POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Gestionale

Tesi di Laurea Magistrale

IMPLEMENTAZIONE DI DEEP LEARNING SU SISTEMI EMBEDDED: THE ALOHA EXPERIENCE

Relatore
prof. Emilio Paolucci

Tutor aziendale
Cristina Chesta

Candidato
Vincenzo Roma

Aprile 2021

Sommario

L'Intelligenza Artificiale è avanzata al punto di poter trasformare la maggior parte dei settori negli anni a venire e tra i diversi approcci, certamente il Deep Learning rappresenta uno dei più promettenti. Infatti, esso ha già dimostrato la sua efficacia in numerose applicazioni come la classificazione delle immagini e il riconoscimento vocale. Tuttavia, l'implementazione di queste applicazioni su sistemi embedded richiede la soddisfazione di un importante trade-off: ottenere un'elevata precisione nonostante le limitate risorse energetiche e computazionali. L'approccio tradizionale per assolvere tale compito prevede due distinte fasi che consistono, nello sperimentare diverse configurazioni di rete fino ad ottenere un modello che incontri gli obiettivi di qualità prefissati, e nel cercare di ottimizzarlo rispetto alla specifica architettura target. In questo modo l'intero flusso di lavoro potrebbe richiedere mesi di codifica, compromettendo significativamente la produttività a seguito di continui rework, estenuanti fasi di tuning e sovraccarico di lavoro per i membri del team. Il lavoro di tesi, svolto presso Concept Reply, si pone come obiettivo: da un lato, l'ideazione e successiva implementazione di un caso studio volto a dimostrare i benefici derivanti dall'adozione di ALOHA, toolflow di automazione del processo di design del modello, mappatura ottimale rispetto all'hardware di riferimento e generazione del codice per il successivo porting; dall'altro, l'espletamento di un'analisi di mercato volta a determinare un potenziale target di clienti e fornire consapevolezza circa il contesto competitivo.

Ringraziamenti

Dedico la presente pagina dell'elaborato per esprimere la mia gratitudine nei confronti delle persone che mi hanno supportato nella redazione dello stesso e durante questa fantastica permanenza in un luogo magico chiamato Politecnico di Torino.

A Cristina Chesta, mio tutor aziendale, e Luca Rinelli, mio collega, per la solidarietà e disponibilità mostrate nel corso del tirocinio formativo e del lavoro di tesi.

Al mio relatore prof. Emilio Paolucci, grazie alle cui critiche costruttive ho appreso grandi insegnamenti e grazie ai cui suggerimenti sono riuscito a redigere un elaborato che sento essere maggiormente "mio".

Alla mia famiglia, che con straordinario spirito di sacrificio ha reso possibile il conseguimento di questo importante traguardo e alla quale vanno i meriti della persona che oggi rappresento. In particolare, a nonna Lena e alle sue infinite preghiere.

Ai miei fratelli per scelta, Antonio, mio mentore, e Vito, sempre pronto a darmi conforto.

Ai miei amici più cari, storici e recenti. In particolare, ad Ari e Cicciuzzu, presenti dal giorno della mia immatricolazione, a Thomas e Marco, coinquilini più unici che rari, ai miei marsicani, a "quei bravi ragazzi".

Al mio alter ego me medesimo, la cui ambizione ha superato di gran lunga il suo talento.

Lascio infine questo messaggio in eredità al me futuro:

"Without committment you will never start but more importantly without consistency you will never finish. It's not easy so, keep working, keep striving, never give up, fall down seven times get up eight. Ease is a greater threat to progress than hardship so, keep moving, keep growing, keep learning, see you at work."

Indice

Capitolo 1

Introduzione

Andrew Ng, co-founder di Coursera, professore aggiunto a Stanford e founder del Google Brain Deep Learning project ha affermato: “AI is the new electricity”; ha poi proseguito dicendo: “About 100 years ago, electricity transformed every major industry. AI has advanced to the point where it has the power to transform every major sector in coming years” (Urlini et al. 2019). Tra i vari possibili approcci all’intelligenza artificiale, certamente quello del Deep Learning rappresenta in assoluto uno dei più promettenti avendo già dimostrato di poter assolvere egregiamente numerosi compiti, specie negli ambiti del riconoscimento vocale e della classificazione di immagini. Tuttavia, progettare simili applicazioni avendo cura di ottenere un calcolo efficiente “on the edge” rappresenta ancora oggi un’ardua sfida.

La scelta di questo argomento di tesi risiede nella curiosità e naturale propensione ad operare in contesti altamente innovativi. È opinione diffusa e condivisa che l’intelligenza artificiale interesserà molte delle professioni del futuro, dal momento che la digital transformation di interi settori industriali è in atto e l’interazione uomo-macchina è ormai una relazione immanente e con la quale occorre raffrontarsi. Inoltre, se in accordo con una delle previsioni di Gartner i developers rappresentano il cuore pulsante e la maggiore forza in gioco nel campo dell’intelligenza artificiale (Goasduff 2020), tuttavia, anche l’ingegnere gestionale, storicamente figura di raccordo tra gli aspetti più tecnici e quelli legati al business, che debba interfacciarsi con tali tematiche non può che trarre beneficio dal conoscere le dinamiche che si celano all’interno della “black box” chiamata AI. Come la storia insegna, l’innovazione tecnologica non è mai neutrale; laddove introduca benefici potrebbe indurre dei possibili mis-use. Ecco che l’alfabetizzazione e la disseminazione assumono dunque un aspetto rilevante, anzi essenziale, per cogliere tutto il potenziale che la tecnologia ha da offrire, limitandone eventuali effetti indesiderati.

La tesi è stata svolta presso Concept Reply, società di consulenza informatica che si occupa di supportare i propri clienti attraverso lo sviluppo di soluzioni end-to-end relativi ad Internet of Things e Cloud Computing. In particolare, il lavoro si colloca all'interno di un progetto internazionale finanziato dalla comunità europea. Denominato ALOHA[1], esso coinvolge 14 partners sia accademici che industriali distribuiti in 7 differenti Paesi e, l'oggetto prevede la realizzazione di una toolflow volta a facilitare l'implementazione di algoritmi di Deep Learning su piattaforme eterogenee dotate di limitate risorse energetiche e computazionali.

Sia questo il contesto di riferimento, il lavoro di tesi si pone un duplice obiettivo: da un lato, l'espletamento di un'analisi di mercato volta a determinare un potenziale target di clienti e fornire consapevolezza circa il contesto competitivo; dall'altro, l'ideazione e successiva implementazione di un caso studio volto a dimostrare i benefici derivanti dall'adozione di ALOHA, toolflow di automazione del processo di costruzione, addestramento e distribuzione su hardware di modelli di Deep Learning.

L'elaborato di tesi è stato strutturato in modo tale da coinvolgere il lettore ed esporre nel modo più efficace possibile un tema assai complesso quanto affascinante al contempo. Per tali ragioni, nel primo capitolo si è reso necessario fare chiarezza su ciò che il Deep Learning rappresenta, contrapponendolo a concetti come Artificial Intelligence e Machine Learning coi quali sovente viene confuso. Successivamente, è stata introdotta una breve panoramica sull'evoluzione storica. Dopodiché sono stati riassunti in pochi fattori chiave, le ragioni in base alle quali è possibile desumere l'esplosione di interesse a cui si sta assistendo di recente. Nell'intento di fornire un po' di contesto, si è poi analizzato l'ecosistema che ruota attorno all'intelligenza artificiale. Infine, dello spazio è stato riservato alla discussione delle prospettive future, allo scopo di fornire degli spunti di riflessione.

Il capitolo secondo apre con la definizione del problema che concerne l'implementazione di Deep Learning su sistemi embedded ed introduce preliminarmente la tecnologia ALOHA. In seguito, contrapponendo il flusso di lavoro in ALOHA a quello tradizionale, si è cercato di enfatizzare le criticità a quest'ultimo connesse. Dopodiché, si è ritenuto opportuno porre la lente di ingrandimento su ALOHA, commentando una breve parte di ingegneria del software. A tal proposito, in prima battuta sono stati introdotti e discussi i vari tools costituenti ed il modo in cui questi interagiscono tra di essi; successivamente invece si è cercato di emulare l'esperienza utente al fine di fornire una visione complessiva ed intuitiva del prodotto e del suo funzionamento. Successivamente sono stati riportati gli insight frutto delle analisi di mercato condotte e che hanno consentito di valutare il posizionamento di ALOHA, sia dal punto di vista di potenziali segmenti in termini di tipologia, settore ed area geografica, a cui rivolgere la propria offerta commerciale, che da quello dei principali competitors coi quali contendersi il mercato.

---

[1] https://www.aloha-h2020.eu/

Il terzo capitolo verte sull'ideazione e successiva implementazione del caso studio "KeyWord Spotting", volto ad effettuare un assessment della toolflow e dimostrare i benefici che derivano a seguito della sua adozione. Per prima cosa si è deciso di trattare due temi quali: cos'è il keyword spotting e il perché di una tale scelta come dominio di riferimento. In particolare, essendo l'input di tipo audio peculiare per queste tipologie di applicazioni, si è ritenuto opportuno fornire un breve cenno sul background teorico che ad esso compete. Proseguendo, si è declinato ulteriormente l'obiettivo che s'intende raggiungere attraverso il caso studio. A tal proposito, è stato formalizzato un opportuno set di KPI's i quali fungeranno da parametri di controllo per l'esperimento e consentiranno di validare i vantaggi che scaturiscono dall'integrazione di ALOHA nel processo di design. Chiarito il perché lo si fa si è discusso del come, riportando la metodologia di cui ci si è dotati per far fronte a tale intento. Specificatamente, è stato definito il punto di partenza ovvero il tipo di applicazione, è stato introdotto il dispositivo target scelto ed è stata presentata la strategia con la quale approcciare alle attività di sviluppo, ivi compresa la sequenza dei passi da seguire. Infine, le principali evidenze empiriche derivanti dall'espletamento del caso studio sono state riportate e discusse.

1.1 Premessa

Prima di fornire una panoramica generale sul Deep Learning, esplorando brevemente l'evoluzione che lo ha interessato sino al come oggi esso è concepito, è doveroso far luce su una consuetudine oramai radicata nell'accezione comune del termine, vale a dire la sua interscambiabilità coi termini Artificial Intelligence e Machine Learning, d'ora in avanti indicati coi rispettivi abbreviativi AI e ML. Sebbene questi siano molto spesso pensati come sinonimi, in realtà costituiscono degli "oggetti" molto differenti tra loro. Volendo utilizzare una metafora, l'AI rappresenta nel concetto di matrioska la madre del seme DL. L'intuizione, sottostante a tale asserzione, è facilmente riconoscibile in figura 1.

Figura 1: La matrioska dell'AI

Giunti a tale premessa, è possibile fornire una definizione di ciò che ognuno di questi termini rappresenta.

Sin dagli albori della nascita dei computer, ci si è chiesti se mai questi avessero potuto un giorno pensare al pari di una mente umana. Il funzionamento cardine di tali dispositivi prevedeva la risoluzione di problemi complessi per l'uomo ma facili da formalizzare attraverso regole ben precise. La vera sfida è sorta quando problemi cognitivamente semplici ma di difficile interpretazione, hanno messo a dura prova l'abilità delle macchine. Di conseguenza, il significato dell'intelligenza artificiale si rifà alla capacità di un sistema di eseguire funzioni cognitive che tipicamente vengono associate alla mente umana come l'apprendimento, la percezione, l'interazione col circostante, il ragionamento.

Il Machine Learning rappresenta una delle tecniche dell'AI e fa sì che un sistema, sia in grado di apprendere da sé anziché ricevere istruzioni di programmazione esplicite. In sostanza, gli algoritmi di ML attraverso il processamento di dati riescono ad estrarre dei patterns grazie ai quali anticipare ciò che accadrà oppure fornire delle raccomandazioni a supporto del processo decisionale, altresì noti rispettivamente come predizioni e prescrizioni.

Quanto al Deep Learning invece, esso rappresenta un approccio all'AI e consta di un sottoinsieme del ML all'interno del quale, attraverso una combinazione di tecniche avanzate di training e componenti architetturali di reti neurali, si cerca di mimare il modo in cui il cervello umano opera. In sostanza, le architetture di DL si compongono di diversi layers a seguito dei quali vengono provocati determinate proprietà e comportamenti. Dando origine così ad una rete neurale costituita da molti strati interconnessi tra loro e per questo detta profonda, con un approccio di tipo gerarchico, quindi a livelli, rappresentazioni più complesse vengono costruite ed apprese in relazione ad altre più semplici. In questo modo, una rete neurale processando un'ingente quantità di dati è in grado di acquisire esperienza, comprendere ad esempio l'aspetto di un oggetto e riconoscerlo in un nuovo contesto. A tal proposito vale la pena notare che, proprio la classificazione delle immagini rappresenta una delle applicazioni a cui meglio si presta l'apprendimento profondo e nella quale ha già dimostrato dei netti miglioramenti rispetto ai metodi tradizionali alternativi. A scopo meramente esemplificativo, una generica rappresentazione è mostrata qui di seguito.

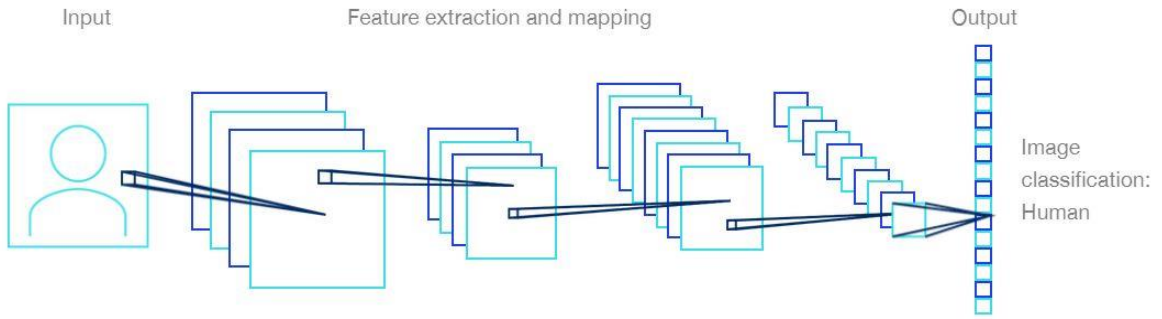


Figura 2: Esempio di classificazione attraverso una CNN[2]

[2]https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/an-executives-guide-to-ai#

1.2 Contesto storico

Sebbene il Deep Learning venga comunemente considerato come uno dei paradigmi emergenti della tanto citata industria 4.0, in realtà le sue radici risalgono al 1940. Le ragioni di tale percezione sono da associarsi alla relativa poca popolarità negli anni passati così come alla diversa connotazione assunta nel tempo, frutto dell'influenza dei diversi ricercatori e punti di vista. Fondamentalmente, tre sono state le correnti che in ordine cronologico vanno sotto il nome di cybernetics, connectionism e l'attuale Deep Learning.

La prima corrente, quella della cybernetics, ebbe origine nel 1940 e si protrasse sino al 1960 circa. È in questo periodo che ebbero luogo gli sviluppi relativi alla teoria dell'apprendimento biologico, basata dunque su una prospettiva neuroscientifica, e furono realizzati i primi modelli in grado di addestrare un unico neurone. Il meccanismo basale di questi modelli consta nell'apprendimento di un set di pesi $w_1, \ldots, w_n$ attraverso i quali poter associare ad un set di input $x_1, \ldots, x_n$ un output y mediante una funzione $f(\boldsymbol{x}, \boldsymbol{w}) = x_1 w_1 + \ldots + x_n w_n$. Tra questi, il perceptron sviluppato da Rosenblatt nel 1958 rappresenta il primo algoritmo di autoapprendimento mentre intorno al 1965, il matematico ucraino Alexey Grigorevich Ivakhnenko sviluppò il primo modello generale di apprendimento la cui forma evoca in maniera abbastanza fedele l'architettura delle moderne reti neurali. Si noti che tali modelli vanno sotto il nome più generale di modelli lineari.

A partire dal 1980 seguì l'ondata del connectionism, il cui contesto di riferimento fu quello delle scienze cognitive e l'idea di fondo che, un significativo numero di piccole unità computazionali potessero dare origine ad un comportamento intelligente dal momento in cui fossero messe in contatto le une con le altre formando una rete. I due principali contributi riconducibili a tale corrente furono la cosiddetta rappresentazione distribuita e il successo della backpropagation. Con la prima ci si rifà al concetto secondo il quale, ciascun input debba essere rappresentato da caratteristiche elementari disaccoppiate, ognuna delle quali possa a sua volta essere applicata nella rappresentazione di altri input. Così, ad esempio i neuroni dediti al riconoscimento del colore potranno essere utilizzati a prescindere dalla categoria di oggetti introducendo una sostanziale forma di efficientamento. Con la backpropagation invece, s'intende quell'algoritmo volto ad allenare reti neurali profonde ovvero caratterizzate da più layer nascosti, il quale rappresenta tutt'oggi l'approccio dominante. Tuttavia, le ricerche connesse a tale movimento continuarono fino alla prima metà degli anni Novanta cui ben presto seguì il sorgere di una fase di declino come si evince in figura 3.

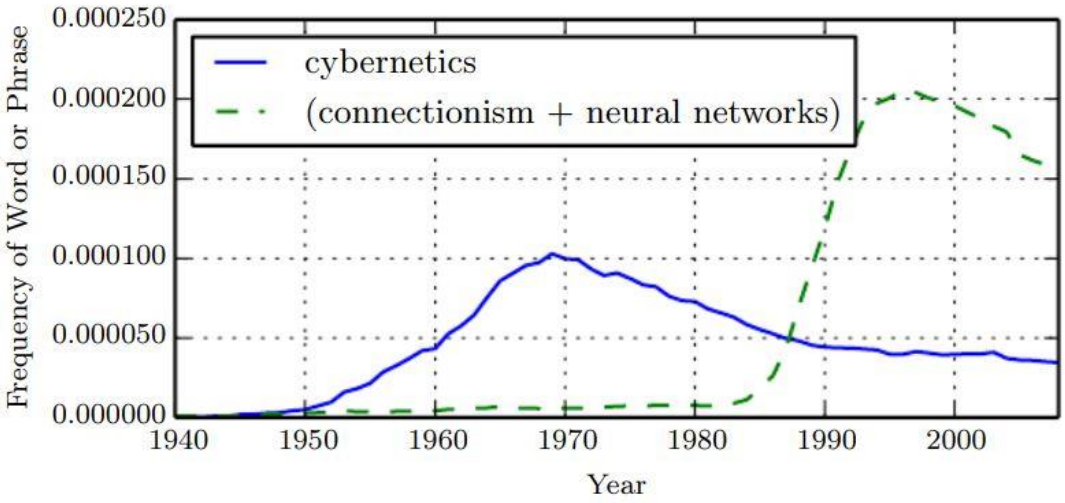


Figura 3: Correnti passate a confronto[3]

Le ragioni di un simile effetto, ovvero la disillusione delle aspettative degli investitori, sono da imputarsi molto probabilmente al mismatching tra i risultati ottenuti e quanto promesso nell'intento di suscitare interesse e accaparrare risorse finanziarie durante le campagne per la raccolta fondi da devolvere alla ricerca.

Nonostante le difficoltà riscontrate all'epoca nell'addestrare reti neurali così profonde, col senno di poi chiaramente connesse anche alle limitazioni indotte dalle tecnologie disponibili in quel momento, oggi è possibile affermare e riconoscere quanto gli stessi algoritmi di quegli anni funzionino abbastanza bene. Semplicemente, l'ecosistema che ruotava attorno al tema non era sufficientemente maturo da estrarre un sostanziale valore. Il punto di svolta avvenne nel 2006. È a questa data che si riconduce il Deep Learning così come viene inteso oggi e il cui merito per aver coniato il termine è da molti attribuito a Geoffrey Hinton il quale, attraverso una nuova modalità di pretrain, stimolò nuovamente l'utilizzo di modelli di apprendimento profondo rendendoli popolari in tutto il mondo. Nella prossima sezione, verranno introdotte e discusse le ragioni che hanno permesso la transizione dall'hype alla realtà e ai quali si deve dunque il successo cui si sta assistendo oggi.

[3]https://www.deeplearningbook.org/

1.3 Drivers del cambiamento

Dave Coplin, Microsoft's chief envisioning officer, considera l'intelligenza artificiale come "the most important technology that anybody on the planet is working on today." (Urlini et al. 2019). Tra i vari approcci all'intelligenza artificiale, sicuramente quello del Deep Learning rappresenta uno dei più promettenti. Ma a cosa è imputabile la recente esplosione di interesse a cui si sta assistendo e la tendenza ad etichettarlo come un qualcosa di innovativo?

Negli ultimi anni sono stati condotti enormi passi avanti che hanno portato i maggiori leaders e thinkers di settore a ritenerlo in assoluto uno dei più promettenti trend del futuro. Volendo racchiudere in pochi concetti chiave le ragioni di un tale successo, non si può prescindere dal menzionare la fortunata convergenza tra i progressi sugli algoritmi, la proliferazione dei dati e la crescita esponenziale di capacità di calcolo e spazio di archiviazione.

Figura 4: Fattori chiave

Il tema relativo agli avanzamenti sugli algoritmi e l'ottenimento di infrastrutture adatte, che hanno fatto sì che le dimensioni dei modelli crescessero notevolmente, seppur ad alto livello è stato presentato e discusso in precedenza. Volendo fornire un breve richiamo, sostanzialmente le reti neurali sono passate dai primi perceptrons con un singolo strato ed un singolo neurone, ad architetture molto articolate caratterizzate da più livelli e talvolta da connessioni ricorrenti. Tali modelli costituiscono oggi un ampio range di grafici computazionali e topologie adatti a prestarsi a diverse applicazioni. A sottolineare il livello di progresso raggiunto nel corso degli anni, si pensi che la dimensione delle reti è esplosa di un fattore 3000X superando così i 170 miliardi di parametri (Yuan 2020). A questo punto si vuole dunque porre l'attenzione sui nuovi fattori appena emersi, vale a dire l'importanza dei dati e l'enorme crescita circa le capacità di storage oltre che le abilità computazionali.

