Internet of Things based smart systems

Valutazione dell'impatto della riduzione della precisione dei pesi in una rete neurale nei confronti dell'accuratezza di classificazione

Alessandro Messina, matricola O55000354
Orazio Scavo, matricola O55000414

ANNO ACCADEMICO 2018/2019

Introduzione

Approximate Computing

- Riscoperta dell'Approximante Computing
- Motivazioni
  - Necessità di nuove tecniche per migliorare le performance
  - Elevato numero di applicazioni con *forgiving nature*
  - Tecnica per ridurre il consumo energetico
    - Aumento dell'autonomia dei dispositivi mobili
- Approximate Computing nelle Neural Networks
  - Forgiving nature
  - Self healing grazie al retraining

Obiettivi

- Problemi delle Neural Network in dispositivi vincolati
  - Elevata quantità di memoria occupata per la memorizzazione dei pesi
  - Necessità di risorse computazionali elevate per il training della rete
  - Elevata energia dissipata per effettuare i calcoli
- Obiettivi
  - Approssimazione del numero di bit necessario per rappresentare i pesi
    - Possibilità di ridurre gli elementi circuitali per effettuare i calcoli (sommatori, moltiplicatori).
      - Area occupata minore
      - Energia dissipata minore
      - Frequenza di clock maggiore
  - Valutazione del tradoff tra memoria risparmiata e perdita di accuratezza

Ambito applicativo

- Applicazione IoT per la gestione di un parcheggio con fotocamere
  - Necessità di rilevare oggetti non opportuni all'interno del parcheggio
  - Necessità di rilevare i posti realmente liberi

- Uso di una CNN per l'object recognition
  - Rilevamento degli oggetti a partire dall'immagine

Flusso dell’analisi

1. Progettazione, allenamento e valutazione dell’accuratezza della rete originale
2. Applicazione dell’Approximate Computing sulla rete realizzata, spiegazione delle scelte relative alle approssimazioni effettuate sulla rete e esposizione dell’applicazione NNAXIM
3. Confronto dei risultati ottenuti per le diverse configurazioni
4. Conclusioni sul lavoro svolto

Analisi della rete originale

Progettazione della rete neurale

- Ricerca della rete neurale per la classificazione
- TinyDNN
- CIFAR 10
  - Dataset: 60000 immagini 32x32 (training e test)
  - Classificazione di: automobili, camion, aeroplani, navi, gatti, cervi, cani, rane, cavalli, uccelli
- Struttura della rete:
  - 3 blocchi di layer (input 32x32):
    - Layer convoluzionale
    - Pooling layer
    - Relu activation layer
  - Fully-connected layer
  - Relu activation layer
  - Fully-connected layer
  - Softmax layer (classificazione)

Scelte operative

- Realizzazione del tool NNAXIM
- Complessità elevata della rete
  - Difficoltà nel training e nel test della rete
  - Tempi di sviluppo troppo lunghi senza il giusto hardware
- Lavoro su rete più semplice
  - Approssimazione di una funzione sinusoidale (sinus fit, TinyDNN)
- Porting su CIFAR 10
  - Singola esecuzione dell'algoritmo per ottenere i risultati

