<Modello NN 2+1D

cosa sono le residual connections, sono difficili per me ma ho capito l'idea molto generale penso: Da quello che ho visto praticamente Il problema principale sta nella back propacation ovvero visto che la propagazione si fa in base alla funzione di attivazione dei neuroni Questo vuol dire che se abbiamo una funzione come la relu che per l'x minore di zero ha derivato zero vuol dire che il neurone non cambierà con la backpropagation, Perciò si fa che dopo qualche layer di relu neuroni viene aggiunto di nuovo il segnale originale all'output di quei neuroni e tutto ciò viene in input al prossimo layer relu.

Il dataet

Il dataset senza cui è stato trainato un modello è composto di set di video strutturati come 30 immagini di dimensioni quadrate.

Una funzione molto importante per il riconoscimento della gesture è stata “load\_most\_moving\_frames\_center\_crop”. Il suo scopo è selezionare automaticamente dai video registrati un set di 30 frame che catturano la parte del video con più movimento. Questo consente di isolare le parti del video più rilevanti per l'analisi del gesto, eliminando porzioni piu statiche. Inoltre viene fatto un crop su ogni frame tenendo solamente il quadrato centrale e scartando le fasce sopra e sotto.

Tutto cio è stato fatto sfruttando la libreria OpenCV con lo scopo di rendere una qualsiasi clip o video il piu simile possibile al dataset su cui il modello è stato trainato.

<Immagine: grafico movimento->video->30 frame->crop centrale>

Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

Il modello

Descrizione generale

Il modello è una rete neurale convoluzionale 3D pensata per riconoscere gesti a partire da video.

Ogni input è una sequenza di 30 frame RGB, e l'obiettivo è classificare il gesto presente nel video, sfruttando sia le informazioni spaziali (nei singoli frame) che temporali (nel movimento tra i frame).

Conv 2+1D

Spiegare brevemente cosa fa una convoluzione 3D: opera su tempo, altezza e larghezza per catturare il movimento nel tempo.

Introduzione alla tecnica “2+1D”: spezzare la convoluzione 3D in due operazioni separate — prima spaziale (2D), poi temporale (1D).

→ Vantaggi: riduce la complessità, migliora la non linearità, e rende più facile l’ottimizzazione.

Per creare il un modello che potesse analizzare video, quindi una sequenza di frame, è stata usata una rete convoluzionale. Di solito una CNN viene usata per classificare immagini singole e quindi viene fatta una interpolazione 2D. Per i video è diverso poiche presentano un concetto in piu: sequenza temporale di frame. Per questo serve un kernel 3D che è capace non solo di interpretare frame singoli ma anche una sequenza di essi. Infatti, a differenza di un filtro 2D, il filtro 3D è capace di muoversi anche nella dimensione temporale.

2D altezza larghezza -> classifica immagini

3D altezza larghezza tempo -> classifica video

Percio se immaginiamo di avere un set di 30 frame uno affianco al altro creerebbero un parallelepipedo (array) di dimensioni: (30, altezza, larghezza). Perciò un filtro 3D funzionerebbe esattamente come uno 2D.

Per la struttura del modello è stato usato il metodo

Residual connections

ResNet sono sono architetture di rete neurale progettate per risolvere un grosso problema che emerge quando le reti diventano molto profonde: il degrado delle prestazioni. Questo è dovuto a:

* Vanishing/exploding gradients
* Difficoltà nel propagare informazioni utili all’indietro (backpropagation)

La somma x + F(x) serve a **non perdere l’informazione iniziale** di x, mentre aggiungi quello che hai imparato in F(x). In pratica ad ogni passo modifichiamo leggermente x con le nuove informazioni apprese dalla convoluzione.

Di solito una rete prende una immagine con 3 canali RGB e ne estrae n (ad esempio 16) feature: bordi verticali, orizzontazi, curve… quindi si fa del pooling e l’immagine (le 16 feature) viene ridotta di dimensioni (questa riduzione di dimensioni simula il catturare feature sempre piu estese e grandi dell’immagine, noi riduciamo l’immagine ma in realtà è come se stessimo rendendo i filtri piu grandi: Aumenta il campo visivo e ogni unità nel layer successivo guarda un’area più ampia dell’immagine originale) e così via. Si passa dai pixel → a bordi → a texture → a parti del viso → a una testa → a un cane o gatto. All’ultimo passo si prendono tutte le feature estratte (teste, zampe…) e con “Flatten + Dense + Softmax”(capire) si decide se è un cane o un gatto. Con l’aumentare dei layer Il gradiente si disperde → l’input originale non viene più "sentito" e La rete diventa difficile da ottimizzare. Invece le reti con residual block funzionano portandosi la rappresentazione x dell’immagine originale lungo tutta la rete e i vari layer potenziano le aree dove trovano certe feature come occhi o orecchie ma mantenendo comunque x che viene ritoccata ad ogni layer. Evito di alterare completamente l’immagine iniziale come farebbero le calssiche CNN ma modifico l’input pian piano. i residual block invece, ti porti dietro la rappresentazione precedente e ci sommi modifiche incrementali.

Le residual connections sono una tecnica architetturale introdotta per rendere più facile l’addestramento di reti neurali molto profonde. In un modello ogni blocco di layer deve imparare da zero la trasformazione richiesta, ma quando la profondità cresce questa operazione diventa instabile: il gradiente può annientarsi o esplodere, e le funzioni come la ReLU rischiano di cancellare informazioni utili. Le residual connections risolvono questi problemi creando una scorciatoia che trasmette direttamente l’input di un blocco alla sua uscita, facendo sì che il blocco debba imparare solo la differenza (il residuo) rispetto a ciò che riceve in ingresso. In questo modo si riducono degradazione e vanishing/exploding gradients, e diventa praticabile l’addestramento di più di strati con benefici in termini di accuratezza e convergenza più rapida.