Relativamente al tema dei dati, tutto ebbe inizio il 1991 quando il CERN decise di aprire al pubblico il web. Decisamente importante fu la nascita, nel 2004, del cosiddetto web 2.0 che segnò il passaggio dalla visualizzazione passiva alla creazione e fruizione di contenuti interattivi e collaborativi, aprendo le porte ai dati generati direttamente da ciascun individuo. Solo un anno dopo si contavano utenti in numero superiore ad un miliardo ma, il vero punto di svolta si ebbe nel 2007 con l'avvento dello smartphone; il resto è storia. Oggi, il numero di dispositivi mobile sovrasta quello dell'intera popolazione mondiale e si stima che ciascun utente, attraverso l'utilizzo di dispositivi elettronici, produca circa 2,5 quintilioni di byte al giorno, facendo sì che il 90% dei dati mondiali sia stato pressoché generato negli ultimi due anni (Chui et al. 2020). Una così ingente quantità di dati immessa in rete e resa disponibile ha determinato la nascita del noto paradigma dei big data, abilitando un approccio data driven stante ad indicare l'attitudine a basare i processi decisionali su dati oggettivi anziché fare affidamento sulle intuizioni personali e soggettive. La diretta implicazione col mondo del Deep Learning sussiste nella netta riduzione delle abilità necessarie ad ottenere delle buone prestazioni a fronte di un significativo aumento dei dati disponibili. In sostanza, ragionando puramente in termini statistici, l'accesso a così tanti dati ha facilitato di gran lunga l'addestramento di reti neurali profonde, superando quello che rappresentava una delle maggiori difficoltà di un tempo, ovvero il riuscire a generalizzare bene l'apprendimento a partire però da un set di dati davvero piccolo. Per chiarezza espositiva e per enfatizzare l'intensità della crescita, qui di seguito si riporta un benchmark tra alcuni dei più noti dataset.

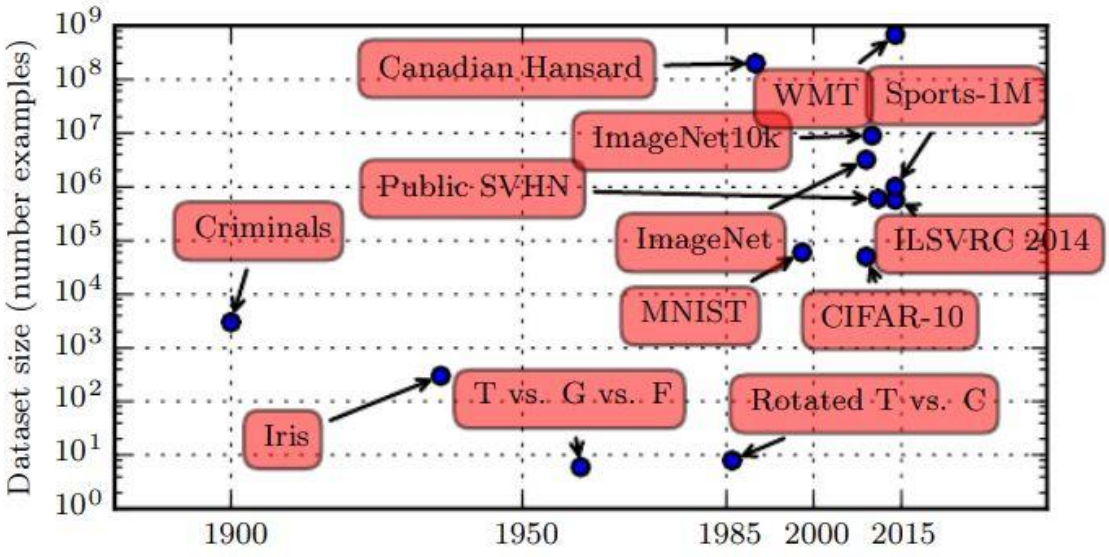


Figura 5: Benchmark sulla dimensione dei dataset nel tempo[4]

[4]https://www.deeplearningbook.org/

Unitamente ai dati, gli altri fattori abilitanti la tecnologia sono stati il continuo aumento delle capacità di calcolo e della memoria disponibile. Potendo far leva su tali risorse e consentendo così di eseguire reti di dimensioni sempre maggiori, compiti più complessi sono stati resi accessibili, peraltro con un elevato grado di precisione. In particolare, tra le principali invenzioni vale la pena annoverare il Cloud e l'unità di elaborazione grafica siglata come GPU.

Circa quest'ultima, introdotta sul mercato nel 1999, se ne intuì l'utilità nel mondo dell'apprendimento profondo e che ben si prestasse a certe tipologie di applicazioni solamente nel 2009 quando, Andrew Ng, ingegnere informatico statunitense nonché attuale professore presso l'università di Stanford, rivelò il suo potenziale dimostrando che per reti con più di 100 milioni di parametri, il training su GPU rendeva possibile una riduzione di 70X nelle tempistiche rispetto alla tradizionale CPU (Chui et al. 2020).

D'altro canto, altrettanto importante è stata la nascita del Cloud computing. Per la prima volta nella storia, l'utilizzo di sistemi IT altamente performanti e solitamente presenti nelle grandi aziende, è stato esteso e reso accessibile a milioni di persone grazie al Cloud. La prima società ad offrire spazio di archiviazione e capacità di calcolo, inoltre a costi accessibili, è stata Amazon nel 2002. A questa, altri player come Microsoft e Google si sono susseguiti. Si pensi che il costo medio per un gigabyte di archiviazione su disco, il quale inizialmente si attestava intorno 280 dollari, ha subito una rapida discesa sino a toccare un prezzo pari a poco meno di 1 dollaro. Analogamente, considerazioni simili valgono per la capacità di calcolo (Chui et al. 2020).

Alla luce delle considerazioni fatte, riassumendo, le determinanti del cambiamento e le sinergie che si sono venute ad instaurare tra queste hanno dato origine ad un circolo virtuoso, facendo sì che gli algoritmi potessero disporre di quanto necessitassero ai fini del raggiungimento del successo e conseguentemente per la generazione di valore.

1.4 Ecosistema oggi

Dopo aver introdotto e brevemente discusso le motivazioni che spingono a percepire l'apprendimento profondo come un qualcosa di innovativo, si vuole a questo punto cercare di fornire una visione olistica del medesimo in senso lato, andando cioè a considerare tutto ciò che ruota attorno al mondo dell'intelligenza artificiale. Nel far fronte a tale intento, per facilitare la fruizione dei contenuti sono state individuate quattro aree quali ricerca, investimenti, persone e strumenti, ciascuna delle quali sarà analizzata minutamente.

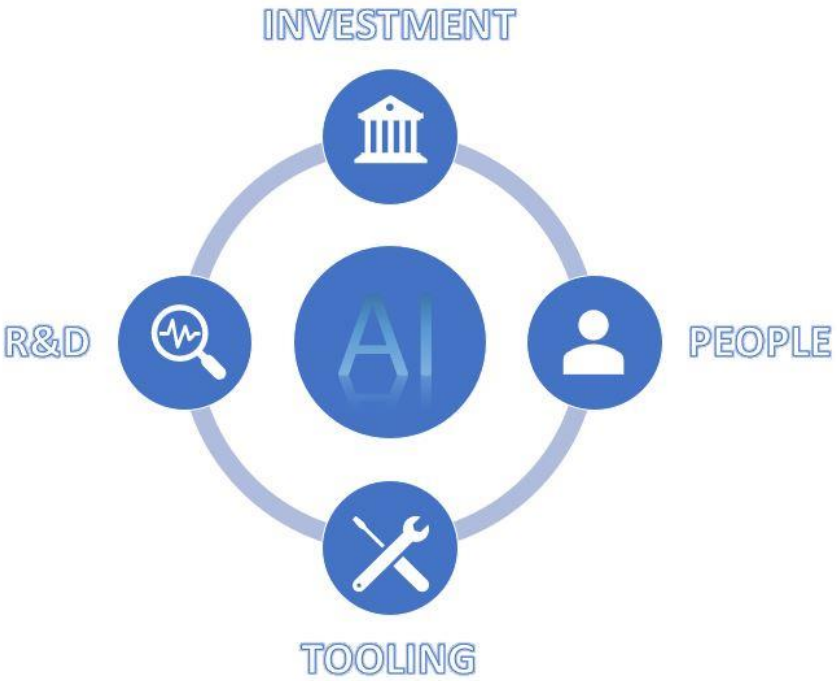


Figura 6: L'ecosistema dell'AI

1.4.1 Ricerca e sviluppo

Sin da quando vi è memoria, l'R&D ha costituito una fondamentale guida per la creazione di una qualsivoglia innovazione, sia essa radicale o incrementale, indipendentemente dal dominio di riferimento. L'intelligenza artificiale oggi, costituisce un campo di ricerca estremamente fertile e in continuo movimento. Numerosi sono i dati a sostegno di una tale asserzione. Prescindendo dal voler mostrare un insieme esaustivo e onnicomprensivo della ricerca condotta, si riportano qui di seguito alcuni degli insights ritenuti particolarmente interessanti.

A livello globale, assumendo come orizzonte temporale il periodo di tempo tra il 1998 e il 2018 e prendendo in considerazione le pubblicazioni di papers sull'intelligenza artificiale rispetto al numero complessivo di pubblicazioni presenti all'interno di Elsevier's Scopus, il più grande database al mondo di abstract e citazioni, parrebbe che tale numero sia stato soggetto ad una crescita pari a circa il 300%. In particolare, tra le regioni che hanno contribuito in misura maggiore all'ottenimento del risultato vi è l'Europa, la quale negli ultimi anni è riuscita a produrre annualmente più di 20000 articoli, mentre la Cina rappresenta quella con il maggiore potenziale per gli anni a venire data la sostanziale crescita che ha subito nel corso degli anni (Perrault et al. 2019). Inoltre, vale la pena notare come sempre più, nell'ambito dell'AI, le collaborazioni tra aziende ed università stiano diventando pratica molto diffusa e consolidata, specialmente per Paesi come Stati Uniti e Cina.

Ponendo invece la lente d'ingrandimento sul Deep Learning, che come già accennato rappresenta uno dei più promettenti approcci all'AI, il numero di paper pubblicati sull'archivio arXiv è cresciuto per tutte le regioni, con il Nord America in testa alle altre. Una rappresentazione della cumulata tra il 2015 e il 2018 per singolo Paese è mostrata qui di seguito.

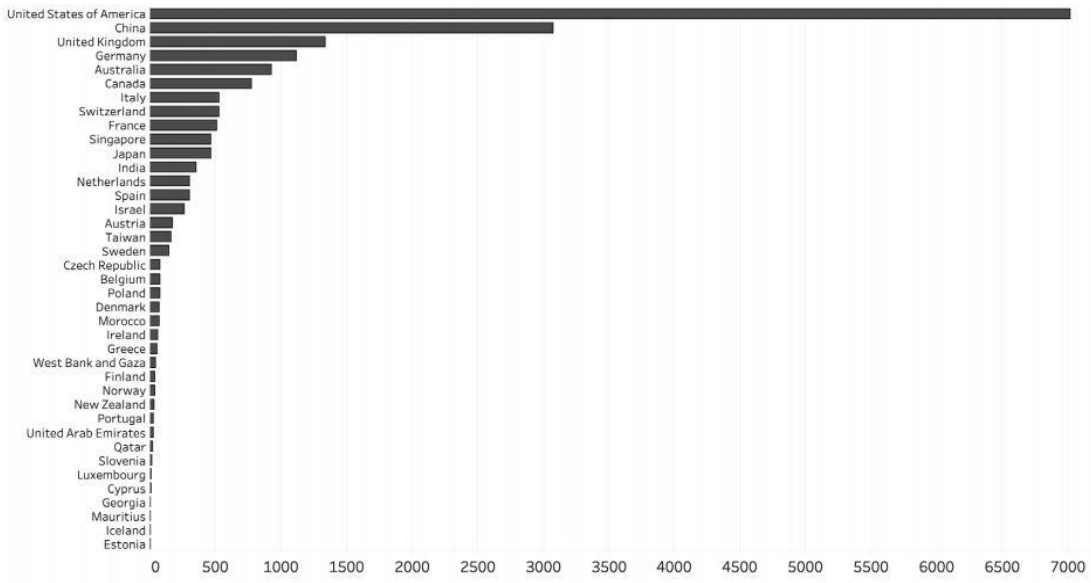


Figura 7: Ranking tra Paesi sull'attività di ricerca nell'AI[5]

[5]https://hai.stanford.edu/sites/default/files/ai_index_2019_report.pdf

1.4.2 Investimenti

Anche dal punto di vista delle decisioni di investimento, quello dell'intelligenza artificiale sembrerebbe essere un target di riferimento molto ambito e nel quale riporre fiducia. Solamente nel 2019 a livello globale sono stati destinati fondi per un ammontare superiore ai 70 miliardi di dollari, peraltro, somma che sembrerebbe destinata a crescere negli anni a venire.

Di questi, almeno un 40% rappresentano investimenti in startup, ciò a dire quanto cruciale e rilevante sia il ruolo di entità così peculiari per via dell'intrinseco carattere innovativo oltre che maggiore flessibilità che le caratterizzano. Relativamente con la stessa intensità sono state concluse operazioni di M&A. Una panoramica dei livelli di investimento negli anni, per ciascuna singola tipologia, è fornita qui di seguito.

Figura 8: Investimenti globali nell'AI per tipologia[6]

Se invece la classificazione viene effettua per area geografica, si nota come la maggior parte del flusso di denaro provenga da Paesi come Stati Uniti, Cina ed Europa mentre, normalizzando i dati rispetto alla dimensione del Paese, spiccano Israele e Singapore come realtà emergenti (Perrault et al. 2019).

[6]https://hai.stanford.edu/sites/default/files/ai_index_2019_report.pdf

1.4.3 Strumenti

Ovviamente, aspetto core dell'ecosistema è l'insieme degli "utensili da lavoro" che vengono adoperati per lo sviluppo di applicazioni basati su intelligenza artificiale. Il merito per una proliferazione così ricca spetta al concetto di open source. Con esso s'intende un approccio allo sviluppo e distribuzione del software secondo il quale, il codice sorgente debba essere reso pubblico e accessibile a chiunque detenga apposita licenza, abilitando la collaborazione tra più soggetti anche qualora siano particolarmente distanti tra loro. A tal proposito, si noti che sempre più, l'intera comunità del software dipende fortemente da tale metodologia, la quale è diventata oramai pratica consolidata nell'ambito del data science.

Volendo fornire una rappresentazione non totalitaria ma comprendente senz'altro i principali strumenti utilizzati nell'ambito del machine learning, si riporta qui di seguito una loro schematizzazione.

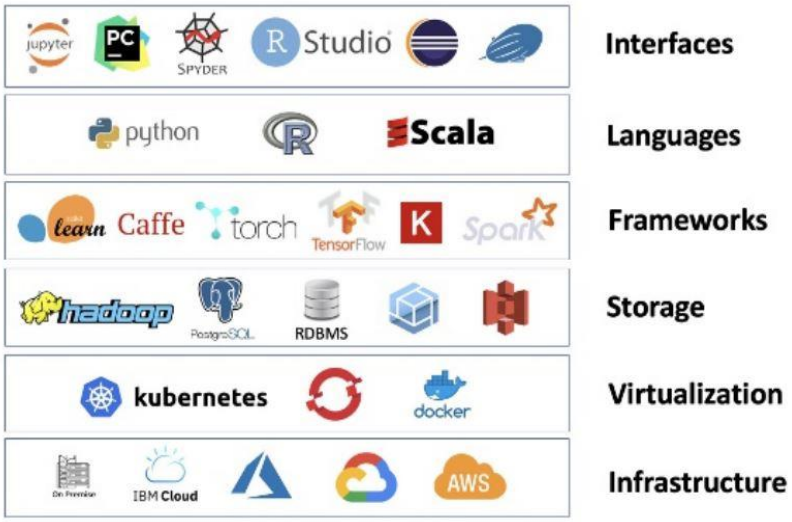


Figura 9: Tooling[7]

Poiché una trattazione completa di questi esula dallo scopo del lavoro di tesi, si ritiene opportuno fornire solamente un breve approfondimento su cosa un framework rappresenti e quale sia stata l'evoluzione verificatasi nel corso degli ultimi anni.

Un framework altro non è che una libreria, basata su uno specifico linguaggio di dominio e dotata di una ricca raccolta di moduli, attraverso il cui impiego sia possibile facilitare e rendere operativa la creazione di applicazioni di machine learning. Nel corso degli anni, accesa è stata la diatriba tra le soluzioni alternative proposte che si è conclusa con l'affermarsi di un duopolio. Infatti, a rappresentare circa il 95% dei casi d'uso sia nell'ambito della ricerca che del business sono proprio Tensorflow e Pytorch, realizzati rispettivamente da giganti tech quali Google e Facebook (Yuan 2020).

---

[7]https://higherlogicdownload.s3.amazonaws.com/IMWUC/ba913cda39b04d8185526a9f6ccedb3f/UploadedImages/9781492074953.pdf

1.4.4 Persone

Naturalmente, il cuore pulsante nonché elemento portante su cui regge l'intero paradigma tecnologico non può che essere la figura dell'uomo, grazie al cui ingegno ha reso realizzabile un tale progresso. Affinché sia possibile sviluppare simili applicazioni, ottenendo inoltre performance notevoli, non si può prescindere dall'aver maturato specifiche skills che consentano un uso appropriato della tecnologia. Sebbene lo specifico mix di competenze necessarie possa variare a seconda del singolo caso d'uso, in termini più generali la parola d'ordine è "diversity", ovvero dare origine a dei gruppi di lavoro nei quali, persone molto diverse tra loro per set di competenze e punti di vista collaborano. A tal proposito, un buon connubio dovrebbe prevedere:

- Competenze di dominio, che consentano di leggere e riportare i bisogni del mercato;
- Competenze matematiche e statistiche tipiche della figura del data scientist, attraverso le quali poter ricavare informazioni a partire dai dati;
- Competenze di programmazione, non potendo prescindere dallo sviluppo software;
- Competenze a livello di sistemi, essendo altrettanto importante tutto il tema relativo all'ottimizzazione hardware data l'esorbitante quantità di risorse computazionali richieste per questo tipo di applicazioni e i vincoli stringenti tipici di tali sistemi.

Uno schema riassuntivo è fornito qui di seguito.

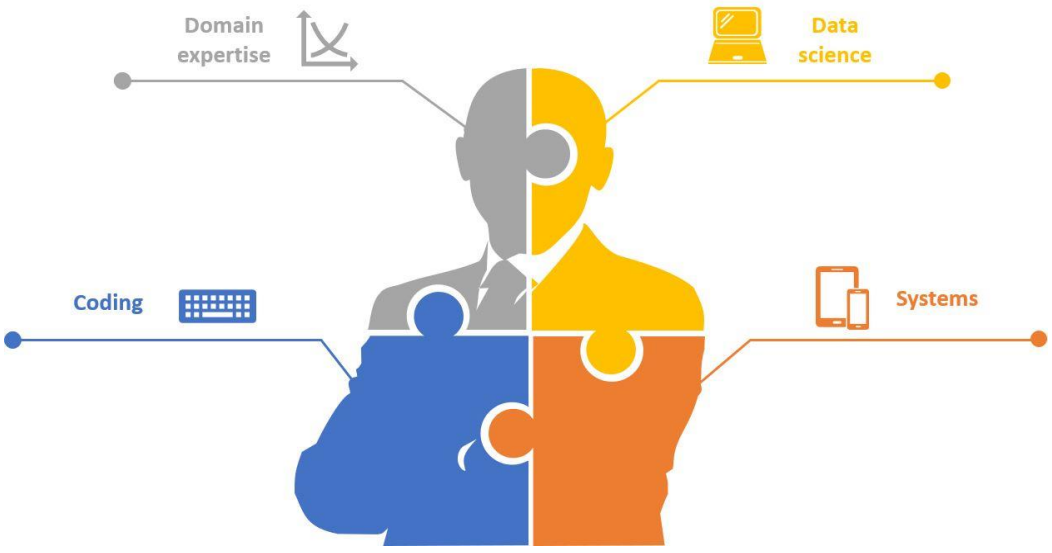


Figura 10: AI skills

Rivolgendo lo sguardo al mercato del lavoro, dai report annuali sull'artificial intelligence index, la crescita delle assunzioni osservata negli anni sembrerebbe destinata a persistere. Tra i settori di maggiore rilievo spiccano servizi hi-tech e manifatturiero. In risposta ad una simile tendenza, di recente l'interesse da parte degli studenti nei confronti dell'AI, inteso in senso lato, è cresciuto drasticamente. Così, dai numeri un'altrettanta crescita sembrerebbe interessare le iscrizioni a corsi in tema di AI e discipline correlate, sia tra le università che tra quelli erogati online da organizzazioni educative come Udacity e Coursera. Per citare un esempio, come mostrato in figura 11, il numero di iscrizioni ad un corso introduttivo sull'intelligenza artificiale tenutosi presso l'università di Stanford, ha subito una crescita pari a 5X rispetto al 2010.

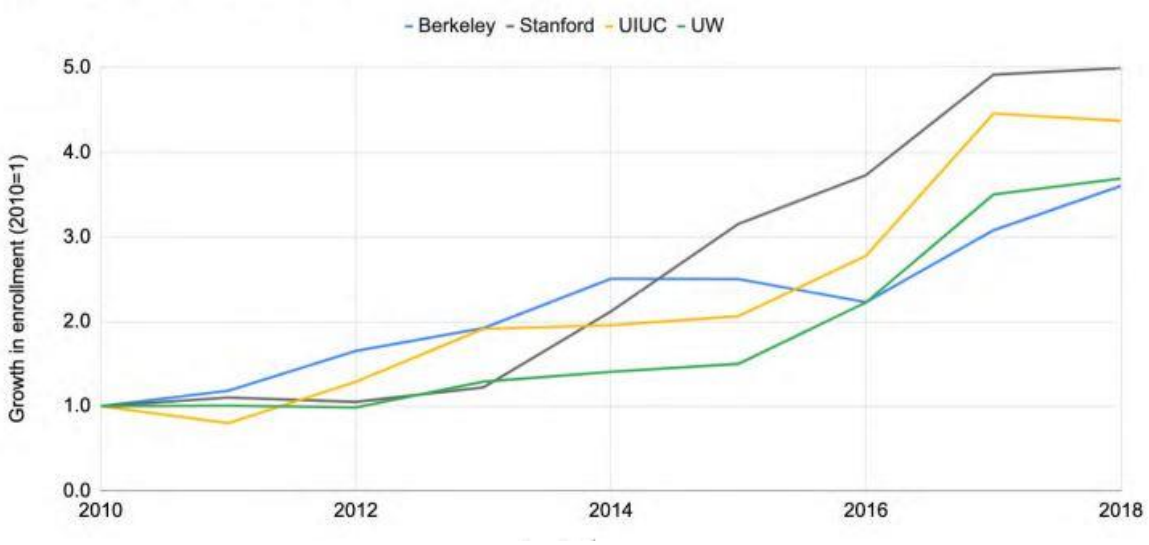


Figura 11: Crescita delle iscrizioni ad un corso introduttivo su AI in alcune Università[8]

[8]https://hai.stanford.edu/sites/default/files/ai_index_2019_report.pdf

1.5 Prospettive future

+14% sul prodotto interno lordo a livello globale, per un equivalente di 15.7 trilioni di dollari, di cui un +26% per la Cina, Paese con il maggior beneficio, è l'impatto stimato dell'AI per il 2030. Queste sono alcune delle proiezioni economiche riportate in uno studio condotto da PwC e che rendono l'AI probabilmente la più grande opportunità commerciale negli anni a venire (Rao et al. 2017).