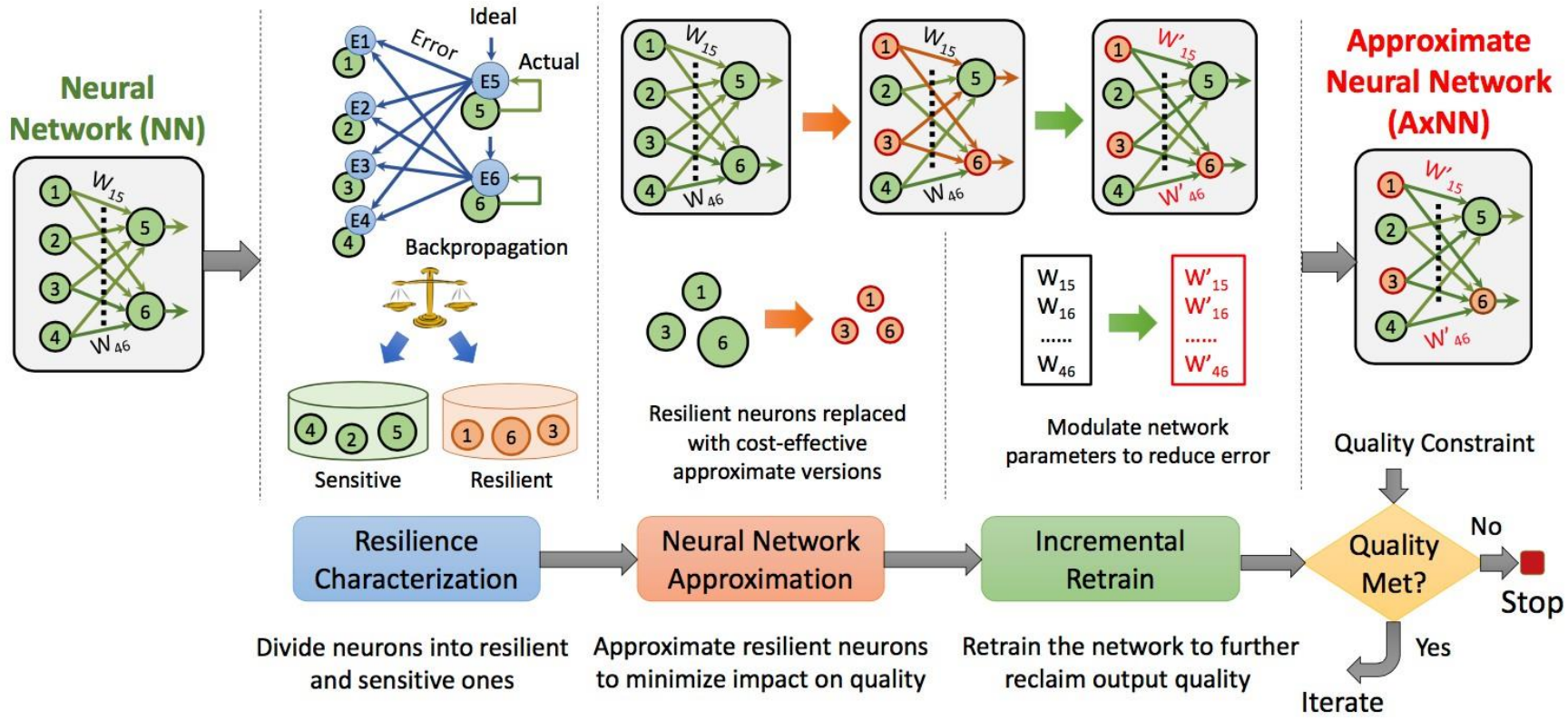
Valutazione dell’accuratezza della rete

- Accuratezza nel caso di classificazione:
  - Percentuale di classificazioni corrette su quelle totali effettuate sul dataset di test
  - Accuratezza originale: intorno al 70% (68.20%)

Approximate Computing sulla rete neurale realizzata

Premessa

- Approccio ideale per l'applicazione dell'approximate computing



Premessa

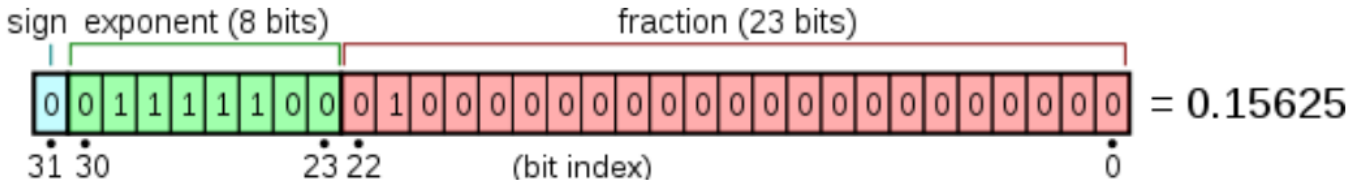
- Uso dei risultati dei papers per la caratterizzazione della resilienza
- Mancanza di hardware e tempo per usare l'approccio ideale
- Scelta di 9 configurazioni
  - Basata sullo studio dello standard per la rappresentazione dei float

