Let F be the model.; $\vec{w}$ be the flattened Parameter vector, X : input data representative of input

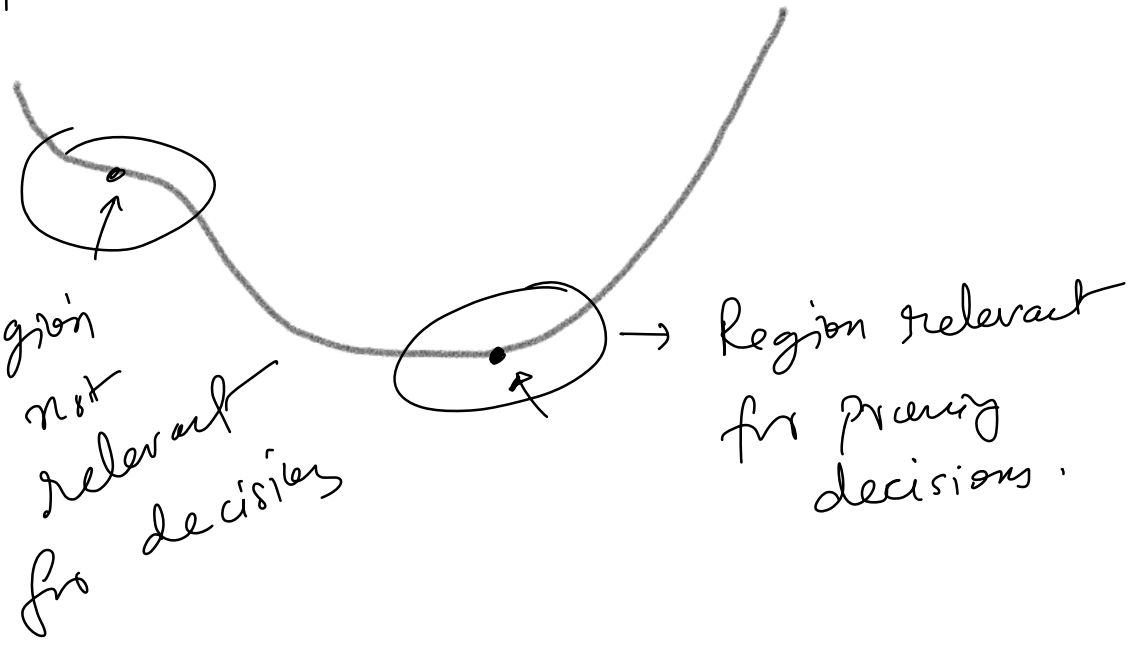
Q Again, what should we be looking at while pruning?

The only "information" that we get about good and bad decisions in ML are through the loss landscape.

↳ "Training depends on the loss landscape"

Q: Which Model should we start at?

→ Since the information you can get about loss landscape is "local", you better start from the "Converged Model"



Let $\mathcal{L}(w^*, X)$ be the loss function and let w^* be the weight at minima (converged model) then to get "information" about the landscape analytically we need to approximate the local landscape

$$\mathcal{L}(w, X) = \mathcal{L}(w^*, X) + \nabla \mathcal{L}(w^*, X)^T \delta w + \frac{1}{2} \delta w^T \nabla^2 \mathcal{L}(w^*, X) \delta w + O\left(\|\delta w\|_2^3\right)$$

$\rightarrow$ "Converged" $\Rightarrow \nabla \mathcal{L}(w^*, X)^T = 0$

"local" / "quadratic" $\Rightarrow O(\|\delta w\|_2^3) \rightarrow 0$

$$\mathcal{L}(w, X) - \mathcal{L}(w^*, X) = \frac{1}{2} \delta w^T H(w^*, X) \delta w.$$

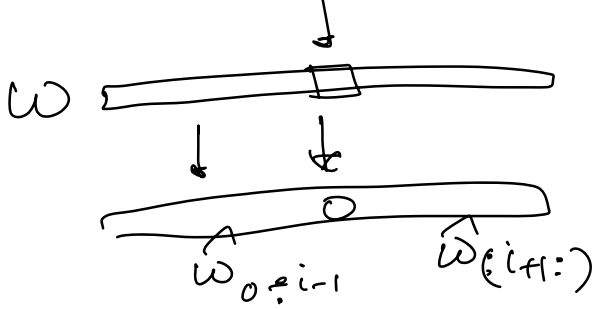
$$E(\delta w) = \frac{1}{2} \delta w^T H(w^*, X) \delta w$$

Note that this is just a quadratic function and thus is easy to analyse

Which "i" to prune

↳ Lowest saliency

↳ Update the rest of the wts.



