LEC3 DSA5204

$\boxed{\text{Recap}}$

Shallow NN ⟶ $\boxed{\text{adaptive}}$ basis model ⟶ { advantage : do not suffer from Curse of Dimensionality / dis-advantage : difficult to train }

adaptive basis model, linear basis model ⟶ universal approximator

$\boxed{\text{Problem}}$ { gradient calculation ⟹ expensive / how to calculate gradient in practice }

{ DNN ⟶ FCNN (Simplest one)
extension of GD-method for large scale problems { SGD / Momentum
BP Algo ⟺ ☆ Computational Graph }

① Shallow NN ⟶ Deep NN (DNN)

$\boxed{\text{Regression}}$

$$h^{(0)} = x$$
$$h^{(1)} = g^{(1)}(W^{(1)}h^{(0)} + b^{(1)})$$
$$h^{(i+1)} = g^{(i+1)}(W^{(i+1)}h^{(i)} + b^{(i+1)})$$
$$\hat{y} = w^T h^{(\ell)} + c$$

$\boxed{i = 0, 1, \ldots, \ell-1}$

Q: How many layers is appropriate?

A: larger # of layers may not result in improvements!

↓

Trade-off { model capacity / difficulty in training process

② Advantages for FCNN Architecture

1. $\boxed{\text{sequential model}}$ can be viewed as one PRIOR

$\updownarrow$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad \swarrow\!\nearrow$

☆ $\boxed{\text{built-in property for FCNN}}$ $\qquad\qquad$ $\boxed{\text{inductive bias}}$

2. { better approximation property
   better generalization property

☆

③ $\boxed{\text{Issues}}$

1. Optimization related $\longleftrightarrow$ non-convexity

2. Prone to over-fitting with inappropriate { initialization
   regularization

④ GD $\longrightarrow$ SGD

GD : $\hat{\theta} = \min_{\theta} R_{emp}(\theta)$

$\Rightarrow \theta^{(k+1)} = \theta^{(k)} - \varepsilon \nabla_\theta R_{emp}(\theta^{(k)})$

$\boxed{\text{Mini-batch SGD}}$ (combine) $\rightarrow$ what we use in real application

SGD $\quad R_{\gamma_k}(\theta) \longrightarrow \gamma_k \sim Unif(1, 2, \ldots K)$

unbiased estimator with large variance

☆ $\boxed{\nabla_\theta R_{\gamma_k}(\theta^{(k)}) \longrightarrow \text{estimator of } \boxed{\nabla_\theta R_{emp}(\theta^{(k)})}}$

$\Rightarrow \theta^{(k+1)} = \theta^{(k)} - \varepsilon \nabla_\theta R_{\gamma_k}(\theta^{(k)})$

$\boxed{\text{estimator of } \boxed{\nabla_\theta R_{pop}(\theta^{(k)})}}$

$$\begin{cases} R_{emp}(f) = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i), y_i) \\ R_{pop}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(f(x), y)] \end{cases}$$

☆ **Dynamics of SGD [Example] ⟷ hand analysis**

$$\Rightarrow R(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}(\theta - \theta^{(i)})^2$$

$$\boxed{\{\theta^{(i)}\}_{i=1}^{N} \text{ satisfies that } \begin{cases} \frac{1}{N}\sum_{i=1}^{N}\theta^{(i)} = 0 \\ \frac{1}{N}\sum_{i=1}^{N}(\theta^{(i)})^2 = 1 \end{cases}}$$

Data Point

then $R(\theta) = \frac{1}{2}\theta^2 + \frac{1}{2}$

GD iterates : $\theta_k^{GD} = \theta_{k-1}^{GD} - \varepsilon\,\theta_{k-1}^{GD}$

$$= (1-\varepsilon)^k\,\theta_0$$

SGD iterates : $\theta_k^{SGD} = \theta_{k-1}^{SGD} - \varepsilon(\theta_{k-1}^{SGD} - \theta^{(\gamma_{k-1})})$

$$= (1-\varepsilon)\,\theta_{k-1}^{SGD} + \varepsilon\,\theta^{(\gamma_{k-1})}$$

$$\boxed{\gamma_{k-1} \sim \mathrm{Unif}([1, 2, \ldots, N])}$$

$\boxed{\text{Convergence?}}$ $\Rightarrow \theta_k^{SGD} = (1-\varepsilon)^2\,\theta_{k-2}^{SGD} + (1-\varepsilon)\cdot\varepsilon\,\theta^{(\gamma_{k-2})} + \varepsilon\,\theta^{(\gamma_{k-1})}$

$$= \cdots\cdots$$

$$= \underbrace{(1-\varepsilon)^k\,\theta_0}_{\text{deterministic}} + \underbrace{\varepsilon\sum_{j=1}^{k}(1-\varepsilon)^{j-1}\,\theta^{(\gamma_{k-j})}}_{\text{randomness}}$$

