Unità Sensore Intelligenti per Applicazioni Industriali

Danilo Pau, Giusy Tomarchio, Alessandra Di Pietro
System Research and Applications
STMicroelectronics, Italy
danilo.pau@st.com, giusy.tomarchio@st.com, alessandra.dipietro@st.com

Abstract

Questo documento presenta le soluzioni hardware e software che STMicroelectronics ha sviluppato nel contesto dell'industria 4.0. In particolare esse si sviluppano intorno alle applicazioni della manutenzione predittiva delle macchine.
Vengono descritti soluzioni hardware e software utilizzati e dimostrati da STMicroelectronics e da un partner a Electronica 2018 che sono estendibili a diversi casi d'uso, nonchè le modalità di implementazioni di una rete neurale a livello di microcontrollore del nodo sensore.
In particolare la soluzione STEVAL-BFA001V1B è un reference design kit per applicazioni come il monitoraggio delle condizioni operative; la soluzione STM32Cube.AI è il tool, unico nel suo genere, che aumenta la produttività degli sviluppatori embedded generando automaticamente codice Ansi C a partire da reti neurali pre-addestrate.

1 Introduzione

L'Intelligenza Artificiale (AI) nacque dalla domanda che Alan Turing si pose nel 1950 [1]: *puo' una macchina pensare ?* Poiche, sfortunatamente, una definizione accurata di pensiero umano sfugge tutt'ora, Turing intuendo la difficoltà riformulo' la domanda chiedendosi se potessero esistere calcolatori digitali capaci di imitare l'essere umano qualora gli fosse assegnato un compito, come ad esempio rispondere a domande o riprodurre il parlato etc.
Piu' recentemente l'AI è stata definita come la capacità assegnata ad un calcolatore digitale di acquisire conoscenza dai dati, approssimando funzioni di trasferimento nascoste in essi e non lineari, mostrando capacita di adattare il proprio apprendimento ad un ambiente tempo variante.
Da molti l'AI è considerata una singolarità tecnologica che cambierà molto presto il contesto industriale nella gestione delle informazioni, a prescindere da quale sia la sorgente che le genera.
In particolare le reti neurali artificiali (ANN) offrono l'opportunità di innovare pratiche di trattamento del segnale basate sulla programmazione di regole pensate dall'uomo e codificate in algoritmi di processamento. Grazie al moderno e produttivo linguaggio di programmazione python, ed ad una sempre maggiore disponibilità di librerie che astraggono i componenti base dell'ANN, sono stati costruiti numerosi ambienti di programmazione software che sono vastamente utilizzati dalle comunità di sviluppatori (Keras, Tensorflow, Pytorch Caffe, Lasagne, e molti altri), che offron primitive per trattare le topologie ANN e farle "apprendere" dai dati, in modalità supervisionata. Il risultato della procedura di "apprendimento" è un modello ovvero un insieme di coefficienti che parametrizzano i vari layers delle ANN altresi rappresentate da una propria topologia, composta da strati organizzati in grafi connessi che elaborano un algebra non lineare manipolando tensori.
Confortati dal teorema dell'approssimatore universale di Cybenko [2], l'ANN codifica una funzione di approssimazione accurata, non lineare che opera in uno spazio iperdimensionale. E' capace di apprendere la funzione di trasferimento dalla relazione nascosta tra i dati di ingresso e uscita di un qualunque sistena input-output. Questa relazione puo' essere cosi complessa dal renderla addirittura impossibile da modellizzare analiticamente. Utilizzando questo approccio le ANN trasferiscono la complessita della creazione di modelli analitici in modelli auto-appresi dai dati, a patto che si sia capaci di architettare una toplogia che massimizzi l'accuratezza aspettata, di raccogliere i notevoli quantità di dati e etichettare i risultati aspettati, di inzializzarle correttamente e portale a convergenza rispetto ad un criterio obiettivo.
Le ANN sono state ampiamente applicate in numerosi contesti quali ad esempio l'elaborazione delle immagini e della voce. Recentemente le ANN sono state utili anche in applicazioni nel settore industriale dove l'introduzione di nuove pratiche come la manutenzione predittiva possono rivoluzionare le attuali pratiche basate su controlli a scadenze temporali fisse. Altresi l'automazione industriale, cosi come la costruzione di ambienti di lavoro intelligenti possono creare opportunità che nuovi attori impresa potrebbero cogliere.
Poiche nel settore industriale, è necessario impiegare diverse tecnologie per monitorare lo "stato di salute" dei macchinari, essendo più sensori montati su ogni singola macchina, l'ammontare di dati sensori da inviare per una successiva elaborazione nel cloud, non solo richiede notevoli capacità di calcolo, ma anche onerosi capacità di banda. Ad esempio supponendo di utilizzare 4 accellerometri tri-assiali per monitorare le vibrazioni dei cuscinetti di un albero motore rotante a 2000rpm con una frequenza di campionamento di 20 KHz si possono generare 41.472 Gbyte al giorno da inviare al cloud, nell'ipotesi di un

