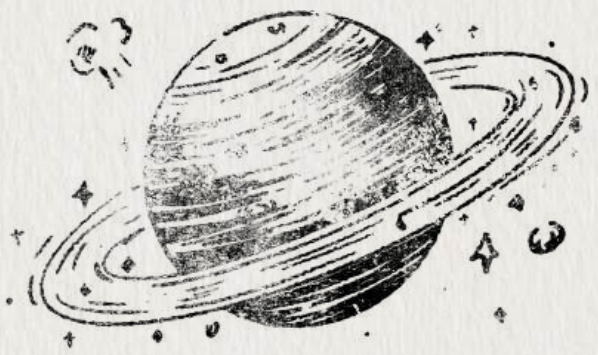
ITEC 5050

Fr 1:30 - 4:20

150, E1



NOTEBOOK

CATCH THE STAR THAT HOLDS YOUR DESTINY. THE ONE THAT FOREVER TWINKLES WITHIN YOUR HEART. TAKE ADVANTAGE OF PRECIOUS OPPORTUNITIES WHILE THEY STILL SPARKLE BEFORE YOU. ALWAYS BELIEVE THAT YOUR ULTIMATE GOAL IS ATTAINABLE AS LONG AS YOU COMMIT YOURSELF TO IT.

Machine Learning and

Artificial Intelligence

Title Introduction

Chapter Date

Review / / / / /

Rating ○ ○ ○ ○ ○

Define Learning Problem

- Task, T
  Predicting whether or not this startup will receive the series A round of investment.
- Performance measure, P
  Percent of "not bad+" responses received in the interview process.
- Experience, E
  Examples of <question, answer, feedback> triplets

Task: Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$.

$\equiv$ Construct **prediction rule** $f : \mathcal{X} \to \mathcal{Y}$

Performance: Risk $R(f) \equiv \mathbb{E}_{XY}\,[\text{loss}(Y, f(X))]$

$(X, Y) \sim P_{XY}$

Experience: Training Data $\{(X_i, Y_i)\}_{i=1}^n \overset{i.i.d.}{\sim} P_{XY}$ **(unknown)**

Supervised vs Unsupervised

Supervised Learning – Learning with a teacher

$\{(X_i, Y_i)\}_{i=1}^n$ →  → $\hat{f}_n$

Documents, topics — Mapping between documents and topics

Unsupervised Learning – Learning without a teacher

$\{X_i\}_{i=1}^n$ →  → $\hat{f}_n$

Documents — Model for word distribution OR clustering of similar documents

Linear Regression

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

A simplified model:

$y = w_1x_1 + w_2x_2 + w_3x_3 + b$

Home Owner | Marital Status | Annual Income

A vector version:

$y = \mathbf{w}^T\mathbf{x} + b$

- y is the prediction
- $\mathbf{w}^T$ is the weight
- b is the bias
- $\mathbf{w}^T$ and b together are the parameters

How to quantitatively handle nominal data?
A common solution: one-hot encoding
(1) Home Owner: Yes → [0, 1] No → [1, 0]
(2) Marital Status: Married → [1,0, 0], Single → [0, 1, 0], Divorced → [0, 0, 1]

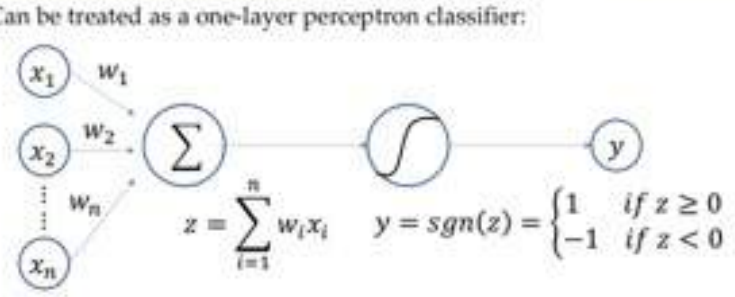
Modern typology:
https://dl.acm.org/doi/pdf/10.1145/2533988

61

Linear Regression

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- Can be treated as a one-layer perceptron classifier:



- Deep network: $y = f_0(f_1(\cdots f_n(x)))$
- Unlike linear regression model, $f_i(x)$ is non-linear function

62

Parameter Estimation and Loss Function

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Loss: $L = \frac{1}{N}\sum_n e_n$

where e_n represents the prediction error of the n-th instance

- Loss: how good a set of values is.
- Loss is a function of parameters

$e = |y - \hat{y}|$ L is mean absolute error (MAE)

$e = (y - \hat{y})^2$ L is mean square error (MSE)

If y and $\hat{y}$ are both probability distributions ⟹ Cross-entropy

63

Optimization

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

$w^* = arg\min_w L$

Gradient Descent

- (Randomly) Pick an initial value w^0
- Compute $\frac{\partial L}{\partial w}|_{w=w^0}$

65

Optimization

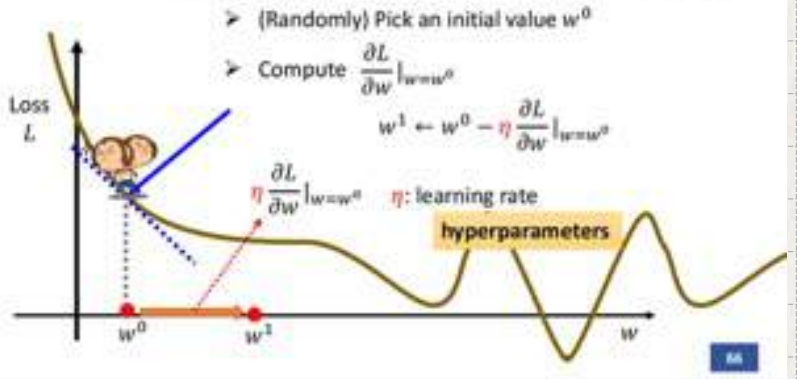
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

$w^* = arg\min_w L$

Gradient Descent

- (Randomly) Pick an initial value w^0
- Compute $\frac{\partial L}{\partial w}|_{w=w^0}$

$w^1 \leftarrow w^0 - \eta\frac{\partial L}{\partial w}|_{w=w^0}$



66

Optimization

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

$w^* = arg\min_w L$

Gradient Descent

- (Randomly) Pick an initial value w^0
- Compute $\frac{\partial L}{\partial w}|_{w=w^0}$

$w^1 \leftarrow w^0 - \eta\frac{\partial L}{\partial w}|_{w=w^0}$

- Update w iteratively

Does local minima truly cause the problem?

67

Optimization

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Gradient Descent

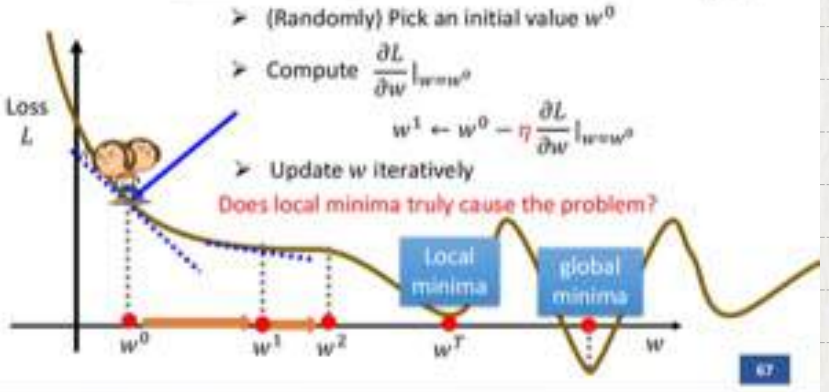
$$w^* = arg\min_w L$$

- (Randomly) Pick an initial value w^0
- Compute $\frac{\partial L}{\partial w}|_{w=w^0}$

$$w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w}|_{w=w^0}$$

- Update w iteratively

Does local minima truly cause the problem?