Per utilizzare una metafora, quello dell'intelligenza artificiale rappresenta uno straordinario spazio costellato da innumerevoli applicazioni pratiche e campi di ricerca attivi. Essa si presta ad assolvere una miriade di task la cui utilità si esplica nell'industria così come nella routine quotidiana di ognuno di noi. Infatti, a far riflettere sulla portata rivoluzionaria del tema in questione vi è proprio l'eterogeneità dei settori coinvolti: robotica e automazione industriale, semiconduttori, automotive, healthcare e pharma, telecomunicazioni, retail, servizi finanziari, ecc.

Tra le tecnologie di punta, rileva la guida autonoma con circa il 10% degli investimenti privati a livello mondiale nell'anno 2018/19 (Perrault et al. 2019). Numerosi sono i prototipi che si stanno sviluppando in giro per il mondo e vivace è l'attività di regolamentazione da parte dei vari Paesi.

In accordo con uno studio condotto nel 2019 da McKinsey & Company e attraverso il quale sono state coinvolte 2360 aziende, il 58% degli intervistati ha affermato che la propria organizzazione impiega l'intelligenza artificiale in almeno una funzione aziendale o business unit. Questo, a conferma di un andamento più generale circa la crescita del tasso di adozione dell'AI da parte delle diverse società presenti nel mondo. Tra i benefici derivanti da tale scelta, i risultati riportati in figura 12 suggeriscono un sostanziale miglioramento delle performance che, a livello di singola funziona aziendale, si esplicano principalmente nella riduzione dei costi legati al manufacturing e alla gestione della supply chain, così come nell'incremento delle entrate relative all'area marketing & sales. (Cam et al. 2019).

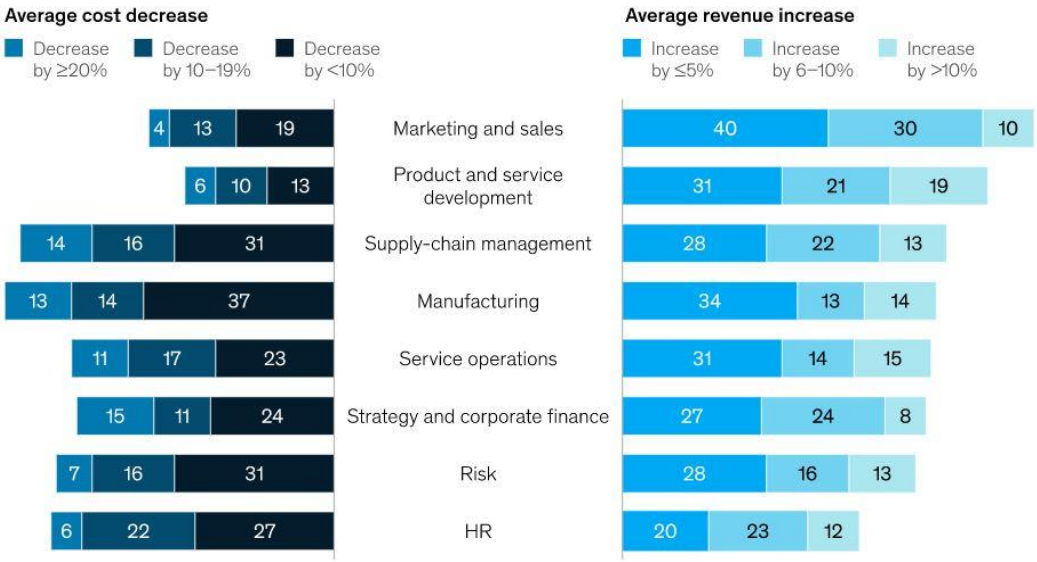


Figura 12: Benefici dall'AI per funzione aziendale *(% di rispondenti)*[9]

Tuttavia, nonostante una simile tendenza, da un progetto di ricerca realizzato di concerto tra il MIT Sloan Management Review e BCG, dai dati sembrerebbe emergere che solo l'11% delle imprese è in grado di conseguire sostanziali benefici finanziari a seguito degli sforzi effettuati (Ransbotham et al. 2020). A tal proposito, dalle indagini condotte allo scopo di desumere quali possano essere state le ragioni del successo ottenuto dalle poche organizzazioni ritenute leaders, e che dovrebbero dunque divenire spunto di riflessione per tutte le altre, sono state indentificate tre caratteristiche essenziali qui di seguito riportate:

- Facilitare l'apprendimento sistematico e continuo tra uomini e macchine, poiché è questa reciprocità a rappresentare il segreto per migliorare il processo decisionale e quindi garantire il successo;
- Sviluppare molteplici modalità attraverso le quali uomini e macchine possano interagire tra loro, perché a seconda del contesto un approccio sarebbe preferibile rispetto ad un altro. Così, in alcuni casi a fronte di raccomandazioni da parte del sistema, l'uomo dovrebbe decidere se e cosa implementare mentre in altri, dovrebbe essere l'uomo a generare soluzioni da sottoporre a valutazione da parte del sistema;
- Cambiare al fine di imparare ed imparare evolvendo. In altre parole, mutare i processi al fine di instaurare i presupposti corretti per lo sviluppo di un apprendimento di tipo organizzativo.

[9]https://www.mckinsey.com/featured-insights/artificial-intelligence/global-ai-survey-ai-proves-its-worth-but-few-scale-impact#

Capitolo 2

ALOHA toolflow

Nell'ambito delle intelligenze artificiali, senz'altro gli algoritmi di Deep Learning rappresentano uno strumento dal potenziale molto elevato. Essi, hanno già dimostrato la loro efficacia in un'ampia varietà di applicazioni come lo speech recognition o l'image classification. Inoltre, in alcuni specifici task cognitivi sono riusciti ad eguagliare, e talvolta perfino a superare, le capacità di noi essere umani.

Nel favorire però un'adozione capillare ed estendere la portata ad innumerevoli nuovi applicazioni e mercati, si rende necessario un cambio di paradigma "from Cloud to Edge". Infatti, si prevede che i dispositivi connessi, anche noti come IoT, raggiungeranno 1 trilione entro il 2035 (Liangzhen et al. 2018). Tali sistemi sono dotati di un certo numero di sensori in grado di rilevare un'ingente quantità di dati come audio, video, temperatura, localizzazione GPS ecc. che a loro volta vengono elaborati e condivisi ad altri nodi nel Cloud. Una simile crescita però introduce seri problemi di larghezza di banda, congestione della rete, latenza, privacy e sicurezza informatica. Così, il passaggio al cosiddetto edge computing, vale a dire un modello di calcolo distribuito ove l'elaborazione dei dati avviene in prossimità del luogo in cui essi vengono richiesti, rappresenta probabilmente la soluzione ottimale per far fronte a tali criticità.

Tuttavia, se da un lato la maggior parte delle ricerche così come dell'effort impiegato nell'implementazione di soluzioni basate su algoritmi di Deep Learning, si pongono come principale obiettivo quello di migliorare le performance in termini di accuratezza del modello di predizione e classificazione; dall'altro, questa tendenza che si esplica dunque nell'apporto di una maggiore complessità computazionale, finisce col gravare sul dispendio energetico dei dispositivi hardware. Se questo di per sé non rappresenta un vincolo così stringente in fase di training del modello, generalmente eseguito a mezzo di infrastrutture estremamente performanti, potrebbe diventarlo nel momento in cui le inferenze andrebbero eseguite su sistemi embedded caratterizzati da limitate risorse energetiche e computazionali.

Ecco che a questo punto pare ormai chiaro quale sia il trade-off in gioco. In accordo col 2nd postulato di Triz, sovente le soluzioni innovative emergono dalla risoluzione delle contraddizioni. Il superamento di quest'ultime rappresenta il meccanismo cardine dell'evoluzione tecnologica ed è proprio a questo che mira la toolflow ALOHA, sfruttando come principio risolutivo quello della soddisfazione del trade-off.

L’acronimo sta per “software framework for runtime-Adaptive and secure deep Learning On Heterogeneous Architectures”. L’idea nasce da una presa di coscienza secondo la quale, il bisogno di piattaforme embedded in grado di eseguire in modo efficiente, on the edge, complesse applicazioni basate sul Deep Learning è cresciuto notevolmente negli ultimi anni. Nonostante si disponga di una moltitudine di microprocessori, talvolta multi core, ottimizzare tali architetture eterogenee rendendole efficienti sotto il profilo energetico senza però compromettere l’accuratezza delle inferenze, rappresenta ancora oggi un’ardua sfida. Purtroppo, definito un certo problema e assegnato un opportuno set di dati, sia questo costruito ad hoc o estratto da piattaforme open source come Kaggle, UCI Machine Learning Repository, Google Dataset Search, ecc., la ricerca di un modello che soddisfi i requisiti di qualità richiesti e la successiva implementazione su un dispositivo embedded, richiede competenze molto elevate, non sempre disponibili o facilmente reperibili all’interno delle organizzazioni, oltre che significativi tempi, costi e sforzi. Così interviene ALOHA, toolflow che attraverso l’automazione di processo mira da un lato ad apportare efficienza riducendo time to market e costi associati allo sviluppo di tali applicazioni; dall’altro a sopperire alla mancanza o carenza di risorse dotate di competenze molto specifiche, essendo una tecnologia che si presta ad essere utilizzata anche da programmatori non familiari con i dettagli hardware e le strategie di ottimizzazione di determinate piattaforme, rappresentando in questo modo un concreto tentativo di “democratization of DL”.

2.1 Flussi di lavoro a confronto

Prima di passare in rassegna i vari tools che compongono ALOHA e come questi operano ed interagiscono tra di essi, si vuole dapprima analizzare con un maggior livello di dettaglio l'intero processo di data science e le relative fasi che lo compongono e, successivamente, porre a confronto i due approcci, tradizionale e automatizzato, che si possono adottare per questo tipo di attività. Le ragioni di tale scelta risiedono in una duplice motivazione: da un lato, fornire una panoramica ad alto livello del processo al fine di evidenziare quali sono le fasi interessate dall'automazione apportata dalla toolflow e rendere più agevole la comprensione del suo funzionamento; dall'altro, far emergere attraverso il confronto con il flusso di lavoro tradizionale la criticità ad esso imputabile e in che modo ALOHA è in grado di risolverla.

Pertanto, tipicamente un processo di data science può essere così schematizzato.

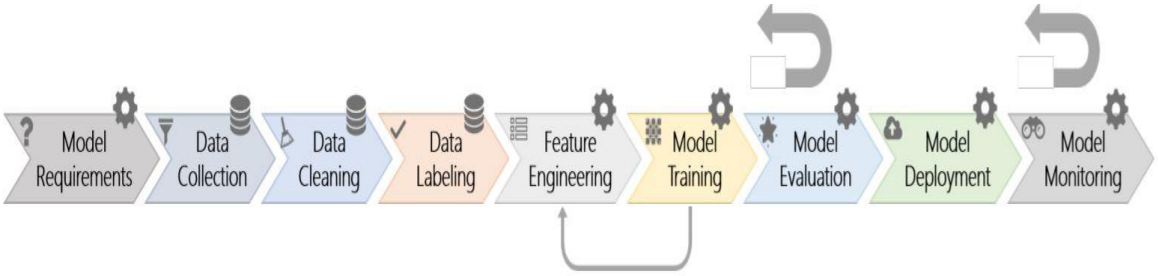


Figura 13: Ciclo di vita di un progetto di machine learning[10]

Come si nota in figura 13, esso è stato suddiviso in nove distinte fasi alcune delle quali orientate ai dati, altre ai modelli. Contrariamente a quanto si possa dedurre da una lettura superficiale, il flusso in realtà non risulta lineare bensì prevede numerosi feedback e iterazioni. Infatti, corposa è la sperimentazione necessaria a far convergere il processo verso una soluzione soddisfacente. Data la natura del problema che s'intende risolvere, il primo passo consiste nella definizione delle tipologie di modelli più adatti così come delle funzionalità che dovranno essere implementate e da essi supportate. A questo punto, seguono tutte le attività rivolte ai dati. La prima di queste consiste nella raccolta. Quale che sia l'origine, di dominio pubblico o privata, questi come tali non possono essere impiegati a causa del disordine insito all'interno del dataset. Essenziale, dunque diventa l'attività di preparazione degli stessi volta a garantire l'ottenimento di un dataset contenente dati "puliti" e utilizzabili. A tal proposito, numerose e più o meno sofisticate sono le tecniche per assolvere un tale compito ma il cui approfondimento esula dalla presente trattazione.

[10]https://www.microsoft.com/en-us/research/uploads/prod/2019/03/amershi-icse-2019_Software_Engineering_for_Machine_Learning.pdf

Una volta che i dati sono stati correttamente processati e organizzati, un'ulteriore attività preliminare è richiesta. Essa consiste nella partizione del dataset in tre sottoinsiemi rispettivamente indicati come training, validation e testing.

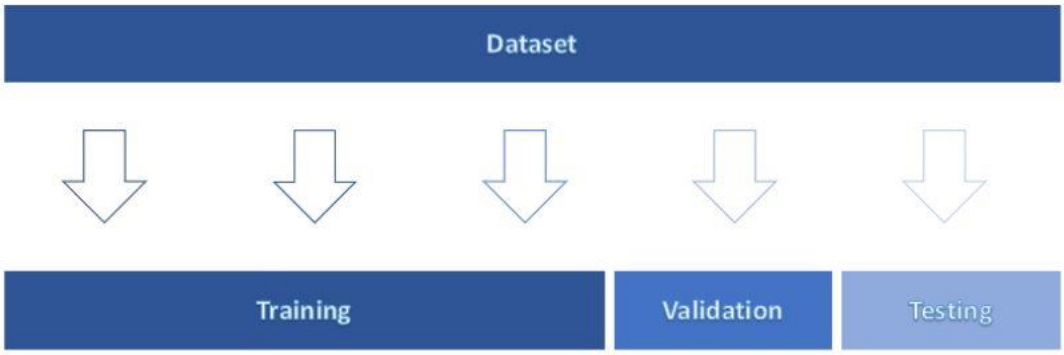


Figura 14: Partizione dataset

Nello specifico, durante la fase di training del modello saranno disponibili solo i due primi sottoinsiemi, vale a dire training e validation. Essi serviranno rispettivamente ad addestrarlo e a regolarne i parametri a seguito di una preliminare valutazione del medesimo sui cosiddetti dati invisibili, cioè non noti al sistema. Una volta che il modello è stato completamente specificato, allora si procederà ad una vera e propria fase di valutazione attraverso il sottoinsieme di testing, contenente al suo interno dati invisibili e in alcun modo disponibili durante la fase di training, allo scopo di verificare in concreto le abilità apprese.

Chiarita la prima parte del processo, quella cosiddetta "data-oriented", è possibile a questo punto spostarsi su quella relativa ai modelli, dove per inciso interviene la toolflow ALOHA. Fondamentalmente, con una logica di tipo ricorsiva diverse configurazioni di rete vengono esplorate e successivamente perfezionate, intervenendo sui cosiddetti hyperparameters, fino ad ottenere una soluzione che incontri le aspettative. In particolare, si noti che a guidare il processo decisionale tipicamente è proprio l'accuratezza del modello. Una volta che il modello ottimale è stato identificato, si passa alla sua implementazione sui dispositivi target e al successivo monitoraggio delle performance nel mondo reale al fine di risolvere eventuali peggioramenti che potrebbero verificarsi nel corso del tempo. Infatti, contrariamente al codice informatico, i modelli godono di una propria "vita" ed essendo soggetti a deterioramento necessitano di una periodica manutenzione che di fatto, si esplica in un retraining dello stesso.

Sintetizzata, in termini del tutto generici, la parte del processo di design relativa ai modelli, si vuole a questo punto fornire un maggior grado di profondità all'analisi al fine di far emergere la criticità insita nell'approccio tradizionale. Nel favorire tale intento, si consideri la seguente figura nella quale si riporta: nella parte sinistra, una schematizzazione del flusso di lavoro tradizionale, cioè con un approccio prettamente di tipo manuale; in quella destra, la logica di funzionamento della toolflow ALOHA che opera in modo completamente automatizzato.

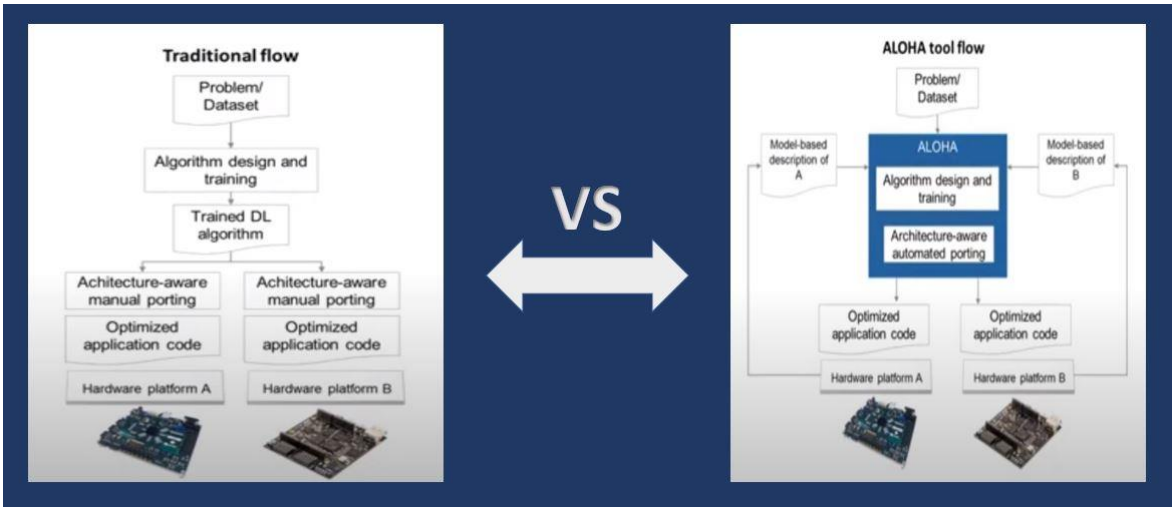


Figura 15: Flussi di lavoro a confronto

Come si nota dallo schema relativo al flusso tradizionale, solo in seguito al dispendioso processo di design e training del modello si inizia a tener conto del sistema embedded sul quale dovrà essere implementato, peraltro, attraverso un porting manuale che richiede elevate competenze oltre che un significativo effort. La criticità di un simile approccio risiede dunque proprio nella dicotomia tra le fasi di training ed inference, infatti, ponendosi durante la prima come unico obiettivo progettuale la massimizzazione dell'accuratezza e, non tenendo conto delle specifiche caratteristiche del sistema sul quale andrà effettuata l'inferenza, il rischio che si corre è quello di compromettere seriamente la fattibilità dell'applicazione o comunque limitarne notevolmente le potenzialità. Altresì, una simile metodologia mina significativamente la produttività dando origine a continui rework, estenuanti fasi di tuning e sovraccarico di lavoro per i membri del team.

Dal confronto tra i due flussi invece, si nota come l'utilizzo di ALOHA permetta di ovviare al problema, tenendo conto del dispositivo target sin dal principio e consentendo una selezione del modello consapevole, tenuto conto di alcuni vincoli in termini di performance ritenuti idonei per il tipo di applicazione che s'intende realizzare. Parimenti, la mappatura e il porting avverranno in modo automatico e complessivamente tempi e sforzi associati all'attività di sviluppo verranno significativamente abbattuti.

2.2 Descrizione prodotto

Se finora si è cercato di mantenere un approccio ad alto livello sulla tecnologia ALOHA, al fine di mettere in evidenza i tratti essenziali e poter cogliere il ruolo che essa ricopre senza però appesantire troppo la trattazione, in questa sezione invece, prevalente sarà la parte di ingegneria del software. In particolare, con un maggior grado di profondità verrà dapprima introdotto il framework e la concatenazione di tools e flussi tra questi, cui seguirà l'emulazione dell'esperienza utente volta a semplificare la comprensione del prodotto e presentare al contempo le principali funzionalità supportate.

2.2.1 Framework

Prima di fornire una descrizione dell'obiettivo generale di ciascuno step così come, in modo più dettagliato ed esaustivo, dei principi di funzionamento dei vari tools, una rappresentazione del framework è mostrata in figura 16.