Note sullo Standard IEEE sulla rappresentazione dei float

- Motivazioni
  - Simulazione dell'uso di float approssimati in NNAXIM
- Codifica dei float
  - Segno: 1 bit
  - Esponente: 8 bit (intero con segno in complemento a 2)
  - Mantissa o significando: 23 bit (24, il bit *i* è implicito)

$$VAL = (-1)^{Segno} \cdot 2^{Esponente-127} \cdot [i, < Significando >]_{base2}$$

$$VAL = (-1)^{Segno} \cdot 2^{esponente-127} \cdot \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i}\right)$$



Scelta delle configurazioni approssimate

- Approssimazione tramite arrotondamento
  - Dimezzamento dell'errore introdotto
- Scelta dell'entità dell'approssimazione sulla base dell'errore
  - Calcolo dell'errore introdotto dalle varie approssimazioni
    - Pesi della rete dell'ordine di grandezza delle unità
    - Errore molto basso nell'approssimazione
  - Ambito applicativo (classificazione)
  - Entità dell'approssimazione da 16 a 21 bit
- Approssimazione sui layer della rete
  - Resilienza degli hidden layer
  - Sensibilità dei layer di output

Configurazioni approssimate

Configurazione	Numero di bit di ogni peso dei neuroni dei layer di input e output	Numero di bit di ogni peso dei neuroni degli hidden layer
Originale	32	32
1	16	16
2	14	14
3	12	12
4	32	16
5	32	14
6	32	12
7	16	14
8	14	12
9	12	11

Tabella 2. Configurazioni scelte per le approssimazioni del numero di bit dei pesi.

Flusso dell'approssimazione: algoritmo per l'approximate computing

1. Approssimazione dei pesi secondo la configurazione scelta
2. Test dopo l'approssimazione
   - Valutazione dell'accuratezza della rete
3. Retraining
   - Per mitigare l'effetto dell'approssimazione
   - Capacità di self healing della rete
4. Test dopo il retraining
   - Valutazione dell'accuratezza della rete
   - Valutazione del numero di bit risparmiati

