Apprendimento di Reti Neurali su Dispositivi a Risorse Limitate

Franco Maria Nardini[1], **Lorenzo Valerio**[2], **Andrea Passarella**[2], **Raffaele Perego**[1]
[1]ISTI-CNR Pisa, [2]IIT-CNR, Pisa.
f.nardini@isti.cnr.it, l.valerio@iit.cnr.it, a.passarella@iit.cnr.it, r.perego@isti.cnr.it

Abstract

La trasformazione digitale che stiamo vivendo negli ultimi anni è trasversale a tutti i settori della società. In ambito produttivo, tale trasformazione sta portando verso la quarta rivoluzione industriale caratterizzata da i) grandi quantità di dati raccolti e ii) decentralizzazione delle risorse computazionali lungo la linea produttiva. In tale contesto l'uso dell'intelligenza artificiale (AI) è spesso subordinato all'adozione di soluzioni distribuite caratterizzate dall'utilizzo di hardware di capacità limitate. In questo articolo descriviamo una tecnica di apprendimento di reti neurali ottimizzata per dispositivi con risorse limitate. Una prima sperimentazione su dataset MNIST conferma la bontà dell'approccio che consente di ridurre efficacemente la dimensione della rete durante l'addestramento senza perdite significative della precisione del metodo.

1 Introduzione

Le aziende italiane e mondiali stanno affrontando una importante transizione verso l'adozione del paradigma Industry 4.0. Tale paradigma assegna all'ICT, ed in particolare alla gestione ed analisi dei dati, un ruolo centrale nello sviluppo e nell'ottimizzazione del processo produttivo. Questo ruolo è abilitato da crescente capacità di calcolo distribuita su tutta la linea produttiva e da una sempre maggiore disponibilità di dati prodotti a tutti i livelli del processo produttivo. Questi due ingredienti sono alla base della derivazione di conoscenza utile al processo produttivo stesso. Si pensi ad esempio a due casi di studio tipici in Industria 4.0: i) ottimizzazione del processo, al fine di migliorare l'efficienza di una catena produttiva, ii) manutenzione predittiva di un impianto usando i dati di uso dell'impianto stesso.

Tuttavia, dato l'incremento esponenziale di dati prodotti e raccolti all'*edge* della rete, si prevede che la capacità delle infrastrutture di comunicazione e computazione (nonostante l'avvento del 5G e l'evoluzione dei data centre) non riuscirà a stare al passo con il "data tsunami" previsto nei prossimi anni, rendendo il paradigma basato unicamente su soluzioni cloud non più sostenibile nel medio-lungo periodo [Cisco Systems, 2018]. Inoltre, si stima che rispetto a tutto il volume di dati prodotto all'edge, solo una piccola parte (circa il 10%) avrà una valenza di carattere generale, mentre, la parte rimanente ha valore limitato all'area geografica o al momento temporale in cui è stato generato. A fronte di queste considerazioni, l'opportunità di affiancare a meccanismi di AI basati su sistemi cloud centralizzati, soluzioni di AI decentralizzate basate su edge e fog computing, secondo cui la computazione e l'estrazione di conoscenza da questa enorme mole di dati viene spostata verso l'edge della rete [Conti *et al.*, 2017].

In tale paradigma, i dispositivi all'edge della rete spesso sono caratterizzati da risorse molto limitate in termini di capacità di calcolo, memoria e rete, il che rende complicato eseguire tecniche di AI per derivare conoscenza da tali dati, a causa dei tipici requisiti di computazione di cui necessitano. Si pone quindi l'esigenza di sviluppare tecniche di AI che possano essere utilizzate su dispositivi a risorse limitate per poterne abilitare un utilizzo efficiente, in ambito Industria 4.0 e non solo, senza perdite significative di accuratezza. Più in particolare, questa necessità non comprende solamente la creazione di modelli di learning efficienti dal punto di vista del loro utilizzo, ma anche lo sviluppo di tecniche di *addestramento* che siano eseguibili in contesti in cui i dispositivi preposti non dispongano di molte risporse computazionali. Esempi di questi dispositivi sono: dispositivi mobili quali smartphone e tablet ma anche mini-pc come Intel Edison, Raspberry-Pi ecc.

In questo articolo ci focalizziamo su tecniche efficienti ed efficaci di machine learning specificamente pensate per dispositivi caratterizzati da risorse limitate. In dettaglio, le competenze che stiamo sfruttando, adattandole al contesto descritto, riguardano tecniche di compressione di reti neurali *deep* (DNN) che permettono una riduzione significativa, già durante la fase di apprendimento, della dimensione della rete neurale.

Gli approcci alla compressione (pruning) di reti neurali seguono di fatto due linee principali. Da una parte il pruning può essere realizzato eliminando alcuni dei neuroni di una DNN, mentre dall'altra, eliminando opportunamente i pesi ritenuti superflui. Allo stato attuale, la maggior parte dei lavori si focalizza principalmente sul secondo approccio, per cui sono state presentate tecniche basate i) sulla fattorizzazione delle matrici dei pesi, ii) sulla magnitudine e/o rappresentazione dei valori dei pesi e iv) metodi di *distillation*. La principale critica ai lavori appartenenti a questo gruppo è che la compressione avviene quasi sempre dopo la conclusione dell'addestramento della rete. Una via alternativa e relativamente

recente, invece, prevede di integrare la compressione direttamente durante la fase di apprendimento. Tipicamente, in tali approcci, si tenta di apprendere durante la fase di allenamento la distribuzione di attivazione (binaria) dei singoli neuroni finalizza alla rimozione degli stessi.