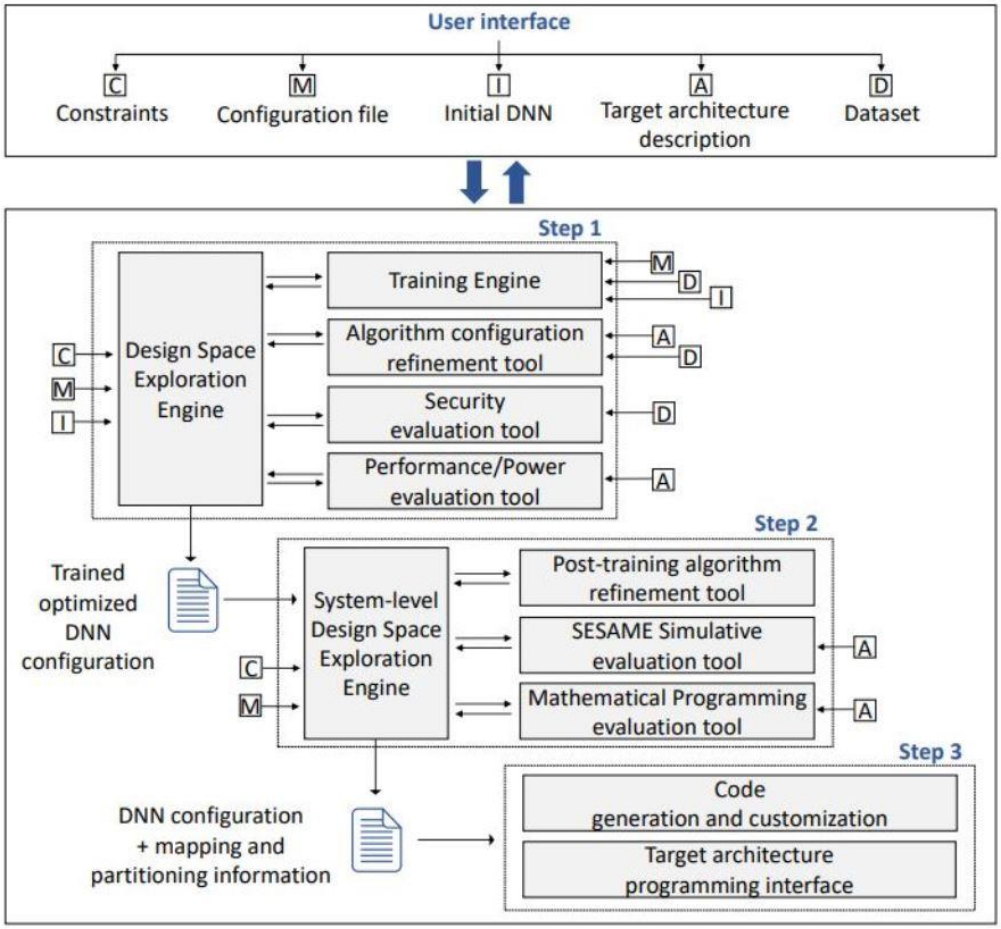


Figura 16: ALOHA Framework[11]

[11] https://dl.acm.org/doi/10.1145/3310273.3323435

Come si evince da quest'ultima, il flusso di lavoro è suddiviso in componenti più piccoli, indipendenti l'uno dall'altro ma che interagiscono e si influenzano vicendevolmente. La logica adottata è di tipo waterfall dove, l'output dello stadio a monte funge da input per quello a valle. Ad ognuna delle tre fasi individuate, è associato un macro-tool volto all'esecuzione della rispettiva. Il tutto viene armonizzato attraverso una piattaforma di orchestrazione mentre per la condivisione dei modelli di Deep Learning tra i vari componenti della toolflow, ci si rifà allo standard ONNx altrimenti detto Open Neural Network Exchange, il quale garantisce l'interoperabilità tra i vari frameworks, tools, ecc.

Affinché la toolflow possa dar luogo all'automazione del processo è necessario che tramite l'interfaccia utente vengano dapprima trasmessi come input:

- Una rete neurale iniziale, la quale risulta opzionale qualora si decida di ricorrere ad un algoritmo generico oppure si voglia sfruttare il gridsearch engine per generare lo spazio di progettazione che si vuole esplorare;
- Il dataset contenente i dati rispetto ai quali effettuare la fase di training;
- Un file contenente la descrizione dell'architettura target che dovrà operare nella fase di inferenza;
- Un file di configurazione con all'interno informazioni sull'applicazione target;
- Un set di vincoli che l'applicazione dovrà essere in grado di soddisfare, relativi a parametri quali consumo energetico, sicurezza, tempo di esecuzione, memoria occupata ed accuratezza.

Una volta che tali operazioni preliminari sono state opportunamente realizzate e le informazioni in input correttamente condivise, è possibile avviare il processo di design vero e proprio.

Lo step 1 si pone come obiettivo quello di progettare, addestrare e selezionare un algoritmo che sia compliant con lo specifico task, il set di vincoli e l'architettura target che dovrà eseguire l'inferenza. Per assolvere a tale compito si rende necessario l'utilizzo dei seguenti tools:

- DSE engine: guida il processo di ottimizzazione. A partire dalla configurazione di rete neurale fornita come input dall'utente o generandone una random da sé, esplora lo spazio di progettazione valutando i vari punti di design, ciascuno dei quali corrispondenti ad una data candidata configurazione dell'algoritmo. Per svolgere tale attività si avvale dell'intervento dei successivi tools in grado di performare delle valutazioni sotto determinati punti di vista;
- Training engine: rappresenta la principale utlity dedita all'attività di training in ALOHA. Supporta varie tecniche ed è in grado di operare sia partendo da zero che utilizzando un nuovo dataset per aggiornare i pesi di un modello già addestrato per altri scopi. Fornisce valutazioni sotto il profilo dell'accuratezza;
- Algorithm configuration refinement for parsimonius inference: attraverso tecniche di ottimizzazione come pruning e quantization, mira a ridurre l'effort computazionale associato all'esecuzione dei punti di progettazione;
- Security evaluation: come suggerito dal nome stesso, è in grado di determinare la sicurezza di ciascun modello candidato, quest'ultima intesa come resilienza agli attacchi espressa attraverso un livello tipicamente basso, medio o alto;
- Performance and power evaluation: considera come metriche da valutare la velocità nell'eseguire l'inferenza così come l'energia spesa per il funzionamento.

Lo step 2 consiste nel perseguire l'ottimizzazione del processo di inferenza sull'architettura target di destinazione. In particolare, la consapevolezza circa gli elementi di elaborazione presenti sulla piattaforma così come delle strutture di memorie ed interconnessioni disponibili, abilita la ricerca della migliore partizione in sub-tasks del modello iniziale e successivamente, la mappatura ottimale di quest'ultimi sulle diverse unità di elaborazione presenti sull'hardware target.
A tal proposito, vengono utilizzati i tools:

- System level DSE engine: in maniera del tutto similare al DSE engine esaminato allo step 1, esplora lo spazio di progettazione dove però in questo caso, i vari punti di design corrispondono a ben definiti schemi di partizionamento e mapping. Nel compiere ciò, sono richieste per ciascun punto candidato delle valutazioni, le quali vengono eseguite dai successivi tools satellite;
- Post-training algorithm configuration refinement for parsimonius inference: in base alle specifiche caratteristiche dell'architettura target, si preoccupa di effettuare degli step addizionali di riduzione del carico di lavoro qualora sia possibile oppure in alternativa, notifica al DSE engine di proseguire con il modello iniziale;
- Sesame: simulando per ciascun candidato lo scheduling delle varie operazioni da eseguire sulle differenti risorse, fornisce delle valutazioni in termini di tempo di esecuzione, consumo energetico, utilizzo dell'hardware ed eventuali conflitti;
- Mathematical programming evaluation: svolge un ruolo analogo al tool Sesame precedentemente discusso ma, anziché restringere lo spazio di ricerca ed effettuare una simulazione più mirata, utilizza un approccio analitico andando ad esplorare l'intero spazio di progettazione.

Infine, lo step 3 si occupa di completare il porting dell'applicazione sull'architettura target automatizzando la generazione del codice. Nello specifico, un'interfaccia di programmazione riceve come input le informazioni su partizionamento e mappatura ottimali derivanti dal precedente step, le quali vengono tradotte in adeguate calls alle primitive di elaborazione e comunicazione esposte dall'architettura. A questo punto, ove è possibile, lo stesso codice generato è a sua volta sottoposto ad ulteriori tecniche di ottimizzazione volte a ridurre i consumi energetici ed incrementare le performance.

2.2.2 Interfaccia utente

Che si tratti di un prodotto o servizio, la progettazione dell'interfaccia che separa quest'ultimo dall'utente rappresenta un punto cruciale. È dalla buona riuscita di quest'attività che scaturisce un'elevata affordance nell'interazione e di conseguenza una migliore soddisfazione del cliente. In quest'ottica, nel concepire l'interfaccia utente ci si è posti come principale obiettivo quello di garantire accessibilità e facilità d'uso anche nel caso in cui l'utente sia meno esperto e poco confidente con tali tematiche. Sostanzialmente, esso viene accompagnato durante tutto l'iter e guidato a partire dalla fase di definizione dell'esperimento; mostrandogli il work in progess, dato il ruolo cruciale che riveste il feedack di sistema circa la user experience ed infine, consentendogli di visualizzare a schermo, sotto forma di tabelle e grafici, i risultati emersi. Sia questo l'obiettivo principale, al pari della trattazione sul framework, seguirà un approfondimento nell'intento di completare la product overview.

Avviando ALOHA, aprendo l'interfaccia utente dal web ed effettuando il login, la home si presenta nel seguente modo.

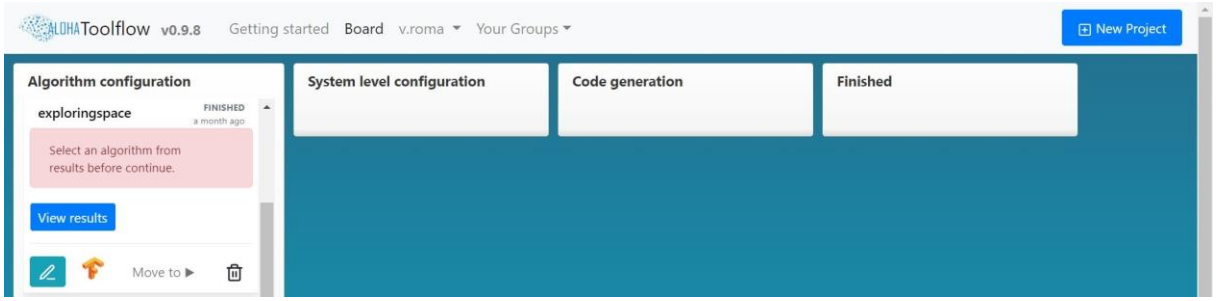


Figura 17: Home

Come si può notare, essa assume le vesti di un Kanban Board dove ciascun progetto viene rappresentato da una card che evidenzia, oltre che lo stadio al quale il progetto è giunto anche le informazioni più importanti. In particolare, sono tre le colonne dove ad ognuna delle quali è associata il rispettivo step di sviluppo del processo di design. Tipicamente il flusso avviene da sinistra verso destra sebbene, venga lasciata facoltà all'utente di retrocedere qualora voglia reiterare uno specifico step cambiandone le modalità. Conclusa l'ultima fase, l'esperimento può considerarsi terminato e l'applicazione generata può essere implementata nel mondo reale traendone i benefici auspicati.

Ci si concentri ora sulla creazione di un nuovo esperimento. Semplicemente cliccando sulla voce *new project* nell'angolo in alto a destra della home, si passa all'ambiente nel quale avviene la sua configurazione.

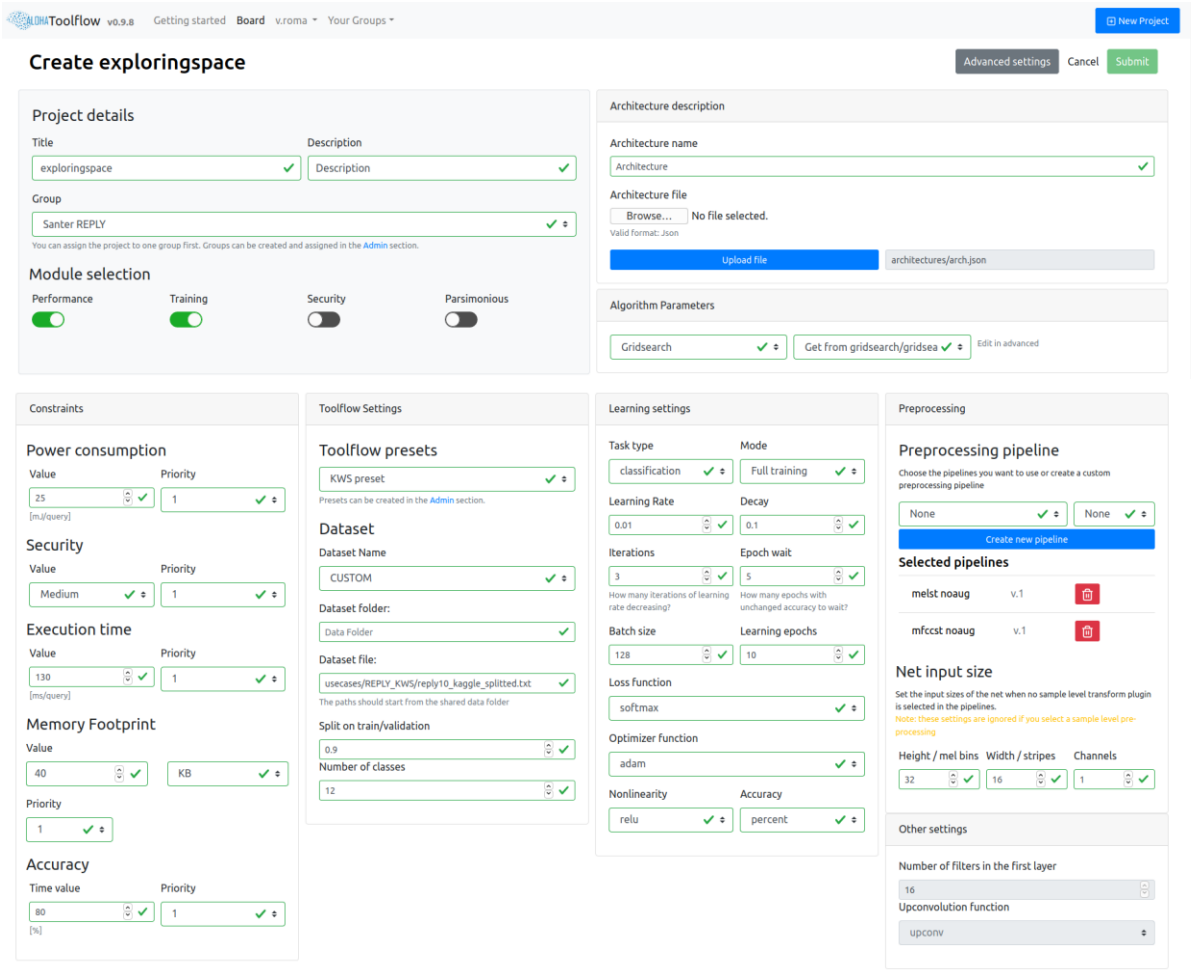


Figura 18: Configurazione esperimento

Una volta indicato il titolo del progetto, una breve descrizione e il gruppo di utenti che ci lavoreranno o con i quali lo si vuole condividere, si passa a definire quali tools si intendono adoperare attraverso la selezione dei rispettivi moduli performance, training, security e parsimonius.

Successivamente, occorre selezionare l'architettura target e caricare il file contenente una sua descrizione in termini di numero di processori, connettività e modalità operative disponibili, al fine di conferire consapevolezza dell'hardware sin dalle primissime fasi.

Segue la definizione dei parametri che caratterizzeranno la struttura delle reti neurali come numero di layers convoluzionali, completamente connessi, frequenza di pooling, tipologia di non linearità ecc. A tal proposito, si può optare per fornire il percorso che riporta alla cartella contenente i modelli desiderati in formato .onnx o magari, sfruttare il gridsearch per esplorare lo spazio di progettazione. Per questo secondo approccio occorre dapprima configurarlo in un file .json come il seguente,

```
{
  "CH":1,
  "H":32,
  "W":16,
  "CL":12,
  "KS":"3x3",
  "softmax":false,
  "batch_norm":true,
  "leakyrelu":false,
  "yolo":false,
  "boxes":5,
  "double_conv":false,
  "poolCadence":[2,3,4],
  "convLayers":[2,3,4,4],
  "channels_rules": [[1], [2,1], [1,0.5,1], [2,0.5,1]],
  "fcLayers":[1,2],
  "Mvalues":[16, 24, 32],
  "Jvalues":[32, 64]
}
```

Figura 19: Configurazione gridsearch

dopodiché, caricarlo in *advanced settings* raggiungibile tramite l'omonimo pulsante nell'angolo in alto a destra.

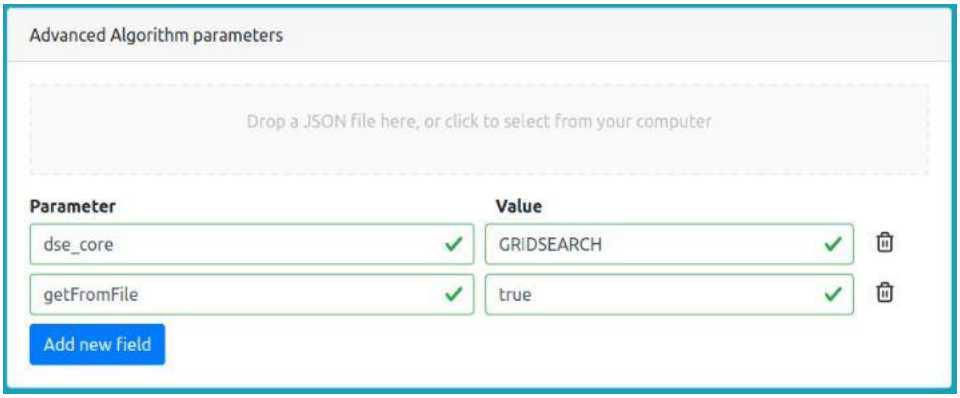


Figura 20: Caricamento gridsearch

A questo punto, si procede fissando i vincoli e le relative priorità in termini di consumo energetico, livello di sicurezza, velocità di esecuzione, spazio occupato e accuratezza che si intendono rispettare in funzione dell'obiettivo che ci pone. In questo modo, verranno filtrati gli algoritmi che non soddisfano tali soglie riducendo così lo spazio di esplorazione e di conseguenza le tempistiche associate a tale attività. Tuttavia, potrebbe anche accadere che modelli con un minor grado di accuratezza rispetto a quello prefissato ma che risultino compliant con gli altri vincoli vengano addestrati comunque.

Dopodiché si passa al setting della toolflow. Nello specifico, è possibile partire da uno specifico preset come ad esempio il *KWS preset*, sfruttando quindi una configurazione standard che potrà essere personalizzata attraverso aggiustamenti minoritari oppure, impostare il tutto dall'opzione di default indicata con *None*. Segue la definizione del dataset funzionale alla fase di training. In particolare, è possibile utilizzare sia alcuni tra i più noti e utilizzati datasets all'interno di progetti di Machine Learning, tra i quali si annoverano *MNIST*, *CIFAR10*, ecc.; altrimenti, qualora si voglia utilizzare un proprio dataset, è sufficiente selezionare la voce *custom* è successivamente indicare il percorso che conduce alla cartella che lo contiene. Indipendentemente dalla scelta, è necessario poi specificare la partizione dei dati in training e validation attraverso un parametro compreso tra 0 e 1. Questo perché come è noto dalla letteratura, splittare il dataset ed effettuare una validazione del modello ancor prima della fase di test vero e proprio, rappresenta la best practice volta ad assicurarsi che non si sia semplicemente verificato overfitting cioè, che il modello si adatti troppo bene al particolare set di dati con il quale è stato addestrato ma non sia in grado di generalizzare su dati aggiuntivi e quindi classificare in modo affidabile osservazioni future. Vale la pena sottolineare che una volta definito il parametro, ALOHA esegue questa attività in modo completamente automatico avendo cura di soddisfare la proprietà di speaker Independence. Infine, il numero di classi che dipenderà chiaramente dal tipo di applicazione che s'intende sviluppare va specificato.

Dunque, si procede con il learning settings. In questa fase sono definiti sia alcune specifiche informazioni inerenti all'applicazione che si vuole implementare che i cosiddetti hyperparameters, ovvero tutti quei parametri che non possono essere appresi ma occorre fissare a priori. In particolare, essi dipendono spesso dallo specifico problema oltre che dalla tipologia di dati di cui si dispone. A scopo meramente indicativo la configurazione include:

- Task type, sia esso di classificazione, segmentazione o rilevamento;
- Mode, che si tratti di un full training, fast training o esclusivamente validation di un modello già addestrato;
- Learning rate, rappresentante il passo con il quale ci sposta verso il punto di ottimo, corrispondente alla minima perdita;
- Decay, fattore moltiplicativo volto a ridurre il passo quando per un numero di epoche pari ad epoc wait, l'accuracy non migliora significativamente;
- Epoc wait, il cui ruolo è stato appena esposto;
- Iterations, stante ad indicare il massimo numero di volte che si vuole ridurre il passo perché superata tale soglia, non si riscontrerebbero miglioramenti nell'accuratezza apprezzabili;
- Batch size, caratterizzante la dimensione dei gruppi in cui viene partizionato il dataset e quindi la modalità con cui vengono sottoposti i dati al modello. Ciò consente di aggiornare i pesi in modo intelligente, cioè sulla base di un maggior numero di nuovi elementi;
- Learning epochs, vale a dire il numero di volte che l'intero dataset viene processato attraverso la rete;
- Loss function, sia essa softmax, sigmoid o altre minoritarie;
- Optimizer function, che si tratti di Stochastic Gradient Descent, Adam o altre note in letteratura;
- Nonlinearity, tra cui relu e tanh.

Infine, per ultimo ma non per importanza vi è la definizione del preprocessing. Tale attività ricopre un ruolo estremamente importante e consiste nel manipolare e modellare i dati, rendendoli fruibili per lo scopo cui devono prestarsi, ivi inclusa la data augmentation volta ad irrobustire il dataset. A tal proposito, dopo aver fissato la dimensione dell'input occorre configurare le pipelines. Per far fronte a tale intento sono disponibili due modalità: scrivere i propri plugins, creare la propria pipeline e caricarla tramite *advanced settings* oppure comporla tramite l'interfaccia utente. Per la prima modalità, una volta create è sufficiente caricarne il file .json qui.

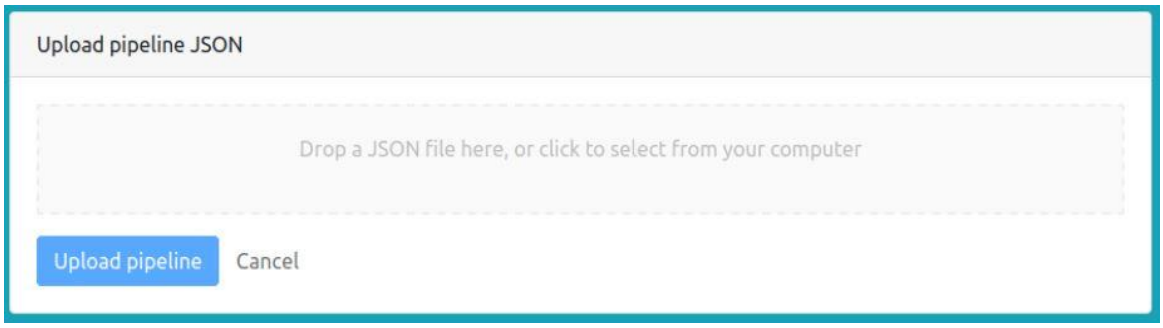


Figura 21: Caricamento Pipeline

Circa la seconda invece, cliccando su *create new pipeline* in basso a destra, una finestra come questa dovrebbe apparire.

