Lecture 2-1

频率 interpretation: prob are long term relative frequencies

Bayesin interpretation: logically consistent degrees of belief 信念. 主观层面

$L(H|E)=P(E|H)$ $L(M|D)=P(D|M)$ ← model explains data怎么样

$P(H|E)=\frac{P(H)P(E|H)}{P(E)} \propto P(H)L(H|E)$

↑假设有多个假设 $P(H_i|E)$ 分母都是同一个 $P(E)$ 可以忽略.

1. MLE. 假设θ固定. 完全依赖 Data 来找最能解释的参数. $m_h+m_t=m$

$D=\{D_1,\dots D_m\}$ $L(\theta|D)=\prod_{i=1}^{m}\theta^{m_h}(1-\theta)^{m_t}$ log-下 $m_h\log\theta+m_t\log(1-\theta)$

对θ求导 $\frac{m_h}{\theta}-\frac{m_t}{1-\theta}=0 \Rightarrow \hat{\theta}_{MLE}=\frac{m_h}{m_t+m_h}=\frac{m_h}{m}$ $L(\theta|D)=\theta^{m_h}(1-\theta)^{m_t}$.

1.1 sufficient statistics: function 整合所有信息. 如 ~~s(D)=L~~

(m_h, m_t) 是 sufficient statistic

MLE无法提供置信度. 10(3:7) 认为随机波动. 因为有强烈生活中bias. 因该 coin 是 unbiased 的.

↓

Bayesin: 小数据时先验重要. 大数据时数据压倒一切.

{ MLE: $\theta^*=\arg\max P(D|\theta)$ 点估计. X可信程度 X uncertainty

Bayesin: 把θ当成随机变量 得到的是 $p(\theta|D)$ 有 prior $p(\theta)$

Bayesin: $P(D_{m+1}=H|D)=\int P(D_{m+1}=H,\theta|D)d\theta=\int P(D_{m+1}=H|\theta,D)p(\theta|D)d\theta$

↑在知道θ后, 与D无关.

Bayesian MAP: $P(D_{m+1}=H|D)=\theta^*=\arg\max_\theta p(\theta|D)$ 退化成 $=\theta\int\theta p(\theta|D)d\theta$

↑简化成求后验最大的

posterior distribution $p(\theta|D)\propto p(\theta)L(\theta|D)=\theta^{m_h}(1-\theta)^{m_t}p(\theta)$

有 Beta分布做 $p(\theta)$ 先验 $p(\theta)=\frac{\Gamma(\alpha_t+\alpha_h)}{\Gamma(\alpha_t)\Gamma(\alpha_h)}\theta^{\alpha_h-1}(1-\theta)^{\alpha_t-1}$

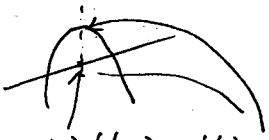
$\Gamma(\alpha)=(\alpha-1)!$ 这里的 α_h 和 α_t 是虚构(prior)认为在 $\alpha_h+\alpha_t$ 次 game 中有 α_h 次 head α_t 次 tail. α. α越大, 对prior越自信

提变成 $\theta^{m_h+\alpha_h-1}(1-\theta)^{m_t+\alpha_t-1}$

$P(D_{m+1}=H|D)=\int_0\theta p(\theta|D)d\theta=c\int\frac{m_h+\alpha_h}{m+\alpha}$

↑还是Beta. Beta期望 $E[\theta]=\frac{m_h+\alpha_h}{m+\alpha}$

Lecture 2-2



1. Jensen's Inequality

若 f 是定义在 I 上凹函数，$\forall x,y\in I$，$\lambda\in[0,1]$ 满足 $\lambda f(x)+(1-\lambda)f(y)\le f(\lambda x+(1-\lambda)y)$

对于概率分布 $p_i\in[0,1]$，$\sum_{i=1}^{n} p_i=1$ $x_i\in I$.

$\sum_{i=1}^{n} p_i\{f(x_i)\}\le f(\sum_{i=1}^{n} p_i x_i)$ 若 f 强凹

证明：显然 $n=2$ 时成立. 那么扩展到 n. 有. $\sum_{i=1}^{n} p_i f(x_i)\le f(\sum_{i=1}^{n} p_i x_i)$

$\sum_{i=1}^{n+1} p_i f(x_i)=\sum_{i=1}^{n} p_i f(x_i)+p_{n+1} f(x_{n+1})$ 设 $\sum_{i=1}^{n} p_i=\lambda\in[0,1]$.

~~$\le f(\sum_{i=1}^{n} p_i x_i)$~~ 变成 $1\times\frac{1}{\lambda}\cdot\lambda$.

$=\lambda\sum_{i=1}^{n}\frac{1}{\lambda}p_i f(x_i)+p_{n+1}f(x_{n+1})$

$\le\lambda f(\sum_{i=1}^{n}\frac{p_i}{\lambda}f(x_i))+p_{n+1}f(x_{n+1})$ $\Rightarrow\lambda+p_{n+1}=1$.

$\le f(\lambda\sum_{i=1}^{n}\frac{p_i}{\lambda}x_i+(1-\lambda)x_{n+1})=f(\sum_{i=1}^{n+1}p_i x_i)$

特例. 对数函数 $\log(x)$ 在 $(0,+\infty)$ 严格凸凹 $\sum_{i=1}^{n} p_i\log(x_i)\le\log(\sum_{i=1}^{n} p_i x_i)$

即 $E[\log(X)]\le\log E[X]$

2. Entropy $H(X)=-\sum_{x\in X}P(x)\log P(x)=-E[\log P(x)]$ 约定 $0\log 0=0$

性质：1. $H(X)\ge 0$

2. $H(X)=0$ iff $\exists x\in X$ $P(X=x)=1$

3. $H(x)\le\log|X|$ ← 取值空间基数 iff 均匀分布. 用 Jensen.

$H(X)=\sum_{x}P(x)\log\frac{1}{P(x)}\le\log(\sum_{x}P(x)\frac{1}{P(x)})=\log|X|$

这里 f 是 $\log$ x_i 是 $\frac{1}{P(x)}$

条件熵 $H(Y|X=x)=\sum_{y}P(y|x)\log\frac{1}{P(y|x)}$ 这可自

其是就是把 $P(X)$ 换成 $P(Y|X=x)$

$H(Y|X=x)$ 可能大于 $H(Y)$ 因为 $X=x$ 既可能带来确定性.

也可能带来不确定性.

$H(Y|X)=\sum_{x}P(x)H(Y|X=x)$ 平均条件熵：已知 X 后，Y 剩余的平均不确定性.

$H(X,Y)=-\sum_{x,y}P(x,y)\log(P(y)P(x|y))=H(Y)+H(X|Y)$

~~$H(Y)+H(X|Y)=-\sum_{y}\log P(Y)+\sum_{y}P(y)H(X|Y=y)$~~ $P(Y)$ X

联合熵

$H(X,Y)=-\sum_{x,y}P(x,y)\log P(x,y)$ X

~~$=-\sum_{xy}P(y)\log P(y)+\sum_{xy}P(y)\sum P(x|Y=y)\log\frac{1}{P(x|Y=y)}$~~

3. Divergence 衡量分布差异

1. $KL(P\|Q)=\sum_x P(x)\log\frac{P(x)}{q(x)}$ ；若 $P(x)>0$ 且 $Q(x)=0$，则 $KL=\infty$

$KL(P\|Q)\geq 0$.　　$\log x$ 有负号，所以大于等于.

proof $\sum_x P(x)\log\frac{P(x)}{q(x)}=-\sum_x P(x)\log\frac{Q(x)}{P(x)}\geq -\log\left(\sum_x P(x)\frac{Q(x)}{P(x)}\right)$

Cross entropy $H(P,Q)=-\sum_x P(x)\log\frac{1}{Q(x)}=-E[\log Q(X)]$　　$=-\log 1=0$

分布

$KL(P\|Q)=E[\log P(X)]-E[\log Q(X)]=H(P,Q)-H(P)$

推理：$H(P,Q)\geq H(P)$　因为 $KL(P\|Q)\geq 0$.

即 $\sum_x P(x)\log Q(x)\leq\sum_x P(x)\log P(x)$　负号.

令 $f(x)$ 为非负函数，有 $\sum_x f(x)\log Q(x)\leq\sum_x f(x)\log P(x)$

$P^*(x)=f(x)/\sum_x f(x)$　　↑ 可以归一化回去

2. JS Divergence. $JS(P\|Q)=\frac{1}{2}KL(P\|M)+\frac{1}{2}KL(Q\|M)$

其中 $M=\frac{P+Q}{2}$　　对称，有界 $[0,\log 2]$　$JS(P\|Q)=0\Leftrightarrow P=Q$

$JS(P\|Q)=\log 2$ 当 P和Q完全不相交.

↑ 算 $p(x)$ 时，$m(x)$ 退化为 $\pm p(x)$ 整个退化为 $\sum_x p(x)\log 2=\log 2$.

非 unsupervised-learning：最小化 $KL(P\|Q)=H(P,Q)-H(P)$

↑ 固定.