67

Optimization

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- (Randomly) Pick initial values w^0, b^0 $\quad w^*, b^* = arg\min_{w,b} L$
- Compute

$$\frac{\partial L}{\partial w}|_{w=w^0,b=b^0} \quad w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w}|_{w=w^0,b=b^0}$$

$$\frac{\partial L}{\partial b}|_{w=w^0,b=b^0} \quad b^1 \leftarrow b^0 - \eta \frac{\partial L}{\partial b}|_{w=w^0,b=b^0}$$

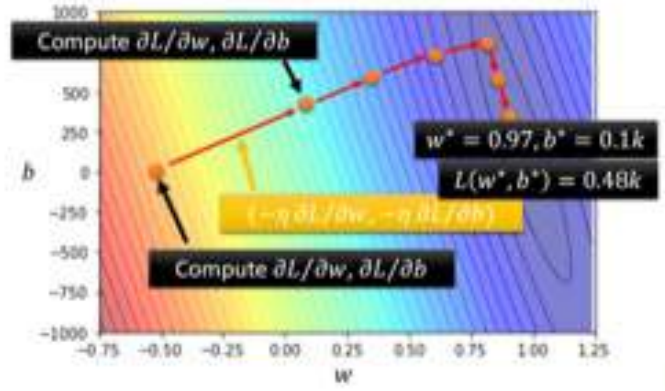
Can be done in one line in most deep learning frameworks

- Update w and b interatively

68

Optimization

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)



69

Model Assessment

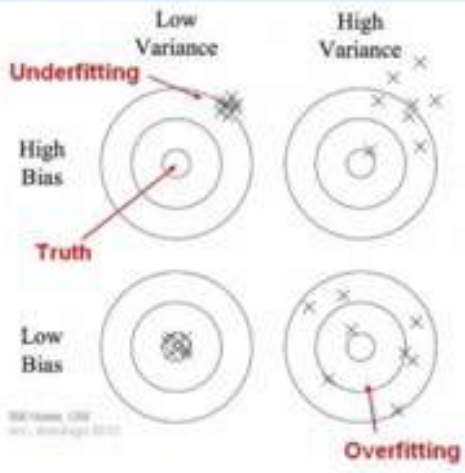
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- Generalization of a ML model refers to how well the rules/patterns/functions learned by the ML model, apply to examples not seen by the model.
- The generalization performance of a machine learning method relates to its prediction capability on independent test sets.
- Assessment of this performance is extremely important in practice, since it guides the choice of the machine learning method or model.
- Further, this gives us a measure of the quality of the ultimately chosen model.

Bias and Variance

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- **Bias**: the di… correct valu…
- High bias – …
- **Variance** is … tells us spr…
- High variar… → perform…



…r model and the …aining and test sets. …data point which …t hasn't seen before …rates on test data

71

Bias and Variance

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

If we assume that $Y = f(X) + \epsilon$ and $E[\epsilon] = 0$ and $Var(\epsilon) = \sigma_\epsilon^2$ then we can derive the expression for the expected prediction error of a regression fit $\hat{f}(X)$ at an input $X = x_0$ using squared error loss

$$Err(x_0) = E[(Y - \hat{f}(x_0))^2 | X = x_0]$$

For notational simplicity let $\hat{f}(x_0) = \hat{f}$, $f(x_0) = f$ and recall that $E[f] = f$ and $E[Y] = f$

$$\begin{aligned} E[(Y - \hat{f})^2] &= E[(Y - f + f - \hat{f})^2] \\ &= E[(y - f)^2] + E[(f - \hat{f})^2] + 2E[(f - \hat{f})(y - f)] \\ &= E[(f + \epsilon - f)^2] + E[(f - \hat{f})^2] + 2E[fY - f^2 - \hat{f}Y + \hat{f}f] \\ &= E[\epsilon^2] + E[(f - \hat{f})^2] + 2(f^2 - f^2 - fE[\hat{f}] + fE[\hat{f}]) \\ &= \sigma_\epsilon^2 + E[(f - \hat{f})^2] + 0 \end{aligned}$$

For the term $E[(f - \hat{f})^2]$ we can use a similar trick as above, adding and subtracting $E[\hat{f}]$ to get

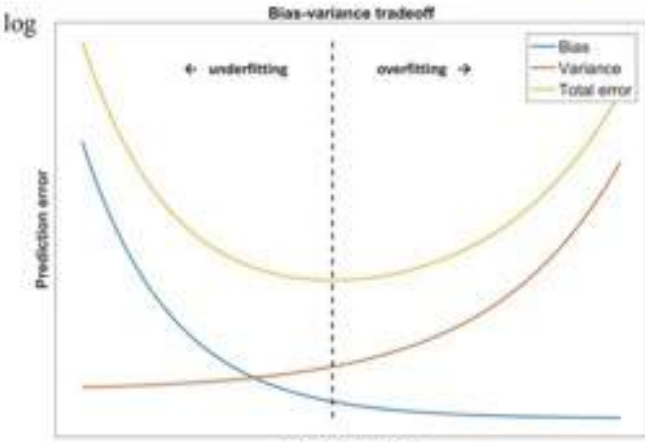
$$\begin{aligned} E[(f - \hat{f})^2] &= E[(f + E[\hat{f}] - E[\hat{f}] - \hat{f})^2] \\ &= E\left[f - E[\hat{f}]\right]^2 + E\left[\hat{f} - E[\hat{f}]\right]^2 \\ &= \left[f - E[\hat{f}]\right]^2 + E\left[\hat{f} - E[\hat{f}]\right]^2 \\ &= Bias^2[\hat{f}] + Var[\hat{f}] \end{aligned}$$

72

Bias and Variance

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Putting it log



73

Title

Chapter Date

Review / / / / /

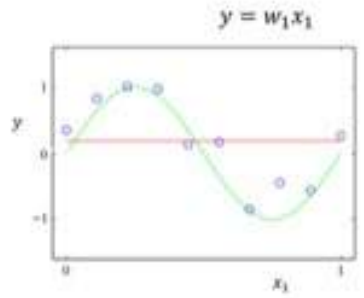
Rating ○ ○ ○ ○ ○

Model Assessment

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$y = w_1 x_1$

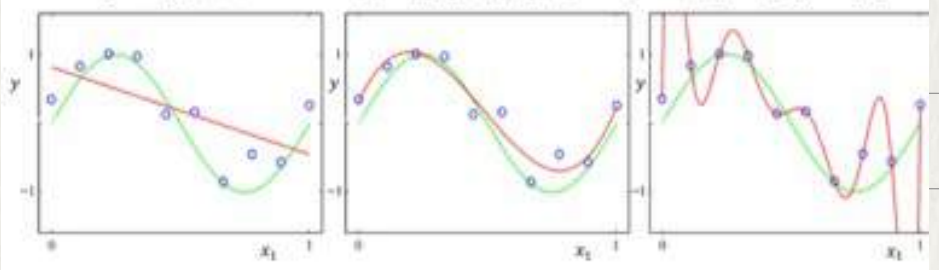


74

Model Assessment

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

$y = w_1 x_1 + b$ $y = w_1 x_1 + w_2 x_1^2 + b$ $y = w_1 x_1 + w_2 x_1^2 + w_3 x_1^3 + b$



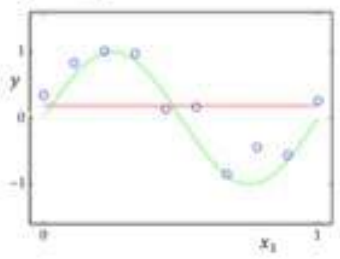
75

Model Assessment

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

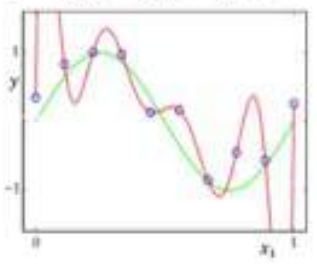
Underfitting : model is too simple — does not fit the data.

$y = w_1 x_1$



Overfitting : model is too complex — fits perfectly, does not generalize.

$y = w_1 x_1 + w_2 x_1^2 + w_3 x_1^3 + b$



76

Regularization

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting.

Original Loss

New Loss: $L = \frac{1}{N}\sum_{n} e_n + \lambda J(f)$

where λ is a positive hyperparameter, $J(f)$ represents the model's complexity. The term $\lambda J(f)$ is called the regularizer or penalty term.