Note that $\boxed{\mathbb{E}_{\gamma_k}[\theta^{(\gamma_k)}] = \frac{1}{N}\sum_{i=1}^{N}\theta^{(i)} = 0}$

⇒ Property :

1. SGD iteration $\xleftrightarrow{\text{equivalent}}$ GD iteration w.r.t expectation

$$\mathbb{E}_\gamma[\theta_k^{SGD}] = (1-\varepsilon)^k \theta_0 = \theta_k^{GD}$$

2. What about variance ? (Second - Moment)

$$\longrightarrow \mathbb{E}_\gamma[(\theta_k^{SGD})^2] = (1-\varepsilon)^{2k}\theta_0^2 + 2\varepsilon(1-\varepsilon)^k \theta_0 \sum_{j=1}^{k} (1-\varepsilon)^{j-1} \underline{\mathbb{E}_{\gamma_{k-j}}[\theta^{(\gamma_{k-j})}]}$$

$$+ \varepsilon^2 \sum_{j,\ell=1}^{k} (1-\varepsilon)^{j+\ell-2} \boxed{\mathbb{E}_{\gamma_{k-j},\gamma_{k-\ell}}[\theta^{(\gamma_{k-j})}\theta^{(\gamma_{k-\ell})}]}$$

(if $\gamma_1, \dots, \gamma_K$ i.i.d)

$$= (1-\varepsilon)^{2k}\theta_0^2 + \varepsilon^2 \cdot \sum_{j=1}^{k}(1-\varepsilon)^{2j-2}$$

$$\mathbb{E}_{\gamma_i,\gamma_j}[\theta^{(\gamma_i)}\theta^{(\gamma_j)}] = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

$$= (1-\varepsilon)^{2k}\theta_0^2 + \frac{\varepsilon}{2-\varepsilon}[1-(1-\varepsilon)^{2k}]$$

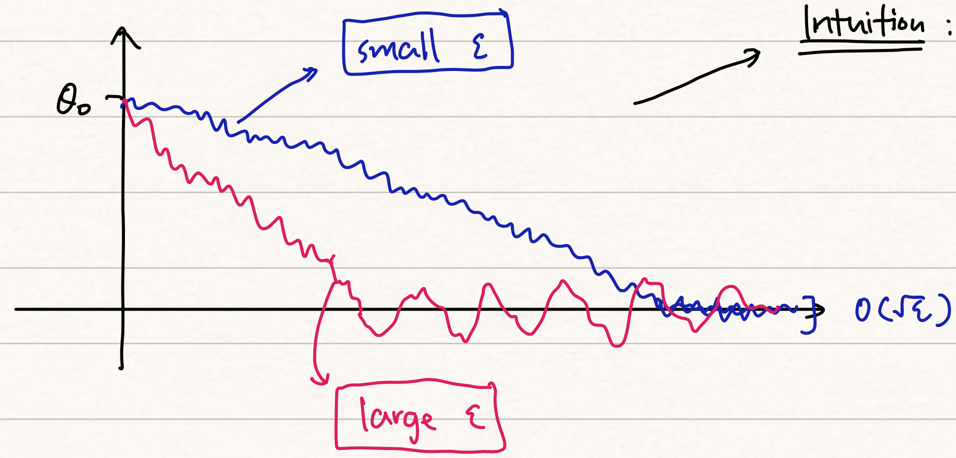
$$\longrightarrow \mathrm{Var}_\gamma[\theta_k^{SGD}] = \mathbb{E}_\gamma[(\theta_k^{SGD})^2] - (\mathbb{E}_\gamma[\theta_k^{SGD}])^2$$

$$= \frac{\varepsilon}{2-\varepsilon}[1-(1-\varepsilon)^{2k}] \xrightarrow{k\to\infty} \frac{\varepsilon}{2-\varepsilon} > 0$$

$$\longrightarrow \boxed{\text{when } \begin{cases} \varepsilon \to 0 \\ k \to \infty \end{cases}, \text{ then } \mathrm{Var}_\gamma[\theta_k^{SGD}] \approx \frac{\varepsilon}{2}}$$

$\boxed{\text{std} \to O(\sqrt{\varepsilon})}$ as $k \to \infty$

Diagram



Intuition : learning rate decay

⑤ SGD with varying learning rate

↓

Learning Rate Decay

Condition
$$\begin{cases} \sum \varepsilon_k = \infty \\ \sum \varepsilon_k^2 < \infty \end{cases} \longrightarrow \varepsilon_k = \frac{1}{k+1}$$