$H(P,Q)=-\int p(x)\log Q(x)dx\approx-\frac{1}{N}\sum_{i=1}^{N}\log Q(x_i)=-\frac{1}{N}\log Q(D)$

↑ 最大化 $Q(D)$

supervised-learning. $P(x,y)\xrightarrow{\text{sample}}D\xrightarrow{\text{学}}Q(y|x)$

希望缩小误差，即 $H(P,Q)$ 变成最大化 $\sum_{i=1}^{N}\log Q(y_i|x_i)$

4. Mutual information

$I(X;Y)=H(X)-H(X|Y)$　在得知Y后对X减少的不确定度

$=\sum_x p(x)\log\frac{1}{p(x)}-\sum_{x,y}p(x,y)\log\frac{1}{p(x|y)}=\sum_{x,y}p(x,y)\frac{p(x|y)}{p(x)}=\sum_{xy}p(x,y)\frac{p(x)}{p(x)}$

↑ $p(x)=\sum_y p(x,y)$

$=KL(P(X,Y)\|P(X)P(Y))$

对称

$=H(Y)-H(Y|X)$　从KL性质来

$\Rightarrow I(X,Y)\geq 0$　$=0$ iff $X\perp Y$　　$H(X|Y)\leq H(X)$ iff $X\perp Y$

Lecture 2-3

X, Y, $c: X \to Y$: target concept to learn C: concept class

D: target distribution S: training samples H: set of hypotheses (↙ hypothesis, 学到的)

1. 泛化误差 $R(h) = \Pr_{x\sim D}[h(x) \neq c(x)] = \mathbb{E}_{x\sim D}[I_{h(x)\neq c(x)}]$

   经验误差 $\hat{R}_S(h) = \frac{1}{m}\sum_{i}^{m} I_{h(x)\neq c(x)}$

   $\mathbb{E}_{S\sim D^m}[\hat{R}_S(h)] = R(h)$ 经验期望为泛化

2. Probably Approximately Correct Learning (PAC)

   如果说 concept class C 是 PAC-learnable，那么 ∃ 算法 A 和 $poly(\cdot,\cdot,\dots)$

   s.t. $\forall \varepsilon, \delta > 0$, $\forall D$ on X, $\forall c \in C$. 都有

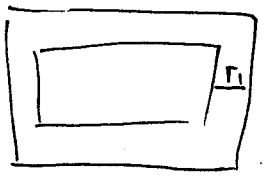
   $m \geq poly(1/\varepsilon, 1/\delta, n, size(c))$:

   $\Pr_{S\sim D^m}[R(h) \leq \varepsilon] \geq 1-\delta$ 等价于 $\Pr[R(h) > \varepsilon] \leq \delta$

   当算法 A 存在，则称为一个 PAC learning algo for C

   最多错 ε 的概率大于 $1-\delta$

   2.1 若 $H = C$ 即假设空间包含 target class



   希望 $\Pr(r_i) \leq \frac{\varepsilon}{4}$ 这样 $\Pr[\cup_i r_i] \leq \sum_i \Pr[r_i]$ 放大了，有重叠.

   Bad case 有些超过 $\frac{\varepsilon}{4}$

   $\Pr[R(R) > \varepsilon] = \Pr[\cup_{i=1}^{4} \Pr[r_i] > \varepsilon/4] \leq 4(1-\varepsilon/4)^m \leq 4e^{-m\varepsilon/4}$

   某个大 strip 一个样本没进去的 prob 小于 $1-\frac{\varepsilon}{4}$ ↑ 4个strip $(1-x \leq e^{-x})$

   2.2 $H \neq C$ $R(h_S) = 0$. consistent: 在训练数据上能做到 $\hat{R}_S(h) = 0$ 即 $\hat{R}_M = 0$

   $\Pr_{S\sim D^m}[R(h) \overset{\leq \varepsilon}{\geq} 1-\delta]$ holds if $m \geq \frac{1}{\varepsilon}(\log|H| + \log\frac{1}{\delta})$

   ↙ hypothesis 数

   $R(h) \leq \frac{1}{m}(\log|H| + \log\frac{1}{\delta})$

   Proof: $\Pr[\exists h \in H: \hat{R}_S(h) = 0 \wedge R(h) > \varepsilon]$

   $\leq \sum_{h\in H} \Pr[\hat{R}_S(h) = 0 \wedge R(h) > \varepsilon]$

   $= \sum_{h\in H} \Pr[\hat{R}_S(h) = 0 \mid R(h) > \varepsilon] \Pr[R(h) > \varepsilon]$

   单个样本不犯错 $\leq 1-\varepsilon$.

   放成 1

   $\leq |H|(1-\varepsilon)^m \leq |H|e^{-m\varepsilon} = \delta \iff m \geq \frac{1}{\varepsilon}(\log|H| + \log\frac{1}{\delta}) \iff R(h) \leq \frac{1}{m}(\log|H| + \log\frac{1}{\delta})$

   bound $O(\frac{1}{m})$

   只要 hypothesis set finite. ERM 就是 PAC 算法

2.3. Inconsistent：常见. $H \neq C$ 且 $R(h_S) \neq 0$

Hoeffding's inequality: X_i 独立. $\in [a_i, b_i]$ $\varepsilon > 0$ $\Rightarrow \Pr(\sum X_i - E[\sum X_i] \geq \varepsilon) \leq e^{-2\varepsilon^2 / \sum (b_i - a_i)^2}$

$\Pr(\sum X_i - E[\sum X_i] \leq -\varepsilon) \leq e^{-2\varepsilon^2 / \sum (b_i - a_i)^2}$

显然 $E[\sum X_i]$ 就是 $R(h)$. 前者是 $\hat{R}(h)$ $\forall$ hypothesis $X \to \{0, 1\}$ 代入 $a=0$ $b=1$

$\Rightarrow \Pr[|\hat{R}(h) - R(h)| \geq \varepsilon] \leq 2e^{-2m\varepsilon^2} \Leftrightarrow 2e^{-2m\varepsilon^2} \leq \delta \Leftrightarrow |\hat{R}_S(h) - R(h)| \leq \sqrt{\frac{\log \frac{2}{\delta}}{2m}}$

$\forall h \in H$, $|\hat{R}_S(h) - R(h)| \leq \sqrt{\frac{\log |H| + \log \frac{2}{\delta}}{2m}}$

$\forall h \in H$, $R(h) \leq \hat{R}_S(h) + O\left(\sqrt{\frac{\log(|H|)}{m}}\right)$ error bound in $O(\frac{1}{\sqrt{m}})$

3. Infinite Hypothesis Set (之前 $\log |H| \to \infty$, 失效)

引入 a. growth function b. VC dimension 用来 compute grow function

$\Pi_H : N \to N$ for a hypothesis H

$\forall m \in N$, $\Pi_H(m) = \max_{\{x_1, \dots, x_m\} \subseteq X} |\{(h(x_1), \dots, h(x_m))\}, h \in H\}|$.

最多的用在 H中的 hypothesis h 把 m个点分成不同方式的数

若 H为二分类. 则 $\Pr[|R(h) - \hat{R}(h)| \geq \varepsilon] \leq 4 \Pi_H(2m) e^{-\frac{m\varepsilon^2}{8}}$

即 $R(h) \leq \hat{R}(h) + \sqrt{\frac{2 \log \Pi_H(m)}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$

复杂度 置信度.

$A \subseteq B$ 那么 $d_{VC}(A) \leq d_{VC}(B)$

3.1 VC Dimension: ① Dichotomy：一种label点的方式 ② shattering 能够被打散说明H能实现任意 Dichotomy

$VC\dim(H) = \max\{m : \Pi_H(m) = 2^m\}$ ↑最大能被H打散的点集大小 $\Pi_H(m) = 2^m$

$VC\dim(R^d$的超平面$) = d+1$

$VC\dim(\sin) = +\infty$

3.2 Sauer's lemma：令 H with $VC\dim(H) = d$, $\forall m \in N$ 这里 $m < d$

$$\Pi_H(m) \leq \underbrace{\sum_{i=0}^{d} \binom{m}{i}}_{①} \leq \sum_{i=0}^{d} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \quad \frac{m}{d} \text{ 大于1}$$

$$= \binom{m}{d} \sum_{i=0}^{d} \binom{m}{i} \left(\frac{d}{m}\right)^{i} \leq \binom{m}{d} \sum_{i=0}^{dm} \binom{m}{i} \left(\frac{d}{m}\right)^{i} = \left(\frac{m}{d}\right)^{d} \left(1 + \frac{d}{m}\right)^{m} \leq \left(\frac{em}{d}\right)^{d}$$

要么 $VC\dim(H) = d < +\infty$ and $\Pi_H(m) = O(m^d)$

or $VC\dim(H) = +\infty$ and $\Pi_H(m) = 2^m$

3.3 令H为 $\{-1, +1\}$ $VC\dim(H) = d$, $\forall \delta > 0$. 至少有 $1-\delta$ 的概率 $|R(h) - \hat{R}(h)| \leq \sqrt{\frac{8d \log \frac{2em}{d} + 8 \log \frac{4}{\delta}}{m}}$.