Negli scenari Industry 4.0 e IoT considerati in questo lavoro, quest'ultima linea appare promettente perchè apre alla possiblità di eseguire l'addestramento di modelli di DNN direttamente su dispositivi con risorse limitate. La progressiva diminuzione della dimensione del modello da addestrare può infatti tradursi in un risparmio incrementale di risorse del dispositivo che esegue l'addestramento. In questo articolo la nostra attenzione è focalizzata a questa classe di soluzioni.

2 Hard pruning di reti neurali

Per questo lavoro ci stiamo basando su un approccio presente in letteratura in cui il pruning della rete avviene apprendendo la distribuzione di attivazione/spegnimento dei neuroni della rete. In praticolare l'approccio proposto in [Louizos *et al.*, 2017] addestra una DNN minimizzando una funzione di costo composta dalla somma di due termini che sono funzione dei parametri della rete $\boldsymbol{\theta}$: i) l'errore di accuratezza della rete e ii) un regolarizzatore $\|\boldsymbol{\theta}\|_0$ che, tramite la norma L_0, consente di trovare l'insieme di parametri di minor cardinalità che al contempo massimizzi l'accuratezza della rete.

Dato che l'apprendimento della rete è effettuato tramite *backpropagation*, un algoritmo di ottimizzazione basato sulla discesa del gradiente, la funzione di costo composta dai due termini sopra indicati non è utilizzabile direttamente a causa della non derivabilità del regolarizzatore introdotto. Tale problema può essere superato mediante un approccio probabilistico: approssimando la distribuzione dei neuroni accesi e spenti attraverso una distribuzione continua e derivabile, *Hard Concrete Distribution*. Il risultato è che, legando opportunamente il parametro che determina la probabilità di accensione dei neuroni ai parametri della rete, diventa possibile minimizzare congiuntamente l'errore di predizione e il numero di neuroni attivi, epoca per epoca. Tuttavia, la riduzione introdotta da tale soluzione di *pruning* è solo virtuale e indicata comunemente con il termine *soft pruning*. Infatti, sebbene di epoca in epoca il numero di neuroni attivi decresca, a causa del processo di estrazione randomica, i neuroni attivi di epoca in epoca sono sempre differenti. Dal punto di vista dell'utilizzo di risorse, questo non garantisce nessun risparmio di memoria dato che la dimensione della rete rimane sempre costante.

Questo problema può essere risolto introducento un meccanismo di *hard pruning*. L'idea di base è che, durante l'apprendimento, se un neurone viene spento esso non possa più riaccendersi. Addestrare con hard pruning ha l'effetto di azzerare, a training time, interi gruppi di pesi e di conseguenza, parti di rete. Differentemente dal soft pruning, questo meccanismo permette di ottenere una efficace riduzione dell'occupazione di memoria della rete. Infatti, è sufficiente eliminare, per ogni livello della rete, le righe delle matrici dei pesi i cui neuroni sono azzerati per ottenere matrici con un numero di elementi inferiore. L'approccio che proponiamo per spegnere i neuroni è probabilistico: durante ogni epoca calcoliamo la probabilità empirica di accensione di ogni singolo neurone attraverso l'uso della Hard Concrete Distribution opportunamente normalizzata per il numero di estrazioni eseguite in ogni epoca. Dalle probabilità ottenute creiamo una maschera binaria che viene applicata alla rete durante le epoche successive. Questa maschera, aggiornata epoca per epoca, contribuisce a spegnere i nodi della rete precedentemente accesi mantenendo consistente lo stato globale dell'hard pruning durante l'addestramento.

3 Risultati preliminari e conclusioni

Abbiamo verificato l'efficacia dell'hard pruning descritto in Sezione 2 su un task di classificazione multi-label: il riconoscimento in una immagine di un numero scritto a mano. Si chiede di assegnare ad ogni immagine del dataset la classe che identifica il numero scritto a mano nell'immagine stessa. A tal proposito, si è impiegato il noto dataset MNIST [LeCun *et al.*, 1998] e abbiamo addestrato per 500 epoche un Multilayer Perceptron (MLP) con due hidden layer di 300 e 100 nodi rispettivamente. La dimensione iniziale della rete è di circa 266.610 parametri per un totale di circa 1 MiB di memoria occupata. I risultati della rete addestrata con hard pruning sono poi confrontati con quelli del MLP addestrato utilizzando soft pruning.

I risultati preliminari ottenuti ci dimostrano che, applicando hard pruning, è possibile ridurre di circa il 70% le dimensioni della rete durante il training con un degrado delle performance di accuratezza inferiore al 0.2% rispetto all'addestramento di una rete equivalente che utilizza soft-pruning.

Il raffinamento del metodo proposto ed una sua sperimentazione su larga scala sono gli obiettivi per i prossimi mesi di lavoro. Le competenze descritte sono maturate all'interno di due progetti di ricerca H2020 e di un progetto PON (bando Fabbrica Intelligente): i) Wireless Autonomous, Reliable and Resilient ProductIon Operation ARchitecture for Cognitive Manufacturing (AUTOWARE, GA # 723909), ii) Big Data to Enable Global Disruption of the Grapevine-powered Industries (BigDataGrapes, GA # 780751), iii) Operational Knowledge from Insights and Analytics on Industrial Data (OK-INSAID, ARS01_00917).

Riferimenti bibliografici

[Cisco Systems, 2018] Inc. Cisco Systems. Cisco Global Cloud Index: Forecast and Methodology, 2016–2021. *White Pap.*, page 46, 2018.

[Conti *et al.*, 2017] Marco Conti, Andrea Passarella, e Sajal K. Das. The Internet of People (IoP): A new wave in pervasive mobile computing. *Pervasive Mob. Comput.*, 41:1–27, 2017.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Louizos *et al.*, 2017] Christos Louizos, Max Welling, e Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.