació.

a interpretar posteriorment els resultats dels models, ja que aquestes classes van aparèixer sovint entre les més difícils de classificar.

Finalment, la definició de classes es va contrastar amb

la seva traducció a partitura. L'objectiu del projecte no era només classificar sons de bateria, sinó generar una sortida interpretable com a transcripció musical. Per aquest motiu, es va evitar definir classes que fossin acústicament diferenciables però poc útils en notació, i també es va evitar agrupar instruments que, tot i tenir certa similitud tímbrica, havien de representar-se de manera diferent en la partitura. El resultat final va ser un conjunt de catorze classes: *kick*, *xstick*, *snare*, *hihat_closed*, *hihat_open*, *hihat_pedal*, *tom*, *floor_tom*, *vibraslap*, *crash*, *ride_bow*, *ride_bell*, *chinese_cymbal* i *splash_cymbal*.

A.2 Anàlisi de finestres i desbalanceig

Durant el desenvolupament també es va analitzar la distribució de les finestres temporals utilitzades per entrenar els models. Aquesta anàlisi tenia l'objectiu de comprovar fins a quin punt el problema estava desbalancejat no només entre classes, sinó també entre finestres amb activitat i finestres sense cap *onset* al centre. Els resultats van confirmar que la majoria de finestres no contenien un esdeveniment central, fet que justificava l'ús d'estratègies específiques de mostreig durant l'entrenament.

TAULA 6: RESUM DE L'ANÀLISI DE FINESTRES DEL CONJUNT DE DADES.

Paràmetre	Valor
Finestres temporals analitzades	2000
Duració de les finestres	1.0 s
Frames per finestra	86
Tolerància	1 frame
Pistes analitzades	400
Finestres amb centre actiu	22.00%
Finestres amb centre inactiu	78.00%

Com es mostra a la Taula 6, només un 22% de les finestres analitzades contenien un *onset* al centre, mentre que el 78% restant corresponia a finestres sense activitat central. Aquesta distribució és problemàtica perquè, si el model s'entrena directament sobre aquestes dades, pot aprendre a predir absència d'esdeveniment de manera sistemàtica i obtenir una pèrdua aparentment baixa sense aprendre correctament la detecció de cops de bateria.

També es va analitzar quines classes apareixien amb més freqüència al centre de les finestres actives. Aquesta anàlisi va mostrar que les classes més habituals eren *snare*, *hihat_closed* i *kick*, mentre que altres instruments tenien molta menys presència. Aquest resultat coincideix amb la distribució global del dataset i explica per què les classes més freqüents tendeixen a obtenir millors resultats en els experiments.

TAULA 7: CLASSES MÉS FREQUËNTS AL CENTRE DE LES FINESTRES ANALITZADES.

Classe	Freqüència
<i>snare</i>	209
<i>hihat_closed</i>	88
<i>kick</i>	70
<i>floor_tom</i>	60
<i>tom</i>	58
<i>ride</i>	22
<i>crash</i>	14
<i>hihat_open</i>	14

Finalment, es van estudiar les combinacions de classes

més freqüents al centre de les finestres. Aquesta informació era important perquè el problema no es podia formular com una classificació multiclasse simple: en una mateixa finestra podien aparèixer diversos instruments simultàniament. Per exemple, combinacions com *snare* amb *hihat_closed*, o *kick* amb *hihat_closed*, són habituals en patrons de bateria.

TAULA 8: COMBINACIONS DE CLASSES MÉS FREQUËNTS AL CENTRE DE LES FINESTRES ANALITZADES.