3.4 For $|R(h) - \hat{R}(h)| \leq \varepsilon$ to hold for all $h \in H$ with p at least $1-\delta$. 满足 $m = O(d)$

模型越复杂，数据量要求越大

DL Lecture 3

1. 从传统 → DL

传统 supervised $\{x_i, y_i\}_{i=1}^N \to P(y|x)$

DL: $\{x_i, y_i\}_{i=1}^N \to h = f(x), P(y|h) \Rightarrow P(y|x,\theta)$

多层线性复合仍线性

2. Weight Initialization (避免前向信号, GD BP层间爆/消失)

Xavier $w \sim U[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$ n输入维度: 每层输入方差大致稳定

He $w \sim N[0, \frac{2}{\sqrt{n}}]$ 或 $w \sim N(0, \frac{2}{n})$ 通常用于ReLU. (负值截断, 需更大方差)

↑ 标准差

3. Activation

3.1 sigmoid $g(z) = \sigma(z) = \frac{1}{1+\exp(-z)}$ $\sigma'(z) = \sigma(z)(1-\sigma(z))$

$z \gg 0$ or $z \ll 0$ $\sigma(z) \to 0/1$ 导数接近0, vanishing gradient

不推荐作为 hidden 的 act. 适合二分类输出层

3.2 ReLU $g(z) = \max\{0, z\}$ $g'(z) = \begin{cases} 1, & z>0 \\ 0, & z \le 0 \end{cases}$

pros: $z>0$, $\nabla = 1$ 快训 cons: 若某 neuron 对所有样本都 $z<0$, $\nabla = 0$ 无法更新

缓解: 把 bias 初始化为小的正数 dead neuron

3.2.1 ReLU variant: 负半边留 ∇

$g(z, \alpha) = \max\{0, z\} + \alpha \min\{0, z\}$ $\begin{cases} \alpha = -1 & \text{absolute value rectification} \\ \alpha = 0.01 & \text{Leaky ReLU} \\ \alpha \text{可学} & \text{Parametric ReLU} \end{cases}$

3.3 Hyperbolic Tangent (Tanh) 零中心, 但仍会 saturate

$g(z) = \tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \in [-1, 1]$

相比 sigmoid, gradient 通常更大.

但当 $|z|$ 很大, 仍饱和

3.4 Swish, Mish, GeLU

$\begin{cases} \text{Swish: } g(z) = z\sigma(\beta z) & \beta \text{固定或可学} \\ \text{Mish: } g(z) = z\tanh(\log(1+\exp(z))) = z\tanh(\zeta(z)) & \text{softplus } \zeta(z) = \log(1+\exp(z)) \\ \text{GeLU: } g(z) = z\Phi(z) & \Phi(z) \text{是标准高斯的 CDF} \end{cases}$

光滑, 平滑, 负半非0 ∇, 非单调. 上方无界, 一般比 ReLU 稳

3.5 Softplus $g(z) = \zeta(z) = \log(1+\exp(z))$

4. 输出层概率模型 logit $z = W^T h + b$

4.1 Linear Gaussian 当y是实值 $p(y|x) = N(y|z, I)$ $L(x, y, \theta) = -\log P(y|x, \theta) \propto \frac{1}{2}\|y - z\|_2^2$

$\frac{\partial L}{\partial z_k} = z_k - y_k$

4.2 Sigmoid输出 当 $y \in \{0,1\}$ 二分类

$P(y|x) = \text{Bernoulli}(y|\sigma(z)) = \sigma(z)^y (1-\sigma(z))^{1-y}$

$L(x,y,\theta) = -[y \log \sigma(z) + (1-y)\log(1-\sigma(z))]$

Saturation 只有在 model 正确且非常确信时有有

$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial \sigma(z)} \frac{\partial \sigma(z)}{\partial z} = \left(-\frac{y}{\sigma(z)} + \frac{1-y}{1-\sigma(z)}\right)\frac{\partial \sigma(z)}{\partial z}$

$= \left(-\frac{y}{\sigma(z)} + \frac{1-y}{1-\sigma(z)}\right)\sigma(z)(1-\sigma(z))$

$= -y(1-\sigma(z)) + (1-y)\sigma(z)$

$= \sigma(z) - y$

4.3 Softmax 输出 $\{1,2,\cdots,C\}$ 多分类

$P(y=k|x) = \frac{\exp(z_k)}{\sum_{i}^{C}\exp(z_i)}$ k是正确标签. 对正确 k

令 $S = \sum_{i}^{C}\exp(z_i)$ 随机变量

$L = -\log P(y=k|x) = -z_k + \log S$

$\frac{\partial L}{\partial z_k} = -1 + \frac{\partial \log S}{\partial S}\frac{\partial S}{\partial z_k}$

$= -1 + \frac{1}{S}\cdot\exp(z^k)$

$= -1 + \frac{1}{S}\exp(z^k)$

$\Rightarrow \frac{\partial L}{\partial z_k} = P(y=k) - I(y=k)$ 但 $k \neq y_{true}$

$\frac{\partial L}{\partial z_m} = \frac{1}{S}\exp(z_m)$ $P(y=k)$

$= P(y=m)$

或 $\frac{\partial L}{\partial z_k} = \hat{y_k} - I(y=k)$ $\hat{y_k} = P(y=k)$

↑ y_{true}

5. $J(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(x_i,y_i,\theta) = \frac{1}{N}\sum_{i=1}^{N} -\log P(y_i|x_i,\theta)$

SGD $\theta \leftarrow \theta - \alpha \frac{1}{|B|}\sum_{i\in B}\nabla L(x_i,y_i,\theta)$

6. BP

$z_k = \sum_j u_j W_{kj}$, $u_j = g(z_j)$ $z_j = \sum_i u_i W_{ji}$

↑ 总和k行有关

$\frac{\partial L}{\partial W_{kj}}$ GR $= \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial W_{kj}} = u_j \delta_k$ $\delta_k = \frac{\partial L}{\partial z_k}$

最后一层没激活

↑ 最近层 ↘ 只出来 z_k u_j

$\frac{\partial L}{\partial W_{ji}} = \sum_k \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial W_{ji}} = \sum_k \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial u_j}\frac{\partial u_j}{\partial z_j}\frac{\partial z_j}{\partial W_{ji}}$ ↓ u_i

隐藏层

W_{ji} 只会出来的 $z_j \to u_j \to z_k$ 一部分.

$= \sum_k \delta_k W_{kj} g'(z_j) u_i = u_i\left(g'(z_j)\sum_k \delta_k W_{kj}\right)$

定义 $\delta_j = g'(z_j)\sum_k \delta_k W_{kj}$

$\frac{\partial L}{\partial W_{ji}} = u_i \delta_j$ $\frac{\partial L}{\partial b_j} = \delta_j$

↑ 输入激活 ↑ 目标单元误差信号. 从L到输出的信号

W_{ji} 只和 u_i 相乘输出 z_j

7. Optimization

7.1 SGD 只看当前 ∇ noise大. 路径震荡. 狭长valley 收敛慢

7.2 Momentum SGD

$g \leftarrow \frac{1}{m} \nabla_\theta \sum_i L(f(x^{(i)};\theta), y^{(i)})$

$v \leftarrow \alpha v - \varepsilon g$ velocity v

$\theta \leftarrow \theta + v$

exponentially decaying the average of past gradients $\alpha = 0.5, 0.9, 0.99$

7.3 AdaGrad

$r \leftarrow r + g \circ g$

$\Delta\theta \leftarrow -\frac{\varepsilon}{\delta + \sqrt{r}} \circ g$

$\theta \leftarrow \theta + \Delta\theta$

对稀疏特征友好. 历史习小的参数 lr 大

cons: r单增. lr↓ 后期学不动

7.4 RMSProp : 遗忘太久远历史

$r \leftarrow \rho r + (1-\rho) g \circ g$

$\Delta\theta \leftarrow -\frac{\varepsilon}{\sqrt{\delta + r}} \circ g$

$\theta \leftarrow \theta + \Delta\theta$

指数滑动平均. 不会无限累积

有 adaptive scaling 但无显式 momentum

noise仍大

7.5 Adam : Momentum + RMSProp

$s \leftarrow \rho_1 s + (1-\rho_1) g \circ g$

$r \leftarrow \rho_2 r + (1-\rho_2) g \circ g$ ↓ 类似RMSProp $\varepsilon = 0.001$ $\rho_1 = 0.9$ $\rho_2 = 0.999$ $\delta = 10^{-8}$

类似动量 $\hat{s} = \frac{s}{1-\rho_1^t}$ $\hat{r} = \frac{r}{1-\rho_2^t}$

$\Delta\theta \leftarrow -\varepsilon \frac{\hat{s}}{\sqrt{\hat{r}} + \delta}$ $\theta \leftarrow \theta + \Delta\theta$

8. LR Scheduling

固定 / step decay / exponential decay / cosine annealing / warm restart

周期降低再个模拟重新开搜. 但有好base

9. Overfit Dropout

$\tilde{J}(\theta) = J(\theta) + \lambda \|\theta\|_2^2$

9.1 Bagging 训多个投票. 费劲

9.2 Dropout 廉价 Bagging. $m_j \sim \text{Bernoulli}(p)$ $\tilde{u}_j = m_j u_j$

大网中随机训小网. 防止神经元间产生过强 co-adaption

convex $O(\frac{1}{\sqrt{T}})$ strongly convex $O(\frac{1}{T})$

DL Week 4 CNN

1. MLP处理图时参数爆炸，丢失空间结构，缺乏平移泛化能力
2. CNN基本架构 INPUT → CONV → RELU → POOL → FC
3. Convolution

3.1 连续卷积 function f, g

$$(f*g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)\,d\tau = \int_{-\infty}^{+\infty} f(t-\tau)g(\tau)\,d\tau$$