Immagine che contiene Carattere, testo, schermata, Elementi grafici

Il contenuto generato dall'IA potrebbe non essere corretto. Immagine che contiene Carattere, schermata, calligrafia, simbolo

Il contenuto generato dall'IA potrebbe non essere corretto.

Nella rete 3D per il riconoscimento dei gesti, i residual block sono stati fondamentali per garantire che i pattern spaziali e temporali appresi ai livelli più bassi restassero disponibili fino alla parte finale del modello. Grazie all’AddResidualBlock, ogni volta che un blocco di convoluzioni rischiava di “spegnere” alcuni feature (ad esempio a causa di normalizzazione o ReLU), l’input originale veniva reiniettato integralmente all’uscita. Questo ha evitato che informazioni sottili sul movimento — essenziali per distinguere gesti simili — andassero disperse durante la discesa di scala e la profondità del network. In pratica, senza residual block la rete avrebbe dovuto ricostruire da zero ogni elemento del segnale ad ogni livello, compromettendo stabilità e precisione; con gli AddResidualBlock, invece, ho ottenuto una formazione più robusta, una convergenza più veloce e un riconoscimento dei gesti più affidabile anche in presenza di sequenze complesse.

Una resinette permette di trasportare l’input lungo tutta la rete e calcolare ad ogni passo, in questo modo si evita che durante la rete venga “scordato” l’input.

Una fattore importante è appunto rimanere costanti con le dimensioni dell’input e output di ogni Residual block.

Spiegare qui alternanza residual block resize video così spiego in generale cosa facciamo poi andiamo nel profondo a capire i sottoblocchi.

Spiegare la strategia architetturale: ogni blocco convoluzionale è seguito da un downsampling (ResizeVideo) per estrarre feature a livello sempre più astratto.

Questa alternanza serve a costruire una rappresentazione gerarchica, riducendo la risoluzione spaziale ma mantenendo l’informazione grazie ai residual.

Ogni livello opera a una scala diversa, ma l’informazione originale resta disponibile.

Blocchi principali

Breve spiegazione dei componenti principali:

ARCHITETTURA DEL MODELLO

È un modello di deep learning che usa Keras ed è composto da diversi livelli grazie a Functional API: classi Python che ereditano da ekras.layers.Layer che poi vanno a specializzarsi; quindi, vengono assemblate e riutilizzate per comporre il modello finale. Qui, infatti, un punto fondamentale che poi verrà approfondito è il fatto di poter costruire piu un grafo anzi che il classico modello lineare di keras.sequential. Infatti c’è òa pèossibilità di mettere i layer in sequenza ma anche in parallelo.

<immagine modello>

INPUT MODELLO

Immagine che contiene testo, Carattere, ricevuta, documento

Il contenuto generato dall'IA potrebbe non essere corretto.Una pipeline di input al modello serve innanzitutto per inserire l’immagine e passarla attarverso i nodi. Un punto fondamentale è un layer di pre-processing che usa data augmentation per migliorare la generalizzazione del modello. Infatti questo layer applica ad ogni video una variazione di luminosità costante su tutti i frame simulando condizioni di luce diverse. Questa strategia rende il modello meno prono all’overfitting rispetto alle condizioni di luminosità viste durante l’addestramento.

<immagini io che divento chiaro>

Conv2Plus1D: un layer che implementa la convoluzione 2+1D descritta di sopra.

Quindi i dati vengono normalizzati e passati ad una relu.

Dioi qyesta fase iniziale ci sono una serie di blocchi con lo scopo di ridurre man mano la risoluzione dei video e nel frattempo inserire blocchi di ResNet come descritto di sopra.

Dopodiche viene fatto un pooling sul tensore finale che aggrega i dati lungo tutte e 3 le dimensioni e ne fa una media. Ogni feature è rappresentato da un unico valore. Infine c’è un Dense layer semplicemente per la classificazione del gesto.

Conv2Plus1D: scomposizione spaziale-temporale.

LayerNormalization: stabilizzazione delle attivazioni.

ReLU: attivazione non lineare.

Project: adattamento dimensionale per la somma residuale.

Menzione della GlobalAveragePooling + Dense finale per ottenere la classificazione.

La rete sfrutta la combinazione di convoluzioni 2+1D e residual block per apprendere in modo efficiente movimenti complessi. L’alternanza con Resize consente di catturare informazioni a diverse scale spaziali.

RISULTATI

Il modello si comporta bene sia durante l’addestramento che nella fase di test. È abbastanza generalizzato e non ci sono segni di overfitting, riesce a predirre le classi in modo eguale senza prediligenre una e raggiunge una una accuracy del 90%.Immagine che contiene schermata, testo, quadrato, Rettangolo

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, diagramma, schermata, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Detto ciò, è importante fare una considerazione sul dataset utilizzato: i dati con cui il modello è stato addestrato contengono un gruppo omogeneo di persone e questo limita la capacità del modello di generalizzare quando viene testato su soggetti con caratteristiche differenti. Si può dire che è un overfitting indotto dal dataset. Dai risultati la CNN non sembra essere il problema ed è plausibile aspettarsi risultati migliori con una ampliazione del dataset introducendo appunto diversità nel set di training.