Combinació de classes	Freqüència
<i>none</i>	1560
<i>snare</i>	160
<i>hihat_closed</i>	59
<i>tom</i>	48
<i>floor_tom</i>	35
<i>kick</i>	34
<i>snare, floor_tom</i>	17
<i>snare, hihat_closed</i>	13
<i>ride</i>	11
<i>kick, hihat_closed</i>	9

A partir d'aquests resultats, es va aplicar una estratègia combinada per reduir el biaix cap a la classe negativa. En primer lloc, es va augmentar la presència de finestres positives durant l'entrenament. En segon lloc, es va utilitzar mostreig ponderat segons la freqüència de cada classe, per donar més oportunitats a instruments menys representats. Finalment, es van incorporar *hard negatives*, és a dir, finestres properes a un *onset* però sense un esdeveniment central. Aquestes mostres són especialment útils perquè obliguen el model a distingir entre l'energia residual d'un cop i l'aparició real d'un nou esdeveniment.

A.3 Adaptació tècnica de l'exemple fonètic d'e-prop

L'adaptació de l'exemple fonètic d'e-prop va ser una de les parts més tècniques del projecte. El punt de partida va ser el codi dels autors de l'e-prop, que implementava una tasca de classificació fonètica *framewise*. En aquesta tasca, cada instant temporal de l'entrada acústica s'associava a una única classe fonètica, i la xarxa produïa una predicció per *frame*. Aquesta formulació era útil com a referència perquè també treballava amb àudio representat com una seqüència temporal de característiques, però diferia del problema de la bateria en diversos aspectes.

El primer canvi va ser la reimplementació del model en PyTorch. Aquesta decisió es va prendre per mantenir la coherència amb la resta del projecte, ja que el model CNN i el pipeline de dades també s'havien desenvolupat amb aquesta llibreria. La traducció no va consistir només a canviar la sintaxi de TensorFlow a PyTorch, sinó que va requerir reproduir explícitament els mecanismes interns de l'e-prop. En particular, es van implementar la dinàmica de les neurones ALIF, el càlcul de les traces d'elegibilitat, el senyal d'aprenentatge local i la substitució manual del gradient dels pesos recurrents abans del pas de l'optimitzador.

En una xarxa recurrent entrenada amb *backpropagation through time*, el gradient es calcula retropropagant l'error a través de tota la seqüència temporal. En canvi, l'e-prop proposa una alternativa més local, on l'actualització dels pesos recurrents es construeix a partir de dos termes: les traces d'elegibilitat i el senyal d'aprenentatge. Les traces

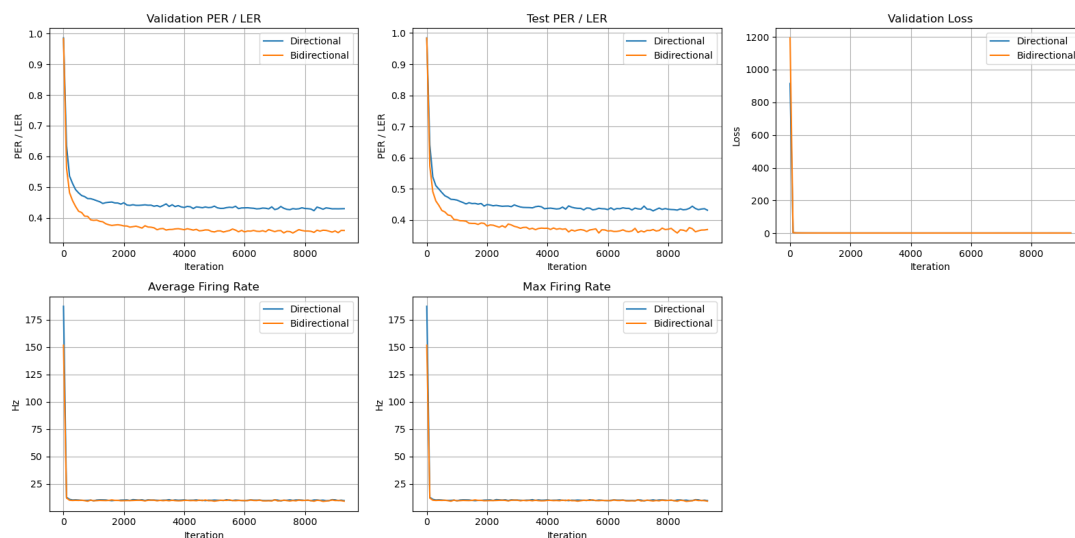


Fig. 15: Resultats de l'execució de l'exemple fonètic d'e-prop en TensorFlow, comparant l'arquitectura original bidireccional amb l'adaptació a una única xarxa *forward*. En aquesta prova, el model amb només xarxa *forward* presenta un error aproximadament 10 punts superior al de la configuració bidireccional.

