Validation Error: viene calcolato per ogni epoch durante il training (ma su un set di dati diverso). Nel nostro codice viene chiamato “Selection error” presumo perché può essere usato per selezionare il migliore modello

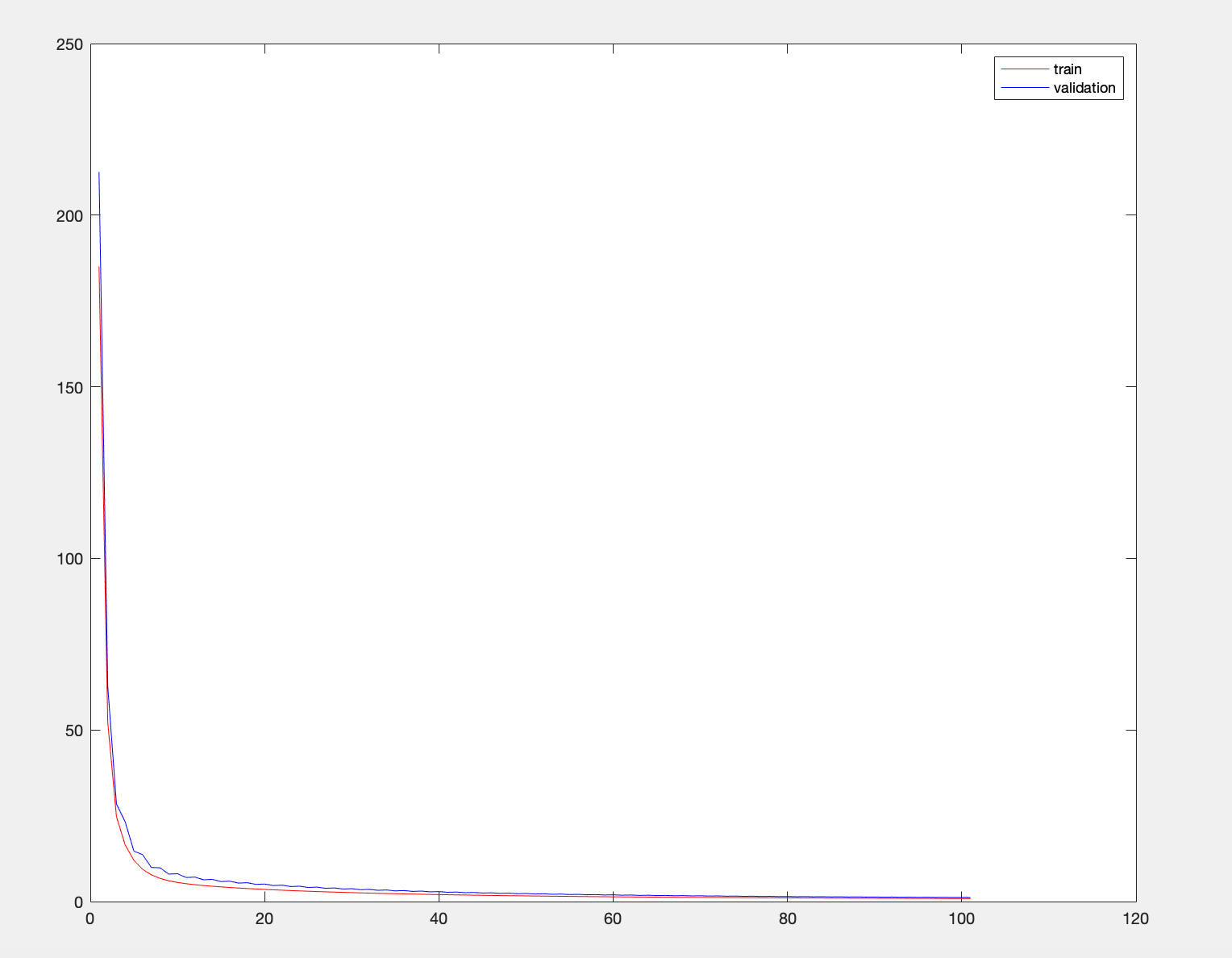
Overfitting: se l’errore sul validation set ricomincia a crescere mentre quello sul training set decresce oppure se quello sul validation set rimane sempre molto maggiore di quello sul training set (Anche se decrescono entrambi)

Nel constructor di Gradient\_descent ho impostato

reserve\_error\_history = true;

reserve\_selection\_error\_history = true

In modo da salvare l’errore sul traiing e validation set (quello che poi ho plottato)



Nel nostro caso anche riducendo il dataset sembra che non ci siano problemi di overfitting, ciò potrebbe essere dovuto alla semplicità della funzione che stiamo approssimando.

Domanda anche aumentando di molto il massimo numero di epoch (ad esempio a 1000) il criterio per cui il training viene interrotto è sempre il raggiungimento di questo limite, e mai ad esempio incremento minimo dei parametri o loss sufficentemente bassa.

In perform training c’è un criterio di stopping che riguarda l’over fitting: se il selection error è maggiore di quello dell’iterazione precedente viene aggiornato il contatore “selection\_failures”, il training si ferma se “selection\_failures” supera la soglia “maximum\_selection\_error\_decreases” ( di default impostata a 1000 in gradient\_descent) e se il flag early\_stopping è attivo ( di default lo è, sempre in gradient descent)

Inoltre è possibile far restituire al training la rete con il minimo selection error impostando il flag “return\_minimum\_selection\_error\_neural\_network” su true (di default è false)

In realtà nel nostro caso sarebbe comunque l’ultima iterazione ad essere restituita in quanto l’errore decresce solamente.