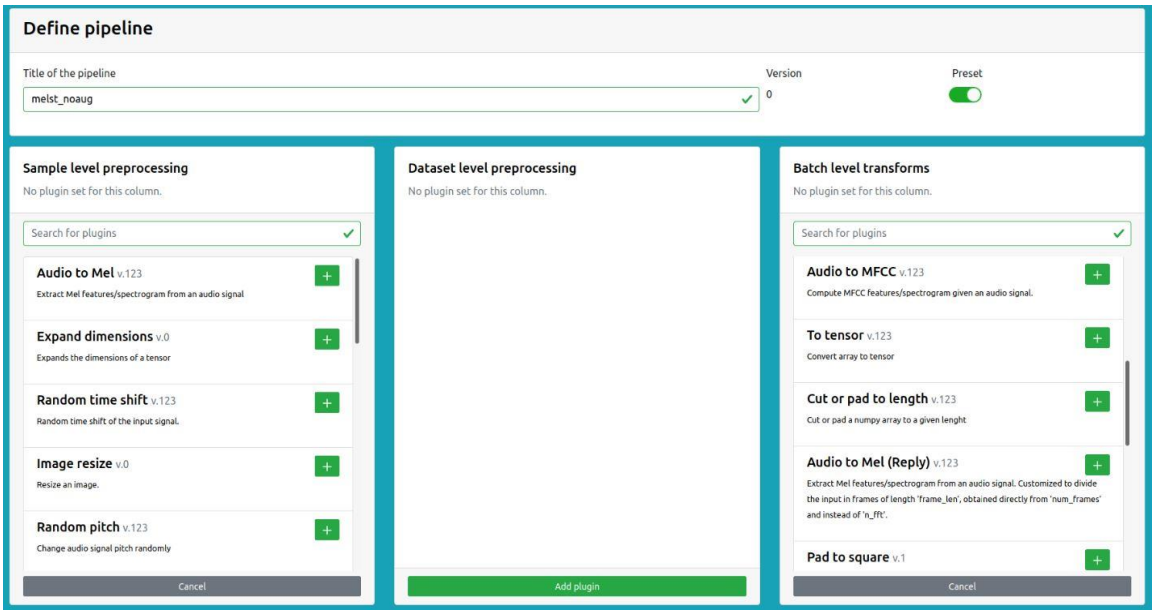


Figura 22: Pipeline e plugins tramite l'interfaccia utente di ALOHA

Come si nota in figura 22, una pipeline ha un titolo, una versione e tre elenchi di plugins uno per ciascuno stadio in cui ALOHA suddivide la pipeline, vale a dire:

- A livello di sample, cioè per ciascun singolo dato. È il caso più comune in cui vengono eseguiti ridimensionamenti, riempimenti o caricamento dei dati da file system e così via;
- A livello di dataset, quindi utilizzandolo in blocco. Solitamente utilizzato per attività come la normalizzazione;
- A livello di batch, ovvero a gruppi di dati la cui numerosità è decisa nel learning setting. Qui ha luogo qualsiasi altra trasformazione necessaria prima di fornire in input il dato alla rete, inclusa la data augmentation.

Per ciascuno di questi la toolflow offre un elenco dinamico di plugins già pronti all'uso quindi, basta selezionare semplicemente quelli a cui si è interessati e personalizzare eventualmente alcuni parametri correlati. Si ricordi che le scelte sono dettate dalla natura dei dati e dal tipo di operazioni cui si vogliono assoggettare. Vale la pena notare che una volta creata, una data pipeline resterà salvata e potrà essere richiamata per esperimenti successivi facendo così risparmiare tempo e sforzi.

Una volta che il form è stato completato e tutto è stato correttamente settato, è possibile cliccare sul tasto *submit* e una nuova card dovrebbe apparire. A questo punto, è sufficiente cliccare sul tasto *start* avviare la valutazione. Nell'attesa che quest'ultima termini, è possibile monitorarne i risultati parziali cliccando sull'icona di TensorBoard, toolkit che consente di tracciare e visualizzare le metriche di maggiore interesse a sostegno del processo decisionale. Quando lo status della card passa da *running* a *finished* allora lo step 1 può dirsi terminato e si passa così all'analisi dei risultati. Questi, qualora siano soddisfacenti condurranno alla scelta dell'algoritmo ottimale che verrà sottoposto agli step successivi; in alternativa, supporteranno la definizione dei nuovi esperimenti con un approccio di tipo "trial and error". Nello specifico, la visualizzazione dei risultati a mezzo di grafici e tabelle contenenti le principali metriche di riferimento, viene resa possibile cliccando sulla voce *view results.*

Qui, oltre a poter selezionare il modello ritenuto più promettente e quindi destinato ad esser sottoposto allo step successivo, è presente la funzionalità *download* che consente di scaricare eventualmente le configurazioni di reti neurali desiderate. Peraltro, qualora si ritenga necessario disporre di ulteriori dettagli circa i risultati ottenuti, è possibile affiancare alle statistiche già fornite da ALOHA quelle di TensorBoard. A titolo esemplificativo, si vedano le figure 23 e 24 che riportano i risultati di un esperimento che non si avvale dei moduli security e parsimonius, ragion per cui risulta giustificata l'assenza delle relative metriche.

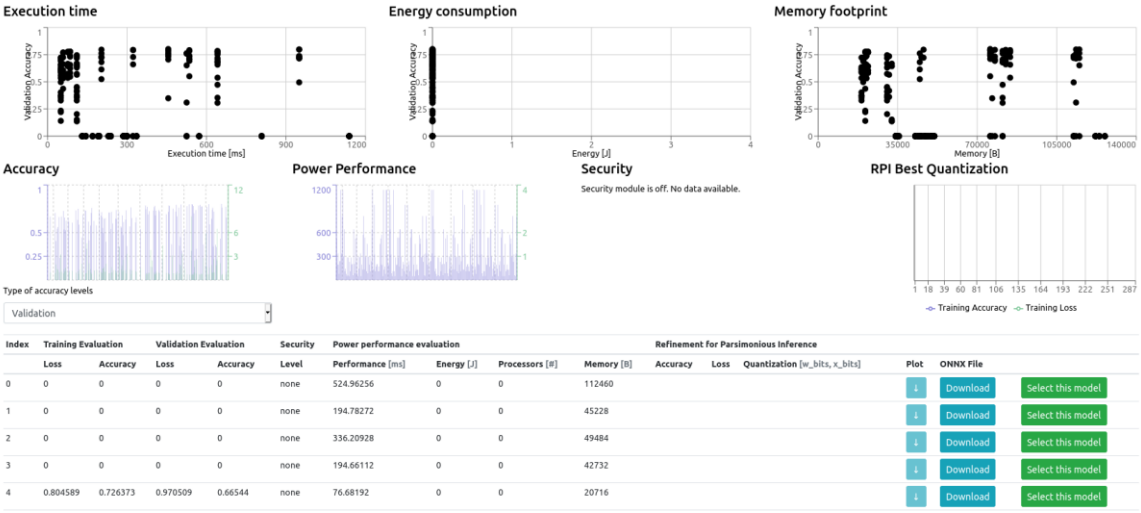


Figura 23: Risultati in ALOHA

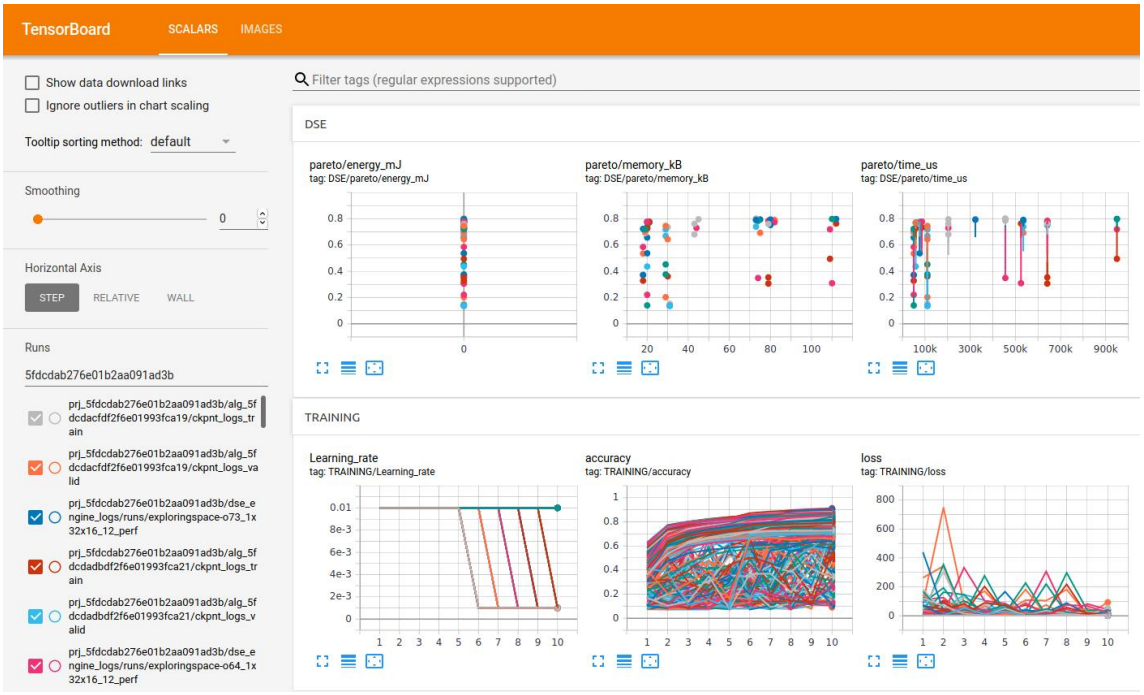


Figura 24: Risultati in TensorBoard

2.3 Posizionamento di mercato

Senza dubbio nell'ambito dell'intelligenza artificiale, il cosiddetto apprendimento profondo rappresenta ad oggi una delle più grandi opportunità commerciali che qualsivoglia società dovrebbe considerare a fondamento del proprio vantaggio competitivo. A segnalare la veridicità di una affermazione così forte basti pensare alla portata rivoluzionaria di interi settori, dal retail al manifatturiero, che l'introduzione di una simile tecnologia è in grado di apportare. In accordo con uno studio condotto da Technavio, società londinese, con un CAGR del 45% il mercato del Deep Learning a livello mondiale è destinato a crescere di 7.2 miliardi di dollari nell'orizzonte temporale 2020-2024 (Technavio 2020). Una stima più conservativa secondo Emergen Research, riporta invece un CAGR del 21.4% ed un aumento di 5.98 miliardi di dollari nel periodo di riferimento 2020-2027 (Emergen Research 2020). Quale che sia la stima più realistica certo è che entrambe le ricerche suggeriscono una sostanziale crescita del mercato. D'altro canto, per le limitazioni che una soluzione cloud-based comporta, in precedenza già discusse, si sta assistendo ad una transizione verso il paradigma dell'edge computing. Dunque, sempre più le applicazioni di Deep Learning prenderanno vita all'interno dei core di sistemi embedded i quali, una volta integrati, renderanno gli "oggetti" intelligenti.

Se da un lato Svetlana Sicular, VP Analyst in Gartner, ha affermato: "AI is starting to deliver on its potential and its benefits for businesses are becoming a reality" (Goasduff 2020), dall'altro Chirag Dekate, Senior Director Analyst in Gartner, sostiene: "Launching pilots is deceptively easy but deploying them into production is notoriously challenging" (Costello 2020). A rappresentare probabilmente il più grande ostacolo alla diffusione su larga scala è la carenza di competenze tecniche. Tale barriera, che vale a livello globale, lo è ancor di più relativamente allo scenario italiano dove, uno studio condotto dall'Osservatorio Artificial Intelligence del Politecnico di Milano, riporta unitamente alla mancanza di budget e al commitment da parte del top o middle management, la difficoltà nel creare un'organizzazione digitale quali principali fattori frenanti l'adozione.

Il superamento di una così ardua sfida risiede nella capacità di scalare il processo di data science e senz'altro l'automazione rappresenta la chiave per il successo. Una tale asserzione trova riscontro nella recente proliferazione di numerose piattaforme di autoML all'insegna di un tentativo di "democratization" ed accelerazione del processo di design. Infatti, è proprio il garantire accessibilità ad un bacino utenti con background molto diversi tra loro e magari non esperti di dominio, uno dei propositi principali a cui mirano tali sistemi. Altresì, aumentare la produttività ed ottenere soluzioni che generino valore economico nel minor tempo possibile e a costi contenuti risulta essere altrettanto fondamentale. A conferma di quanto appena discusso, un'analisi condotta da Mordor Intelligence sostiene come l'adozione di framework open source a supporto della progettazione sia diventata pratica molto diffusa tra le imprese, mentre, un report redatto da Technavio riporta quanto sia sostanziale la preferenza da parte delle imprese a dotarsi di software che, attraverso un'interfaccia di programmazione ad alto livello, forniscano sostegno nell'espletamento delle attività di data science.

È all'interno del contesto sopra descritto che si colloca la toolflow ALOHA e la cui proposta di valore è racchiusa in questo statement:

"Facilitare l'implementazione di algoritmi di Deep Learning su sistemi embedded, caratterizzati da limitate risorse energetiche e computazionali, attraverso l'automazione del processo di design, mappatura e successivo porting del modello, riducendo notevolmente tempi e costi associati a tale attività e rendendola accessibile anche ad utenti meno esperti, superando così la principale barriera che la stragrande maggioranza delle imprese odierne si ritrova a dover affrontare."

Nelle prossime sezioni si cercherà di fornire una risposta alla domanda: "proposta di valore per chi?", cercando di declinare in modo più dettagliato quali sono quei soggetti che meglio si identificano nella proposta suggerita e verso i quali orientare la propria offerta commerciale e, particolare attenzione sarà rivolta all'analisi del contesto competitivo volta ad individuare i principali attori attualmente presenti sul mercato e a fornire consapevolezza circa il proprio posizionamento rispetto a questi.

2.3.1 Customers

Riprendendo una citazione di Dave Coplin, Microsoft's chief envisioning officer, l'intelligenza artificiale rappresenta "the most important technology that anybody on the planet is working on today" (Urlini et al. 2019). Come diretta conseguenza, in prima battuta si potrebbe pensare di considerare come potenziali clienti:

Figura 25: Potenziali clienti

Tuttavia, adottare un simile approccio alquanto approssimato ed indirizzare la propria offerta commerciale verso un bacino utenti così ampio ed eterogeneo, potrebbe rivelarsi fallimentare oltre che una strada difficilmente praticabile. Pertanto, si rendono necessari approfondimenti ed ulteriori considerazioni per ciascuna delle macrocategorie identificate.

Per quanto riguarda gli individui, siano essi studenti, ricercatori o più in generale sviluppatori e data scientist, ben potrebbero questi rappresentare uno dei target di riferimento, specie nelle fasi iniziali di lancio del prodotto. Infatti, la tecnologia ALOHA potrebbe rappresentare un valido sostegno per l'avanzamento dello stato dell'arte e fungere da stimolo per l'ideazione di nuovi possibili use case.

Relativamente alle startups, il focus andrebbe posto chiaramente su quelle operanti in ambito AI. A tal proposito, cospicua è la numerosità specie di provenienza dagli Stati Uniti, dove, nel 2018 se ne contavano quasi 1400 (Liu 2020). A titolo puramente illustrativo si riporta qu di seguito un ranking delle 100 startup più promettenti al 2020, tra le circa 5000 analizzate a livello globale.

Figura 26: Le 100 AI startups più promettenti al mondo nel 2020[12]

Essendo native digital, entusiaste della tecnologia, propense all'innovazione e tipicamente soggette a budget ridotti in termini di tempi, costi e risorse umane, ALOHA potrebbe suscitare senz'altro il loro interesse e rappresentare un valido supporto nelle attività di sviluppo di nuove soluzioni, efficientando i processi e consentendo di concentrarsi maggiormente su altri aspetti come il marketing, la ricerca di fondi o l'instaurazione di opportune partnership.

[12]https://www.cbinsights.com/research/artificial-intelligence-top-startups/

Infine, circa le imprese sarebbe opportuno introdurre delle variabili di segmentazione che consentano di cogliere le peculiarità di ciascuna categoria. A tal proposito si è scelto di utilizzare la dimensione ed il settore di appartenenza ottenendo così una matrice come la seguente:

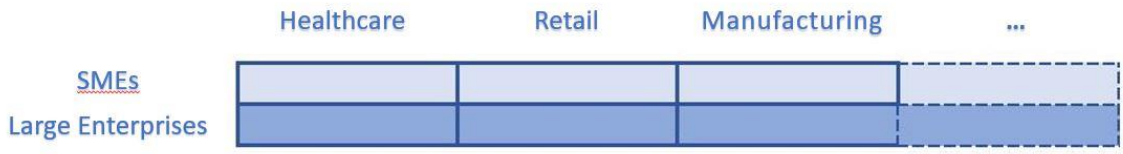


Figura 27: Variabili di segmentazione imprese

Secondo la normativa europea si considerano piccole e medie imprese tutte quelle aventi un numero di dipendenti inferiore a 250 ed un fatturato al di sotto dei 50 milioni di euro; di conseguenza, saranno ritenute grandi imprese tutte le altre.

- Tra le prime, probabilmente quelle molto piccole saranno più restie ad investire in una tecnologia come ALOHA perché interessate più alla sopravvivenza che all'attivazione di progetti così innovativi, mentre, quelle di maggiori dimensioni invece, che dispongano di sufficienti risorse economiche per l'acquisizione di un software come ALOHA ma non abbastanza per permettersi un servizio di consulenza, e che abbiano già all'interno delle figure IT che possano farne un buon uso, ben potrebbero rappresentare dei futuri utenti della toolflow.
- Relativamente alle seconde invece, più strutturate e dotate di maggiori capitali, facendo leva sulla rete di contatti di cui dispongono i diversi partner coinvolti all'interno del progetto, si potrebbe pensare di utilizzare ALOHA internamente al fine di proporre soluzioni chiavi in mano e su misura rispetto alle esigenze del cliente. Infatti, in accordo con uno studio condotto da Market Research Future, c'è una tendenza da parte delle imprese a fornire in outsourcing lo sviluppo di soluzioni basati su AI piuttosto che realizzarle internamente (Market Research Future 2021). Tuttavia, non si esclude la possibilità che alcune di queste possano decidere di voler adottare la tecnologia ed utilizzarla autonomamente, pertanto, l'esigenza o meno di un servizio di consulenza potrà essere gestita caso per caso.

Da un punto di vista dei settori di riferimento invece, come più volte è stato ribadito, l'intelligenza artificiale coinvolgerà i più disparati fino a raggiungere probabilmente la quasi totalità delle industry. Nel concentrare i propri sforzi commerciali, in termini di adoption timeline, si potrebbe pensare di assegnare la priorità a tutti quei settori in cui già ad oggi si riportano dei casi di successo per poi progredire estendendo la portata a nuovi. Si noti che, dalle ricerche condotte, sembrerebbero essere tali:

- Retail, dove nel prediligere i servizi online diventa ad esempio essenziale individuare e soddisfare le aspettative dei clienti attraverso la profilazione e successiva fornitura di raccomandazioni personalizzate;
- Healtcare, con applicazioni che spaziano dalla diagnosi medica allo studio di nuovi farmaci;
- Financial Services, in cui soluzioni basate su Deep Learning consentono di analizzare velocemente documenti, identificare il rischio di probabili frodi, migliorare l'esperienza utente attraverso l'impiego di chatbots oppure incrementare la sicurezza attraverso sistemi di videosorveglianza smart;
- Manufacturing, nel quale attraverso la visione artificiale è possibile migliorare notevolmente il rilevamento e successiva classificazione dei difetti, oppure, grazie ai progressi nel riconoscimento del parlato, diventa possibile l'interazione a distanza tra cobot e personale impegnato nello svolgimento di altre attività.

Quanto all'area geografica, poiché il progetto ALOHA così come i partners coinvolti confinano all'interno della comunità europea, è ragionevole pensare che il canale d'ingresso più realistico sia proprio il mercato europeo, il quale, si aggiudica peraltro il secondo posto a livello globale in termini di market share, subito dopo il Nord America ad oggi leader nel panorama dell'intelligenza artificiale.

Se queste finora discusse rappresentano le ipotesi iniziali fondate sulla ricerca secondaria oltre che sul proprio know-how e personale percezione del mondo che ci circonda, chiaramente della ricerca primaria andrebbe pianificata ed eseguita al fine di validare o confutare quanto è stato espresso così da poter indirizzare in modo analitico la propria offerta commerciale.

2.3.2 Competitors

Come più volte si è cercato di sottolineare, l'intelligenza artificiale rappresenta probabilmente una delle più grandi opportunità commerciali del ventunesimo secolo e, a seguito dell'interesse riscosso da parte di utenti, sviluppatori ed investitori, vivace è l'attività all'interno di questo dominio. Alcune organizzazioni aziendali, seppur in numero esiguo, hanno già iniziato a fare leva su intelligenza artificiale e machine learning per promuovere l'innovazione e creare valore sia per esse che per i vari stakeholders, tra i quali i propri clienti. Si è ampiamente discusso di quali siano le criticità più comuni legate alla scalabilità, prima fra tutte la difficoltà nel reperire opportune competenze tecniche, e si è anche detto di come l'automazione possa rappresentare la soluzione ad un tale problema. Infatti, senza alcun segno di recessione, nel corso degli ultimi anni l'autoML ha continuato ad affermarsi ed imporsi sul mercato, grazie anche ad i numerosi progressi in termini di funzionalità e grado di automazione che la ricerca in tale ambito ha consentito di apportare alle numerose piattaforme esistenti.

Da un preliminare screening del contesto competitivo è emerso come ad oggi il mercato risulti decisamente frammentato e popolato da attori quali alcune delle principali big tech come Google, Microsoft, Amazon Web Services e IBM, oltre che da molteplici nuovi entranti tra cui si annoverano Dataiku, DataRobot, H20.ai, BigML, dotData, ecc. A scopo meramente rappresentativo e senza alcuna pretesa di esaustività, si riporta qui di seguito il posizionamento di alcuni dei principali players tecnologici secondo Gartner.