↓

Sufficient Condition for convergence

⑥ Variants of SGD

↓

Momentum method ↔ reduce zig-zag behaviour

GD with momentum

$$\begin{cases} V_{k+1} = \alpha V_k - \varepsilon \nabla_\theta R_{emp}(\theta_k) \\ \theta_{k+1} = \theta_k + V_{k+1} \end{cases}$$

Note: if $\alpha = 0$, then we go back to conventional GD

7. Back Propagation Algo (BP)

↓

Question: How to calculate $\nabla_\theta R_{emp}(\theta)$

$$= \frac{1}{N} \sum_{i=1}^{n} \nabla_\theta L(f_\theta(x_i), y_i)$$

$\boxed{\text{TensorFlow Architecture}}$

Computational Graph ⟷ formulate the network architecture

↓

$\boxed{\text{DAG}}$ Directed Acyclic Graph

↓

visualize the parameter dependency

Chain Rule

1. scalar function

$$h = g \circ f$$

$$\frac{dh}{dx} = \frac{dg}{df} \cdot \frac{df}{dx}$$

2. vector function

$$h = g \circ f \qquad h: \mathbb{R}^m \longrightarrow \mathbb{R}$$

$$f: \mathbb{R}^m \longrightarrow \mathbb{R}^n \quad g: \mathbb{R}^n \longrightarrow \mathbb{R}$$

$$\nabla_x h(x) = \nabla_x f(x) \nabla_f g(f)$$

$$\begin{cases} \nabla_x h(x) \in \mathbb{R}^m \\ \nabla_x f(x) \in \mathbb{R}^{m \times n} \quad \nabla_f g(f) \in \mathbb{R}^n \end{cases}$$

$$\frac{dh}{dx} = \frac{dg}{df} \cdot \frac{df}{dx} \longrightarrow \text{Jacobian } \frac{dh}{dx} = \nabla_x h(x)^T$$

Note: Jacobian = Gradient^T

Toy Example :

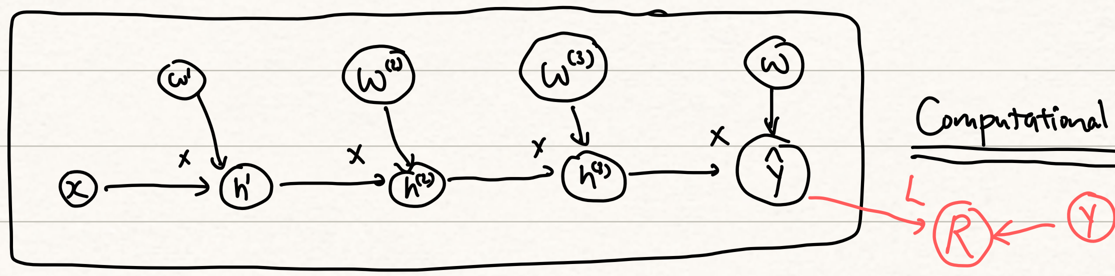
- $h^{(1)} = w^{(1)} x$
- $h^{(2)} = w^{(2)} h^{(1)}$
- $h^{(3)} = w^{(3)} h^{(2)}$
- $\hat{y} = w h^{(3)}$

$\theta = \{ w^{(1)}, w^{(2)}, w^{(3)}, w \}$

$\mathcal{D} = \{ (x, y) \}$

$R(\theta) = L(\hat{y}, y)$

$= L(\hat{y}(\theta), y)$



Computational Graph

→ **Foward Propagation**

given $\underline{x, w^{(i)}, w}$, calculate $\underline{h^{(1)}, h^{(2)}, h^{(3)}, \hat{y}}$

→ **Backward Propagation**

calculate $\frac{dR}{dw}$, $\frac{dR}{dw^{(3)}}$, $\frac{dR}{dw^{(2)}}$, $\frac{dR}{dw^{(1)}}$

a) $\frac{\partial R}{\partial \hat{y}} = \frac{\partial L}{\partial \hat{y}} \longrightarrow p$ (store)

b) $\frac{\partial R}{\partial h^{(3)}} = \frac{\partial R}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h^{(3)}} = p \cdot w \longrightarrow p^{(3)}$ (store)

$\Rightarrow \frac{\partial R}{\partial w^{(3)}} = p^{(3)} \cdot h^{(2)}$

c) $\frac{\partial R}{\partial h^{(2)}} = p^{(3)} \cdot w^{(3)} \longrightarrow p^{(2)}$ (store)

$\Rightarrow \frac{\partial R}{\partial w^{(2)}} = p^{(2)} h^{(1)}$

d) $\frac{\partial R}{\partial h^{(1)}} = p^{(2)} w^{(2)} \longrightarrow p^{(1)}$ (store)

$\Rightarrow \frac{\partial R}{\partial w^{(1)}} = p^{(1)} h^{(0)} = p^{(1)} x$