$(f*g)(t)$：f 输入信号，g filter，t 输出位置

由 f 邻域多加权求和，权重来自 g

3.2 离散卷积 weighted smoothing

$$(f*g)(n) = \sum_{m=-\infty}^{+\infty} f[m]g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m]g[m]$$

4. 2D conv 和 Cross-Correlation

4.1 Real 2D conv

$$Z(i,j) = (I*k)(i,j) = \sum_m \sum_n I(m,n)k(i-m, j-n) = \sum_m \sum_n I[i-m, j-n]k[m,n]$$

I：Image，k：kernel

卷积核被翻转

4.2 Cross correlation

$$Z(i,j) = (I*k)(i,j) = \sum_m \sum_n I(i+m, j+n)k(m,n)$$

在 CNN 中，kernel 是可学习，是否翻转不影响表达能力

5. CNN 三大优势
   1. Sparse connectivity：一个神经元只连接 receptive field 里的 region（计算可行）
   2. Parameter Sharing：不同空间相同权重（是等变性前提）
   3. Equivariance：平移等变性 先移再卷 = 先卷再移

Invariance 是 $f(g(x)) = f(x)$　　$f(g(x)) = g(f(x))$

6. 卷积层公式（feature map 索从1开始，i 是 channel，i 是对应输出 channel 的）

$$Z_{j,k}^{(i)} = \sum_{l,m,n} I_{j+m-1, k+n-1}^{(l)} K_{m,n}^{(i,l)} + b^{(i)}$$

加入 Stride（也是从1开始，对齐位置）

$$Z_{j,k}^{(i)} = \sum_{l,m,n} I_{(j-1)s+m, (k-1)s+n}^{(l)} K_{m,n}^{(i,l)} + b^{(i)}$$

$$H_{out} = \left\lfloor \frac{H+2P-F_h}{s} \right\rfloor + 1$$

$$W_{out} = \left\lfloor \frac{W+2P-F_w}{s} \right\rfloor + 1$$

参数量 $C_{in} F_h F_w C_{out} + C_{out}$

一个 filter 为 $C_{in} F_h F_w + 1$

7. 反向传播 $J(k,b)$ 设上一层传回梯度张量 G

$$G_{j,k}^{(i)} = \frac{\partial J(k,b)}{\partial Z_{j,k}^{(i)}}$$

7.1 对 kernel 梯度 $\frac{\partial J(k,b)}{\partial k_{j,k}^{(i,l)}} = \sum_{m,n} \frac{\partial J}{\partial Z_{m,n}^{(i)}} \frac{\partial Z_{m,n}^{(i)}}{\partial k_{j,k}^{(i,l)}} = \sum_{m,n} G_{m,n}^{(i)} I_{(m-1)s+j,(n-1)s+k}^{(l)}$

对 input $\frac{\partial J(k,b)}{\partial I_{p,q}^{(l)}} = \sum_i \sum_{m,n} K_{p-(m-1)s,\, q-(n-1)s}^{(i,l)} G_{m,n}^{(i)}$

对 bias $\frac{\partial J(k,b)}{\partial b^{(i)}} = \sum_{m,n} G_{m,n}^{(i)}$

$$k_{j,k}^{(i,l)} = k_{j,k}^{(i,l)} - \alpha \frac{\partial}{\partial k_{j,k}^{(i,l)}} J(k,b) \qquad b^{(i)} = b^{(i)} - \alpha \frac{\partial}{\partial b^{(i)}} J(k,b)$$

8. Pooling
- Max Pooling
- Average Pooling 平滑
- L_2-norm pooling
- prob weighted pooling 强调能量

较为见

作用
① 对小平移不敏感
② 降低计算、显存要求
③ 改善统计效率：关注是否存在某特征，而非精确位置
④ 处理可变尺寸输入

8.1 反传 upsampling operation that ~~increases~~ inverses the subsampling in the forward pass

max pooling. 只取获得 max response 区域

如 $\begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix} \to 3$ 反梯为 $\begin{bmatrix} 0 & \delta \\ 0 & 0 \end{bmatrix}$

9. Data Norm ; Batch Norm

9.1 Why ?

9.1.1 特征尺度不同 ⇒ GD会对某些 weight 猛打五

9.1.2 梯度尺度不同 ⇒ loss contour 狭长，梯度下降 zig-zag 训练变慢

令 $X = [x_{ij}]_{N \times D}$ $\mu_j = \frac{1}{N} \sum_{i=1}^{N} x_{ij}$ $\sigma_j^2 = \frac{1}{N} \sum (x_{ij} - \mu_j)^2$ $\hat{x_{ij}} = \frac{x_{ij} - \mu_j}{\sigma_j}$

每个特征 均值为0，方差为1

9.2 Batch Normalization 前层参数改后，后层输入分布也改 ⇒ covariate shift.

不仅规范化原始输入，也规范化每一层输入/激活

$\hat{x} = \frac{x - \mu}{\sigma}$ $y = \gamma \hat{x} + \beta$ 测试时 μ, σ 来自 moving average

是否恢复某些均值和尺度

pros : ①加速训练 ②降低初始化敏感度 ③轻微正则化 ⑤稳定深层网络

Lec 5 RNN

1. Vanilla RNN

$a^{(t)}$

1.1 更新 $h^{(t)} = \tanh(b + Wh^{(t-1)} + Ux^{(t)})$

$o^{(t)} = c + Vh^{(t)}$

$\hat{y}^{(t)} = \text{softmax}(o^{(t)})$

$W, U, V, b, c, \theta_{em}$

1.2 Loss Input $\{x^{(1)}, x^{(2)}, \dots, x^{(T)}\}$ $\{y^{(1)}, \dots, y^{(T)}\}$

$= -\sum_{t=1}^{T} \log P(y^{(t)} | x_{:t})$ 对所有 timestamp 求和

1.3 BPTT $\nabla_W L, \nabla_U L, \nabla_V L, \nabla_b L, \nabla_c L, \nabla_{\theta_m} L$

$\hat{y}_i^{(t)} = \frac{\exp(o_i^{(t)})}{\sum_j \exp(o_j^{(t)})}$ $L^{(t)} = -\log \hat{y}^{(t)}_{y^{(t)}}$

time t 不同输出

$\frac{\partial L^{(t)}}{\partial o_i^{(t)}} = \hat{y}_i^{(t)} - 1_{i, y^{(t)}}$ (GT) ⇒ 预测 prob - 真实 one-hot 标签

$\because o^{(t)} = c + Vh^{(t)} \Rightarrow \nabla_c L = \sum_{t=1}^{T} \nabla_{o^{(t)}} L$ (所有时间步), $\nabla_V L = \sum_{t=1}^{T} (\nabla_{o^{(t)}} L)(h^{(t)})^T$

$h^{(t)}$ 到总 loss 可以通过 $\begin{cases} \to o^{(t)} \to L^{(t)} \\ \to h^{(t+1)} \cdots \to L^{(t+1)} \cdots \end{cases}$

$\nabla_{h^{(t)}} L = V^T \nabla_{o^{(t)}} L + W^T \delta^{(t+1)}$

定义 $\delta^{(t)} \equiv \nabla_{a^{(t)}} L$

$\frac{\partial h^{(t)}}{\partial a^{(t)}}$ 向 ⇒ matrix $= \text{diag}(1 - (h^{(t)})^2)$ ⇒ $\delta^{(t)} = \text{diag}(1 - (h^{(t)})^2) \nabla_{h^{(t)}} L$ (tanh 是 element wise 导致的向量 ↑, chain rule ↓)

chain rule 未来

⇒ $\delta^{(t)} = \text{diag}(1 - (h^{(t+1)})^2)[V^T \nabla_{o^{(t)}} L + W^T \delta^{(t+1)}]$ 且 $\delta^{(T+1)} = 0$ 边界条件 (超了 ↑)

$\nabla_b L = \sum_{t=1}^{T} \delta^{(t)}$ $\nabla_W L = \sum_{t=1}^{T} \delta^{(t)} (h^{(t-1)})^T$ $\nabla_U L = \sum_{t=1}^{T} \delta^{(t)} (x^{(t)})^T$

1.4 Teacher forcing：把 $y^{(t)}$ 作为输入

$\log p(y^{(1)}, y^{(2)} | x^{(1)}, x^{(2)}) = \log p(y^{(2)} | y^{(1)}, x^{(1)}, x^{(2)}) + \log p(y^{(1)} | x^{(1)}, x^{(2)})$

output recurrence

pros: ① 时间 τ 之间 完全 decouple ⇒ 可以并行

↓

lack hidden to hidden recurrence: less powerful.