Figura 28: Magic quadrant for Data Science and ML Platforms (Gartner)[13]

[13]https://pages.dataiku.com/gartner-2021

Approfondendo lo scenario relativo all'autoML si evince però come gli ideatori di tali piattaforme si ritrovino a dover affrontare un'importante decisione, la quale influenzerà certamente il proprio modello di business:

- Automatizzare l'intero processo di data science partendo dall'esplorazione e preparazione dei dati fino alla messa in produzione del modello;
- Automatizzare specifiche parti del flusso di lavoro, siano esse orientate più sui dati o sui modelli.

Diverse sono le ragioni per credere che, non sempre optare per una soluzione end-to-end e generalista sia la scelta migliore, infatti, la decisione andrebbe adoperata in funzione delle specifiche esigenze che s'intendono soddisfare. Così, ad esempio una società potrebbe non necessitare di alcun supporto nella preparazione del dataset e sarebbe avversa a corrispondere un prezzo per una piattaforma onnicomprensiva che di fatto sfrutterebbe solo in parte, o magari, anziché disporre di modelli preconfigurati e tools che in modo completamente automatico cerchino di customizzarli rispetto alla specifica applicazione, potrebbe gradire una certa autonomia nello svolgere tali attività. Saper cogliere determinate sfaccettature, nonostante possano risultare alquanto sottili, diventa essenziale perché una volta note esse consentono di effettuare una scelta consapevole circa la natura dei bisogni ai quali si mira a rispondere, cui segue dunque un diverso posizionamento di mercato.

Richiamando quanto già ampiamente è stato discusso circa la toolflow ALOHA ed il suo funzionamento, essa interviene nell'automazione della parte del flusso di lavoro "build/train/deploy" relativa ai modelli, pertanto, dovrebbe risultare piuttosto agevole desumere in quale delle due categorie di prodotto essa si colloca. Nello specifico, ci sono altre due peculiarità che occorre considerare al fine di poter correttamente individuare l'area di mercato nella quale si inserisce e i principali contendenti. La prima è che contrariamente alla maggior parte delle piattaforme presenti sul mercato, orientate più sul machine learning in generale, ALOHA risulta finalizzata agli esperimenti di Deep Learning. La seconda invece è relativa al focus sul "deploy on devices at the edge" che prevede solitamente periodi molto lunghi di ottimizzazione manuale al fine di rendere il modello compatibile con i requisiti del dispositivo target.

Alla luce delle considerazioni sinora fatte dovrebbe apparire chiaro perché molte delle piattaforme precedentemente citate e rappresentate nel diagramma di Gartner, che rientrano nella categoria di soluzioni end-to-end intese più come data science in generale, non possano essere considerate propriamente dei competitors. Altresì, tecnologie come shAIp, Appen e Kili Technology, seppur non appartenenti alla famiglia delle piattaforme end-to-end, potrebbero essere considerate al più dei prodotti complementari per via del loro focus sulla parte del flusso di lavoro a monte, quella cioè relativa ai dati.

Procedendo in questo modo, ovvero attraverso una valutazione puntuale dell'offerta commerciale prospettata da ciascuno dei principali attori presenti all'interno nel mercato, sono stati individuati e classificati come potenziali competitors perché più vicini agli obiettivi di ALOHA i seguenti: OctoML, Comet, Deep Cognition, Clarifai, Neural Designer, Allegro AI e Neural Network Intelligence. In tabella 1 si riporta una concisa valutazione delle diverse piattaforme al fine di oggettivarne punti di forza e di debolezza.

Tabella 1. Assessment Competitors

Competitors	Platform evaluation
OctoML https://octoml.ai/	Prevede che si disponga già di un modello addestrato il quale viene automaticamente ottimizzato rispetto ad un certo numero di hardware. Altresì, fornisce un benchmark rispetto ad alcuni modelli più comuni oltre che una stima dei requisiti dimensionali che il sistema embedded deve avere al fine di contenere l'eseguibile del modello.
comet https://www.comet.ml/site/	Consente di monitorare e comparare esperimenti al fine di valutarne le performance e facilitare l'individuazione di eventuali problematiche e criticità. Per essere utilizzata però richiede l'integrazione di una riga di codice nel proprio script, dunque l'attività di costruzione e addestramento del modello deve esser svolta manualmente o attraverso altre piattaforme.
Deep Cognition https://deepcognition.ai/	Specifica per il Deep Learning, presenta un'interfaccia abbastanza intuitiva. Tuttavia la configurazione del modello è svolta graficamente componendo la rete con i relativi blocchi e il deployment avviene sottoforma di servizio e non finalizzato a dispositivi at the edge
clarifai https://www.clarifai.com/	Seppur piattaforma end-to-end pone il focus sulle applicazioni basate su Deep Learning. Fornisce la possibilità di dare in outsourcing la parte relativa ai dati e/o quella sui modelli. Non è chiaro in che modo supporta l'utente nell'ottimizzazione e successivo porting del modello sul target di riferimento.
Neural Network Intelligence https://github.com/Microsoft/nni	Rappresenta un toolkit in grado di facilitare l'automazione del processo di design. Integra al suo interno le principali tecniche a supporto di: ingegnerizzazione features, ottimizzazione hyperparameters, esplorazione architetture di rete e compressione modelli. Tuttavia richiede l'utilizzo da linea di comando e quindi una certa familiarità con la scrittura di codice.
allegro AI https://www.allegro.ai/	Consiste in una suite open source che fornisce supporto nel monitorare, analizzare, comparare, ottimizzare e rendere riproducibili gli esperimenti condotti. Sebbene sia sufficiente una singola linea di codice per integrarlo, è esclusiva per il framework Pytorch e chiaramente richiede elevate competenze tecniche.
neuraldesigner https://www.neuraldesigner.com/	Copre l'intero flusso, dai dati alla messa in produzione del modello. Risulta specializzata sulle reti neurali e non è richiesta la scrittura di codice. Consente di visualizzare sottoforma di grafici e tabelle una serie di informazioni. È dubbia l'usabilità così come la capacità di far si che le caratteristiche del target di riferimento guidino la progettazione.

Mettendo a fattor comune le diverse piattaforme si evince come ciascuna miri a fornire un qualche tipo supporto alla progettazione di applicazioni basate su Deep Learning, tuttavia, alcune di queste presentano delle lacune rispetto all'offerta di ALOHA. Così, OctoML e Comet richiedono che si disponga già di un modello addestrato; Allegro AI risulta limitato a Pytorch e prevede la scrittura di codice; con Deep Cognition viene meno il supporto all'ottimizzazione e distribuzione su sistemi embedded, peraltro una delle attività più critiche e che riveste un ruolo centrale nella proposta di valore di ALOHA; Neural Network Intelligence sebbene abbia molte features in comune ad ALOHA, non risulta essere alla portata di tutti per via delle competenze richieste nel suo utilizzo. Ergo, tra quelle annoverate nell'analisi, al momento le uniche a destare una qualche preoccupazione perché da considerarsi dei validi diretti competitors risultano Clarifai e Neural Designer.

In estrema sintesi, per fare il punto della situazione ed avere una visione d'insieme a supporto delle decisioni più di natura strategica, è stata sviluppata la seguente analisi SWOT all'interno della quale si riportano, punti di forza e di debolezza della toolflow ALOHA nonché le principali minacce e opportunità provenienti dall'ambiente esterno.

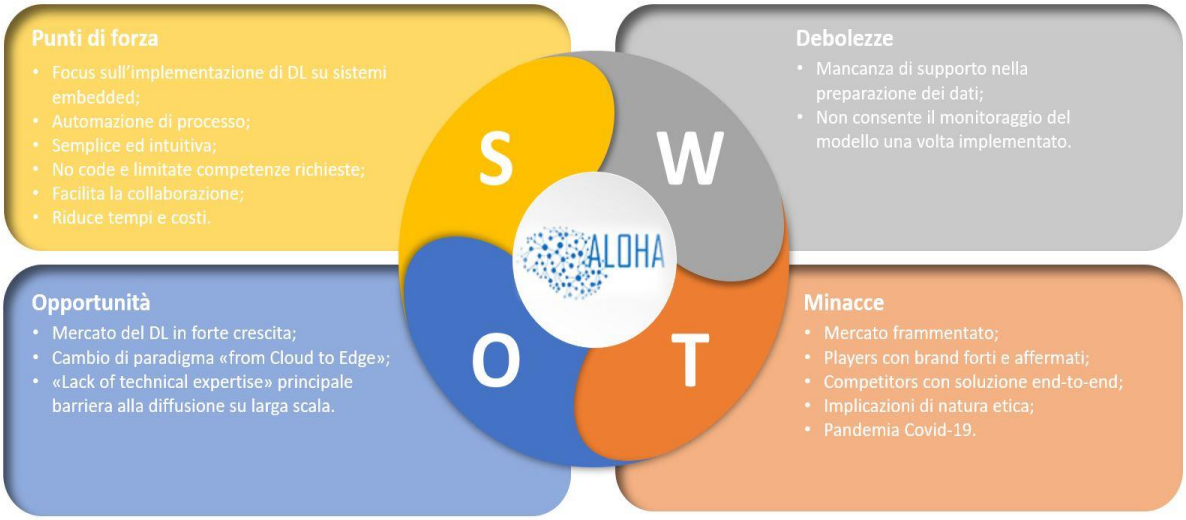


Figura 29: Analisi SWOT

Capitolo 3

KeyWord Spotting's use case

Gli algoritmi di Deep Learning, grazie ai grandi progressi conseguiti nel corso degli ultimi anni, hanno cominciato a riscuotere un certo successo sia nel mondo accademico che industriale. Tra i principali campi applicativi rientra sicuramente il riconoscimento del parlato, il cui mercato a livello globale sembrerebbe destinato a crescere da 10.7 a 27.16 miliardi di dollari nell'orizzonte temporale 2019-2025 (Liu 2021). Come è noto dalla storia, il parlato è stato e rappresenta tuttora uno dei principali mezzi di comunicazione tra gli esseri umani, sicché le prime forme di interazione sono avvenute a mezzo della voce e solo successivamente mediante scrittura. Inoltre, al giorno d'oggi, in un mondo all'insegna del digital e nell'era della "smartification", sempre più "dispositivi intelligenti" di ogni sorta circondano le nostre vite, sia private che lavorative. Queste due considerazioni che apparentemente viaggiano in parallelo, in realtà conducono ad una importante riflessione e ci portano a credere che, ben presto la voce potrebbe divenire la modalità preponderante nel relazionarsi alle macchine. Tale asserzione trova sostegno nei recenti sviluppi tecnologici nell'ambito del riconoscimento vocale così come nel crescente numero di dispositivi, siano essi appartenenti all'elettronica di largo consumo piuttosto che alla robotica industriale, in grado di supportare una simile funzionalità.

Alla luce delle considerazioni fatte, nell'effettuare un assessment della toolflow ALOHA si è scelto di realizzare uno use case avente come dominio di riferimento quello del keyword spotting. Volendone fornire una definizione, esso rappresenta la capacità di un sistema di rilevare all'interno di un flusso audio continuo, specifiche parole rispetto alle quali è stato addestrato e, sulla base di queste attivarsi o compiere delle precise operazioni. La sua natura di sistema "always-on", unitamente ad elevata accuratezza e tempi di risposta pressoché in tempo reale, entrambi richiesti per una buona esperienza utente, lo rende perfettamente allineato agli obiettivi della toolflow ALOHA, dal momento che l'ottimizzazione sotto il profilo energetico e computazionale assume un ruolo cruciale per questo tipo di applicazioni.

Prima di descrivere dettagliatamente l'obiettivo che attraverso il seguente use case s'intende conseguire, oltre che la metodologia adottata per far fronte a tale scopo e quindi le evidenze empiriche che da esso sono scaturite, si ritiene opportuno fornire un cenno sui fondamenti teorici che stanno alla base del riconoscimento del parlato.

3.1 Background teorico

Nel corso degli anni, il riconoscimento vocale è stato oggetto di studi e notevoli sono stati i progressi condotti in tale ambito. Nell'intento di percorrere velocemente la sua storia, non si può prescindere dall'enfatizzare le principali milestone. Tutto ebbe inizio nel 1952 quando il sistema Audrey cominciò a riconoscere le sole cifre. Dieci anni dopo, IBM introdusse Shoebox, un computer in grado di riconoscere oltre alle cifre anche sedici parole. Successivamente, attraverso l'exploitation nel day by day si è giunti a ricoprire un vocabolario di alcune centinaia di parole. In seguito, l'introduzione del metodo statistico noto come Hidden Markov Model ha esteso la portata del riconoscimento vocale, potenzialmente ad un'infinità di termini. A partire dagli anni Novanta, numerose metodologie e modelli sono emersi, affetti però da alcune lacune dovute al passaggio dal laboratorio ad un ambiente arbitrario. Così nel 2008, le principali tecnologie sviluppate sono riuscite ad assolvere a tali problematiche seppur limitatamente ad un ristretto numero di task. Di recente invece, gli sforzi muovono verso il raggiungimento di un vocabolario illimitato, che includa molteplici lingue, volto ad una miriade di task, il tutto confinato in un'ambiente arbitrario.

Un segnale audio però, affinché possa essere implementato all'interno di una rete neurale che è solita operare in due dimensioni, rispetto alle immagini richiede un'attività addizionale di preprocessing. Quest'ultima consiste nel partizionare in frame, di lunghezza l e passo s, il segnale audio in input; applicare un estrattore delle caratteristiche a ciascuno di questi ottenendo dei vettori, i quali una volta impilati andranno a costituire una matrice avente il tempo e la frequenza rispettivamente lungo le due dimensioni, altresì detta spettrogramma. In questo modo, il dato in input divenuto fruibile per la rete neurale potrà essere processato e, a seguito del passaggio all'interno di un modulo di classificazione, saranno generate delle probabilità associate a ciascuna delle classi di output.

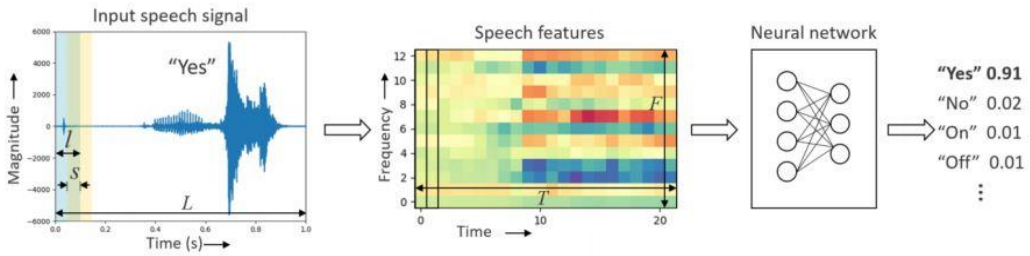


Figura 30: Approfondimento su preprocessing per input di tipo audio[14]

[14]https://arxiv.org/abs/1711.0712

3.2 Obiettivo

Riprendendo brevemente il problema che limita una proliferazione su larga scala di sistemi embedded dotati di un algoritmo di Deep Learning, che consenta loro di abilitare svariate funzionalità come il riconoscimento di difetti o specifici comandi vocali, una volta progettato e addestrato un modello avente una buona accuratezza, ottimizzarlo al fine di ottenere delle buone prestazioni in termini di spazio occupato, velocità di calcolo e consumo energetico, richiede decisamente tempi e sforzi non indifferenti oltre che elevate competenze tecniche. Infatti, dall'analisi della letteratura e dalle ricerche condotte emerge che tipicamente, al fine di eseguire l'inferenza in locale su sistemi caratterizzati da limitate risorse energetiche e computazionali, gli sviluppatori incorrono in mesi di lavoro da dedicare a simili attività. Come si evince dalla proposta di valore di ALOHA, l'adozione della toolflow dovrebbe consentire di risolvere il suddetto problema, facilitando l'implementazione su hardware e rendendola un'attività decisamente più rapida e accessibile a tutti grazie all'automazione di processo apportata.

Nel voler fornire evidenza della promessa fatta, si è deciso di ideare un singolo caso studio volto ad effettuare una valutazione sul campo di ALOHA durante l'espletamento di un esperimento completo. Come suggerisce il nome dello use case e come in parte è già stato accennato, esso verte sul riconoscimento del parlato, precisamente sul keyword spotting, e si pone come obiettivo la quantificazione dei benefici derivanti dall'utilizzo della tecnologia. In particolare, seppur limitatamente ad uno specifico ambito, le evidenze emerse potranno essere poi generalizzate seppur magari non parimenti anche ad altri scenari e casi applicativi.

Per esprimere formalmente i vantaggi dichiarati e poter disporre di un insieme di parametri di controllo che da un lato, consentano di monitorare le performance di processo e forniscano supporto decisionale, e dall'altro, rendano possibile affermare il successo o meno della prova, il primo passo da percorrere consiste nell'individuazione di un opportuno set di KPIs. In generale, la convenienza nell'adozione di un indicatore deriva dall'analisi costi-benefici che dall'implementazione dello stesso scaturiscono. A tal proposito, le linee guida suggeriscono di utilizzare il minimo numero di indicatori che consenta di controllare in modo efficace l'intero processo. Poiché il principale beneficio per l'utente, si esplica in un'accelerazione delle tempistiche richieste al fine di ottenere, una soluzione ottimizzata e pronta per essere implementata su una data architettura target, riducendo così significativamente il time to market, sono stati definiti dei KPIs rispetto a due aree d'intervento:

- Sotto il profilo gestionale, la variabile di riferimento risulta essere il tempo impiegato affinché il progetto possa dirsi terminato. Questo computa al suo interno sia le ore uomo conseguenti all'intervento manuale dell'utente che i tempi macchina dedicati ai vari automatismi. In particolare, per un maggior livello di dettaglio si è preferito scindere il tempo totale lungo le tre macrofasi quali la progettazione dell'algoritmo, l'ottimizzazione rispetto allo specifico target ed infine la generazione del codice;
- Dal punto di vista informatico invece, le metriche ritenute rilevanti e delle quali si vuole tener traccia sono la capacità del modello di classificare correttamente il dato in input, lo spazio da esso occupato che ricordiamo debba essere compatibile con il requisito di memoria del dispositivo di riferimento ed infine, la velocità di esecuzione del medesimo durante la fase di inferenza.

In tabella 2 si riporta per ciascun indicatore definito una breve descrizione, una stima del valore target che l'utilizzo di ALOHA dovrebbe consentire di ottenere ed infine, un commento oggettivo sui pregi e i difetti ad essi associati.

Tabella 2. KPIs per la valutazione delle performance del processo di design

Indicatore	Descrizione	Target	Pregi	Difetti
KPI 1_1 Tempo progettazione (TP)	Tempo speso per progettare, addestrare e selezionare un algoritmo adatto alla specifica applicazione (utilizzando i tools dello step 1)	Giorni	Fornisce una stima del tempo impiegato per lo step 1 e quindi rappresenta una proxy dell'effort richiesto	Dipende molto dalla complessità del problema, dalle competenze dell'utente e dall'infrastruttura sulla quale viene eseguito il training
KPI 1_2 Tempo ottimizzazione (TO)	Tempo speso per identificare la migliore partizione dell'algoritmo selezionato e definire la mappatura ottimale sull'architettura target (utlizzando i tools dello step 2)	Ore	Fornisce una stima del tempo impiegato per lo step 2 e quindi rappresenta una proxy dell'effort richiesto	È influenzato dalla specifica configurazione di rete oltre che dalla dimensione dello spazio di progettazione (numero di processori e possibili mappature)
KPI 1_3 Tempo generazione codice (TGC)	Tempo speso per generare e customizzare il codice per il porting dell'algoritmo sull'architettura target (utilizzando i tools dello step 3)	Ore	Fornisce una stima del tempo impiegato per lo step 3 e quindi rappresenta una proxy dell'effort richiesto	Varia a seconda che l'adattamento del middleware per una data architettura target sia già stato effettuato per altre applicazioni pregresse oppure no
KPI 2_1 Accuratezza (A)	Numero medio di osservazioni etichettate correttamente	In base al tipo di applicazione si è soliti fissare un limite inferiore, superato il quale non si ritiene soddisfacente	Rappresenta una stima dell'affidabilità del sistema e quindi della qualità del servizio oltre che della soddisfazione dell'utente	Obiettivi troppo ambiziosi potrebbero finire col gravare sulla fattibilità tecnica dato il dispedioso fabbisogno energetico richiesto
KPI 2_2 Memoria (M)	Spazio occupato dall'intera rete neurale, dunque prendendo in considerazione input/output, pesi, ecc.	In funzione della capacità dell'architettura target	Consente di assicurarsi che l'implementazione sul dispositivo sia realmente fattibile	È influenzata dall'applicazione o meno della quantization oltre che dal numero di bit utilizzato
KPI 2_3 Tempo esecuzione inferenza (TEI)	Tempo di risposta da parte del sistema, inteso anche come il numero totale di operazioni per ciasun layer della rete	A seconda del contesto ma tipicamente pressochè in tempo reale	Fornisce una stima della prontezza	Data una rete e quindi il numero totale di operazioni necessarie e ad essa associate, il tempo impiegato dipenderà dallo specifico hardware sulla quale andrà eseguita

3.3 Metodologia

Nel concepire il caso studio ci si è ispirati alla TensorFlow Speech Recognition Challenge promossa nel 2017 su Kaggle, la più grande community al mondo di data science. Intenzionalmente si è cercato di individuare e replicare ex novo qualcosa che fosse già stato fatto in passato anziché ideare un esperimento ad hoc, al fine di poter disporre di un termine di paragone con il quale raffrontarsi. Come suggerisce il titolo della challenge, l'ambito è quello del riconoscimento vocale e la sfida, consiste nella creazione di un modello capace di classificare dei semplici comandi unitamente alle classi *silence* ed *unknown,* quest'ultima intesa come tutto ciò che esula dalle keywords specificate.

yes	up	left	on	stop	unknown
no	down	right	off	go	silence

Figura 31: Definizione classi

Tipicamente, e questo vale sia per l'apprendimento profondo che per il Machine Learning più in generale, estendere l'accesso ad appositi dataset favorisce la collaborazione tra gruppi e consente una sorta di normalizzazione tra i diversi approcci utilizzati, rendendo così i risultati omogenei e confrontabili tra di loro. Per tali ragioni, diverse iniziative sono state intraprese lungo questa direzione al fine di favorire il progresso in questo ambito. Così, al pari di ImageNet e raccolte simili nell'ambito del computer vision, lo Speech Command Dataset di Google è diventato lo standard di riferimento per il KWS. Esso include al suo interno un set di file audio .wav da un secondo, ciascuno contenente una singola parola in inglese pronunciata da migliaia di persone differenti. Per ovvie ragioni, anche il caso proposto si basa su di esso e nello specifico su un suo sottoinsieme. Infatti, avendo come features le 12 classi precedentemente introdotte, prima di avviare l'esperimento in ALOHA si rende necessario una sua configurazione. Essa prevede di estrarre i soli comandi di interesse e popolare le rispettive cartelle. Così, dopo aver scaricato il dataset originale si è utilizzato uno script, scritto dal team di ALOHA e condiviso nel repository di progetto su GitLab, per eseguire in pochi minuti ed in modo completamente automatico il seguente task.