monitoraggio constante. Questa criticità, unità alla necessita di elaborare in tempo reale i dati onde generare un'attuazione a bassisima latenza, suggerisce che i dati debbano essere valutati immediatamente dopo la loro generazione e non nel could onde poter prendere decisioni rapide. La validità di questa proposizione è ancor maggiore qualora vengano utilizzati una moltitudine di sensori, come nelle aspettative di crescita del mercato IoT. Poichè è eccessivamente dispendioso inviarli al cloud, è urgente elaborarli in modo decentralizzato, ovvero làddove sono generati. Conseguentemente alcune decisioni possono essere prese localmente onde evitare alte latenze di counicazioni con il cloud riducendone drammaticamente il carico computazionale e il consumo di energia.
Cosi', l'elaborazione delle informazioni con l'AI è delegata a unità sensori dotate di risorse limitate (per esempio 100KB – 1 MB RAM, 1-2 MB Flash e 100MHz di frequenza di calcolo) e poste direttamente a contatto o prossime al sensore stesso in modo da costituire un nodo sensore intelligente [3, 4] realizzando la cosidetta Edge AI .
In un esempio applicativo, cio' significa che una unità sensore di vibrazioni posto vicino ad un cuscinetto a sfere di un asse ferroviario possa essere dotato dell'intelligenza necessaria per rilevare in largo anticipo e prima che si verifichino, eventuali danni o un affaticamento precoce del materiale (la base tecnologica per la manutenzione predittiva). Successivamente queste informazioni compatte possono essere quindi inviate ad un sistema superiore di elaborazione ad esempio nel cloud. Inoltre in esso vengono raccolte e valutate le informazioni provenienti da tutti i sensori posti sui diversi assi dei vagoni del treno, monitarando cosi le condizioni dell'intero treno o addirittura del tratto binario persorso. Cosi, i sistemi di monitoraggio continuo e di manutenzione predittiva comprendono numerosi nodi di sensori intelligenti collegati tramite una rete composta da un'architettura stratificata su più livelli composti da numerose entità come uno o più coordinatori, gateway connessi ad un server oppure direttamente connessi ad un servizio cloud come mostrato nella Figura 1.

