Lecture 15

• I.O.U: Constrained Optimization

$$\min f(x)$$
$$\text{s.t. } g(x) \le 0 \leftarrow \text{primal constraint}$$

$$\mathcal{L}(x, \lambda) = f(x) + \lambda g(x)$$

① $\min_{x} \max_{\lambda \ge 0} \mathcal{L}(x, \lambda) \xrightarrow{②} \max_{\lambda \ge 0} \boxed{\min_{x} \mathcal{L}(x,\lambda)} \rightarrow \max_{\lambda \ge 0} Q(\lambda)$

$Q(\lambda)$

primal → dual

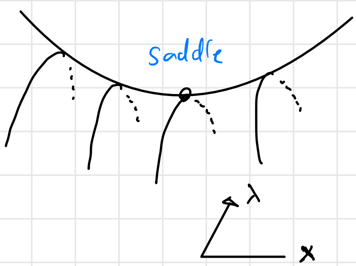
Dual constraint

Note: ① $\max_{\lambda \ge 0} \lambda \boxed{g(x)} \rightarrow \infty$

> 0

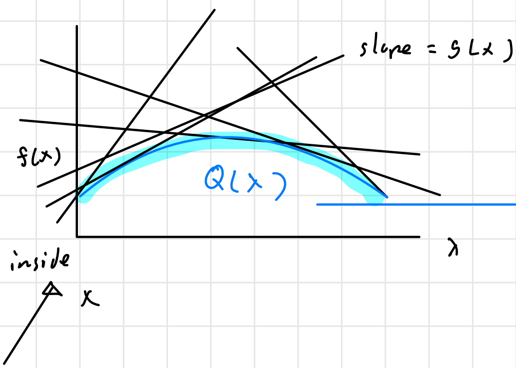
② Von Neumann

convex - concave

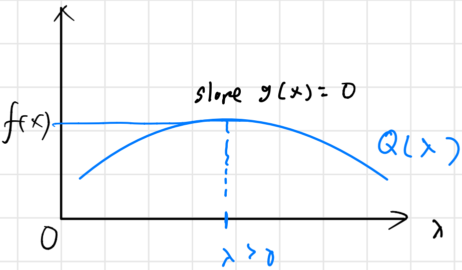


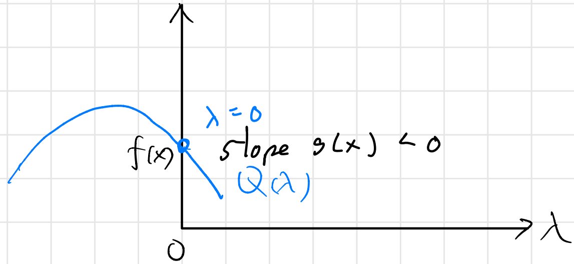
The Lagrangian is linear in λ

③



Lower Envelope
Concave





- What if we had

$$\left.\begin{aligned} &\min f(x) \\ &\text{s.t. } g(x) = 0 \end{aligned}\right\}$$

Then no need for $\lambda \geq 0$

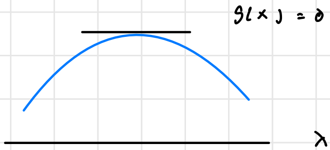
$$\min_x \max_\lambda \mathcal{L}(x, \lambda)$$

$$\max_\lambda \lambda g(x)$$

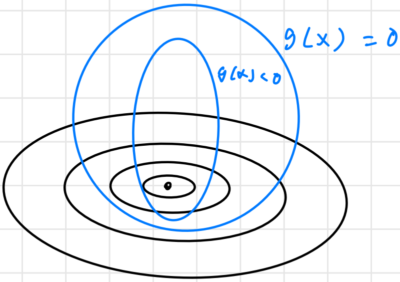
if $g(x) > 0$ $\xrightarrow{\lambda \to \infty}$ $+\infty$

if $g(x) < 0$ $\xrightarrow{\lambda \to -\infty}$ $+\infty$

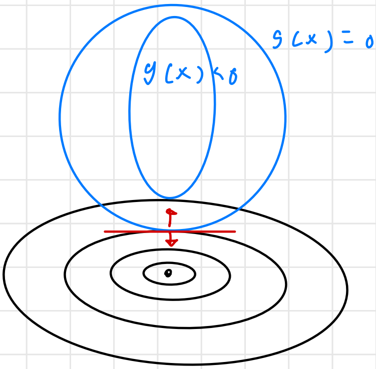
③



- Consider the Contour Plot:



$\lambda = 0$



$f'(x) = -\lambda g'(x)$

$f'(x) + \lambda g'(x) = 0$

$\frac{\partial}{\partial x} \mathcal{L}(x, \lambda) = 0$

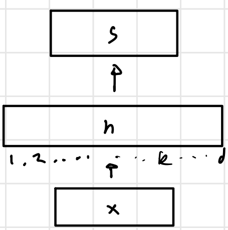
- KKT conditions

(1) primal constraints
(2) Dual constraints
(3) Stationarity $\quad \frac{\partial}{\partial x} \mathcal{L} = 0$
(4) Complementary slackness ; $\lambda g(x) = 0$

- Deep Learning :

  - slogan 1 : approximator

    - ReLU : $f(x)$ piecewise linear
    - Sigmoid : $f(x)$ piecewise constant



$$s = f(x) = \frac{1}{\sqrt{d}} \sum_{k=1}^{d} \beta_k \, \sigma(x^T \alpha_k)$$

- Initialization : $\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} \overset{iid}{\sim} P_0(\alpha_k, \beta_k)$, $k = 1, \dots, d$

$$k(x, x') = \mathrm{cov}(f(x), f(x')) = \mathrm{cov}\left(\frac{1}{\sqrt{d}} \sum \beta_k \sigma(x^T \alpha_k), \frac{1}{\sqrt{d}} \sum \beta_k \sigma(x'^T \alpha_k)\right)$$

$$= \frac{1}{d} \sum_{k=1}^{d} E\left(\beta_k^2 \, \sigma(x^T \alpha_k) \, \sigma(x'^T \alpha_k)\right)$$

$$= E\left(\beta_k^2 \, \sigma(x^T \alpha_k) \, \sigma(x'^T \alpha_k)\right)$$

Central limit theorem

So $f(x) \sim NNGP(\quad) \xrightarrow{d} \infty$ CLT converges to GP

Neural net Gaussian process

1. Gradient Descent Learning

- freeze $\alpha_k \overset{iid}{\sim} P_0(\alpha)$

$$f(x) = h(x)^T \beta$$

$$h_k = \frac{1}{\sqrt{d}} \sigma(x^T \alpha_k) \quad \text{random feature}$$

learn β

$$\text{Loss}(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - h(x_i)^T \beta)^2$$

$$\beta_0 = 0$$

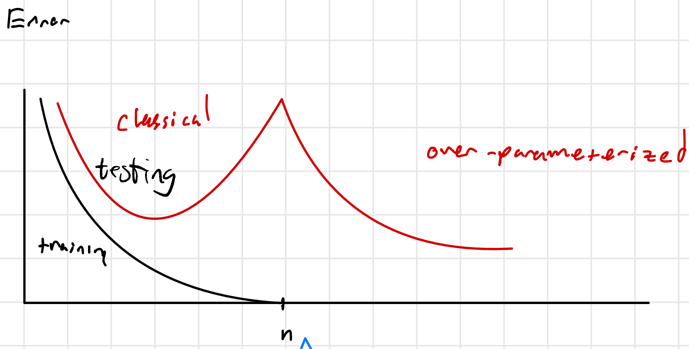
$$\beta_{t+1} = \beta_t + \eta_t \frac{1}{n} \sum_{i=1}^{n} (\underbrace{y_i - h(x_i)^T \beta_t}_{e_i}) h(x_i)$$

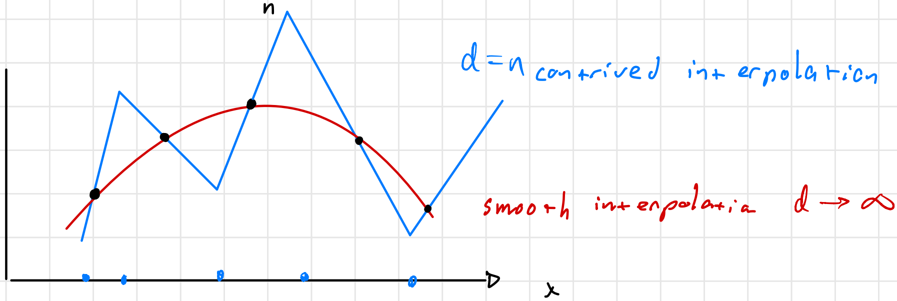
$$\longrightarrow \hat{\beta} = \sum_{i=1}^{n} c_i h(x_i) \quad \text{representer.}$$

$$\hat{f}(\underset{\text{query}}{x}) = h(x)^T \hat{\beta} = \sum_{i=1}^{n} c_i k(x_i, x) \quad \text{kernel regression} \quad \lambda \to 0$$