La finestra temporale concessa per la progettazione del modello era stata fissata pari a due mesi e, sebbene non vi sia traccia dell'effettivo effort impiegato dai vari gruppi partecipanti, è ragionevole assumere questa come proxy del tempo necessario allo sviluppo di simili applicazioni con un approccio manuale; peraltro, stima che risulta confermata dall'analisi della letteratura scientifica così come dalle ricerche condotte.

Le candidature presentate dai partecipanti nel rispetto della deadline di consegna sono state valutate in base all'accuratezza riscontrata in fase di test. A tal proposito, qui di seguito si riporta un estratto della classifica finale includente i migliori dieci team e il rispettivo punteggio, stante ad indicare il numero medio di osservazioni etichettate correttamente, espresso in termini percentuali. Nello specifico si noti che il primo classificato, nonché vincitore della challenge, è riuscito ad ottenere un modello caratterizzato da un'accuratezza del 91.06 %.

#	△pub	Team Name	Notebook	Team Members	Score
1	▲ 3	Heng-Ryan-See * good bug? *			0.91060
2	▼ 1	Thomas O'Malley			0.91048
3	—	Little Boat			0.91013
4	▼ 2	high five			0.90931
5	▲ 5	은주니(ttagu99) & sjv			0.90896
6	▲ 1	S4		+4	0.90825
7	▼ 2	GREAT@SHU			0.90790
8	▲ 3	VAZ			0.90649
9	▼ 1	Gold Gazua			0.90637
10	▼ 4	but			0.90532

Figura 32: Estratto classifica TensorFlow Speech Recognition Challenge[15]

[15]https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/leaderboard

Tuttavia, alla luce della trattazione finora condotta si può facilmente desumere quale sia il principale limite di un esperimento così definito se calato in un contesto reale. Infatti, ponendosi come unico obiettivo progettuale la massimizzazione dell'accuratezza in maniera stand alone e, non curandosi delle performance sotto il profilo energetico e computazionale, l'implementazione su hardware potrebbe essere compromessa significativamente non garantendone così la fattibilità. Pertanto, se la TensorFlow Speech Recognition Challenge rappresenta il punto di partenza attraverso il quale è stato possibile definire tipologia e numero di classi, dataset e stima dei tempi necessari al solo sviluppo del modello mediante approccio tradizionale, al netto cioè della relativa ottimizzazione e generazione dell'eseguibile al fine di consentirne il porting su un sistema embedded, con ALOHA si vuole estendere il punto di arrivo rendendo l'applicazione concreta all'atto pratico.

Ora che la tipologia di applicazione è stata definita, il passo successivo consiste nella scelta dello specifico hardware sulla quale dovrà essere implementata. A tal proposito, si è deciso di adottare come dispositivo target il modulo STEVAL-STLCS01V1, anche noto come SensorTile, di produzione della STMicroelectronics.

Figura 33: Sensortile[16]

Esso è un modulo IoT molto compatto, di dimensioni pari a quelle di un quadrato di lato 13,5 mm, caratterizzato da potenti capacità di elaborazione, connettività Bluetooth a basso consumo energetico e una varietà di sensori, tra cui un microfono.

[16]https://www.st.com/resource/en/data_brief/steval-stlkt01v1.pdf

A questo punto, a complemento della determinazione dello use case, non resta che pianificare da un punto di vista metodologico i passi da compiere mediante ALOHA. Relativamente allo step 1, volto alla ricerca del modello ottimale, si adotterà un approccio a due stadi:

- Un primo esperimento piuttosto rapido e senza il ricorso alla data augmentation, volto ad esplorare lo spazio di progettazione attraverso la configurazione di un opportuno gridsearch con diversi gradi di libertà come: il numero di layers, le mappe delle caratteristiche di ciascuno di essi e le diverse preelaborazioni. Procedendo in questo modo, sarà possibile identificare velocemente configurazioni di algoritmi promettenti. A tal proposito, si segnala che grazie alla stima delle performance per ogni potenziale candidato, eseguita in ALOHA dal tool power performance, solo i modelli che soddisfano i vincoli fissati dall'utente verranno addestrati efficientando così il processo di training;
- Un secondo esperimento più intenso e che utilizza la data augmentation, focalizzato esclusivamente sui migliori modelli addestrati nell'esperimento precedente. L'obiettivo è quello di far emergere ulteriormente il loro potenziale migliorandone le prestazioni da un lato, grazie al dataset irrobustito e, dall'altro, attraverso la regolazione ad hoc dei cosiddetti hyperparameters.

Successivamente, una volta individuata la configurazione di rete più promettente per la specifica applicazione, si procederà con gli step 2 e 3 rispettivamente volti ad ottimizzare il modello rispetto alla SensorTile e a generare il codice necessario al porting su hardware, completando così lo use case. Riguardo a quest'ultimo vale la pena ricordare che, in accordo con l'obiettivo prefissato, durante il suo decorso verrà tenuta traccia dei KPI's in precedenza introdotti, cosicché i risultati ottenuti in termini di performance possano essere confrontati opportunamente con le evidenze mostrate su Kaggle.

Sia questa una descrizione ad alto livello della metodologia adottata, ora seguirà un approfondimento da un punto di vista operativo della procedura relativa allo step 1 poiché, contrariamente agli step 2 e 3 completamente automatizzati, è previsto al suo interno l'intervento umano. L'ambiente in cui avviene la creazione dell'esperimento si presenta nel seguente modo.

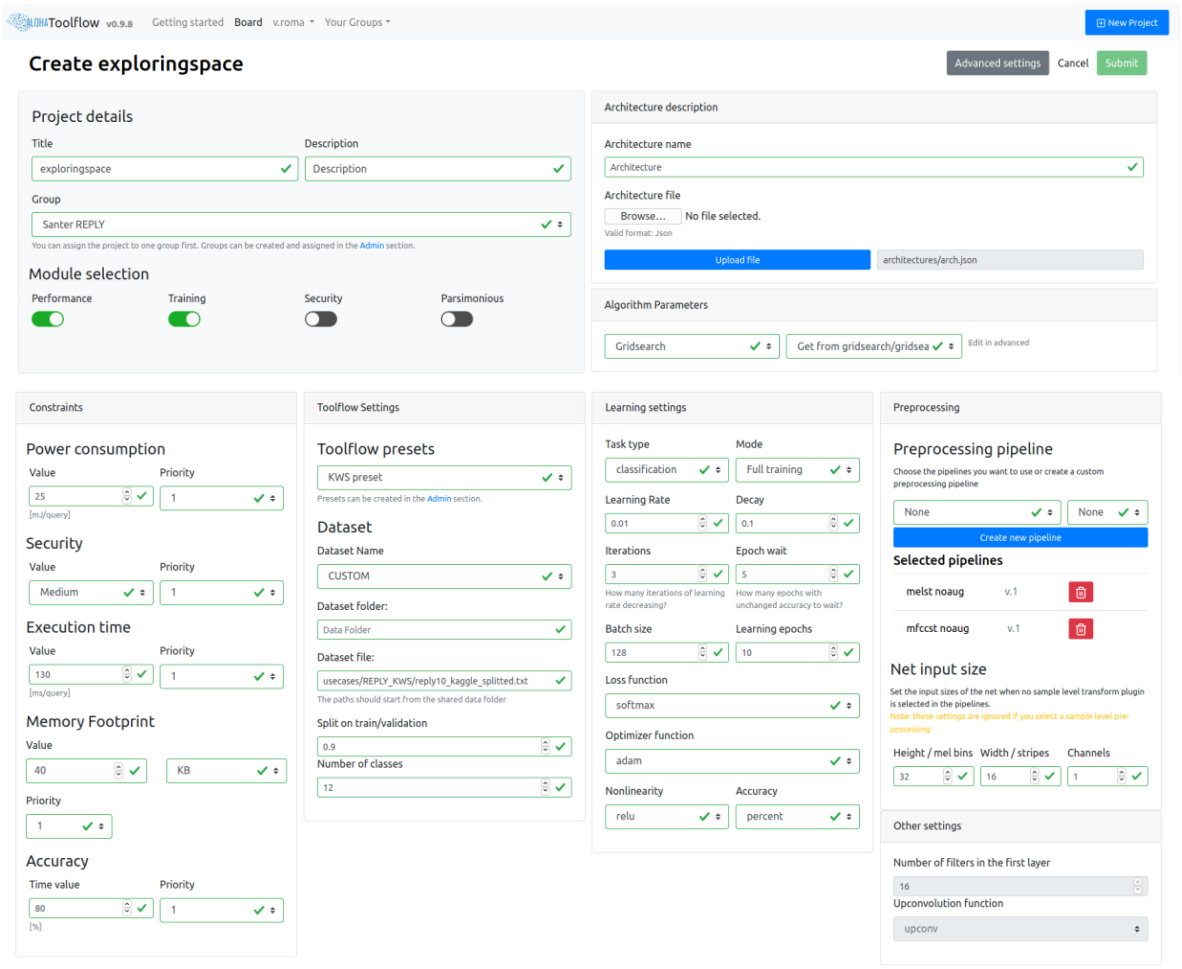


Figura 34: Creazione esperimento

Come si nota in figura 33, prima di poter avviare la fase di training del modello occorre eseguire alcune scelte e settare diversi parametri. Prescindendo da una descrizione puntuale e analitica, verranno enfatizzati solamente i tratti essenziali.

Per far fronte alla configurazione dell'esperimento ci si è avvalsi di diverse fonti d'informazione. Innanzitutto, per familiarizzare col tema è stato seguito il corso CS231[17] sulle reti neurali dell'università di Stanford. Successivamente, si sono analizzati vecchi esperimenti eseguiti in ALOHA, seppur relativi ad applicazioni differenti da quella prevista in questo use case, così come la documentazione presente nel repository di progetto su GitLab[18]. Infine, sono stati poi raccolti e analizzati alcuni papers tratti dalla letteratura scientifica, primo fra tutti Hello Edge[19], oltre che i documenti relativi alla SensorTile[20].

Integrando opportunamente le suddette informazioni è stato possibile procedere con la configurazione. Così, relativamente ai *coinstraints* sono stati fissati i seguenti valori, comuni ad entrambi i due stadi.

Constraints	Value	Unit of measurement	Priority
Execution time	130	ms/query	1
Memory footprint	40	kB	1
Accuracy	80	%	1

Figura 35: Constraints

A tal proposito, si noti che i valori di *Accuracy, Memory footprint* ed *Excution Time,* rappresentano il target di riferimento dei rispettivi KPI's, i quali nel framework generale riportato in tabella 2 erano rimasti indefiniti perché da personalizzare rispetto allo specifico use case.

[17]https://www.youtube.com/watch?v=vT1JzLTH4G4&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv

[18]https://gitlab.com/aloha.eu

[19]https://arxiv.org/abs/1711.07128

[20]https://www.st.com/content/st_com/en/products/evaluation-tools/solution-evaluation-tools/sensor-solution-eval-boards/steval-stlkt01v1.html#

Per quanto riguarda la sezione *Learning settings* sono stati utilizzati i seguenti setup rispettivamente per i due stadi.

First stage	
Task type:	classification
Mode:	full training
Learning rate:	0.01
Decay:	0.1
Iteration:	3
Epoc wait:	5
Batch size:	128
Learning epochs:	10
Loss function:	softmax
Optimizer function:	adam
Non linearity:	relu
Accuracy:	percent

Second stage	
Task type:	classification
Mode:	full training
Learning rate:	0.01
Decay:	0.1
Iteration:	3
Epoc wait:	10
Batch size:	128
Learning epochs:	50
Loss function:	softmax
Optimizer function:	adam
Non linearity:	relu
Accuracy:	percent

Figura 36: Learning settings

Si noti che le uniche differenze sono date dal *learning epochs,* ovvero il numero di volte in cui il dataset è processato attraverso la rete e, dall'epoc wait, inteso come il numero di epoche che occorre far trascorrere nonostante non ci sia un significativo miglioramento dell'accuratezza, prima di ridurre il *learning rate* moltiplicandolo per il *decay.* Infatti, come precedentemente discusso, lo scopo al primo stadio è quello di individuare velocemente mediante gridsearch le configurazioni di rete più promettenti, ragion per cui è si è deciso di eseguire un numero ridotto di epoche, nello specifico 10, rispetto alle 50 previste al secondo stadio.

Relativamente al gridsearch utilizzato al primo stadio, è stato configurato il seguente file .json dal quale scaturiscono 3 x 4 x 2 x 3 x 2 = 144 tipologie di reti neurali, risultato della combinazione dei livelli di *poolCadence*, *convLayers*, *fcLayers*, *Mvalues* e *Jvalues*.

```
{
  "CH":1,
  "H":32,
  "W":16,
  "CL":12,
  "KS":"3x3",
  "softmax":false,
  "batch_norm":true,
  "leakyrelu":false,
  "yolo":false,
  "boxes":5,
  "double_conv":false,
  "poolCadence":[2,3,4],
  "convLayers":[2,3,4,4],
  "channels_rules": [[1], [2,1], [1,0.5,1], [2,0.5,1]],
  "fcLayers":[1,2],
  "Mvalues":[16, 24, 32],
  "Jvalues":[32, 64]
}
```

Figura 37: Configurazione gridsearch

Infine, per la configurazione del preprocessing sui dati, utilizzando alcuni dei plugins già presenti in ALOHA e messi a disposizione dell'utente, sono state create le due seguenti pipelines per il primo stadio.

melst noaug		
Sample level	Dataset level	Batch level
Load audio from wav		To tensor
Cut or pad to lenght		
Audio to Mel (Reply)		

mfccst noaug		
Sample level	Dataset level	Batch level
Load audio from wav		To tensor
Cut or pad to lenght		
Audio to MFCC		

Figura 38: Pipelines primo stadio

Mentre, per il secondo stadio sono state generate le seguenti.

melst aug		
Sample level	Dataset level	Batch level
Load audio from wav		Random time shift
Cut or pad to lenght		Random pitch
		Random speed
		Random sum noise
		Audio to Mel (Reply)
		To tensor

mfccst aug		
Sample level	Dataset level	Batch level
Load audio from wav		Random time shift
Cut or pad to lenght		Random pitch
		Random speed
		Random sum noise
		Audio to MFCC
		To tensor

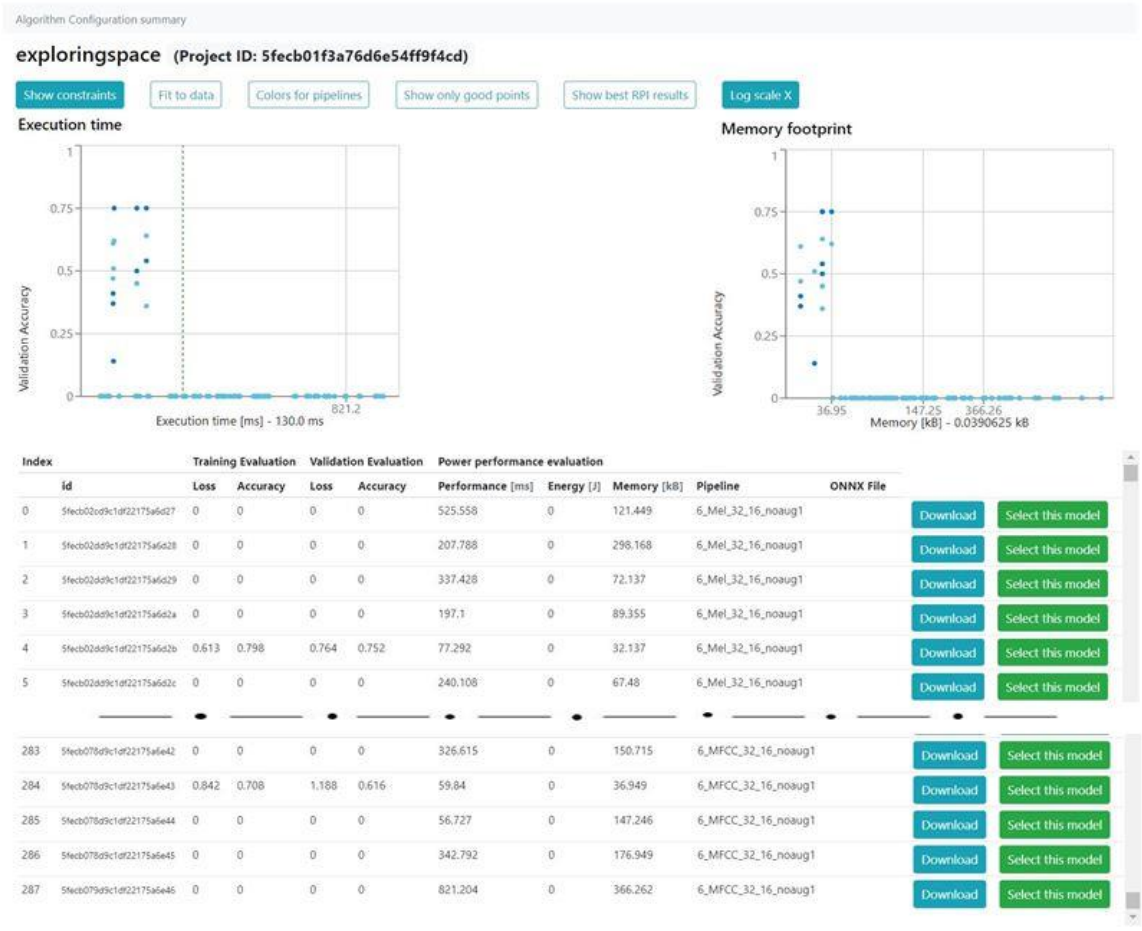
Figura 39: Pipelines secondo stadio

Si nota come la principale differenza, consista nell'introduzione al secondo stadio di alcuni plugins dediti alla data augmentation, volta ad irrobustire il dataset ed incrementare così le performance dei modelli a maggior potenziale.

3.4 Principali evidenze

Descritta la metodologia adottata nell'affrontare lo use case ed avendo a mente l'obiettivo che attraverso il suo espletamento s'intende conseguire, la seguente sezione sarà dedicata alla discussione dei risultati empirici ottenuti a seguito delle prove condotte.

Come già descritto in precedenza, lo step 1 della toolflow ALOHA, volto alla ricerca del modello ottimale rispetto agli obiettivi di qualità prefissati, è stato suddiviso in due stadi posti in serie tra loro. Relativamente al primo, quello mediante gridsearch, si riportano qui di seguito alcuni grafici ed una tabella contenente un elenco dei vari modelli generati, oltre che delle principali caratteristiche.



Index		Training Evaluation		Validation Evaluation		Power performance evaluation					
	id	Loss	Accuracy	Loss	Accuracy	Performance [ms]	Energy [J]	Memory [kB]	Pipeline	ONNX File	
0	5fecb02cd9c1df22175a6d27	0	0	0	0	525.558	0	121.449	6_Mel_32_16_noaug1		Download / Select this model
1	5fecb02dd9c1df22175a6d28	0	0	0	0	207.788	0	298.168	6_Mel_32_16_noaug1		Download / Select this model
2	5fecb02dd9c1df22175a6d29	0	0	0	0	337.428	0	72.137	6_Mel_32_16_noaug1		Download / Select this model
3	5fecb02dd9c1df22175a6d2a	0	0	0	0	197.1	0	89.355	6_Mel_32_16_noaug1		Download / Select this model
4	5fecb02dd9c1df22175a6d2b	0.613	0.798	0.764	0.752	77.292	0	32.137	6_Mel_32_16_noaug1		Download / Select this model
5	5fecb02dd9c1df22175a6d2c	0	0	0	0	240.108	0	67.48	6_Mel_32_16_noaug1		Download / Select this model
283	5fecb078d9c1df22175a6e42	0	0	0	0	326.615	0	150.715	6_MFCC_32_16_noaug1		Download / Select this model
284	5fecb078d9c1df22175a6e43	0.842	0.708	1.188	0.616	59.84	0	36.949	6_MFCC_32_16_noaug1		Download / Select this model
285	5fecb078d9c1df22175a6e44	0	0	0	0	56.727	0	147.246	6_MFCC_32_16_noaug1		Download / Select this model
286	5fecb078d9c1df22175a6e45	0	0	0	0	342.792	0	176.949	6_MFCC_32_16_noaug1		Download / Select this model
287	5fecb079d9c1df22175a6e46	0	0	0	0	821.204	0	366.262	6_MFCC_32_16_noaug1		Download / Select this model

Figura 40: Risultati primo stadio

Essendo 144 le configurazioni di rete generate dal gridsearch ed avendo definito 2 pipelines per il relativo preprocessing, sono stati complessivamente esplorati 288 alternative progettuali. Com'è facilmente desumibile a vista d'occhio, grazie alla stima ex ante delle performance eseguita da ALOHA, solo i modelli compatibili con i vincoli imposti sono stati addestrati.

Dopo un'attenta analisi dei risultati, sono stati selezionati come modelli più promettenti da sottoporre allo stadio successivo i seguenti. A tal proposito, si noti che la configurazione di rete i-esima corrisponde in realtà a due modelli addestrati con la relativa pipeline mentre, in figura 41 si riporta solamente quello con la migliore accuratezza ottenuta.

	Training		Validation			
Index	Loss	Accuracy	Loss	Accuracy	Performance [ms]	Memory [kB]
4	0.613	0.798	0.764	0.752	77.3	32.9
88	0.646	0.788	0.789	0.755	86.1	32.9
131	0.793	0.737	2.592	0.415	59.1	23.7
140	0.587	0.805	0.783	0.753	59.8	37.8

Figura 41: Modelli più promettenti

Procedendo con il secondo stadio, le 4 configurazioni di rete sono state addestrate incrementando il numero di epoche e ricorrendo alla data augmentation, consentendo così l'ottenimento dei seguenti risultati.

