Main.cpp

Dal file di input legge due variabili come targets

92 outputs\_number -> impostato da noi uguale a 1

Gradient\_descent.cpp

1164 perform\_training()

1256 calculate\_training\_loss() ->

NormalizedSquaredErrore.cpp 290 calculate\_training\_error

317 calculate outputs ->

aggiunto calculate\_our\_outputs

1283 calculate\_training\_loss\_gradient()

chiama

NormalizedSquaredErrore.cpp 715 calculate\_training\_error\_gradient

751 output\_gradient ->calculate\_output\_gradient 987

qui dentro chiamo la mia funzione f che calcola le soluzioni dato outputs( che per noi è tau)

e poi calcolo f(outputs)-targets

moltiplichiamo per il gradiente della f calcolato in

calculate\_gradient\_our\_outputs 1017

calculate directional point usa calculate training error(parameters) in cui ho aggiunto nel calcolo degli outputs la chiamata alla funzione calculate our outputs

256 Testing Analysis ,calculate\_target\_outputs()

Anche qui per il calcolod egli output da confrontare con i target, va inserita la nostra funzione