**Organizzazione di code**

nella cartella training results sono presenti tutti gli addestramenti fatti, in particolare questa si divide in sottocartelle a seconda del tipo di addestramento, le sottocartelle terminali contengono:

-file python per condurre quel addestramento

-file dei pesi di quell’addestraemento

-una serie di file csv dove sono contenuti i risultati

-un file json dove si vede loss e accuracy che ha avuto quell’addestramento epoca per epoca.

**Addestramenti principali**

Gli addestramenti principali sul dataset riorganizzato sono:  
Addestramento lstm finale: “training\_results\reorganized\_dataset\split\_60\_30\_10\REAL\_Gradnorm\15\_epoch\lstm\CNN\_starter\_multihead\classic\_norm\_in\_epoch”

Addestramento modello efficient-not b0 + tre heads (multi head model) con GradNorm:

“training\_results\reorganized\_dataset\split\_60\_30\_10\REAL\_Gradnorm\8\_epoch\starter\_multihead\starter\_multihead\_no\_dummy\_with\_gradnorm\_REAL\_normalize\_in\_epoch\_first\_time”

Addestramento baseline single head: “training\_results\reorganized\_dataset\split\_60\_30\_10\15\_epochs\_training\paper\_baseline\NO\_grad\_norm\_official”

Gli addestramenti principali sul dataset fornito dal paper sono:

baseline:

training\_results\paper\_dataset\paper\_baseline\paper\_dataset\_csv

Gli addestramenti principali sul dataset rielaborato ma con lo split del paper:

training\_results\reorganized\_dataset\paper\_split\paper\_baseline\batch\_not\_splitted\batch\_size\_128

**Test delle reti**

Gli script utili a valutare le reti sono nella cartella “test\_nets”, in particolare è presente:

-script di test modello a single head

-script di test modello multihead

-script di test modello lstm

in questi codice vi è un parametro che può essere impostato sia su “validation”, che su ”test” a seconda di se si vuole testare la rete su set di validation o di test.

NB

è importante che sia nei file di addestramento che di test vanno cambiati i path, sia dei pesi del modello, che dei csv, che del dataset.