The commonly used regularization techniques are :

1. Lasso Regularization – L1 Regularization
2. Ridge Regularization – L2 Regularization

77

Regularization

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- **L1 Regularization**
- Adds the *"absolute value of magnitude"* of the coefficient as a penalty term to the loss function.
- Helps achieve feature selection by penalizing the weights to approximately equal to zero if that feature does not serve any purpose in the model.

$$L = \frac{1}{N}\sum_{n} e_n + \lambda \sum_{i=1}^{m} |w_i|$$

n = number of instances
m = number of features

- **L2 regularization**
- Adds the *"squared magnitude"* of the coefficient as a penalty term to the loss function.

$$L = \frac{1}{N}\sum_{n} e_n + \lambda \sum_{i=1}^{m} w_i^2$$

78

Summary – Linear Regression

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

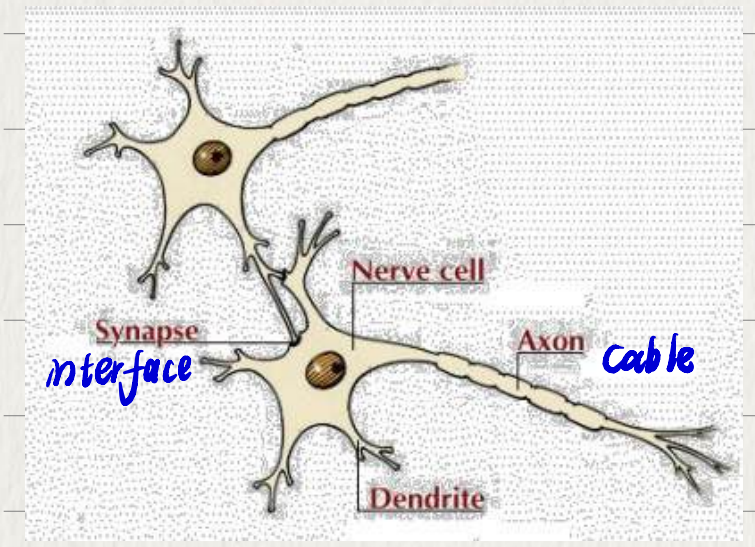
Linear regression exemplifies recurring themes of supervised learning:

- formulate an optimization problem
- choose a model and define a loss function
- improve the generalization by adding a regularizer
- direct solution (set derivatives to zero) gradient descent
- update parameters to solve the optimization problem

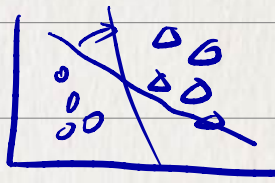
79

Deep Feedforward Networks

1. Perceptron 感知器



"分类"



$X = (x_1, \dots, x_n)$

$d_i = \begin{cases} 1 & \text{线上方} \\ -1 & \text{线下方} \end{cases}$

线 l: $w_0 + w_1 x_1 + w_2 x_2 = w^T x = 0$

$f(x) = w^T x_i = \begin{cases} >0 & \text{线上方} \\ \le 0 & \text{线下方} \end{cases}$

Intialize $w=0$ and $b=0$.

repeat

if $y_i [\langle w, x_i \rangle + b] \le 0$ then

$w \leftarrow w + y_i x_i$ and $b \leftarrow b + y_i$

end if

until all classified correctly

- Nothing happened if classified correctly
- Weight vector is linear combination $w = \sum_{i=1}^{n} y_i x_i$
- Classifier is linear combination of inner products $f(x) = \sum_{i=1}^{n} y_i \langle x_i, x \rangle + b$

Convergence Theorem 收敛定理

If there exists some (w^*, b^*) with unit length and $y_i [\langle x_i, w^* \rangle + b^*] \ge \rho$ for all i,

then the perceptron converges to a linear separator after a number of steps

bounded by $(b^{*2}+1)(r^2+1)\rho^{-2}$ where $\|x_i\| \le r$

- ① Dimensionality independent
- ② Order independent (i.e., also worst case)
- ③ Scales with 'difficulty' of problem

Proof: Starting Point: $w_i = 0$, $b_i = 0$

Step 1: Bound on the increase of alignment

Denote by w_i the value of w at step i

Canalogously b_i. Alignment: $\langle (w_i, b_i), (w^*, b^*) \rangle$

For error in observation (x_i, y_i) we get

$$\begin{aligned} &\langle (w_{j+1}, b_{j+1}) * (w^*, b^*) \rangle \\ &= \langle [(w_j, b_j) + y_i (x_i, 1)], (w^*, b^*) \rangle \\ &= \langle (w_j, b_j), (w^*, b^*) \rangle + y_i \langle (x_i, 1) * (w^*, b^*) \rangle \\ &\ge \langle (w_j, b_j), (w^*, b^*) \rangle + \rho \ge j\rho \end{aligned}$$

Alignment increases with number of errors.

Step 2: Cauchy-Schwartz for the Dot Product

$$\begin{aligned} \langle (w_{j+1}, b_{j+1}) \cdot (w^*, b^*) \rangle &\le \|(w_{j+1}, b_{j+1})\| \, \|(w^*, b^*)\| \\ &\le \sqrt{1 + (b^*)^2} \, \|(w_{j+1}, b_{j+1})\| \end{aligned}$$

Step 3: Upper Bound on $\|(w_j, b_j)\|$

$$\begin{aligned} \|(w_{j+1}, b_{j+1})\|^2 &= \|(w_j, b_j) + y_i (x_i, 1)\|^2 \\ &= \|(w_j, b_j)\|^2 + 2 y_i \langle (x_i, 1), (w_j, b_j) \rangle + y_i^2 \|(x_i, 1)\|^2 \\ &\le \|(w_j, b_j)\|^2 + \|(x_i, 1)\|^2 \le j(r^2+1) \end{aligned}$$

Step 4: Combination of first three steps

$$j\rho \le \sqrt{1 + (b^*)^2} \, \|(w_{j+1}, b_{j+1})\| \le \sqrt{j(r^2+1)((b^*)^2+1)}$$

Solving for j proves the theorem.

随机梯度

Stochastic Gradient With Hinge Loss

① Only need to store errors. This gives a compression bound for perceptron.

② This is stochastic gradient descent on hinge loss

$l(x_i, y_i, w, b) = \max(0, 1 - y_i[\langle w, x_i \rangle + b])$

③ Fails miserably with noisy data

2. Loss Function and Objectives

Objectives

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- (Linear) **Function**
  $f(x) = \langle w, x \rangle + b$
  We need to define what to **do** with it
- **Regression**
  Real valued y, goodness measured by y - f(x)

$l(y, f(x)) = \lvert y - f(x) \rvert$	absolute value
$l(y, f(x)) = \frac{1}{2}(y - f(x))^2$	least mean squares
$l(y, f(x)) = \max(0, \lvert y - f(x) \rvert - \epsilon)$	ϵ-insensitive
$l(y, f(x)) = \begin{cases} \lvert y - f(x) \rvert - 0.5 & \text{if } \lvert y - f(x) \rvert > 1 \\ 0.5(y - f(x))^2 & \text{otherwise} \end{cases}$	Huber

27

Objectives

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- **Classification**
  - Binary y, e.g. {apples, oranges}
  - Multiple categories, e.g. {red, green, blue}
  - Ordinal relationship, e.g. {A, B, C, D, Fail}

$l(y, f(x)) = log(1 + e^{-yf(x)})$	logistic
$l(y, f(x)) = \max(0, 1 - yf(x))$	soft-margin
$l(y, f(x, \cdot)) = \log \sum_{y'} e^{f(x,y')} - f(x, y)$	multiclass logistic

30

多层神经网络处理非线性可分 多层+非线性激活函数，采用阶跃函数 三层神经网络可进行任意分类

3. Stochastic Gradient Descent

随机梯度下降

Empirical Risk: $R_{emp}[f] := \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i))$

Stochastic gradient descent (pick random x, y) $w_{t+1} \to w_t - \eta_t \partial_{f(x_i)} l(y_i, f(x_i)) \partial_w f(x_i)$