solu: ~~rand~~ 随机选生成/实际 value 当 input

2. 梯度消失与爆炸

2.1 线性

$$h^{(t)} = Wh^{(t-1)} + Ux^{(t)} + b$$

$$\Rightarrow \frac{\partial h^{(t)}}{\partial h^{(k)}} = \prod_{j=k+1}^{t} \frac{\partial h^{(j)}}{\partial h^{(j-1)}} = \prod_{j=k+1}^{t} W^T = (W^T)^{t-k}$$

若 $W^T = V\,\mathrm{diag}(\lambda)\,V^{-1}$ 则 $= V\,\mathrm{diag}(\lambda)^{t-k}V^{-1}$

若 $|\lambda_i| > 1$ 则 $\to \infty$ 梯度爆炸

若 $|\lambda_i| < 1$ $|\lambda_i|^{t-k} \to 0$ 消失

非线性RNN

$$h^{(t)} = f(Wh^{(t-1)} + Ux^{(t)} + b)$$

$$\frac{\partial h^{(t)}}{\partial h^{(k)}} = \prod_{j=k+1}^{t} \frac{\partial h^{(j)}}{\partial h^{(j-1)}} = \prod_{j=k+1}^{t} W^T \mathrm{diag}(f'(h^{(j-1)}))$$

范数满足

$$\left\| \frac{\partial h^{(t)}}{\partial h^{(j-1)}} \right\| \le \|W^T\| \cdot \|\mathrm{diag}(f'(h^{(j-1)}))\|$$

非线激活作用 { ① bound recurrent dynamics 缓解 exploding gradient
② 若 f' 很小，如 sigmoid / tanh 饱和 ⇒ 更严重 gradient vanish

2.2 Solution

2.2.1 Gradient clipping 解决爆炸

huge difference

$$\hat{g} = \nabla_\theta \xi \qquad \hat{g} \leftarrow \begin{cases} \hat{g}, & \|\hat{g}\| < T \\ \frac{T}{\|\hat{g}\|}\hat{g}, & \|\hat{g}\| \ge T \end{cases}$$

限 gradient 大小 防止更新过大

2.2.2 Identity Init + ReLU 消失

初始化 W 为 $W \approx I$ + ReLU

huge difference.

3. LSTM

3.1 forget gate $f^{(t)} = \sigma(W^f h^{(t-1)} + U^f x^{(t)} + b^f)$ 对之前 $c^{(t-1)}$ 留多少

input gate $i^{(t)} = \sigma(W^i h^{(t-1)} + U^i x^{(t)} + b^i)$ 更新什么 value

output gate $o^{(t)} = \sigma(W^o h^{(t-1)} + U^o x^{(t)} + b^o)$

★ cell state $c^{(t)} = f^{(t)} \circ c^{(t-1)} + i^{(t)} \circ \tanh(W^c h^{(t-1)} + U^c x^{(t)} + b^c)$ candidate values 让 gradient 能沿 cell state 加法路径传播 缓解长程依赖中的梯度消失

final hidden state $h^{(t)} = o^{(t)} \circ \tanh(c^{(t)})$

3.2 GRU

Update gate: $z^{(t)} = \sigma(W^z h^{(t-1)} + U^z x^{(t)} + b^z)$ 留多少旧 hidden state

Reset gate: $r^{(t)} = \sigma(W^r h^{(t-1)} + U^r x^{(t)} + b^r)$ 生成候选记忆是否忽略过去

New Memory Content: $\tilde{h}^{(t)} = \tanh(W(r^{(t)} \circ h^{(t-1)}) + Ux^{(t)} + b$ 候选记忆

Final Memory: $h^{(t)} = z^{(t)} \circ h^{(t-1)} + (1 - z^{(t)}) \circ \tilde{h}^{(t)}$

GRU 参数比 LSTM 少，训练快，结构简单，但 LSTM 在大任务上性能更强且长程依赖↑

4. 双向 RNN

$$\overrightarrow{h_t} = f(\overrightarrow{W}x_t + \overrightarrow{V}\overrightarrow{h_{t-1}} + \overrightarrow{b}) \qquad \overleftarrow{h_t} = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h_{t-1}} + \overleftarrow{b})$$

拼

$$\hat{y} = g(Vh_t + c) = g(V[\overrightarrow{h_t}, \overleftarrow{h_t}] + c)$$ 分类，tagging，encoding

Deep Bidirectional RNN：将上者堆多层 $\overrightarrow{h_t^i} = f(\overrightarrow{W^i}\overrightarrow{h_t^{i-1}} + \overrightarrow{V^i}\overrightarrow{h_{t-1}^i} + \overrightarrow{b_t^i})$ $\overleftarrow{h_t^i} = f(\overleftarrow{W^i}\overleftarrow{h_t^{i-1}} + \overleftarrow{V^i}\overleftarrow{h_{t-1}^i} + \overleftarrow{b_t^i})$

$$\hat{y} = g(V[\overrightarrow{h_t^L}, \overleftarrow{h_t^L}] + c)$$

5. Attention

Decoder 预测公式 $p(y_t|y_1,\cdots y_{t-1},x)=g(y_{t-1},s_t,c_t)$ （当前要生成token；decoder hidden state）

$s_t=f(s_{t-1},y_{t-1},c_t)$ （上一个）

$c_t=\sum_{j=1}^{T}\alpha_{tj}h_j$ （context vector；t对j个输入位置注意力权重）

$\alpha_{tj}=\frac{\exp(e_{tj})}{\sum_{k=1}^{T}\exp(e_{tk})}$ 其中 $e_{tj}=a(s_{t-1},h_j)$

6. RNN问题
- 难以并行
- BP through sequence
- 局部信息与全局信息都通过 hidden state 这个 bottle-neck 传递

	RNN	self-Attention	convolution
单层	$O(nd^2)$	$O(n^2d)$	
顺序操作	$O(n)$	$O(1)$	
最大路长度	$O(n)$	$O(1)$	

总work仍是O(T)

7. Mamba

$h_t=A(x_t)h_{t-1}+B(x_t)x_t$

$y_t=C(x_t)h_t$

重要token强烈更新memory

不重要token几乎不改memory.

inside a structured state-space evolution

Lec 6

1. Generalization gap $S=\{(x_i,y_i)\}_{i=1}^m$ $y_i \in R$

空间 $H=\{h(x)=w^T\phi(x)\mid w\}$

训练误差 $\hat{\varepsilon}(h)=\frac{1}{m}\sum_{i=1}^m (y_i-h(x_i))^2$ 训练过程 $\hat{h}=\underset{h\in H}{\operatorname{argmin}}\ \hat{\varepsilon}(h)$

泛化误差 $\varepsilon(h_S)=E_{(x,y)\sim D}[(y-h_S(x))^2]$

generalization gap $\varepsilon(h_S)-\hat{\varepsilon}(h_S)$

overfit: train error低, generalization high

model 对 train set 随机波动敏感

2. Bias-Variance Decomposition

$$\varepsilon=E_S[\varepsilon(h_S)]=E_S[E_{(x,y)\sim D}[(y-h_S(x))^2]]$$

代入 $y-h_S=y-E_S[h_S]+E_S[h_S]-h_S$.

$$\varepsilon=E_S E_{(x,y)}[(y-E_S[h_S]+E_S[h_S]-h_S)^2]$$

后面这项也来是0

交叉项为0 因为 $E_S E_{(x,y)}[(y-E_S[h_S])(E_S[h_S]-h_S)]$

$$=E_S E_{(x,y)}[(y-E_S[h_S])^2]+E_S E_{(x,y)}[(E_S[h_S]-h_S)^2]$$

$$=E_{(x,y)}[(y-E_S[h_S(x)])^2]+E_S E_x[(E_S[h_S(x)]-h_S(x))^2]$$

Expected Generation error = Bias^2 + Variance

bias. 主要由 H 表达能力决定

3. Ridge 与 LASSO.

Ridge: $J(w,w_0)=\frac{1}{m}\sum_{i=1}^m [y_i-(w_0+w^T\phi(x_i))]^2+\lambda\|w\|_2^2$ 压小权重.