- Over parameterization : $d \to \infty$

- Double Descent





- Consider unfreeze $\alpha_k \overset{iid}{\sim} \rho_0(\alpha)$

$\theta = (\theta_k = (\alpha_k, \beta_k), k = 1, \dots, d)$

initialization $\theta_k^0 = (\alpha_k^0, \beta_k^0) = \rho_0(\alpha, \beta)$

$$f_\theta(x) = f_{\theta_0}(x) + \langle \theta - \theta_0, \nabla_\theta f_{\theta_0}(x) \rangle + 2^{nd} \text{ order}$$

query fixed

$$f_\theta(x) = \frac{1}{\sqrt{d}} \sum_{k=1}^{d} \beta_k \sigma(x^T \alpha_k) = \frac{1}{\sqrt{d}} \sum_{k=1}^{d} h_{\theta_k}(x)$$

$$= f_{\theta_0}(x) + \frac{1}{\sqrt{d}} \sum_{k=1}^{d} \nabla_{\theta_k} h_{\theta_k^0}(x)(\theta_k - \theta_k^0) + \underbrace{\frac{1}{2}\frac{1}{\sqrt{d}} \sum_{k=1}^{d} \nabla_{\theta_k}^2 h_{\theta_k^0}(x)(\theta_k - \theta_k^0)^2}_{\downarrow\, 0}$$

$\Delta = f_\theta(x) - f_{\theta_0}(x)$ so $(\theta_k - \theta_k^0) \propto \frac{1}{\sqrt{d}}$ perturbation

also $(\theta_k - \theta_k^0)^2 \propto \frac{1}{d}$

Thus the second term $\longrightarrow 0$

- Note : As $d \to \infty$ The model behaves as a linear model (Tangent Model)

$$f_\theta(x) = f_{\theta_0}(x) + \langle \theta - \theta_0, \underbrace{\nabla_\theta f_{\theta_0}(x)}_{\tilde{h}(x)} \rangle + 2^{nd} \text{ order}$$

Neural Tangent Kernel Regime

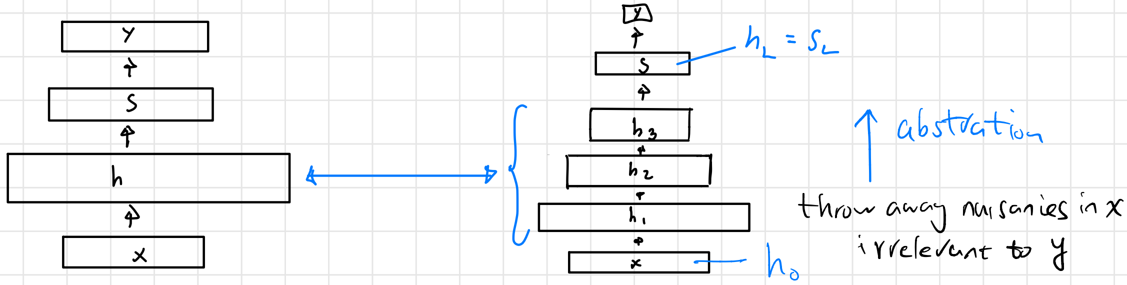
$$\langle \tilde{h}(x), \tilde{h}(x') \rangle = \frac{1}{d} \sum_{k=1}^{d} \nabla_{\theta_k} h_{\theta_k^0}(x) \cdot \nabla_{\theta_k} h_{\theta_k^0}(x')$$

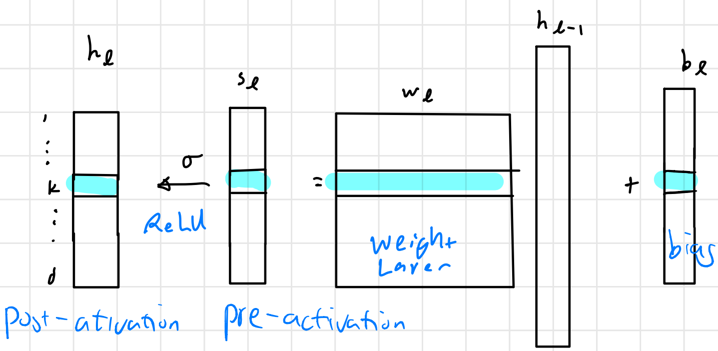
$$\longrightarrow E_{\theta_k}(\qquad)$$

Law of large #

- Multi-Layer Perceptron (MLP)