梯度下降更新公式

Chain rule: $\partial_x [f_2 \circ f_1](x) = [\partial_{f_1} f_2 \circ f_1](x) [\partial_x f_1](x)$

Convergence in Expectation — initial loss

$E_{\bar{\theta}}[l(\bar{\theta})] - l^* \leq \frac{R^2 + L^2 \sum_{t=0}^{T-1} \eta_t^2}{2 \sum_{t=0}^{T-1} \eta_t}$ where $l(\theta) = E_{(x,y)}[l(y, \langle \phi(x), \theta \rangle)]$ and $l^* = \inf_{\theta \in X} l(\theta)$ and $\bar{\theta} = \frac{\sum_{t=0}^{T-1} \theta_t \eta_t}{\sum_{t=0}^{T-1} \eta_t}$

使小y标签和大尺寸能推近

expected loss

parameter average

Proof: Show that parameters converge to minimum

$\theta^* \in \operatorname{argmin}_{\theta \in X} l(\theta)$ and set $r_t := \|\theta^* - \theta_t\|$

$r_{t+1}^2 = \|\pi_X[\theta_t - \eta_t g_t] - \theta^*\|^2$

$\leq \|\theta_t - \eta_t g_t - \theta^*\|^2$

$= r_t^2 + \eta_t^2 \|g_t\|^2 - 2\eta_t \langle \theta_t - \theta^*, g_t \rangle$ — by convexity

hence $E[r_{t+1}^2 - r_t^2] \leq \eta^2 L^2 + 2\eta_t [l^* - E[l(\theta_t)]] \leq \eta^2 L^2 + 2\eta_t [l^* - E[l(\bar{\theta})]]$

- Summing over inequality for t proves claim
- This yields randomized algorithm for minimizing objective functions (try log times and pick the best / or average median trick)

Title 梯度下降法 { 随机选取初始值 $(W^{(0)}, b^{(0)})$; 应用迭代算法求目标函数局部极值.

Chapter Date

Review / / / / /

Rating

Rates

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- Guarantee $\mathbf{E}_{\bar{\theta}}[l(\bar{\theta})] - l^* \leq \frac{R^2 + L^2 \sum_{t=0}^{T-1} \eta_t^2}{2 \sum_{t=0}^{T-1} \eta_t}$
- If we know R, L, T pick constant learning rate

  $\eta = \frac{R}{L\sqrt{(T)}}$ and hence $\mathbf{E}_{\bar{\theta}}[l(\bar{\theta})] - l^* \leq \frac{R[1+1/T]L}{2\sqrt{(T)}} < \frac{LR}{\sqrt{T}}$
- If we don't know T pick $\eta_t = O(t^{-\frac{1}{2}})$

  This costs us an additional log term

  $\mathbf{E}_{\bar{\theta}}[l(\bar{\theta})] - l^* = O(\frac{logT}{\sqrt{T}})$

39

Strong Convexity

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

$l_i(\theta') \geq l_i(\theta) + \langle \partial_\theta l_i(\theta), \theta' - \theta \rangle + \frac{1}{2}\lambda \left\| \theta - \theta' \right\|^2$

- Use this to bound the expected deviation

  $r_{t+1}^2 \leq r_t^2 + \eta_t^2 ||g_t||^2 - 2\eta_t \langle \theta_t - \theta^*, g_t \rangle$

  $\leq r_t^2 + \eta_t^2 L^2 - 2\eta_t [l_t(\theta_t) - l_t(\theta^*)] - 2\lambda\eta_t r_k^2$

  hence $\mathbf{E}[r_{t+1}^2] \leq (1 - \lambda h_t)\mathbf{E}[r_t^2] - 2\eta_t [\mathbf{E}[l(\theta_t)] - l^*]$
- Exponentially decaying averaging

  $\bar{\theta} = \frac{1-\sigma}{1-\sigma^T} \sum_{t=0}^{T-1} \sigma^{T-1-t} \theta_t$

  and plugging this into the discrepancy yields

  $l(\bar{\theta}) - l^* \leq \frac{2L^2}{\lambda T} log[1 + \frac{\lambda R T^{\frac{1}{2}}}{2L}]$ for $\eta = \frac{2}{\lambda T} log[1 + \frac{\lambda R T^{\frac{1}{2}}}{2L}]$

40

Key Components:

- Data & distribution
- Loss Function
- Function class
- Optimization procedure
- Capacity control

Model Selection
- Simple Model
- High Dimensional model
- Even for $d = O(n)$ not enough data to handle noise

4 Backprop

反向传播算法.

Chain Rules

从输出回输入求导.

Backpropagation

Layers & Gradients

$y_i = W_i x_i$

$x_{i+1} = \sigma(y_i)$

$\partial_{x_i} y_i = W_i$

$\partial_{w_i} y_i = x_i$

$\partial_{y_i} x_{i+1} = \sigma'(y_i)$

$\Rightarrow \partial_{x_i} x_{i+1} = \sigma'(y_i) W_i^T$

Backpropagtion

$\partial_{x_i} l(y, y_n) = \partial_{x_{i+1}} l(y, y_n) \partial_{x_i} x_{i+1}$

$= \partial_{x_{i+1}} l(y, y_n) \sigma'(y_i) W_i^T$

$\partial_{w_i} l(y, y_n) = \partial_{y_i} l(y, y_n) \partial_{w_i} y_i$

$= \partial_{x_{i+1}} l(y, y_n) \sigma'(y_i) x_i^T$

$g_n = l'(y, y_n) W_n$

$g_i = g_{i+1} \sigma'(y_i) W_i^T$

$\partial_{w_i} l(y, y_n) = g_{i+1} \sigma'(y_i) x_i^T$

Optimization

Layer Representation

$y_i = W_i x_i$

$x_{i+1} = \sigma(y_i)$

Gradient descent

$W_i \leftarrow W_i - \eta \partial_{w_i} l(y, y_n)$

- Second order method (use higher derivatives)
- stochastic gradient descent (use only one sample)
- Minibatch (small subset)

5. Layers

- Full Connected
  - Forward Mapping
    $y_i = W_i x_i$
    $x_{i+1} = \sigma(y_i)$
    with subsequent nonlinearity
  - Backprop gradients
    $\partial_{x_i} x_{i+1} = \sigma'(y_i) W_i^T$
    $\partial_{W_i} x_{i+1} = \sigma'(y_i) x_i^T$
  - General purpose layer
- Convolutional Layers
  - Feature Locality
    Relevant information only in neighborhood of pixel
    $y_{ij} = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} W_{ij,ab} x_{i+a,j+b}$
  - Translation Invariance
    Weight invariant relative to shift in point of view $y_{ij} = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} W_{ab} x_{i+a,j+b}$

6. Optimization

较大的LR分使参数更新较大，前期收敛较快，但后期会造成loss的震荡，不收敛。

Learning Rate Decay

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- **Constant**
  (requires schedule for piecewise constant, tricky)
- **Useful hack**
  Constant until no more improvement on validation set
- **Polynomial decay**
  $\eta(t) = \frac{alpha^\gamma}{\beta + t}$
  Recall exponent of 0.5 for conventional SGD, 1 for strong convexity. Bottou picks 0.75
- **Exponential decay**
  $\eta(t) = \alpha e^{-\beta t}$
  not recommended since quite aggressive

74

AdaGrad

自适应学习率算法

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

- **Adaptive learning rate** (preconditioner)
  $\eta_{ij}(t) = \frac{\eta_0}{\sqrt{K + \sum_t g_{ij}^2(t)}}$
- For directions with large gradient, decrease learning rate aggressively to avoid instability
- If gradients start vanishing, learning rate decrease reduces, too
- Local variant $\eta_{ij}(t) = \frac{\eta_t}{\sqrt{K + \sum_{t'=t-\tau}^{t} g_{ij}^2(t')}}$

Duchi, Hazan, Singer, 2010
http://www.magicbroom.info/Papers/DuchiHaSi10.pdf

time window

75

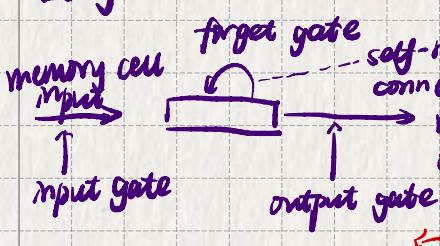
→ 加速模型收敛.