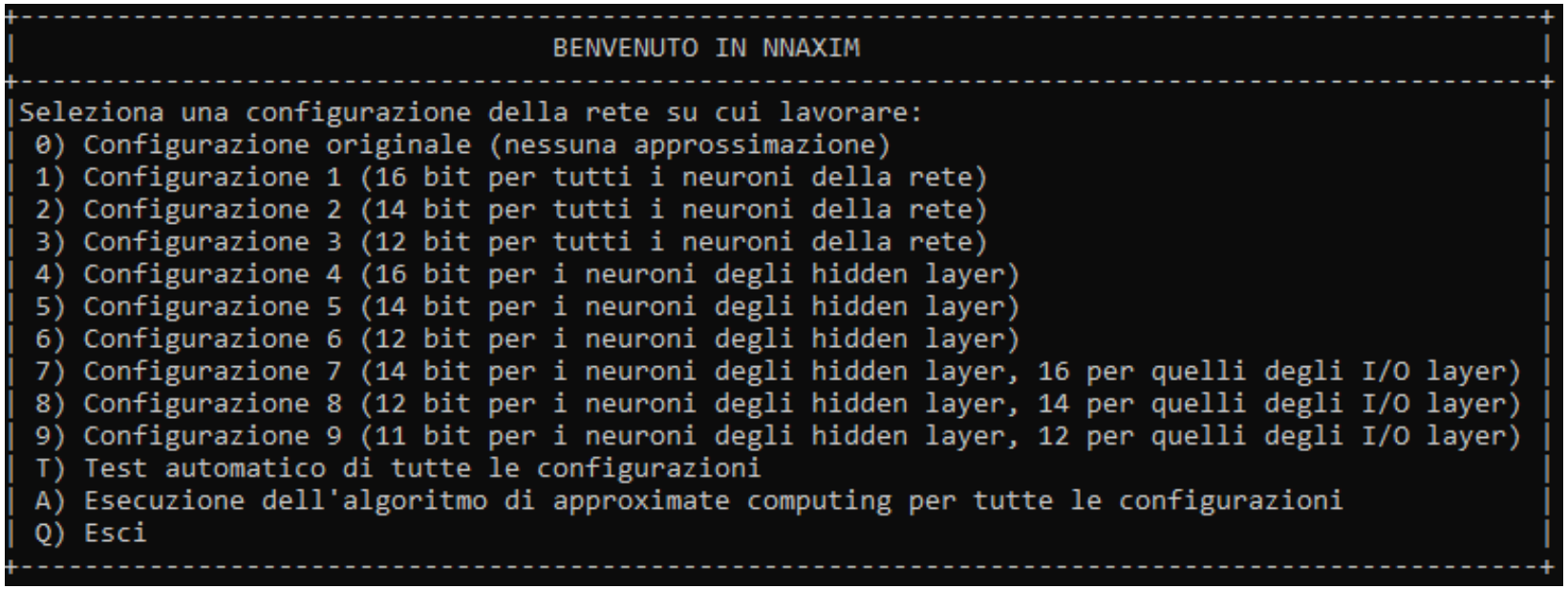
NNAXIM

Neural Network Approximate Computing SIMulator

- Motivazioni
  - Semplificare lo studio
  - Semplificare le operazioni di test e retraining
  - Semplificare l'applicazione dell'algoritmo di approximate computing
- NNAXIM
  - Simulazione dell'applicazione dell'Approximate Computing sul numero di bit usati per rappresentare i pesi di una neural network.
  - TinyDNN
    - CNN per CIFAR10
  - Sviluppato in C++
  - GitHub: https://github.com/Taletex/NNAXIM

Operazioni disponibili

- Allenamento e test di una configurazione
- Esecuzione di un test automatico di tutte le configurazioni
- Esecuzione dell'algoritmo di approximate computing