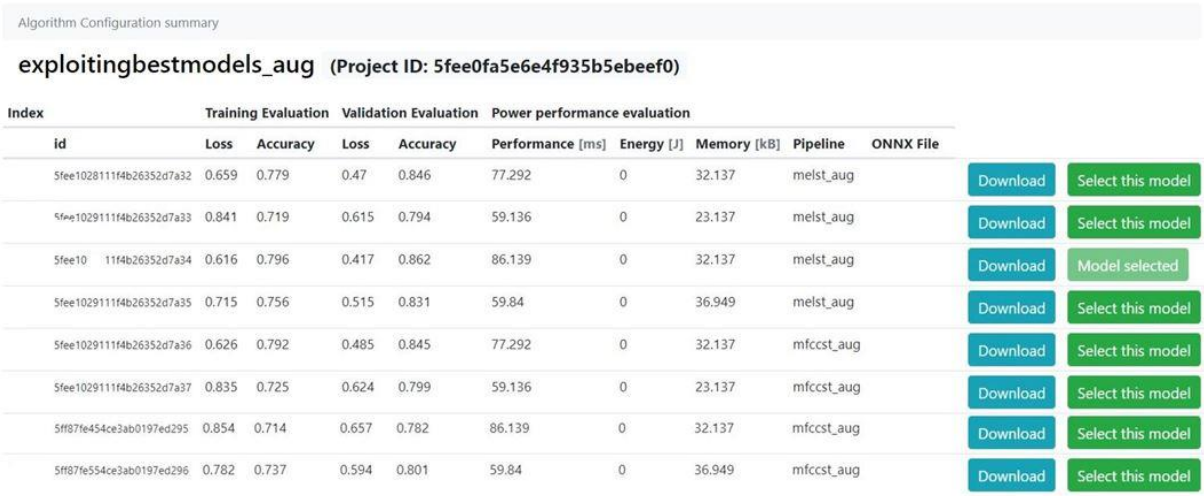


Figura 42: Risultati secondo stadio

Chiaramente, anche in questo caso avendo selezionato 2 distinte pipelines per il preprocessing dei dati, di fatto sono stati addestrati 8 modelli. Tra questi, come risulta indicato in figura 42, è stato scelto come migliore quello che ha una accuratezza in validation, rappresentativa quindi della capacità del modello in un contesto reale, dell'86.2 %, impiega 86.1 ms per inferenza ed occupa solamente 32.9 kB di memoria.

Seppur risulti irrilevante ai fini della prosecuzione dello use case, si ritiene opportuno riportare un benchmark delle 4 configurazioni di rete più promettenti con esse stesse, ovvero con e senza il ricorso alla data augmentation.

	Training		Validation			
Index	Loss	Accuracy	Loss	Accuracy	Performance [ms]	Memory [kB]
1	0.613	0.798	0.764	0.752	77.3	32.9
2	0.646	0.788	0.789	0.755	86.1	32.9
3	0.793	0.737	2.592	0.415	59.1	23.7
4	0.587	0.805	0.783	0.753	59.8	37.8

Whitout data augmentation

	Training		Validation			
Index	Loss	Accuracy	Loss	Accuracy	Performance [ms]	Memory [kB]
1	0.626	0.792	0.485	0.845	77.3	32.9
2	0.616	0.796	0.417	0.862	86.1	32.9
3	0.835	0.724	0.624	0.799	59.1	23.7
4	0.715	0.756	0.515	0.831	59.8	37.8

Whit data augmentation

Figura 43: Benchmark per la valutazione dell'effetto della data augmentation

Come si evince da figura 43, memoria e velocità di esecuzione restano le medesime non dipendendo dalle modalità di addestramento bensì dalla struttura della rete. Invece, dal punto di vista dell'accuratezza si nota come tutti i modelli, a seguito della data augmentation, subiscono un netto miglioramento in validation divenendo così più "bravi" a generalizzare su dati non noti, specie quello rappresentato da un indice pari a 3 il quale, riesce a passare dal 41,5% al 79.9%.

Prima di passare allo step 2 della toolflow ALOHA, un'ultima prova è stata eseguita sul modello ottimo individuato al secondo stadio. A partire dal modello già addestrato, sono state eseguite ulteriori 50 epoche con la medesima pipeline, riducendo però dal 50% al 30% la probabilità che i vari plugins di data augmentation vengano applicati al dato in input i-esimo, in modo tale da renderla meno "aggressiva". Attraverso quest'ulteriore attività addizionale è stato possibile incrementare l'accuratezza di circa 1.5 punti percentuali raggiungendo l'87,7% come mostrato in figura.

Figura 44: Caratteristiche modello ottimo

A scopo meramente illustrativo si riporta qui di seguito una sua rappresentazione.

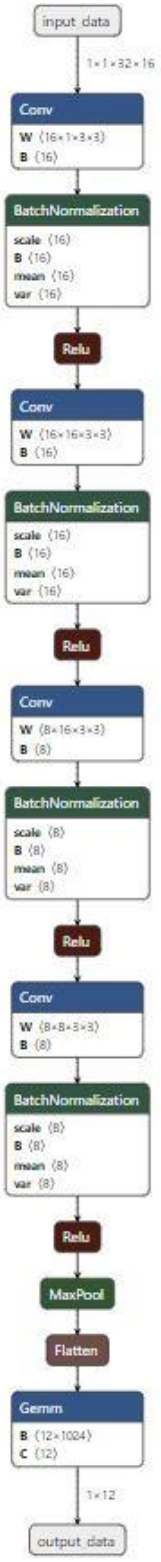


Figura 45: Rappresentazione modello ottimale

Ora che il modello migliore tra quelli esplorati è stato individuato, è possibile passare allo step 2.

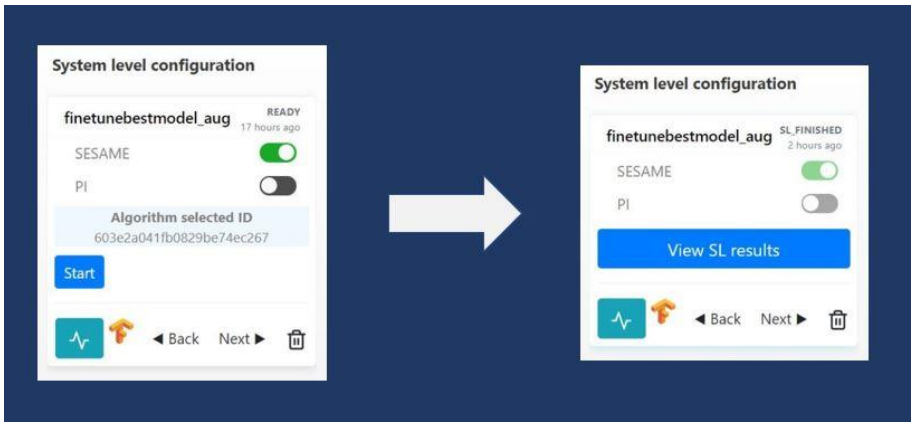


Figura 46: Esecuzione step 2

Attraverso l'utilizzo del tool *SESAME*, in modo completamente automatizzato viene ricercata ed attuata la mappatura ottimale del modello rispetto al target di riferimento. Si noti che questa attività risulta particolarmente utile ed interessante quando si ha a che fare con architetture piuttosto complesse, magari multi core, consentendo di definire su quale core dovrà essere eseguito lo specifico layer migliorando così le performance. A tal proposito, si riporta qui di seguito la mappatura suggerita relativamente al nostro use case.

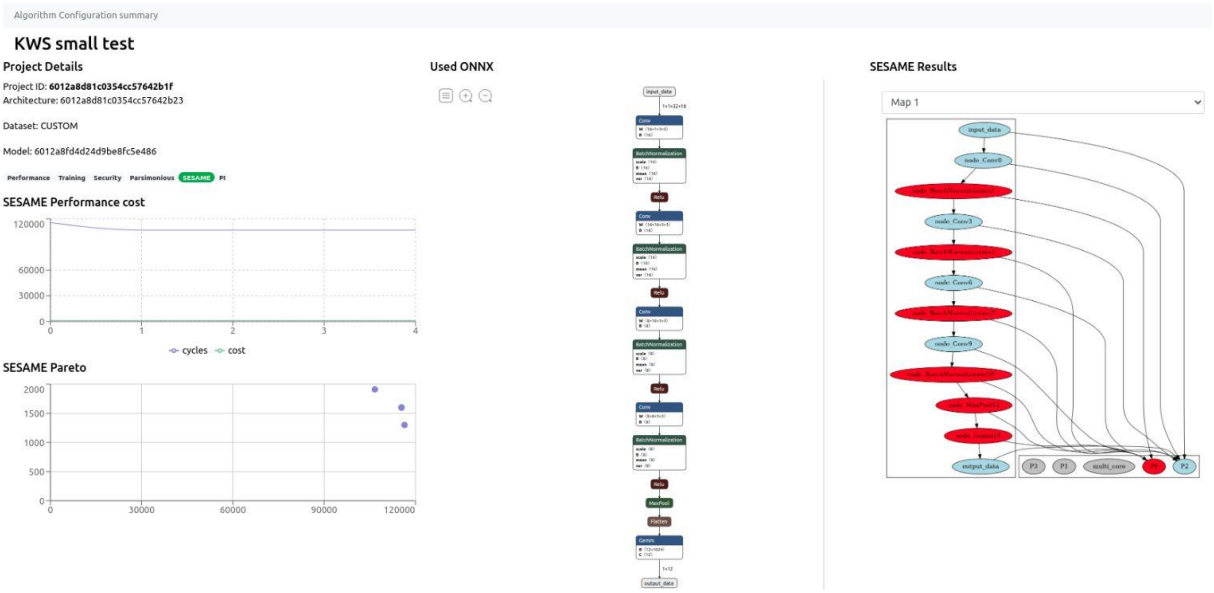


Figura 47: Mappatura ottimale del modello sulla SensorTile

A questo punto, non resta che eseguire il terzo ed ultimo step volto alla generazione del codice, al fine di rendere la soluzione pronta per essere implementata sul sistema embedded di riferimento, ergo la SensorTile nel caso in questione.

Figura 48: Esecuzione step 3

Al pari dello step 2, anche lo step 3 è eseguito in modo completamente automatico. Nello specifico, attualmente sono supportati 4 possibili tools, ciascuno specifico in base al tipo di esigenza. In particolare, per il seguente use case si è scelto di utilizzare *NEURAGHEGP* che consente una semplice implementazione in linguaggio C adatta a qualsiasi processore general purpose.

Terminata la parte sperimentale, per completare lo use case ed assolvere all'obiettivo prefissato, ovvero dimostrare i vantaggi cliente nell'adozione della tecnologia ALOHA, si rende necessaria la valutazione delle KPI's introdotte in precedenza e il confronto con le evidenze empiriche tratte dalla TensorFlow Speech Recognition Challenge, rappresentative dello stato dell'arte. A tal proposito, si riporta qui di seguito una tabella riassuntiva.

Tabella 3. Valutazione KPI's

Indicatore	Target	Effettivo	Stato dell'arte
KPI 1_1 Tempo progettazione	Giorni	9 giorni	60 giorni
KPI 1_2 Tempo ottimizzazione	Ore	5 ore	N.A.
KPI 1_3 Tempo generazione codice	Ore	2 ore	N.A.
KPI 2_1 Accuratezza	80%	87.7%	91.06%
KPI 2_2 Memoria	40 kB	32.137 kB	N.A.
KPI 2_3 Tempo esecuzione inferenza	130 ms	86.139 ms	N.A.

Dal confronto dei valori effettivi, ottenuti a seguito dell'espletamento dello use case, rispetto a quelli target definiti ex-ante, si nota come questi risultino in linea con le aspettative. Mettendo invece a confronto le evidenze empiriche ottenute rispetto a quelle tratte dalla challenge, emerge che sebbene il modello proposto presenti qualche punto percentuale in meno rispetto al top performer su Kaggle, in un tempo decisamente ridotto, nello specifico di un fattore 6 passando da 60 a 10 giorni circa, non solo è stato possibile progettare il modello bensì è stato ottimizzato rispetto alla SensorTile ed infine generato l'eseguibile ottenendo così una soluzione pronta per essere implementata.

Capitolo 4

Conclusioni

L'elaborato di tesi ha avuto il duplice scopo di definire un possibile posizionamento di mercato della toolflow ALOHA e di ideare ed implementare un caso studio al fine di valutare e dimostrare i benefici che derivano della sua adozione.

Innanzitutto, dall'integrazione di numerosi fonti d'informazione è stato possibile documentarsi sul tema del Deep Learning, comprendere il problema inerente all'implementazione su sistemi embedded e familiarizzare con la tecnologia ALOHA ed il suo funzionamento. In questo modo, è stato possibile definire la proposta di valore, a partire dalla quale è stato valutato il product market fit.

Mediante ricerca secondaria sono stati ipotizzati come potenziali segmenti di mercato: individui, startups DL-oriented, SME's e grandi imprese rispondenti a precisi requisiti. Altresì, sono stati individuati come prioritari i settori manufacturing, healtcare, financial services e retail mentre come area geografica l'Europa.

Analogamente, dall'analisi del contesto competitivo è emerso che il mercato risulta decisamente frammentato e popolato da attori quali alcune delle principali big tech oltre che da molteplici nuovi entranti. Tuttavia, a seguito delle peculiarità che la contraddistinguono, si evince come ALOHA si differenzi dalle maggior parte delle piattaforme esistenti. Tra quelle più vicine agli obiettivi di ALOHA sono state individuate: OctoML, Comet, Deep Cognition, Clarifai, Neural Designer, Allegro AI e Neural Network Intelligence. Approfondendo in modo analitico le varie proposte commerciali sono emersi molteplici punti deboli per molte delle società sopra citate, concludendo così come ALOHA possa colmare le principali lacune rivelandosi essere tra le migliori soluzioni ad oggi presenti sul mercato.

Relativamente al caso studio, per prima cosa è stata giustificata la scelta del keyword spotting come dominio di riferimento. Dopodiché, è stato definito l'obiettivo, ovvero quantificare i benefici derivanti dall'utilizzo della toolflow ALOHA, dimostrando la sua valenza rispetto ad un approccio manuale. Successivamente, è stato definito un opportuno set di KPI's allo scopo di valutare le performance, fornire supporto decisionale e poter affermare il successo o meno del caso studio.

Per poter disporre di un termine di paragone è stata replicata la TensorFlow Speech Recognition Challenge promossa su Kaggle, la più grande community al mondo di data science. Se essa ha rappresentato il punto di partenza, consentendo di definire il tipo di applicazione e una stima dei tempi necessari al solo sviluppo del modello mediante approccio manuale, con ALOHA è stato esteso il punto di arrivo introducendo il tema dell'implementazione su hardware. A tal proposito, è stato scelto come sistema embedded target la SensorTile.

Dal punto di vista metodologico, dapprima sono stati esplorati velocemente, mediante gridsearch, 288 potenziali modelli al fine di individuare quelli più promettenti. Quelli evidenziati come tali sono stati sottoposti ad un training più inteso, caratterizzato da un maggior numero di epoche e dal ricorso alla data augmentation, incrementandone così le performance. Infine, analizzando i risultati è stato individuato il modello migliore sul quale è stato attuato un ulteriore training. Scelto il modello, mediante gli step 2 e 3 della toolflow ALOHA, in modo del tutto automatico, è stata ricercata e applicata la mappatura ottimale sulla SensorTile così come è stato generato il codice eseguibile rendendo l'applicazione pronta per essere distribuita.

La soluzione così ottenuta garantisce un'accuratezza dell'87.7 %, impiega 86.1 ms per inferenza ed occupa solamente 32.9 kB di memoria. Dalla valutazione dei KPI's emerge che i risultati ex-post sono perfettamente allineati con i valori target definiti ex-ante, mentre, dal confronto con le evidenze empiriche tratte dalla challenge, si evince che l'utilizzo di ALOHA consente in un tempo 6 volte inferiore, non solo di ottenere un'accuratezza prossima allo stato dell'arte ma una soluzione pronta per essere implementata su hardware.

Concludendo, si può affermare con assoluta certezza che nonostante sia ancora in corso l'attività di sviluppo della toolflow ALOHA, al fine di introdurre nuove funzionalità e perfezionare parte di quelle esistenti, già nella versione attuale ha dato prova della sua utilità, facilitando notevolmente l'implementazione di algoritmi di Deep Learning su sistemi embedded.

Riferimenti

Amershi, S., et al. (2019). *Software Engineering for Machine Learning: A Case Study*. https://www.microsoft.com/en-us/research/uploads/prod/2019/03/amershi-icse-2019_Software_Engineering_for_Machine_Learning.pdf

Appugliese, C., Nathan, P., and Roberts, W. (2019) *Agile AI: A Practical Guide to Building AI Applications and Teams.* https://higherlogicdownload.s3.amazonaws.com/IMWUC/ba913cda-39b0-4d81-8552-6a9f6ccedb3f/UploadedImages/9781492074953.pdf

Cam, A., Chui, M., and Hall, B. (2019). *Global AI Survey: AI proves its worth, but few scale impact.* https://www.mckinsey.com/featured-insights/artificial-intelligence/global-ai-survey-ai-proves-its-worth-but-few-scale-impact

Chavan, K., and Gawande, U. (2015). *Speech Recognition in Noisy Environment, Issues and Challenges: A Review.* https://ieeexplore.ieee.org/document/7292420

Chui, M., et al. (2020). *An executive's guide to AI.* https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/an-executives-guide-to-ai

Costello, K. (2020). *Gartner Predicts the Future of AI Technologies.* https://www.gartner.com/smarterwithgartner/gartner-predicts-the-future-of-ai-technologies/

Emergen Research. (2020). *Deep Neural Network Market Worth USD 5.98 Billion By 2027.* https://www.emergenresearch.com/press-release/global-deep-neural-networks-market

Fujimaki, R., (2020). *AutoML 2.0: Is the Data Scientist Obsolete?* https://www.forbes.com/sites/cognitiveworld/2020/04/07/automl-20-is-the-data-scientist-obsolete/?sh=5fffbbf953c9

Goasduff, L. (2020). *2 megatrends dominate the Gartner hype cycle for artificial intelligence.* https://www.gartner.com/smarterwithgartner/2-megatrends-dominate-the-gartner-hype-cycle-for-artificial-intelligence-2020/

Goodfellow, I., Bengio, J., and Courville, A. (2016). *Deep Learning.* https://www.deeplearningbook.org/

Google Brain. (2018). *TensorFlow Speech Recognition Challenge: Can you build an Algorithm that understands simple speech commands?* https://www.kaggle.com/c/tensorflow-speech-recognition-challenge

Heaton, J., (2020). *Applications of Deep Neural Network.* https://arxiv.org/pdf/2009.05673.pdf

LeCun, Y., Bengio, J., and Hinton, G. (2015). *Deep Learning.* https://doi.org/10.1038/nature14539

Liangzhen, L., Naveen, S., and Vikas, C. (2018). *CMSIS-NN: Efficient Neural Network Kernels for Arm Cortex-M CPUs.* https://arxiv.org/abs/1801.06601

Liu, S., (2020). *Number of AI startup by country.* *https://www.statista.com/statistics/942657/global-ai-startups-by-country/*

Liu, S., (2021). *Global voice recognition market size 2019 and 2025.* https://www.statista.com/statistics/1133875/global-voice-recognition-market-size/

Market Research Future. (2021). *Global Artificial Intelligence as a Service (AIaaS) Market, By Technology, By Vertical – Forecast till 2023. Component, By Application, By End User – Forecast till 2023.* https://www.marketresearchfuture.com/reports/ai-as-a-service-market-7059

Market Research Future. (2019). *Global Deep Learning Market Research Report: By Component, By Application, By End User – Forecast till 2023.* https://www.marketresearchfuture.com/reports/deep-learning-market-6058

Meloni, P., et al. (2019). *Optimization and deployment of CNNs at the edge: the ALOHA experience.* https://dl.acm.org/doi/10.1145/3310273.3323435

Mordor Intelligence. (2020). *Deep Learning Market: Growth, Trends, Forecast 2020-2025.* https://www.mordorintelligence.com/industry-reports/deep-learning

Nisioti, E., (2018). *Automated machine learning: a different notion of deep.* https://towardsdatascience.com/automated-machine-learning-a-different-notion-of-deep-e0f7e5c06fb2

Osservatorio Artificial Intelligence. (2021). *All in: puntare sull'intelligenza artificiale per la ripresa del sistema Paese* https://www.osservatori.net/it/prodotti/formato/video/all-in-puntare-intelligenza-artificiale-ripresa-sistema-paese-video

Pant, A. (2019). *Workflow of a Machine Learning project.* https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94

Palumbo, F., et al. (2020). *Aloha project: software framework for runtime-adaptive and secure deep learning on heterogeneous architectures.* https://www.aloha-h2020.eu/project/project-overview

Perrault, R., et al. (2019). *The AI Index 2019 Annual Report.* https://hai.stanford.edu/sites/default/files/ai_index_2019_report.pdf

Ransbotham, S., et al. (2020). *Expanding AI's Impact With Organizational Learning.* https://sloanreview.mit.edu/projects/expanding-ais-impact-with-organizational-learning/

Rao, A., and Verweij, G. (2017). *Sizing the Prize.* https://www.pwc.com/gx/en/issues/analytics/assets/pwc-ai-analysis-sizing-the-prize-report.pdf

Technavio. (2020). *Deep Learning Market by Type and Geography Forecast and Analysis 2020-2024.* *https://www.technavio.com/report/deep-learning-market-industry-analysis*

Urlini, G., and Loi, D. (2019). *D6.2: Exploitation plan – First Update.* https://www.aloha-h2020.eu/images/Deliverables/D62.pdf

Warden, P. (2018). *Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition.* https://arxiv.org/abs/1804.03209

Yuan, L. (2020). *A Brief History of Deep Learning Frameworks.* https://syncedreview.com/2020/12/14/a-brief-history-of-deep-learning-frameworks/

Yundong, Z., et al. (2018). *Hello Edge: Keyword Spotting on Microcontrollers.* https://arxiv.org/abs/1711.07128

Elenco delle figure