- Momentum
  局部极小值
  Helps with local minima
  Flat (noisy) gradients
  $m_t = (1-\lambda) m_{t-1} + \lambda g_t$
  $w_t \leftarrow w_t - \eta_t g_t - \tilde{\eta}_t m_t$
  Can lead to oscillations for large momentum
  振荡
  Nesterov's accelerated gradient
  加速梯度
  $m_{t+1} = \mu m_t + \varepsilon g(w_t - \mu m_t)$
  $w_{t+1} = w_t - m_{t+1}$
- Capacity Control
  Minimizing loss can lead to overfitting
  Weight Decay $w_t \leftarrow w_t - \eta_t g_t$
  $w_t \leftarrow (1-\lambda) w_t - \eta_t g_t$
- Dropout {Avoid parameter sensitivity
  Distributed representation
  Randomized sparsification

7. Memory & State

- Autoregressive Models AR 自回归
  Time series of observations 考虑时间变化.
  $\cdots, x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}, \cdots$
  Estimate $x_{t+1} = f(x_t, \cdots, x_{t-T})$ e.g. via deep net
- Latent state models
  "记忆"
  $x_{t+1} = f(x_t, \cdots, x_{t-T}, z_t, \cdots, z_{t-T})$
  $z_{t+1} = g(x_{t+1}, \cdots, x_{t-T}, z_t, \cdots, z_{t-T})$
  also called Recurrent Neural Network RNN 循环神经网络.

LSTM Long Short Term Memory



$f_t = \sigma_g(W_f x_t + U_f c_{t-1} + b_f)$
$i_t = \sigma_g(W_i x_t + U_i c_{t-1} + b_i)$
$o_t = \sigma_g(W_o x_t + U_o c_{t-1} + b_o)$
$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_c(W_c x_t + b_c)$
向下传递+更新
$h_t = o_t \odot \sigma_h(c_t)$

Notes / 25 50 75 100

Chapter Date

Rating

Title Regularization and Optimization for Deep Models

Framework of ML:

{ Traing data $\{(x_1, y_1), \cdots (x_n, y_n)\}$
Testing data $\{x^{N+1}, x^{N+2}, \cdots, x^{N+M}\}$

Training
function with unknown – define loss from – optimization
traning data

Testing : Label and used in practice

Guideline

How to determine?
- see right
- large

Loss on training data

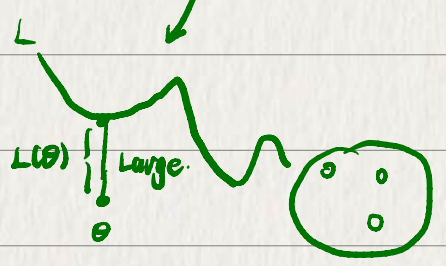
small

model bias
make your model complex
$y = f_\theta, \theta^1, \theta^2 \cdots \theta^{(x)}$ respectively
– too simple
redesign and make it flexible.
more feature $y = b + \sum_j w_j x_j$
deep learning (more neurons, layers)
$y = b + \sum_i c_i \, \text{sigmoid}(b_i + \sum_j w_{ij} x_j)$

optimization
advanced optimization

loss on training data small

large

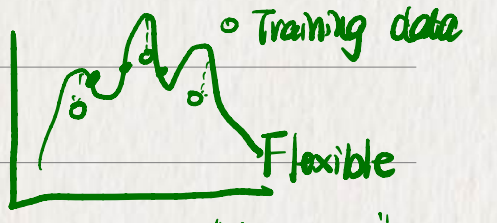


mismatch
Resample
Your training and testing data have different distributions. Be aware of how data is generated.

overfitting
Create more training data (data augmentation)
Limit model flexibility
{ less parameters, sharing parameter
less features
Regularization
Dropout
Early stopping

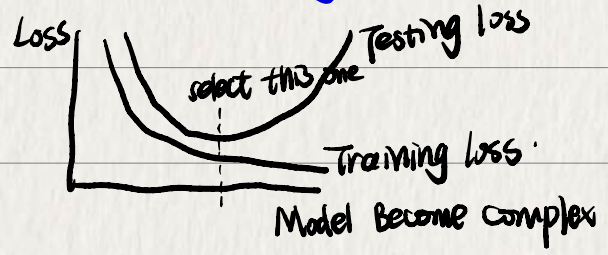
Small loss on training, large loss on testing data

- Testing data
- Training data

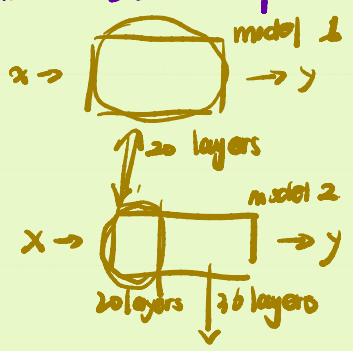


An extreme example $f(x) = \begin{cases} \hat{y}^i & \exists x^i = x \\ \text{random} & \text{otherwise} \end{cases}$
zero training loss, large testing loss

Bias-Complexity Trade-off



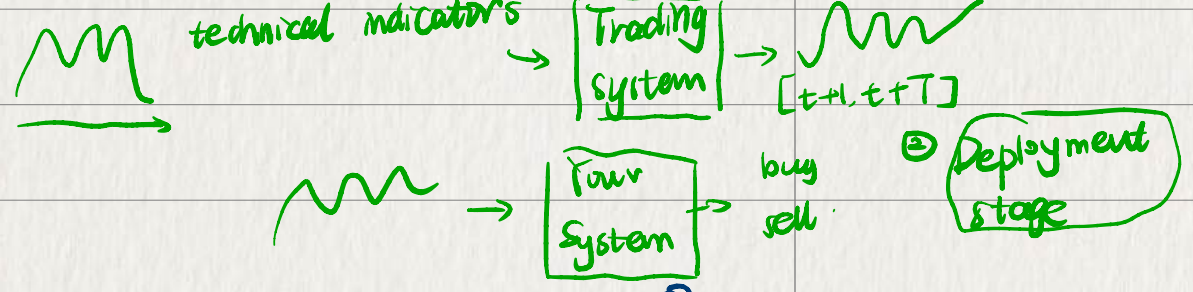
Model Bias vs Optimization Issue



- start from simple model (shallower networks) which are easier to optimize.
- If deeper networks does not obtain smaller loss on training data, then there is optimization issue.

model2 => model 1 + Identify Function

e.g.

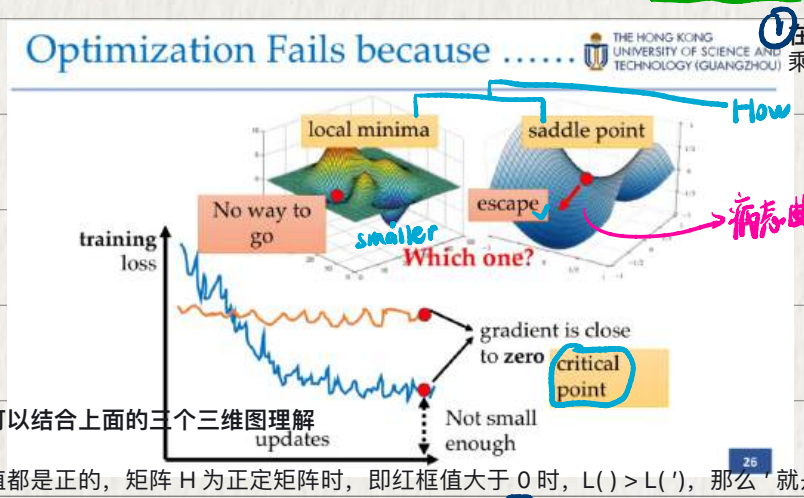


N-fold Cross Validation

Training Set

			Model 1	2	3
Train	Train	Val	mse=0.2	0.4	0.4
Train	Val	Train	0.4	0.5	0.5
Val	Train	Train	0.3	0.6	0.3
			avg=0.3	0.5	0.4