$\phi(x_i)$ 特征映射

Lasso: $J(w,w_0)+\frac{1}{m}\sum_{i=1}^m [y_i-(w_0+w^T\phi(x_i))]^2+\lambda\|w\|_1$ 诱导稀疏性

⇒ 统一形式 $\tilde{J}(\theta)=J(\theta)+\alpha\Omega(\theta)$ $\alpha\in[0,\infty)$ 正则强度

4. L2 regularization / weight decay

$\tilde{J}(w)=J(w)+\frac{\alpha}{2}\|w\|_2^2$

$\nabla_w\tilde{J}(w)=\nabla_w J(w)+\alpha w$

$w\leftarrow w-\eta(\nabla_w J(w)+\alpha w)=(1-\eta\alpha)w-\eta\nabla_w J(w)$ weight decay

在未正则最优点 w^* 附近做二阶近似

$f(x)\approx f(x_0)+(x-x_0)f'(x_0)+\frac{1}{2}(x-x_0)^2 f''(x_0)$

$\tilde{J}(w)=J(w^*)+(w-w^*)\nabla_w J(w^*)+\frac{1}{2}(w-w^*)^T H(w-w^*)$ $H=\nabla_w^2 J(w^*)$

因为 w^* 最优, $\nabla_w J(w^*)=0$

$$=J(w^*)+\frac{1}{2}(w-w^*)^T H(w-w^*)$$

H 只与 w^* 有关, w^* 已确定. H const

$\tilde{J}(w)$ 在梯度为0时等于0最小值. 对w求导 $\nabla_w\tilde{J}(w)=H(w-w^*)=0$

加入 L^2 regu 之后 有 $\alpha\tilde{w}+H(\tilde{w}-w^*)=0\Leftrightarrow\tilde{w}=(H+\alpha I)^{-1}Hw^*$ $(ABC)^{-1}=C^{-1}A^{-1}B^{-1}$

当 $\alpha\to 0$, $\tilde{w}\to w^*$

$\alpha\uparrow$ 分解 $H=Q\Lambda Q^T$ Q orthogonal $Q^TQ=I$ $QQ^T=I$ $Q^T=Q^{-1}$

$\tilde{w}=(Q\Lambda Q^T+\alpha I)^{-1}Q\Lambda Q^T w^*=(Q(\Lambda+\alpha I)Q^T)^{-1}Q\Lambda Q^T w^*=Q(\Lambda+\alpha I)^{-1}\Lambda Q^T w^*$

$\tilde{w} = Q(\Lambda+\alpha I)^{-1}\Lambda Q^T w^*$ 在第 i 个特征方向上，权重被缩放为 $\frac{\lambda_i}{\lambda_i+\alpha}$.

L^2 对低曲率方向惩罚更强，对高曲率方向影响较弱

动一点对 loss 影响小，不重要　　动一点对 loss 影响大，重要

$J(w;X,y) = (Xw-y)^T(Xw-y) \Rightarrow w^* = (X^TX)^{-1}X^Ty$.

加入 L^2 $\Rightarrow$ $(X^TX+\alpha I)^{-1}X^Ty$

防止 X^TX 奇异

Bayesian: L^2 reg 等价于对权重施加 Gaussian prior 后做 MAP estimation

即一开始假设

5. L_1-norm

$\nabla_w \tilde{J}(w) = \nabla_w J(w) + \alpha \operatorname{sign}(w)$ 不可导，用 subgradient.

对 w^* 附近似，设 Hessian 是 diagnal. $H = \operatorname{diag}(H_{1,1},\dots H_{n,n})$, $H_{i,i}>0$.

$\tilde{J}(w) = J(w^*) + \sum_i \frac{1}{2}H_{ii}(w_i - w_i^*)^2 + \alpha|w_i|$

each dim: $w_i = \operatorname{sign}(w_i^*)\max\{|w_i^*| - \frac{\alpha}{H_{i,i}}, 0\}$. soft-threshdding.

$|w_i^*| \le \frac{\alpha}{H_{i,i}} \Rightarrow w_i = 0$ 否则 $|w_i| = |w_i^*| - \frac{\alpha}{H_{i,i}}$ 向0移动.

$\log p(w) = \sum_i \log \text{Laplace}(w_i; 0, \frac{1}{\alpha}) = -\alpha\|w\|_1 + n\log\alpha - n\log 2$

6. Norm Penalty 作为 constrained optimization

$\min J(\theta)$, s.t. $\Omega(\theta)-k\le 0$. 开拉 $L(\theta,\alpha) = J(\theta)+\alpha(\Omega(\theta)-k)$ $\alpha\ge 0$.

$\theta^* = \arg\min_\theta \max_{\alpha,\alpha\ge 0} L(\theta,\alpha)$

7. Data Augmentation { random crop　crop → resize
Drop block　遮挡，用灰色/随机彩色

more valid data $\Rightarrow$ less variance

8. Noise Injection 等价于对权重加惩罚

~~$\tilde{J}_w = E_{p(x,y,\varepsilon_w)}[(\hat{y}_{\varepsilon_w}(x)-y)^2] = E_{p(x,y,\varepsilon_w)}[\hat{y}$~~

$\hat{y}(w+\varepsilon_w) \approx \hat{y}(w,x) + \varepsilon^T\nabla_w\hat{y}(w,x) + \frac{1}{2}\varepsilon_w^T\nabla_w^2\hat{y}(w,x)\varepsilon_w$

$\hat{y}_{\varepsilon_w}(x) \approx \hat{y}(x) + \varepsilon_w^T\nabla_w\hat{y}(x)$

$\tilde{J}_w = E_{x,y,\varepsilon_w}[(\hat{y}(x)+\varepsilon_w^T\nabla_w\hat{y}(x)-y)^2] = E_{x,y,\varepsilon_w}[(\hat{y}(x)-y)^2 + 2(\hat{y}(x)-y)\varepsilon_w^T\nabla_w\hat{y}(x) + (\varepsilon_w^T\nabla_w\hat{y}(x))^2]$

$(\hat{y}(x)-y)^2$ 与 ε 无关

$2(\hat{y}(x)-y)\varepsilon_w^T\nabla_w\hat{y}(x)$ 期望为 0

$(\varepsilon_w^T\nabla_w\hat{y}(x))^2 = \eta\|\nabla_w\hat{y}(x)\|^2$

Lec 7 Transformer

1. Pos emb.

$w_i = \frac{1}{10000^{2i/d_m}}$

$PE_{pos,2i} = \sin(w_i pos)$

$PE_{pos,2i+1} = \cos(w_i pos)$

$\sin(a+b) = \sin a\cos b + \cos a \sin b$

$\cos(a+b) = \cos a \cos b - \sin a \sin b$

$$\begin{bmatrix} \sin(w_i(pos+k)) \\ \cos(w_i(pos+k)) \end{bmatrix} = \begin{bmatrix} \cos(w_i k) & \sin(w_i k) \\ -\sin(w_i k) & \cos(w_i k) \end{bmatrix} \begin{bmatrix} \sin(w_i pos) \\ \cos(w_i pos) \end{bmatrix}$$

$PE_{pos+k} = M_k PE_{pos}$ M_k只依赖偏移k，不依赖绝对位置，self-att可通过线性变换学习相对位置信息

2. Decoder

2.1 Encoder-Decoder Attention

queries来自decoder，k,v来自encoder

$Q_i = X_{decoder} W_i^Q \quad K_i = X_{encoder} W_i^K \quad V_i = X_{encoder} W_i^V$

$Z_i = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}}) V_i$ decoder在当前位置提出query，去encoder检索相关信息

2.2 输出层 $P(y_t = v \mid y_{<t}, X) = softmax(Wh_t + b)_v$ ← 词表中某token

h_t 在位置t的final hidden state

$L = -\sum_t \log P(y_t^* \mid y_{<t}^*, X)$

3. BERT

3.1 Masked Language Model

$P(x_i \mid X_{\setminus i}) = softmax(WT_i + b)$ 被mask位置的输出

$L_{MLM} = -\sum_{i \in M} \log P(x_i^* \mid X_{corrupted})$ ← 被mask后全部input

3.2 Next Sentence Prediction.

50% B在A后 50% B是随机句子.

使用 [CLS] 输出向量 C 预测：不一定是天然是句子内容好 summary.

$P(IsNext \mid A, B) = softmax(WC + b)$ 下游任务finetune

NSP Loss $L_{NSP} = -\log P(y_{NSP}^* \mid C)$

$L_{BERT} = L_{MLM} + L_{NSP}$

3.3 classification → 用[CLS] Token-level task

$\hat{y} = softmax(WC + b)$ 每个token的输出 $\hat{y}_i = softmax(WT_i + b)$

T_i：第i token contextual representation

4. GPT

5. ViT 一个patch $R^{1\times(P^2C)}$ $N = \frac{HW}{P^2}$ 长度为 $\frac{HWC}{HW/P^2} = P^2C$.

$z_0 = [x_{class}; x_p^1 E; \cdots] + E_{pos}$ 额外加入可学习 class token x_{class} 无embedding

E是embedding matrix x_{class}不用乘，但要加 pos emb

标准 encoder $y = \text{MLPHead}(z_L^0)$ ← class token 最终 representation

6. CLIP : Contrasive Language- Image Pre-traing .

图像. 文本都 embed 进同一个 embedding space

让匹配的 text - image 对 vector 相似. 非 match 不相似 .

6.1 Loss 对一个 batch , 含 N pair $(I_1, T_1), \cdots, (I_N, T_N)$

$$L_{CLIP} = -\frac{1}{2N}\sum_{i=1}^{N}\log\frac{\exp(I_i \cdot T_i)}{\sum_{j=1}^{N}\exp(I_i \cdot T_j)} - \frac{1}{2N}\sum_{j=1}^{N}\log\frac{\exp(I_j \cdot T_j)}{\sum_{i=1}^{N}\exp(I_i \cdot T_j)}$$

让 image 和 match text 匹配度在 texts 中最大 对 text 求和

让 text 和 match image 在 images 中匹配度最大, 对 image 求和

6.2 zero-shot classification : ① Create a text prompt for each class

$$\text{Prob}_C = \frac{\exp(I \cdot T_C)}{\sum_{C'=1}^{C}\exp(I \cdot T_{C'})}$$ → only one 可以用 argmax 找最匹配的类.

一类.

