**STRUTTURA DELLA RETE (SLIDE DISEGNO)**

La nostra rete, da come si evince in figura, è composta da 3 layer convoluzionali basati sui quaternioni, 2 layer riccorenti ed infine due output paralleli composti entrambi da due layer fully-connected.

Due output paralleli perché, come ha detto in precedenza il mio illustre collega Marco, i nostri obiettivi sono la classificazione e la localizzazione della sorgente dei file audio.

La classificazione consiste nel individuare a quale classe corrisponde un determinato file audio e questo viene fatto utilizzando la SED, Sound event Detection.

Invece per quanto riguarda la localizzazione del file audio, la nostra rete ritorna le coordinate cartesiane x,y,z della sorgente utilizzando la DOA estimation, direction of arrival.

**STRUTTURA DELLA RETE (SLIDE CONVOLUTIONAL PART.1)**

Abbiamo quindi analizzato la rete nel dettaglio, vedendo che un layer convoluzionale è formato da 3 stage: Convolutional, Detector e Pooling

Il Convolutional stage è il fulcro del layer convoluzionale, nel caso dei quaternioni non viene fatto un semplice prodotto convoluzionale ma il prodotto di Hamilton, dove vado a moltiplicare tutte le componenti di un generico quaternione x per le componenti della matrice dei filtri dei quaternioni W.

Dopo di che questo prodotto viene dato in pasto ad una funzione di attivazione, ed ecco che siamo nel Detector stage, in cui si va a scegliere la funzione di attivazione da applicare. Nel nostro caso una ReLU.

**STRUTTURA DELLA RETE (SLIDE CONVOLUTIONAL PART.2)**

Infine vi è il Pooling in cui attraverso la riduzione della dimensione vado a diminuire il costo computazionale della rete richiesto per processare i dati. Nel nostro caso viene usato il Max-Pooling in cui viene scelto il valore massimo delle sotto regioni prese in considerazione.

Dopo i 3 layer convoluzionali vi sono i due layer ricorrenti, i quali hanno la particolarità di avere un ciclo, infatti essi prendono in input sia l’output fornito dal layer precedente sia l’input di quel layer precedente, formando cosi un ciclo. Nel caso della nostra rete viene utilizzata la tangente iperbolica come la funzione di attivazione.

Infine ci sono i due output che differiscono tra i loro per le funzioni di attivazione.

Nel caso della classificazione viene applicata prima una funzione di attivazione lineare e poi la sigmoid, quest’ultima perché in quanto classificazione si vuole ritornare una distribuzione di probabilità.

Invece nel caso della regressione vengono applicate la funzione lineare e poi la tangente iperbolica.

**STRUTTURA DELLA RETE (SLIDE CONVOLUTIONAL PART.3)**

Per concludere sulla rete, i pesi non devono essere inizializzati in maniera randomica, perché si potrebbe finire in un minimo locale, ma attraverso una distribuzione uniforme. Dato che si tratta di quaternioni, dovrà esser fatto su tutte le componenti dei pesi w, che poi comporranno la matrice W.

**DATASET**

Per quanto riguarda il dataset, ci è stato fornito il dataset TAU nel quale avevamo 500 file audio ed i corrispondenti file CSV. I file CSV contiene le informazioni sul tempo di inizio e di fine del file audio e le coordinate del DOA che vengono rappresentate dagli angoli di azimuth ed elevation.

Questo dataset è composto da un training set con 400 file audio, divisi in 4 split usando la cross validation, ed un test set di 100 registrazioni.

Questa distinzione è importante perché la rete verrà allenata sul training set ed in seguito verrà valutata usando il test set per considerare se il training è stato positivo oppure no.

Le classi prese in considerazioni sono 11, ma in particolare i file audio del dataset hanno 1 o 2 sovrapposizioni del suono, quindi abbiamo considerato questi due sottoinsiemi del dataset, composti ciascuno da 250 file, divisi randomicamente in 5 split in modo tale che ciascuno avesse gli stessi elementi per ogni classe.

Noi abbiamo considerato 4 split per il training e 1 per il test.

**METRICS**

Dopo di che, le metriche utilizzate sono state la SED Score e La DOA Score.

Entrambe medie, una calcolata considerando l’error rate e la f1-score, l’altra considerando la DOA error e la frame recall.

La SED Score deve essere più bassa possibile in modo tale da avere ottimi risultati perché l’error rate, che è la frequenza degli errori, deve essere più piccolo possibile, invece f1-score deve essere alto dato che si basa sui true positive ed i false positive.

Stessa cosa per la DOA score, la quale deve avere un DOA error basso ed invece una frame recall alta dato che prende in considerazione la percentuale di true positive.

La SELD score, che è quella su cui si basiamo noi, è proprio la media dei due score. Più bassa è più i risultati che otteniamo sono buoni.

Ora passo la parola all’altra illustre collega, Claudia.