Testing Set → Testing Set
public private



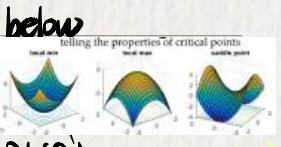
How to determine? ① Tayer Series Approximation

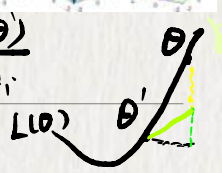
$L(\theta)$ around $\theta = \theta'$ can be appoximated below
A critical point

$L(\theta) \approx L(\theta') + (\theta - \theta')^T g + \frac{1}{2}(\theta - \theta')^T H (\theta - \theta')$

Gradient g is a vector, $g = \nabla L(\theta')$, $g_i = \frac{\partial L(\theta')}{\partial \theta_i}$

Hessian H is a matrix, $H_{ij} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} L(\theta')$





① 在 θ′ 附近的 的 loss 函数可以用泰勒展开式近似表示。下图中的纵坐标是 loss 值，横坐标是向量 θ 。一阶微分乘自变量差值是 y 轴上的绿色长度（可以用 tan 来理解），二阶微分乘两个自变量差值是 y 轴上的红色长度。

③ 下面两句话可以结合上面的三个三维图理解

当所有特征值都是正的，矩阵 H 为正定矩阵时，即红框值大于 0 时，L(θ) > L(θ′)，那么 θ′ 就是局部最小点。

② 在临界点位置，梯度 g 为 0，所以绿框值为 0，只要知道 H 值正负，就知道 L(θ) 与 L(θ′) 的大小关系了。

当特征值有正有负时，即红框有时大于 0 有时小于 0，则 θ′ 就是鞍点。

$v^T H v$

② Hessian: At critical point: $L(\theta) \approx L(\theta') + \frac{1}{2}(\theta-\theta')^T H (\theta-\theta')$

For all v, $v^T H v > 0$ → Around θ': $L(\theta) > L(\theta')$ → Local minima

= H is positive definite = All eigenvalues are positive.

For all v, $v^T H v < 0$ → Around θ': $L(\theta) < L(\theta')$ → Local maxima

= H is negative definite = All eigenvalues are negative.

Sometimes $v^T H v > 0$, sometimes $v^T H v < 0$ → Saddle point

Some eigenvalues are positive, and some are negative.

e.g. $y = w_1 w_2 x$, $x = 1$, w_1, w_2, $y \Leftrightarrow \hat{y} = 1$

$L = (\hat{y} - w_1 w_2 x)^2 = (1 - w_1 w_2)^2$

⑨ $\frac{\partial L}{\partial w_1} = 2(1-w_1 w_2)(-w_2) = 0$

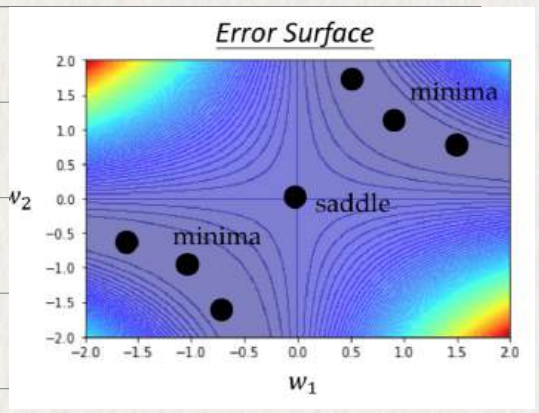
$\frac{\partial L}{\partial w_2} = 2(1-w_1 w_2)(-w_1) = 0$

Ⓗ $\frac{\partial^2 L}{\partial w_1^2} = 2(-w_2)(-w_2) > 0$

$\frac{\partial^2 L}{\partial w_2 \partial w_1} = -2 + 4 w_1 w_2 = -2$

$\frac{\partial^2 L}{\partial w_2^2} = 2(-w_1)(-w_1) > 0$

$\frac{\partial^2 L}{\partial w_1 \partial w_2} = -2 + 4 w_1 w_2 = -2$



Critical point: $w_1 = 0$, $w_2 = 0$

$H = \begin{bmatrix} 0 & -2 \\ -2 & 0 \end{bmatrix}$ $\lambda_1 = 2$, $\lambda_2 = -2$

Saddle point 鞍点

e.g Explain:

y的head是真实值，采用均方误差来计算 loss (L)，计算梯度即一阶微分使之为 0，求出参数 w1和 w2的值，此值对应位置就是临界点；然后将这两个值代入二阶微分求得 H矩阵；再通过 H矩阵去算特征值，算得两个特征值为2和-2，所以该点是个鞍点。

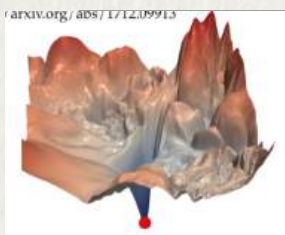
如何确定参数更新的方向?

鞍点处的 H矩阵中，特征值有正有负，这些正负能指导我们更新参数的方向。由下图中间式子推导可知，特征值小于 0，则红色框框里面的值始终小于 0，也就是说L( ) < L(′)，这就是鞍点处loss能继续下降的方向。所以我们只要令 v = u 即可，也就是 − ′ = u，= ′ + u，此时 处的 L 比 ′ 处的 L要低。

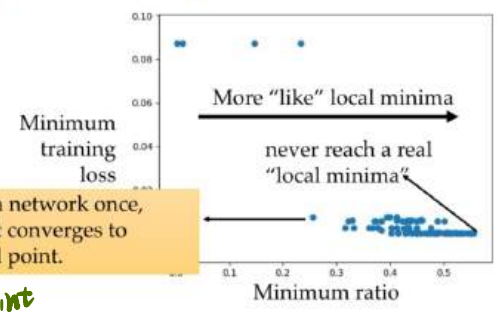
举例说明

还是刚刚那个例子，求出负的特征值对应的特征向量，沿着特征向量的方向更新参数就可以减低 loss。特征向量有无穷多个，取一个即可。

上面讲的利用泰勒展开式判断临界点类型，并用 H矩阵确定参数更新方向的方法。在实际中很少用到，因为二次微分和找特征值、特征向量，需要很大的运算量，不值得。

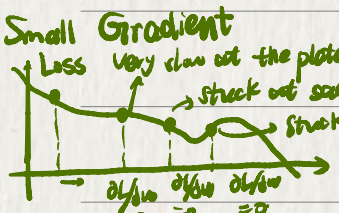


Empirical Study



Train a network once, until it converges to critical point.

$$\text{Minimum ratio} = \frac{\text{Number of Positive Eigenvalues}}{\text{Number of Eigenvalues}}$$



the value of a network parameter w

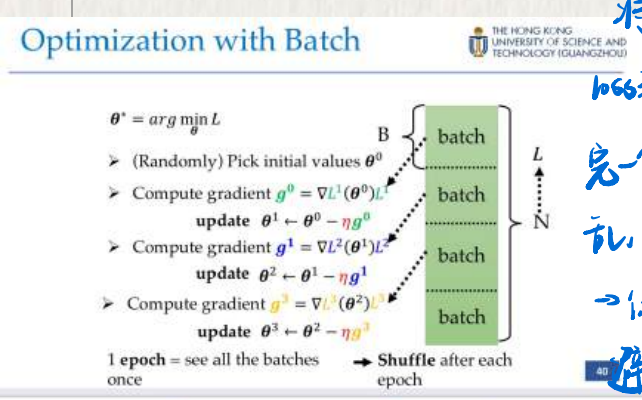
Batch

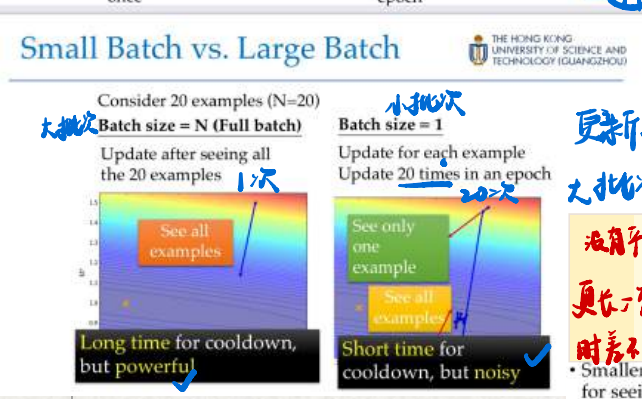
对抗临界点方法 ⊖

epoch batch

将一笔大型资料分若干批次计算 loss和梯度，从而更新参数，每看完一个epoch就把这笔大型资料打乱，然后重新分批次。

→保证每个epoch中batch资料间具有随机性.