DL Lec 8

1. The path to LLMs

1.1 Count-based Models: n-gram

$$P(x_{1:T}) = \prod_{t=1}^{T} P(x_t | x_{1:t-1}) \approx \prod_{t=1}^{T} P(x_t | x_{t-n+1}, \dots, x_{t-1})$$

Markov assumption: 当前词只依赖前面有限的 n-1 个词

问题: ① 稀疏性 ② n个指增 ③ 很多上下文训练集无

X Perplexity $PPL = \exp\left(-\frac{1}{T}\sum_{t=1}^{T} \log P(x_t | x_{<t})\right)$

求用以2为底的log $PPL = 2^{-\frac{1}{T}\sum_{t=1}^{T} \log_2 P(x_t | x_{<t})}$ PPL越低，给真实token概率越高

2. Scaling Law

2.1 Kaplan { Performance 依赖 scale，对 model shape 弱相关
大模型 sample-efficient
Smooth power laws 小模型趋势外推到大模型

超参影响可用小模型预测

2.2 emergent abilities 小模无，有模突然有 但不只由参数量决定，data, 训法, 架构, tokenizer 优化策略都影响

3. Encoder-decoder (T0 | T5) Masked span prediction

$$L_{span} = -\sum_{t=1}^{T_y} \log P_\theta(y_t | y_{<t}, \tilde{x})$$

encoder 双向注意力，decoder causal 翻译、摘要、条件生成强

$\tilde{x}$ 是被 mask span 替换序列，y 是要恢复的序列

decoder-only 强于 encoder-decoder

4. Llama

LayerNorm → RMSNorm

ReLU → $SwiGLU(x) = Swish(xW)\times V = xW\,sigmoid(Ax W)\times V$

Position: Absolute/Relative

MHA → GQA

5. CoT

$$P_\theta(a|x) = \sum_r P_\theta(a, r|x) = \sum_r P_\theta(r|x)\, P_\theta(a|x, r)$$

引入中间变量 r

先推理再回答，在大模上收益明显，size ↑ benifit 越明显

zero-shot CoT 有时甚至超过 few-shot CoT

6. LoRA $W' = W + \Delta W = W + BA$ $W \in R^{d\times r}$ $A \in R^{r\times k}$

$W \in R^{d\times k}$

7. Multimodal LLMs

Continuous repre: 把图|audio encode 成 dense embeddings 与文本 embedding 拼送 LLM

$Z_{image} = f_{vision}(I)$ $Z = [Z_{image}; Z_{text}]$

Discrete 信息丰富，性能强 计算和存储有重

特征量化为 discrete tokens，VQ-VAE 或 clustering

$z_t = \arg\min_j \|h_t - c_j\|^2$

z_t: 离散 unit id; h_t: 连续特征; c_j: codebook center

⇒ 多模态状态输入像文本 token 一样被 LLM 处理

省存储、序列长度↓去重，subword-like modeling

DL Lec 9 VAE & Diffusion

1. Intro

生成模型 ⇒ model distribution → 处理缺失数据 / 测试对高维数据handle / 结合RL

两类 { explicit generative models: GMM, log likelihood. max
implicit generative models: GAN 没有max likelihood

2. Autoencoder

2.1 def: 无监督. target为输入本身. 一般用于降维

希望学 $x \in R^n \to h \in R^d \quad d << n$ h是低维表示,也叫encoding

2.2 结构

encoder: $h = f_\theta(x) = a(Wx+b)$ θ W, b

decoder: $z = g_{\theta'}(h) = a(W'h+b')$ θ' W' b'

$$\theta^*, \theta'^* = \arg\min_{\theta,\theta'} \frac{1}{n}\sum_{i=1}^{n} L(x^{(i)}, z^{(i)}) = \arg\min_{\theta,\theta'} \frac{1}{n}\sum_{i=1}^{n} L(x_i^{(i)}, g_\theta(f_\theta(x^{(i)})))$$

$L(x,z) = \|x-z\|^2$

若x和z是prob的vector, L可以是CE Berndli

$$L_H(x,z) = H(\beta_x \| \beta_z) = -\sum_{k=1}^{d} [x_k \log z_k + (1-x_k)\log(1-z_k)]$$

2.3

Under complete: $\dim(h) < \dim(x)$ 压缩信息

Over complete: $\dim(h) > \dim(x)$ 不一定有提取意义,可能直接学identity matrix

tied weight: decoder权重 $W^* = W^T$ ← encoder 要加regularization: sparsity, denoising contractive penalty

3. Linear Autoencoder与PCA关系

在线性encoder, l_2 reconstruction loss下, 最优linear autoencoder学到的子空间等价于PCA主成分子空间.

$A = U\Sigma V^T$ $B^* = \arg\min_{B:\,rank(B)=k} \|A-B\|_F = U_{\cdot,\le k}\Sigma_{\le k,\le k}V^T_{\cdot,\le k}$ 前k个左奇异值向量 若只能用rank-r近似数据, 最优是留前k的principle directions

Autoencoder 平方误差满足 $\min_\theta \sum_t \frac{1}{2}(x_i^{(t)} - z_{ei}^{(t)})^2 \ge \min_{W',h(X)} \frac{1}{2}\|X - W'h(X)\|_F^2$

最优解为 $W' \leftarrow U_{\cdot,\le k}$ $h(X) \leftarrow \Sigma_{\le k,\le k}V^T_{\cdot,\le k}$ 设 $X = U\Sigma V^T$

$\Rightarrow h(x) = U^T_{\cdot,\le k}x$ $z = U_{\cdot,\le k}h(x)$

$X_{centered} = X - \mu_X$

这正是PCA encoder $x^{(t)} \leftarrow \frac{1}{\Gamma}\left(x^{(t)} - \frac{1}{T}\sum_{i=1}^{T} x^{(i)}\right)$

令 $\tilde{X} = \frac{1}{\Gamma}X_{centered}$

$\tilde{X}\tilde{X}^T = \frac{1}{T}X_{centered}X_{centered}^T = C$ 协方差矩阵

$\Sigma = \frac{1}{m-1}B^TB$

singular values and vectors = eigenvalues/vectors of covariance matrix

4. AE → VAE

4.1 为什么 AE 不能直接生成

① 未显式规定 latent space 的概率分布 ⇒ 破碎、连续、空散

② 训练时 decoder 只见过 encoder 产生的 latent codes，没见过可任意采样的 latent vector

VAE $z \sim p(z) = N(0, I)$

4.2 生成假设

假设训练数据 $X = \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ 由隐变量 z 生成

$z \sim p(z)$，$x \sim p_\theta(x|z)$　$p(z)$ 常取 Gaussian prior $N(0, I)$

Decoder 建模　$p_\theta(x|z) = N(\mu_x, \sigma_x^2)$　由对二值用 Bernoulli

希望最大化　$p_\theta(x) = \int p(z)\, p_\theta(x|z)\, dz$　要对所有 z，intractable，且 posterior $p_\theta(x|z)$ 也不可解

引入 encoder network　$q_\phi(z|x)$ 来近似 $p_\theta(x|z)$　设 $q_\phi(z|x) = N(\mu_z(x), \sigma_z(x))$

4.3 ELBO

$$\log p_\theta(x) = E_{z\sim q_\phi(z|x)}[\log p_\theta(x)]$$

$\log p_\theta(x)$ 与 z 无关

$$= E_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{p_\theta(x|z)}\right] = E_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)} \frac{q_\phi(z|x)}{p_\theta(x|z)}\right]$$

贝叶斯 $p_\theta(x) = \frac{p_\theta(x,z)}{p_\theta(x|z)}$　上下同乘 $q_\phi(z|x)$

$$= E_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + E_{z\sim q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(x|z)}\right]$$

拆开

$$= E_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + D_{KL}(q_\phi(z|x) \| p_\theta(x|z))$$

KL ≥ 0

$$\geq \mathcal{L}(\theta, \phi, x) = E_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$

tractable

最大化 ELBO ⇔ 最大化 $\log p_\theta(x)$ 下界，间接上 approximate posterior 接近 true posterior

$$= E_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)\, p_\theta(z)}{q_\phi(z|x)}\right]$$

$z \sim q_\phi(z|x)$ 其实因有随机节点无法 sample，但可采 $\varepsilon \sim N(0,1)$，$z = \mu_\phi(x) + \sigma(x)\varepsilon$

$$= E_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(z)}{q_\phi(z|x)}\right] + E_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right]$$

$$= -D_{KL}(q_\phi(z|x) \| p_\theta(z)) + E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$$

encoder 输出的 $z|x$ 不要偏离 $p(z)$　reconstruction error − regularization term　让 decoder 从 z 重构 x

4.4 Gaussian KL 闭式解

当 prior 为标准高斯 $p(z) = N(0, I)$　$q_\phi(z|x) = N(\mu_z(x;\phi), \sigma_z^2(x;\phi))$

$$-D_{KL}(q_\phi(z|x) \| p(z)) = \frac{1}{2}(1 + \log \sigma_z^2 - \mu_z^2 - \sigma_z^2)$$

多维时对 latent dimension $j = 1, \dots, J$ 求和

$$-D_{KL}(q_\phi(z|x) \| p(z)) = \frac{1}{2}\sum_{j=1}^{J}(1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2)$$

化简时令 $\mu_z \to 0$　$\sigma_z^2 \to 1$　$\mu_{z,j}$ 是 j-th dim 均值　$\sigma_{z,j}^2$ 是方差

4.5 Representation Trick

$$E_{q_\phi(z|x)}[\log p_\theta(x|z)] \approx \frac{1}{L}\sum_k \log p_\theta(x|z^{(k)}),\quad z^{(k)} \sim q_\phi(z|x)$$

$z^{(k)}$ 是 random sample，BP 不能穿过 sampling node

$$\Rightarrow \mathcal{L}(\theta,\phi,x) \approx \frac{1}{2}\sum_{j=1}^{J}(1+\log \sigma_{zj}^2 - \mu_{zj}^2 - \sigma_{zj}^2) + \frac{1}{L}\sum_k \log p_\theta(x|z^{(k)})$$

$\varepsilon^{(k)} \sim N(0,I)$

$z^{(k)} = \mu_z(x,\phi) + \sigma_z(x,\phi)\cdot\varepsilon^{(k)}$

z 对 μ_x，σ_z 是可微的

4.6 Generation：采样 $z \sim N(0,I)$

Diagonal prior on z ⇒ independent variables

4.7 VQ-VAE：离散 latent representation

动机：token. visual code 等数据具有天然离散结构. VAE 用连续 Gaussian latent 不一定适合.