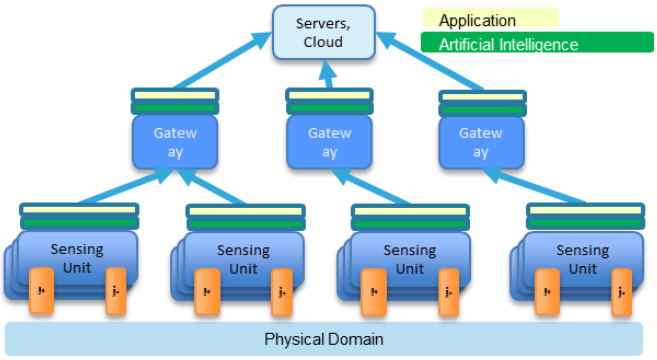

Figura 1 Architettura generale

Qualora i dati fossero elaborati direttamente dai nodi sensori cio' consentirebbe una capacità di risposta a bassa latenza delegando a server remoti o infrastrutture cloud, elaborazioni in grado di correlare dati astratti generati in un arco temporale esteso, richiedenti capacità di storage notevoli, da unità sensore distribuite.
Poichè la risorsa di calcolo colocata al sensore e integrabile in un nodo compatto è un microcontrollore, esso puo' deve fornire capacità di elaborazione locale generando una ridotta quantità di informazioni da comunicare. Il principale compito assegnato al microcontrollore è l'esecuzione dell'ANN e a partire da un pre-processamento di dati sensori sia nel dominio del tempo che in quello delle frequenze (ad esempio basato su Fast Fourier Transform) sempre eseguito dal microcontrollore.
L'apprendimento dell'ANN è dominato dalla disponibilità di set di dati ben annotati, e inoltre il disegno e l'addestramento dell'ANN è ottenuto mediante framework open source quali Keras, TensorFlow, Caffe etc i quali consentono di creare ANN con architetture e complessità differenti. Per l'analisi delle informazioni tempo varianti, ad esempio, sono utilizzate reti con feedbacks ("Rete neurale ricorrente", RNN) mentre per quello non tempo varianti si usano reti senza feedbacks ("Reti convolutive", CNN).

2 L'unità sensore

Un esempio di unità sensore integrato stato dell'arte è basato sulla scheda di valutazione dal fattore di forma compatta (50mmx9mmx9mm) denominata STEVAL-BFA001V1B [3] mostrata in figura 2. E' un kit progettato per il "Condition Monitoring" (CM) e la manutenzione predittiva (PdM). Il kit di sviluppo hardware è costituito da una scheda multi sensore (STEVAL-IDP005V1), un adattatore per lo strumento di programmazione e debug ST-LINK / V2-1 (STEVAL-UKI001V1) e cavi vari utili per l'utilizzo. E' dotato di sensori di movimento, ambientali e acustici in tecnologia MEMS dedicati al monitoraggio dei parametri chiave "dello stato di salute dei macchinari" necessari per la manutenzione predittiva. Il corredo di programmazione include algoritmi dedicati per l'elaborazione avanzata del segnale nel dominio del tempo e della frequenza per l'analisi dell'accelerometro digitale 3D con larghezza di banda piatta a 3 kHz. Integra il microcontrollore a 32 bits STM32F469AI, dotato di unità ARM® Cortex®-M4, possiede una frequenza operativa di 180MHz, 2MB FLASH e 384+4 KB of SRAM con 64-KB RAM CCM. I risultati dell'analisi su base ANN dei dati del sensore sono trasmessi via connettività IO-Link o attraverso una VCOM per monitoraggio diretto su PC. Infine interfaccia utente (GUI) per PC (figura 3) consente di visualizzare i dati elaborati.
Una dimostrazione pratica di tale unità sensore si è tenuta ad Electronica, presso Monaco di Baviera dal 13 al 16 novembre 2018 nel contesto della manutenzione predittiva onde equipaggiare un sistema industriale come, ad esempio, un cuscinetto a sfere di un asse ferroviario.

3 De mistificare l'implementazione dell'ANN

Per molto tempo si è creduto che i microcontrollori in commercio non fossero in grado di addestrare ne implementare l'inferenza di una rete ANN a causa delle loro

limitate capacità di memoria e di calcolo. Questo credo è stato de mistifiacto grazie allo strumento software STM32Cube.AI il quale è stato concepito e utilizzato per convertire ANN pre addestrate in codice ANSI C ottimizzato per i microcontrollori STM32 dotati di unità ARM® Cortex®-M4 e M7. Inoltre, STM32Cube.AI consente la verifica incrementale del codice generato, con il risultato che l'ANN eseguita dall'STM32 garantisce la stessa accuratezza raggiunta in fase di addestramento in virgola mobile.

Figura 2 Il kit STEVAL-BFA001V1B

Figura 3 STEVAL-BFA001V1B GUI

4 Applicazione Intelligente in 5 passi

Le fasi per la realizzazione di una applicazione AI sono essenzialmente 5.