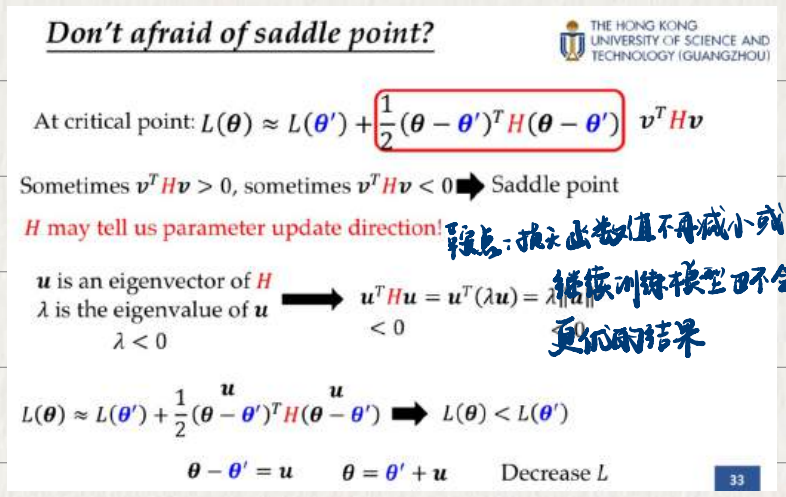


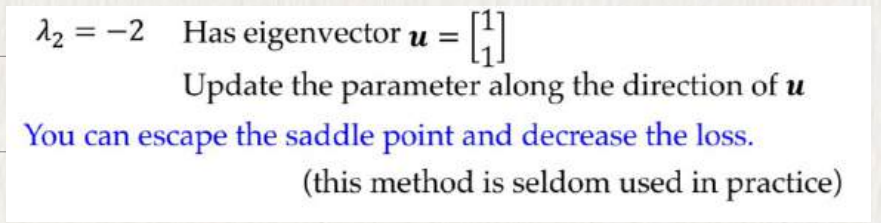


更新速度差不多，小批次系统时长，大批次耗时短。

没有平行计算时，单次更新大批次耗时更长了有平行运算时，单次更新大小批次耗时差不多，但1个epoch中大批次耗时更短。

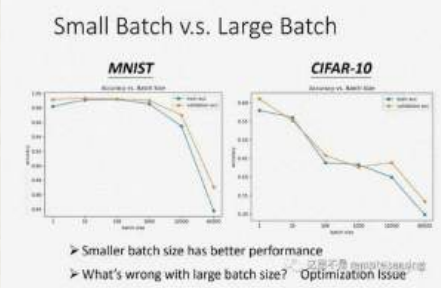
鞍点：指之此类值不再减小或增大，继续训练模型也不会得到更优的结果





大小批次的性能对比

小批次有更好的性能，由图可知同一个模型，同一个网络，training误差随着batch size 的增大而增大，testing 的误差也是。如果是 model bias 的问题，那么在 size 小的时候也会表现差，而不会等到 size 变大才差。所以这是Optimization issue（优化问题）导致大批次性能差。

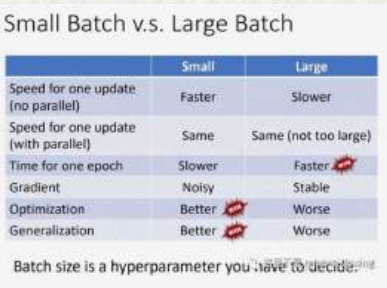


小批次训练时性能好

每次更新的时候，用的 loss 函数会有差异（如右图），因为不同的 batch 用不同的 loss function。换了个 loss 函数就更不容易卡住！！！

总结

小批次在 1 个 epoch 中的速度很慢，耗时很长，但是小批次在优化过程性能更好，在测试时的性能也更好。而批次大小是一个超参数，需要自行设定。



	Small	Large
Speed for one update (no parallel)	Faster	Slower
Speed for one update (with parallel)	Same	Same (not too large)
Time for one epoch	Slower	Faster
Gradient	Noisy	Stable
Optimization	Better	Worse
Generalization	Better	Worse

①BN: $\sigma_i^{(k)} = \sqrt{Var[x_i^{(k)}]}$, $\beta_i^{(k)} = E[x_i^{(k)}]$, 保证整个network的capacity, 还原原特征

②How: chain rule ③Where: 在非线性函数前, 对 $x = Wu + b$ 做规范化

④Why: 克服深度神经网络难以训练: 但均值通常不代表分布

那BN到底是什么原理呢？说到底还是为了防止"梯度弥散"。关于梯度弥散，大家都知道一个简单的栗子：$0.9^{30} \approx 0.04$。在BN中，是通过将activation规范为均值和方差一致的手段使得原本会减小的activation的scale变大。可以说是一种更有效的local response normalization方法（见

⑤ 例如，在神经网络训练中时遇到收敛速度很慢，或梯度爆炸等无法训练的状况时可以尝试BN来解决。另外，在一般使用情况下也可以加入BN来加快训练速度，提高模型精度。

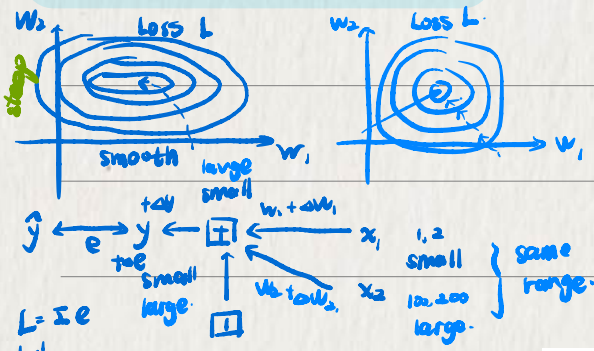
(BN) Batch Normalization

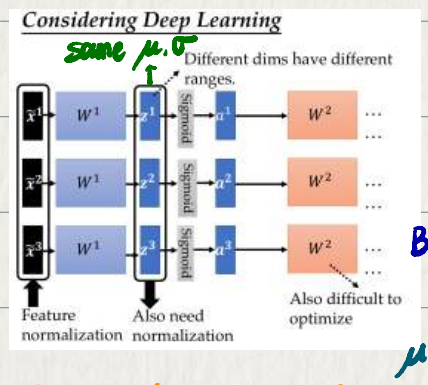
Feature Normalization

$x_i^r \leftarrow \frac{x_i^r - m_i}{\sigma_i} \sim N(0,1)$

↑ Changing

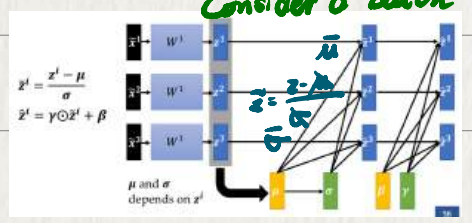
In general, feature normalization makes Landscape gradient descent converge faster.





Batch Normalization

Consider a batch



Batch normalization - Testing

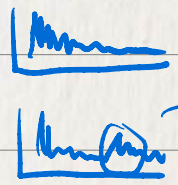
Computing the moving average of μ and σ of the batches during training.

$\bar{\mu} \leftarrow p\bar{\mu} + (1-p)\mu^t$

Others: Layer/ Instance/ Group/ Weight/Spectrum Normalization

Momentum ← 对抗Critical point方法③

Training stuck ≠ Small Gradient



people believe training stuck because the parameters are around a critical point...