d'elegibilitat acumulen informació local sobre com l'activitat de les neurones presinàptiques i l'estat intern de les neurones postsinàptiques afecten cada connexió. El senyal d'aprenentatge, en canvi, indica en quina direcció s'hauria de modificar la sortida de la xarxa per reduir l'error.

En la implementació desenvolupada, cada neurona ALIF mantenia diversos components d'estat: el potencial de membrana, la variable d'adaptació, la sortida binària de *spike* i el període refractari. La variable d'adaptació permetia que una neurona que havia disparat recentment augmentés temporalment el seu llindar efectiu, fet que introdueix una forma de memòria temporal en la dinàmica neuronal. Aquesta propietat és rellevant en senyals d'àudio, ja que la informació no es troba només en un instant concret, sinó també en l'evolució temporal de l'energia i del timbre.

Un cop reproduïda la base neuronal, es va adaptar la tasca. L'exemple fonètic original utilitzava 61 sortides i una funció d'error associada a una classificació multiclasse. En el cas de la bateria, es van definir 14 sortides, corresponents a les classes seleccionades al projecte. A més, es va canviar el tipus de problema: en lloc de predir una única classe activa per *frame*, el model havia de predir un vector binari de catorze posicions, ja que diversos instruments poden aparèixer simultàniament dins de la mateixa finestra temporal.

Aquesta diferència va implicar substituir la interpretació *softmax* per una sortida independent per classe. En termes pràctics, cada neurona de sortida es va tractar com un detector independent d'una classe de bateria. Això va permetre representar combinacions com *kick* amb *hi-hat*, *snare* amb *crash* o altres solapaments habituals en patrons de bateria. El senyal d'aprenentatge local també es va modificar perquè l'error de cada sortida depengués de la seva etiqueta corresponent, i no d'una competició entre classes.

També es va modificar la manera com el model interpretava les dades d'entrada. En la CNN, cada finestra temps-freqüència es processava com una matriu bidimensional completa, de manera semblant a una imatge amb un eix temporal i un eix freqüencial. En canvi, la SNN

no rebia aquesta finestra com una imatge única, sinó com una seqüència temporal: cada columna de la representació acústica, corresponent a un *frame*, s'introduïa successivament com un pas temporal de la xarxa recurrent.

Aquesta formulació era necessària perquè la LSNN manté un estat intern que evoluciona al llarg del temps. Així, la xarxa podia acumular informació dels diferents *frames* de la finestra abans de produir la predicció final, en lloc de fer una classificació directa sobre tota la matriu d'entrada.

Durant el procés d'adaptació del codi també es van fer proves amb la configuració de l'exemple original. En concret, es va estudiar l'efecte d'utilitzar només la xarxa *forward*, eliminant la xarxa *backward*. Aquesta modificació simplificava el model i reduïa la complexitat de la implementació, però també empitjorava el rendiment en l'exemple fonètic. Tot i això, es va considerar una simplificació raonable en el sistema, perquè l'objectiu principal era obtenir una SNN funcional, biològicament plausible i comparable amb la CNN en la tasca de detecció d'*onsets* de bateria.

A partir d'aquesta base, la SNN es va entrenar amb el mateix pipeline de finestres temporals i etiquetes *multi-label* utilitzat per la CNN. Això va permetre comparar els dos models sota condicions equivalents. La diferència principal entre arquitectures no es trobava, per tant, en les dades, sinó en la manera de processar-les i d'actualitzar els pesos: la CNN utilitzava convolucions i *backpropagation*, mentre que la SNN utilitzava dinàmica recurrent de neurones de polsos i aprenentatge local amb e-prop.