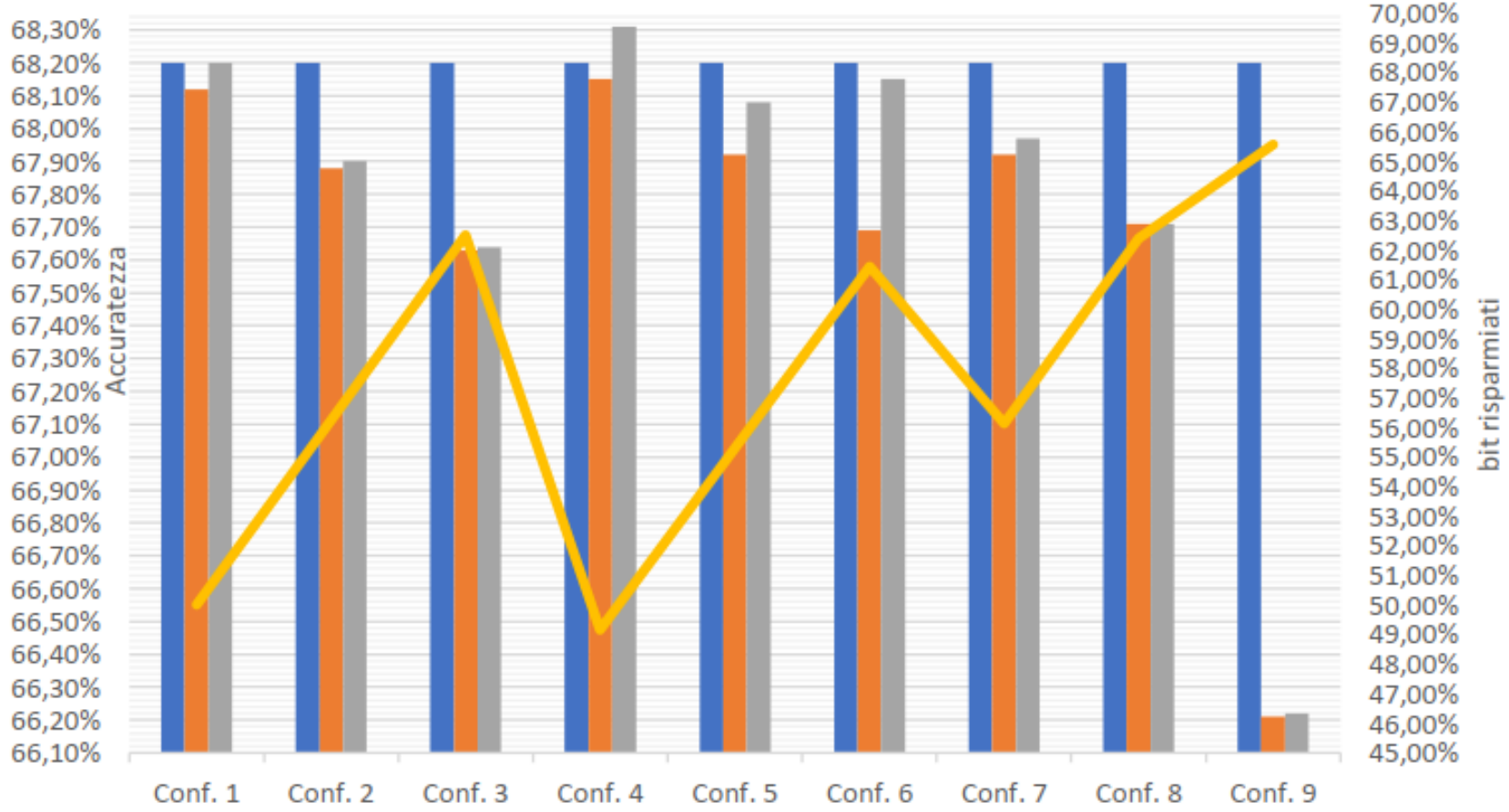
Valutazione dei risultati ottenuti

Processo di valutazione

- Valutazione dei risultati ottenuti per ognuna delle configurazioni
  - Accuratezza della rete
    - Prima del retraining
    - Dopo il retraining
  - Memoria risparmiata

- Comportamento ottenuto pari al comportamento aspettato
  - Maggiore approssimazione comporta maggiore perdita di accuratezza
  - Approssimare anche i layer di I/O comporta una perdita di accuratezza maggiore
  - Il retraining è capace di sanare gli errori introdotti

Confronto risultati della rete neurale



Conf. Originale	68.20%	68.20%	68.20%	68.20%	68.20%	68.20%	68.20%	68.20%	68.20%
Pre retraining	68.12%	67.88%	67.63%	68.15%	67.92%	67.69%	67.92%	67.71%	66.21%
Post retraining	68.20%	67.90%	67.64%	68.31%	68.08%	68.15%	67.97%	67.71%	66.22%
Bit risparmiati	50,00%	56.25%	6 2.50%	49.16%	55.31%	61.45%	56.14%	62.39%	65.57%

Conclusioni

- Conferma delle potenzialità dell'approximate computing
- Enorme diffusione delle applicazioni con forgiving nature
  - Consente un ampio uso dell'approximate computing
- Sviluppo di un tool di simulazione per l'approximate computing
  - Possibilità di ampliare le feature in futuro