1) Confrontare il main con quello su git e recuperare la parte di scaling.

Ho inserito nel main su git la parte di scaling che c'era nel tuo main, pare funzionare anche con due output

2) Cosa ho modificato nei file di opennn? Cos'è testing\_analysis?

nei file di opennn non c'è praticamente niente di modificato (sono solo spazi o cose del genere) eccetto in testing analysis dove avevi aggiunto dei print. Testing analysis è l'istanza che si crea per testare la rete. Sto cercando ora di capire come fare a farla funzionare con più output. Infatti fino ad ora abbiamo inserito lo "sdoppiamento" dell'output fuori dalle operazioni che esegue la rete quando calcola l'output (dato che non abbiamo inserito nuovi parametri), lo abbiamo inglobato nel calcolo della loss e del gradiente.

Ha senso inserire la funzione che calcola l'output multidimensionale a partire da quello della rete anche in Testing\_analysis? (E' già presente in normalized squared error) oppure dovremmo creare una classe che fa questo e le altre operazioni associate tipo calcolare il gradiente?

Ho usato Gradient Descent come training method, a grandi linee i punti in cui ho inserito qualcosa li ho elencati nel file GradientDescent2outputs

Non ho capito come funzionano scaling e unscaling layer. Non facciamo lo scaling usando scale\_inputs/targets\_minimum\_maximum?

PROBLEMA: ho due targets, quando fa lo scaling lo fa in modo indipendente per i due: calcola min e max del primo target e scala e poi idem con il secondo. A noi invece serve che abbiano lo stesso range, dato che calcoleremo il secondo output in funzione del primo (target2= 2\*target1)

inserisco nella funzione scale\_columns\_minimum\_maximum di matrix.h (che è quella in cui effettivamente viene fatto lo scaling) una variabile bool che di default è 0, che viene settata a uno se vogliamo fare lo scaling in modo uniforme

conseguentemente deve essere prima passata alla funzione di data set scale\_targets\_minimum\_maximum -> Adesso sembra funzionare lo scaling dei target ma la loss è altissima: c'è qualcosa che non va

-> cosa non va bene: data la sua trasformazione quando i due target vengono scalati (nello stesso range) non sono più uno il doppio dell'altro, come saranno invece gli output.

Ma non va bene neanche che siano uguali una volta scalati. Nel nostro caso invece può andare bene?