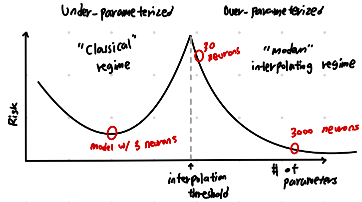
Role of Overparameterization.

Role of Overparameterization.



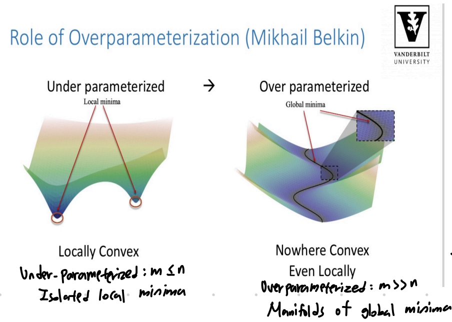
Problem: Neural Network's Loss function is Highly Nonconvex.

But, why does the optimization work? : By the role of overparameterization.

- In overparameterized regression, we are in an interpolation regime. ☆ Interpolation regime → Solving a system of nonlinear equations

$$\begin{cases} f(x_1; w) = y_1 \\ \vdots \\ f(x_N; w) = y_N \end{cases}$$

- In classical view of optimization, non-convex = bad. Many local minima.



① Polyak-Lojasiewicz (PL) Condition (1963)

$: \frac{1}{2}\|\nabla L(w)\|^2 \geq \mu \cdot (L(w) - L(w^*))$

② Alternatively, the μ-PL* condition on a set $S \subset \mathbb{R}^m$ is defined as:

$: \frac{1}{2}\|\nabla L(w)\|^2 \geq \mu L(w)$ because $L(w^*) \approx 0$

☆ PL* condition → Existence of solution and convergence of S(GD) to it.

In Our problem.

$L(w) = \frac{1}{2}\|F(w) - y\|^2$, where $F(w) = [f(x^{(1)}; w), \dots, f(x^{(n)}; w)]^T \in \mathbb{R}^n$

Then, we have:

$\nabla_w L(w) = (F(w) - y)^T DF(w)$ Why?

$L(w) = \frac{1}{2}(F(w) - y)^T (F(w) - y)$, where $F(w) = Xw$, $X \in \mathbb{R}^{N \times M}$, $w \in \mathbb{R}^M$

$$Xw = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_M^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_M^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_M^{(N)} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = F(w)$$

$F(w) \in \mathbb{R}^n$, $y \in \mathbb{R}^n$

$$\begin{aligned} L(w) &= \frac{1}{2}(F(w)^T - y^T)(F(w) - y) \\ &= \frac{1}{2}(F(w)^T F(w) - F(w)^T y - y^T F(w) + y^T y) \\ &= \frac{1}{2}(F(w)^T F(w) - 2F(w)^T y + y^T y) \end{aligned}$$

$$\begin{aligned} \nabla_w L(w) &= D(F(w)^T F(w)) - D(F(w)^T y) \\ &= DF(w)^T F(w) - DF(w)^T y \\ &= DF(w)^T (F(w) - y) \\ &= (F(w) - y)^T DF(w) \end{aligned}$$

☆ The gradient is the special case of Jacobian matrix where the codomain has dimension 1.

When $F: \mathbb{R}^n \to \mathbb{R}$,

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n$$

When $F: \mathbb{R}^n \to \mathbb{R}^m$

$$Jf = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{M \times N}$$

or Df

Then,

$$\|\nabla_w L\|^2 = \|(F(w)-y)^T DF(w)\|^2 = (F(w)-y)^T DF(w)\,DF^T(w)(F(w)-y)$$

Let $K(w) = DF(w)DF^T(w)$, which is referred to as tangent kernel

Then

$L(w) = \frac{1}{2}\|F(w)-y\|^2$

$$\|\nabla L(w)\|^2 \geq \lambda_{min}(K(w))\|F(w)-y\|^2 = \lambda_{min}(K(w)) \cdot (2L(w)) = 2\lambda_{min}(K(w))L(w)$$

and $2\lambda_{min} = \mu$, and μ-PL* condition satisfied

If $L(w)$:

① Is β-smooth (Lipschitz continuous)

② Satisfies μ-PL* condition in a $B(w_0, R)$ with

$$R = \frac{2\sqrt{2\beta L(w_0)}}{\mu}$$

then (S)GD, initialized at w_0, and with learning rate $\mu < \frac{2}{\beta}$ will recover w^* : $F(w^*) = y$.