4.1 Acquisire i dati

Nel contesto applicativo target, si sceglie un sensore (ad esempio un accelerometro o un microfono) dal quale catturare una quantità sufficiente di dati rappresentativi delle classi che descrivono il fenomeno (ad esempio diversi tipi di vibrazioni di un motore). Cio' di solito comporta posizionarlo a stretto contatto del fenomeno fisico da monitorare per registrare il suo stato e le variazioni temporali. Inoltre i parametri fisici da considerare possono essere molteplici come la temperatura, il suono o le immagini a seconda del caso di utilizzo, per cui può essere necessaria anche considerarli simultaneamente.
A tale fine è vitale fornire strumenti per immagazzinare e etichettare i dati in modo accurato e specifico. Questo puo' rappresentare l'80-90% del tempo di progetto.

4.2 Etichettare i dati e disegnare l'ANN

La creazione e l'addestramento di una rete neurale artificiale supervisionata richiede dati sensore accuratamente etichettati. Cosichè per l'"apprendimento supervisionato", il set di dati deve essere caratterizzato in modo che i risultati aspettati siano classificate correttamente per definizione. Questo set rappresenta la "**verità fondamentale**" in un sistema tempo non-variante che verrà utilizzato per addestrare l'ANN e successivamente per validarla. Lo sviluppatore è l'attore principale poichè deve decidere il tipo di topologia dell'ANN che deve poter meglio apprendere dai dati e fornire un utile output per l'applicazione target.

4.3 Addestrare un' ANN

Generalmente gli sviluppatori impiegano framework di apprendimento approfondito comuni ed open source come ad esempio Keras, Tensorflow, Caffe, Pytorch etc. per progettare e addestrare le ANN. Questo comporta il passaggio dei set di dati attraverso l'ANN in modo iterativo addestramento-validazione in modo che le uscite dell'ANN possano ridurre al minimo i criteri di errore desiderati rispetto alla "**verità fondamentale**". Inoltre l'addestramento viene fatto su potenti piattaforma di calcolo, GPU, TPU o CPU, con memoria e potenza di calcolo virtualmente illimitate, onde consentire molte iterazioni di apprendimento in un breve periodo di tempo. Il risultato di questo processo è l'ANN pre-addestrata. Il formato che la rappresenta, sfortunatamente non è ineroperabile con l'ambiente di sviluppo STM32. Si potrebbe eseguire la scrittura manuale del codice ANSI C che implementa l'ANN, tuttavia questo processo è laborioso da realizzare, manutenere e validare.

4.4 Convertire l'ANN in codice ottimizzato per STM32

Il passo successivo consiste nel mappare automaticamente l'ANN pre-addestrata su STM32 tale che il codice ottimizzato generato ed eseguito riduca al minimo i requisiti di complessità e memoria. Questa fase, che altrimenti sarebbe realizzata artigianalmente, è resa automatica e produttiva grazie allo strumento software STM32Cube.AI [5]. Il quale è completamente integrato nell'ecosistema di sviluppo STM32 come estensione del tool noto STM32CubeMX ampiamente utilizzato dagli sviluppatori per la configurazione e generazione di codice C/C++.
STM32Cube.AI guida gli utenti attraverso la selezione del STM32 adatto alla propria applicazione e fornisce un rapido riscontro sulle prestazioni implementative dell'ANN, essendo confortati da una procedura di validazione incrementale sia sul PC che sul STM32 molto precisa.

4.5 Processare nuovi dati con l'ANN

Nella fase finale si applica l'ANN mappata su STM32 nella propria applicazione e la si testa sul campo. Cosi' si semplifica la progettazione di nuovi prototipi, dedicando più tempo alla creazione degli stessi e delle associate applicazioni anche utilizzando pacchetti software funzionali [6] che rappresentano ottimi punti di partenza modificabli a piacere. Questi pacchetti sono esempi completi che incorporano una combinazione di driver di basso livello, librerie middleware e applicazioni di esempio. Infine gli sviluppatori possono anche ottenere supporto e scambiare idee nel forum AI della community STM32.