N: Subproblems

If "i":

$$\delta w = \underset{\delta w_i = -w_i}{\operatorname{argmin}} \left[E(\delta w) \right]$$

So that the i^{th} wt goes to 0

Lagrange's multiplier Method.

$$\delta w = \underset{\delta w}{\operatorname{argmin}} \left[\underbrace{E(\delta w)}_{\downarrow \; \frac{1}{2}\delta w^T H \delta w} + \lambda \left(e_i^T (w + \delta w) \right) \right]$$

e_i: unit vector.

$$\frac{\partial T}{\partial \lambda} = 0 \implies \boxed{\delta w_i = -w}^T$$

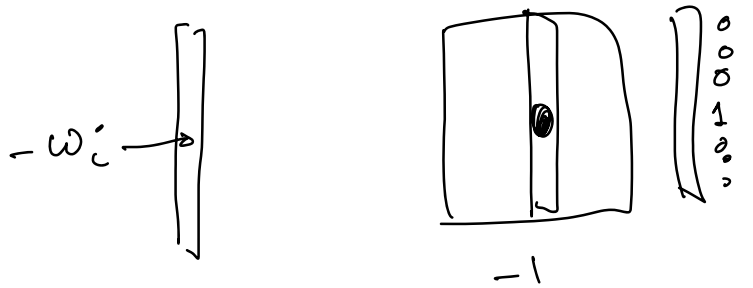
|| for matrix operations look at matrix cook book.

$$\frac{\partial T}{\partial (\delta w)} = H \delta w + \lambda e_i = 0$$

other $\delta\omega_i$ is known

$$H\delta\omega + \lambda e_i = 0$$

$$\delta\omega + \lambda H^{-1} e_i = 0$$



$$-\omega_i = \lambda H^{-1}_{ii}$$

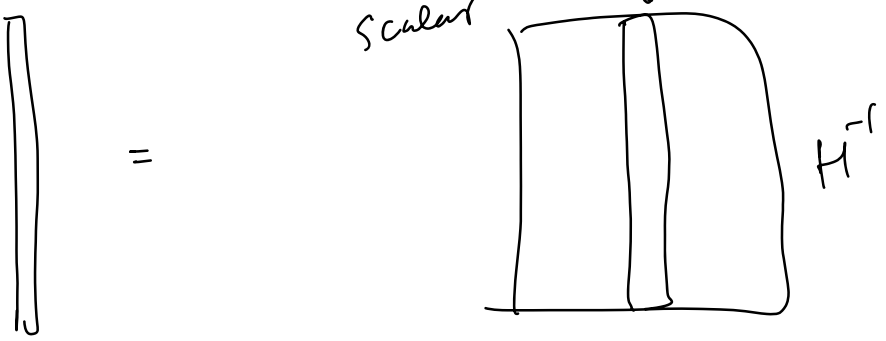
$$\therefore \boxed{\lambda = \frac{-\omega_i}{[H^{-1}_{ii}]}}$$

$$\delta\omega = -\lambda H^{-1} e_i$$

$$\boxed{\delta\omega = \frac{\omega_i}{[H^{-1}_{ii}]} H^{-1} e_i}$$

scalar

i^{th} colmn.



Saving: $\frac{1}{2}\delta\omega^T H \delta\omega$

$$\frac{1}{2}\left[\right]^2$$

$$= \frac{1}{2} \frac{w_i}{(H^{-1})_{ii}^2} \left[(H^{-1} e_i)^T \underbrace{H}\ H^{-1} e_i \right]$$

$$= \frac{1}{2} \left[\frac{w_i^2}{(H^{-1})_{ii}^2} \right] \left(e_i^T \underbrace{(H^{-1})^T} e_i \right)$$

$$= \frac{1}{2} \frac{w_i^2}{H^{-1}_{ii}^2} \cdot H^{-1}_{ii}$$

$$\text{Saliency} = \frac{1}{2} \frac{w_i^2}{H^{-1}_{ii}}$$

$$\delta w = \frac{w_i}{H^{-1}_{ii}} H^{-1} e_i$$

* Algorithm :

(1) Train the full Model.

(2) ($S_i \leftarrow$ saliency of i)

pick i with least saliency.

update

$w \leftarrow w + \delta w$.

If $\{S_i\}_{min}$ is large compared to loss

then Retrain

Redo local approx

If $\{s_i\}_{n \times n}$ is sweet

Complexity :

1 prune : $O(N^3)$