Adaptive Learning Rate (学习率: 步长越大=学习率越高)

Different parameters needs different learning rate

Formulation for one parameter: $\theta_i^{t+1} \leftarrow \theta_i^t - \eta g_i^t$

$g_i^t = \frac{\partial L}{\partial \theta_i}\Big|_{\theta=\theta^t}$, $\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t$

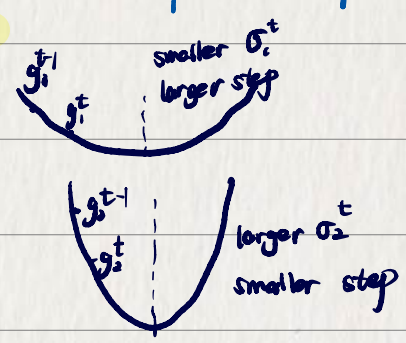
parameter dependent

Root Mean Square

$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t$

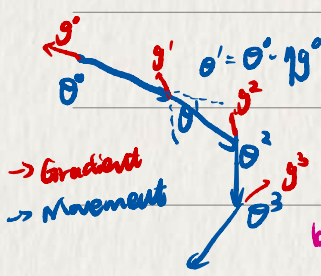
$\sigma_i^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t}(g_i^t)^2}$

Used in Adagrad



Learning rate adapts dynamically

1. (Vanilla) Gradient Descent



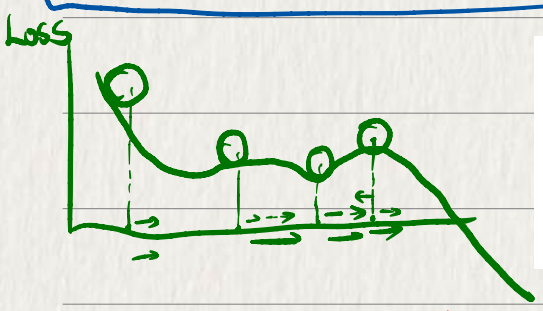
Movement: movement of last step minus gradient at present

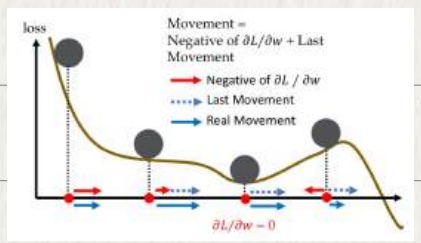
Movement not just based on gradient, but previous movement.

$\theta^0 - m^0 = 0 - g^0 \quad -m^1 = \lambda m^0 - \eta g^0 \rightarrow \theta^1 = \theta^0 + m^1 \quad -g^1 - m^2 = \lambda m^1 - \eta g^1 \rightarrow \theta^2 = \theta^1 + m^2$

movement

movement = Negative of $\partial L/\partial w$ + Last Movement





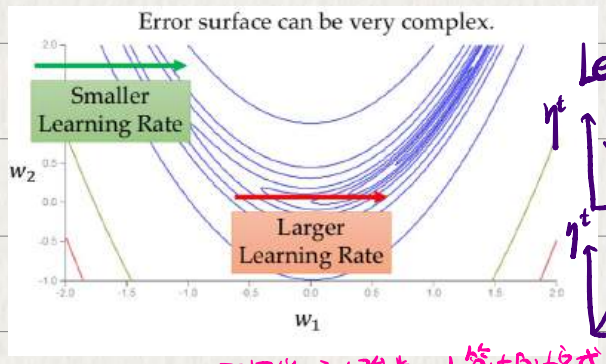
Summary

① Critical points have zero gradients.

② Critical points can be either saddle points or local minima. f can be determined by the Hessian matrix.

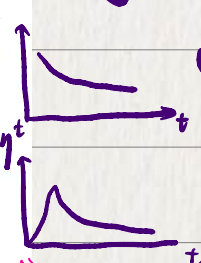
- It is possible to escape saddle points along the direction of eigenvectors of the Hessian matrix.
- Local minima may be rare.

③ Smaller batch size and momentum help escape critical points ✓



Learning Rate Scheduling

$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta^t}{\sigma_i^t} g_i^t$

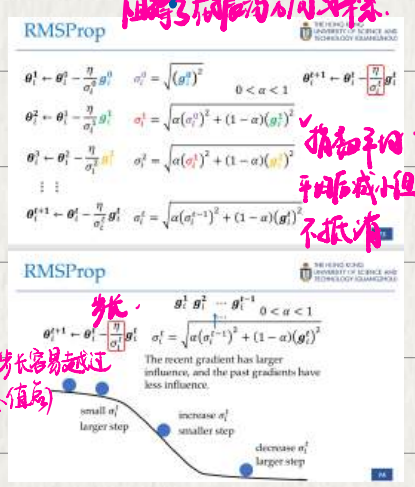


① 不需手动调整学习率超参数, 由算法自动完成

② 为每个参数选择不同学习率

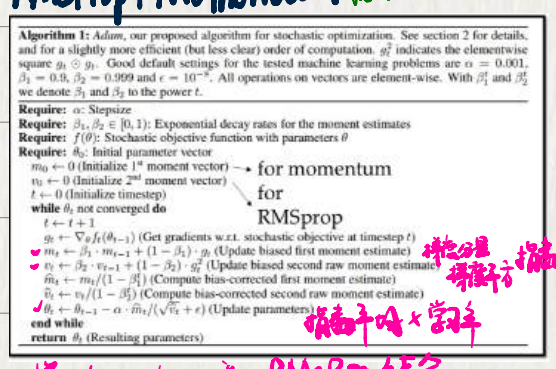
Learning Rate Decay: As the training goes, we are closer to the destination, so we reduce the learning rate.

RMSProp



(大步容易越过极小值点)

RMSProp+Momentum



将 Momentum 和 RMSprop 结合.

Summary of Optimization

(Vanilla) Gradient Descent: $\theta_i^{t+1} \leftarrow \theta_i^t - \eta g_i^t$

Various Improvements

$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta^t}{\sigma_i^t} m_i^t$

- η^t → Learning rate scheduling
- m_i^t → Momentum: weighted sum of the previous gradients
- σ_i^t → root mean square of the gradients (only magnitude)

Consider direction

Notes / 25 50 75 100

Title

Chapter Date

Review / / / / /

Rating ○ ○ ○ ○ ○

Keywords	Notes

Keywords	Notes

Notes / 25 50 75 100

Title

Chapter Date

Review / / / / /

Rating ◯ ◯ ◯ ◯ ◯

Keywords	Notes

Summary

Keywords	Notes

Summary

Notes / 25 50 75 100

Title

Chapter Date

Review / / / / /

Rating ○ ○ ○ ○ ○

Keywords	Notes

Keywords	Notes

Notes / 25 50 75 100

Title

Chapter Date

Review / / / / /

Rating

Keywords	Notes

Summary

Keywords	Notes

Summary

Notes / 25 50 75 100

Title

Chapter Date

Review / / / / /

Rating

No.	Word	Meaning	

No.	Word	Meaning	

Notes / 25 50 75 100

Title

Chapter Date

Review / / / / /

Rating ○ ○ ○ ○ ○

Notes / 25 50 75 100

Chapter 01

Keywords / 关键词

Important and difficult points / 重点难点

Date / 复习日期	**Total Time** / 复习时长	**Efficiency** / 学习效率	**Note** / 备注
		优秀 ○ 良好 ○ 一般 ○ 差 ○	
		优秀 ○ 良好 ○ 一般 ○ 差 ○	
		优秀 ○ 良好 ○ 一般 ○ 差 ○	
		优秀 ○ 良好 ○ 一般 ○ 差 ○	
		优秀 ○ 良好 ○ 一般 ○ 差 ○	
		优秀 ○ 良好 ○ 一般 ○ 差 ○	
		优秀 ○ 良好 ○ 一般 ○ 差 ○	
		优秀 ○ 良好 ○ 一般 ○ 差 ○	