Relazione Tecnica – CIFAR-10 e Reti Neurali Convoluzionali

Analisi, sperimentazione e modellazione di una CNN

Autore: *Alessio Cicilano*

---

1. Introduzione

In questo progetto viene analizzato il dataset CIFAR-10, formato da 60.000 immagini a colori molto piccole (32×32×3) suddivise in 10 categorie diverse.
Per svolgere la classificazione si utilizza una **rete neurale convoluzionale (CNN)**, in quanto una rete interamente densa perderebbe completamente la struttura spaziale dell'immagine trattandola come un vettore, mentre la CNN sfrutta filtri locali capaci di individuare pattern caratteristici come bordi, trame e forme.

La CNN è particolarmente adatta perché mantiene la disposizione dei pixel, usa parametri condivisi e riduce significativamente la dimensionalità rispetto a un MLP.

---

2. Funzionamento della Convoluzione

In una convoluzione, un filtro (kernel) scorre sull'immagine producendo una nuova mappa di attivazioni. Ogni filtro rileva un certo tipo di caratteristica.

Elementi fondamentali:

- **Kernel size:** dimensione del filtro (es. 3×3).
- **Stride:** passo con cui il filtro avanza.
- **Padding:** eventuale aggiunta di bordi per controllare la dimensione dell'output.
- **Numero di canali:** se l'immagine è RGB, anche il kernel deve possedere 3 canali.

Un kernel 3×3 ha 9 pesi; se applicato a un'immagine a 3 canali, i pesi diventano 27 più il bias.

3. Pooling e riduzione delle caratteristiche

Il pooling riduce le dimensioni delle feature map:

- **MaxPooling:** seleziona il valore più alto.
- **AveragePooling:** calcola la media.

Questa operazione conserva le informazioni più rilevanti, riduce i parametri richiesti nelle parti finali della rete e rende il modello meno sensibile a piccole traslazioni.

---

4. Funzione di perdita e metrica

Per la classificazione multiclasse:

- **Loss:** *categorical crossentropy*, ideale con etichette one-hot.
- **Metrica:** *accuracy*, utile per una visione generale delle prestazioni.

Per capire quali classi risultano problematiche si utilizza la **matrice di confusione**, che evidenzia eventuali coppie di categorie frequentemente scambiate.

---

5. Normalizzazione delle immagini

Dividere i valori dei pixel per 255 porta i dati nell'intervallo [0, 1], rendendo più stabile il gradiente.

La **standardizzazione** (media zero e varianza unitaria per ogni canale) è ancora più efficace quando i canali hanno distribuzioni diverse. Entrambe le tecniche accelerano la convergenza e stabilizzano l'apprendimento.

---

6. Limiti delle reti dense su immagini

Utilizzare un MLP dopo aver appiattito l'immagine è inefficiente per due motivi:

1. Il numero di parametri diventa enorme già con un solo layer nascosto.
2. Si perde completamente la struttura bidimensionale dell'immagine.

Le CNN, al contrario, sfruttano filtri locali e pesi condivisi, riducono la dimensionalità e incorporano un "bias induttivo" che le rende naturalmente adatte alla visione artificiale.

---

7. Parametri e capacità di una CNN

Il numero di parametri di una layer convoluzionale dipende da:

- dimensione dei kernel,
- canali in ingresso,
- numero di filtri.

Aumentare il numero di filtri, il numero di layer o la dimensione dei kernel incrementa la capacità della rete ma anche il rischio di sovra-apprendimento.

---

8. Tecniche di regolarizzazione

Per evitare l'overfitting si impiegano:

1. **Data Augmentation:** variazioni realistiche delle immagini.
2. **Dropout:** disattivazione casuale di neuroni durante l'addestramento.
3. **L2 Weight Decay:** penalizzazione dei pesi troppo grandi.
4. **Early Stopping:** interruzione dell'addestramento quando la validazione smette di migliorare.

Una combinazione ben bilanciata di queste tecniche è essenziale.

---

9. Data Augmentation per CIFAR-10

Le trasformazioni devono essere moderate a causa delle dimensioni ridotte delle immagini.
Strategia consigliata:

- flip orizzontale,
- leggere rotazioni (10–15°),
- piccole traslazioni,
- variazioni limitate di colore (luminosità, contrasto, saturazione).

Da evitare:

- flip verticale,
- rotazioni molto ampie o a 90°.

Queste trasformazioni genererebbero immagini irrealistiche e confonderebbero il modello.

---

10. Ottimizzatori e gestione del learning rate

Gli ottimizzatori più comodi ed efficaci sono **Adam** e **AdamW**, capaci di adattare dinamicamente il learning rate di ogni parametro.

Uno scheduler molto utile è **ReduceLROnPlateau**, che riduce la learning rate quando la perdita di validazione non migliora più.

---

11. Interpretazione delle curve di apprendimento

Dalle curve di perte e accuratezza si possono riconoscere tre situazioni:

- **Underfitting:** il modello non riesce ad apprendere.
- **Addestramento corretto:** train e validation progrediscono in modo simile.
- **Overfitting:** la loss di train scende mentre quella di validation sale.

Le curve sono fondamentali per decidere come modificare la rete o i suoi iperparametri.

---

12. Matrice di confusione e classi difficili

Alcune coppie di classi di CIFAR-10 risultano particolarmente complesse da separare:

- gatto ↔ cane,
- auto ↔ camion,
- uccello ↔ aereo.

Per migliorare la distinzione si possono introdurre tecniche come Cutout o utilizzare modelli più profondi.

---

13. Effetti del batch size

Il batch size influenza sia la velocità che la qualità dell'apprendimento:

- batch grandi (es. 128) → convergenza rapida ma possibile peggioramento della generalizzazione;
- batch piccoli (es. 32) → gradienti più rumorosi, miglior generalizzazione ma tempi più lunghi;
- **64** è generalmente il compromesso ideale.

---

14. Buone pratiche per la consegna

Per una documentazione chiara e riproducibile è consigliato includere:

1. fissaggio delle seed di randomizzazione,
2. codice ordinato e suddiviso in funzioni,
3. confronto tra MLP e CNN (parametri e metriche),
4. riepilogo dell'architettura finale,
5. log completi dell'addestramento,
6. grafici delle curve di loss e accuracy,
7. matrice di confusione finale sui dati di test,
8. elenco degli iperparametri utilizzati,
9. analisi dei principali errori del modello,
10. conclusioni e osservazioni personali.