Recall: MLP:





$$h_{\ell,k} = \sigma(s_{\ell,k})$$

$$s_\ell = W_\ell h_{\ell-1} + b_\ell \Rightarrow s_{\ell,k} = W_{\ell,k} h_{\ell-1} + b_{\ell,k}$$

$$\theta = (W_\ell, b_\ell, \ \ell = 1, \dots, L)$$

- Regression: $\text{Loss} = \frac{1}{2} | y - s |^2$

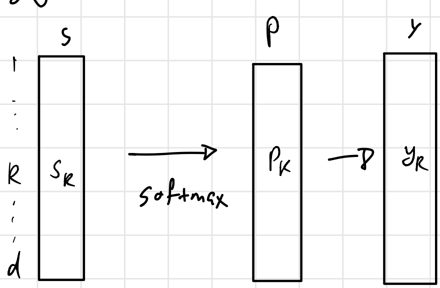
$$\frac{\partial \text{Loss}}{\partial s} = -(y - s) = -e$$

- Classification: $p(y \mid s) = \dfrac{e^{\langle y, s \rangle}}{Z} = \dfrac{e^{y_k s_k}}{\sum_{c=1}^{C} e^{y_c s_c}}$

$$\begin{array}{c} 1 \\ \vdots \\ k \\ \vdots \\ C \end{array} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$\text{Loss} = -\log p(y \mid s)$ cross-entropy

$$\frac{\partial \text{Loss}}{\partial s} = -(y - p) = -e$$

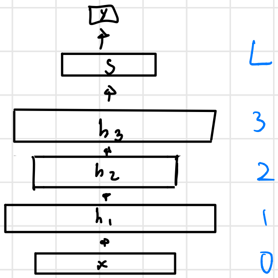


generalized linear model
(GLM)

- Error back-prop

$$-e = \frac{\partial Loss}{\partial h_L}$$

$\downarrow$



Loss

$\vdots$

$\downarrow$

$$\frac{\partial Loss}{\partial h_\ell} \longrightarrow \frac{\partial Loss}{\partial W_\ell}, \frac{\partial Loss}{\partial b_\ell}$$

$\downarrow$

$$\frac{\partial Loss}{\partial h_{\ell-1}}$$

$\downarrow$

$\vdots$

$$\frac{\partial Loss}{\partial h_{\ell-1}}^T = \sum_{k=1}^{d} \frac{\partial Loss}{\partial h_{\ell,k}} \cdot \frac{\partial h_{\ell,k}}{\partial s_{\ell,k}} \quad \frac{\partial s_{\ell,k}}{\partial h_{\ell-1}}^T$$

$$= \sum_{k=1}^{d} \frac{\partial Loss}{\partial h_{\ell,k}} \sigma_k'(s_{\ell,k})\, w_{\ell,k}$$

$$= \begin{array}{c} \left(\cdots \ \dfrac{\partial Loss}{\partial h_{\ell,k}} \ \cdots \right) \\ 1 \quad \cdots \quad k \quad \cdots \quad d \end{array} \begin{pmatrix} \vdots \\ \sigma_k'(s_{\ell,k})\, w_{\ell,k} \\ \vdots \end{pmatrix} \begin{array}{c} 1 \\ \vdots \\ k \\ \vdots \\ d \end{array}$$

$$= \frac{\partial Loss}{\partial h_\ell}^T \sigma_\ell' \odot w_\ell$$

$\downarrow$

row-wise multiply

$$\sigma_\ell' = \begin{pmatrix} \vdots \\ \sigma_k'(s_{\ell,k}) \\ \vdots \end{pmatrix} \begin{array}{c} 1 \\ \vdots \\ k \\ \vdots \\ d \end{array}$$

- $$\frac{\partial Loss}{\partial w_{\ell,k}} = \frac{\partial Loss}{\partial h_{\ell,k}} \frac{\partial h_{\ell,k}}{\partial s_{\ell,k}} \cdot \frac{\partial s_{\ell,k}}{\partial w_{\ell,k}}$$

$$= \frac{\partial Loss}{\partial h_{\ell,k}} \sigma'(s_{\ell,k})\, h_{\ell-1}^T$$

$$= \frac{\partial \text{Loss}}{\partial W} = \begin{array}{cc} \begin{pmatrix} \\ \frac{\partial \text{Loss}}{\partial w_{\ell,k}} \\ \\ \end{pmatrix} & \begin{array}{c} \vdots \\ k \\ \vdots \end{array} \end{array} = \begin{pmatrix} \sigma'(s_{\ell,k}) \frac{\partial \text{Loss}}{\partial h_{k,\ell}} \end{pmatrix} h_{\ell-1}^T =$$

$$= \sigma_\ell' \odot \frac{\partial \text{Loss}}{\partial h_\ell} \cdot h_{\ell-1}^T$$