5 Un esempio di AI per la Manutenzione Predittiva

Grazie ad una collaborazione con Lenord + Bauer [7] si sono sviluppate soluzioni di intelligenza artificiale mediante STM32Cube.AI.
Il caso d'uso era definito nell'ambito delle strutture meccaniche su cui il treno interagisce con le rotaie e l'obiettivo era di riuscire ad identificare comportamenti anomali e relativo range di frequenza a cui si evidenziavano.
La caratterizzazione è stata effettuata tramite un banco prova in cui i disturBi venivamo simulati con un martelletto che agisva per generare vibrazioni sulla struttura.
Da un sensore di vibrazione, basato su un accelerometro MEMS e un microcontrollore STM32 incorporati nel sistema STEVAL-BFA001V1B, sono rilevati e classificati diversi tipi di segnali da una rete neurale direttamente addestrata sulla analisi in frequenza del sensore di vibrazioni. L'ANN è stata addestrata in Keras, il codice generato con STM32Cube.AI, compilato e installato sul microcontrollore per poi essere utilizzata per identificare la probabilità di comportamenti anomali.

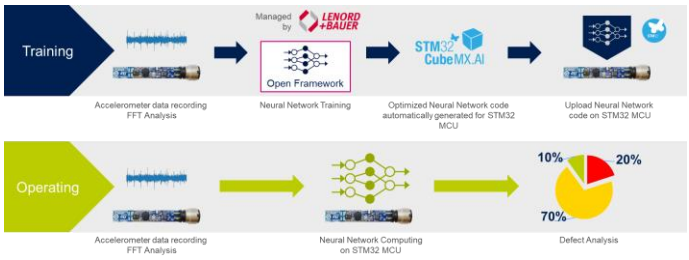

Figura 4 Metodologia di sviluppo mostrata a Electronica 2018

5 Conclusioni

Grazie alla prima di una nuova serie di soluzioni di STMicroelectronics per AI, la comunità di sviluppatori ha la libertà, da Gennaio 2019, di mappare ed eseguire reti neurali artificiali pre-addestrate sull'ampio portafoglio di microcontrolleri STM32. STM32Cube.AI è un pacchetto che estende STM32CubeMX per la generazione automatica di codice C/C++ che consente di realizzare l'AI su microcontrollori STM32 Arm® Cortex®-M-based (M4 e M7).
Le proprietà fondamentali di STM32Cube.AI sono:

1) L'interoperabilità con diversi strumenti di programmazione open source per l'apprendimento approfondito (Keras, Caffe, Lasagne etc.)
2) La compatibilità con gli IDE (IAR, Atollic, Keil etc) e compilatori ANSI-C maggiormente usati
3) È agonostico ai sensori e agli RTOS
4) Abilita ad eseguire più reti neurali artificiali su un singolo micr controllore STM32
5) Offre un supporto completo per micro controllori STM32 a bassissima potenza
6) Aumenta drammaticamente la produttività degli sviluppatori che possono ora focalizzarsi sul valore aggiunto ai propri clienti invece spendere onerosamente tempo nello scrivere codice fatto a mano.

Riferimenti bibliografici

[1, M. Turing, 1950] Computing Machinery and Intelligence. Mind 49: 433-460. https://www.csee.umbc.edu/courses/471/papers/turing.pdf

[2, G. Cybenko, 1989] Approximation by Superpositions of a Sigmoidal Function, Math. Control Signals Systems (1989) 2:303-314; Mathematics of Control, Signals, and Systems 9 1989 Springer-Verlag New York Inc. https://pdfs.semanticscholar.org/05ce/b32839c26c8d2cb38d5529cf7720a68c3fab.pdf

[3] Predictive maintenance kit with sensors and IO-Link capability; https://www.st.com/en/evaluation-tools/steval-bfa001v1b.html

[4] STM32CubeMx - STM32Cube initialization code generator: https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-configurators-and-code-generators/stm32cubemx.html

[5] X-CUBE-AI - AI expansion pack for STM32CubeMX: https://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-ai.html

[6] Le soluzioni STM32 per l'ANN https://www.st.com/content/st_com/en/stm32-ann.html

[7] STM32Cube.AI Edge AI demo with Lenord + Bauer (electronica 2018): https://www.youtube.com/watch?v=iHkEOHqmdb8