用 discrete latent embeddings - posterior 和 prior 都用 categorial distribution

采样结果是整数 index. 用 index 查 embedding table

$$q(z=k|x) = \begin{cases} 1, & k = \arg\min_j \|z_e(x) - e_j\|_2 \\ 0, & \text{otherwise} \end{cases}$$

encoder 输出会被量化到最近 codebook vector

$$L = \log p(x|z_q(x)) + \|sg[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - sg[e]\|_2^2$$

codebook embedding：让 e 靠近 encoder 输出

commitment loss：让 encoder 输出承诺靠近某 embedding

$sg[\cdot]$ 是 stop-gradient operator：前传为 identity，后传 $\nabla = 0$

注意是一个格子选一个代码，而不是整张图选一个

5. Diffusion

5.1 forward：$x_0 \sim q(x)$

$$x_t = \sqrt{1-\beta_t}\,x_{t-1} + \sqrt{\beta_t}\,\varepsilon_t \quad \varepsilon_t \sim N(0,I)$$

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t}\,x_{t-1}, \beta_t I),\quad q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

$$q(x_{0:T}) = q(x_0)q(x_{1:T}|x_0) = q(x_0)\prod_{t=1}^{T} q(x_t|x_{t-1})$$

$z \sim N(\mu_X+\mu_Y, \sigma_X^2+\sigma_Y^2)$

定义 $\alpha_t = 1-\beta_t$，$\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$

$$x_t = \sqrt{\alpha_t}\,x_{t-1} + \sqrt{\beta_t}\,\varepsilon_t = \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\,x_{t-2} + \sqrt{1-\alpha_{t-1}}\,\varepsilon_{t-1}\right) + \sqrt{1-\alpha_t}\,\varepsilon_t$$

$$= \sqrt{\alpha_t\alpha_{t-1}}\,x_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1}) + 1 - \alpha_t}\,\varepsilon = \sqrt{\alpha_t\alpha_{t-1}}\,x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}$$

$$\Rightarrow x_t = \sqrt{\bar{\alpha}_t}\,x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon \quad \varepsilon \sim N(0,I) \quad q(x_t|x_0) = N(x_t; \sqrt{\bar{\alpha}_t}\,x_0, (1-\bar{\alpha}_t)I)$$

$t\uparrow \Rightarrow \bar{\alpha}_t \to 0 \quad q(x_T|x_0) \approx N(0,I)$

5.2 Noise Scheduler

linear：$\beta_t = \beta_1 + \frac{t-1}{T-1}(\beta_T - \beta_1)$

quadratic $\beta_t = \left[\sqrt{\beta_1} + \frac{t-1}{T-1}(\sqrt{\beta_T} - \sqrt{\beta_1})\right]^2$

cosine：$\bar{\alpha}_t = \frac{f(t)}{f(0)}$，$f(t) = \cos^2\left(\frac{\frac{t}{T}+s}{1+s}\cdot\frac{\pi}{2}\right)$

sigmoid $\beta_t = \beta_1 + \frac{\beta_T - \beta_1}{1+e^{\tau(1-2\frac{t-1}{T-1})}}$

5.3 反向 $x_T \sim N(0,I)$ $x_{t-1} = \mu_\theta(x_t,t) + \sigma_t z$, $z \sim N(0,I)$ $p(x_T) = N(x_T; 0, I)$

$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t,t), \sigma_t^2 I)$ $p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$

5.4 Noise Predictor：不直接预测 μ_θ，而是训 network 预测加到 x_0 的 compound noise

↓ time embedding

$\varepsilon_\theta(x_t,t) \approx \varepsilon_t$

noisy image U-Net

$L = E_{x_0\sim q(x_0),\, t\in U(1,T),\, \varepsilon_t\sim N(0,I)}[\| \varepsilon_t - \varepsilon_\theta(\sqrt{\bar\alpha_t}x_0 + \sqrt{1-\bar\alpha_t}\varepsilon_t, t)\|^2]$

- 随机取一真实图，时间步，加噪，预测

5.5 为什么 noise prediction 等价 reverse mean

$q(x_{t-1}|x_t, x_0) = N(x_{t-1}; \tilde\mu_t(x_t,x_0), \tilde\beta_t I)$ $\tilde\mu_t(x_t,x_0) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar\alpha_t}}\varepsilon\right)$

$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\varepsilon_\theta(x_t,t)\right)$ $D_{KL} \propto \|\tilde\mu_t - \mu_\theta\|^2 \Rightarrow E_{x_0,\varepsilon,t}[\|\varepsilon - \varepsilon_\theta(x_t,t)\|^2]$

5.6 ELBO

$E_{x_0\sim q(x_0)}[-\log p_\theta(x_0)] \le E_{x_0, x_{1:T}\sim q(x_{0:T})}\left[-\log \frac{p_\theta(x_0,x_{1:T})}{q(x_{1:T}|x_0)}\right]$

$= E_q\left[-\log p(x_T) - \sum_{t=1}^{T}\log p_\theta(x_{t-1}|x_t) + \sum_{t=1}^{T}\log q(x_t|x_{t-1})\right]$

telescoping ~~$p_\theta(x_{t+1}|x_t) = p_\theta$~~ $q(x_t|x_{t-1}) = q(x_t|x_{t-1},x_0)$

$= \frac{q(x_t|x_0)\, q(x_{t-1}|x_t,x_0)}{q(x_{t-1}|x_0)}$ log 完 $= \sum_{t=1}^{T}\log q(x_t|x_0)$

$= D_{KL}(q(x_T|x_0)\|p(x_T)) + \sum_{t=2}^{T} E_q[D_{KL}(q(x_{t-1}|x_t,x_0)\| p_\theta(x_{t-1}|x_t)]] - E_q[\log p_\theta(x_0|x_1)]$ $+ \sum_{t=1}^{T}\log q(x_{t-1}|x_t) - \sum_{t=1}^{T}\log q(x_{t-1}|x_0)$

让 final noisy distribution 匹配 gaussian prior

reverse transition matching

$= \log q(x_T|x_0) - \log q(x_0|x_0)$ telescoping

reconstruction term $= \log q(x_T|x_0)$

5.7 Sample

① $x_T \sim N(0,I)$

② 对于 $t = T, T-1, \dots, 1$ ↓ denoising direction

$x_{t-1} = \mu_\theta(x_t,t) + \sigma_t z$ ← stochastic sampling term $z\sim N(0,I)$

其中 $\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\varepsilon_\theta(x_t,t)\right)$

理论 $\sigma_t^2 = \tilde\beta_t = \frac{1-\bar\alpha_{t-1}}{1-\bar\alpha_t}\beta_t$ 但实践 $\sigma_t^2 = \beta_t$

若 σ_t 为 0 则 stochastic sampling

若 $\sigma_t = 0$，则 DDIM deterministic

DL Lec 10 Diffusion models

1. ODE SDE

ODE

$\frac{dx}{dt} = f(x,t)$ or $dx = f(x,t)\,dt$

Analytical solu $x(t) = x(0) + \int_0^t f(x,\tau)\,d\tau$

Iterative Num solu $x(t+\Delta t) \approx x(t) + f(x(t),t)\Delta t$

确定性轨迹，给定 $x(0)$ 后未来状态完全确定

无randomness

SDE

$\frac{dx}{dt} = \underline{f(x,t)} + \underline{g(x,t)w_t}$ drift coefficient, diffusion coefficient

$dx = f(x,t)dt + g(x,t)\underline{dw_t}$ random

$x(t+\Delta t) \approx x(t) + f(x(t),t)\Delta t + g(x(t),t)\sqrt{\Delta t}\,N(0,I)$

每次运行都不同，引入概率云

2. 从离散 diffusion → 连续时间 SDE

无限多小步的极限

2.1 forward diffusion 连续极限

$x_t = \sqrt{1-\beta_t}\,x_{t-1} + \sqrt{\beta_t}\,N(0,I)$

令 $\beta_t := \beta(t)\Delta t$

Taylor 当 Δt 很小时：$\sqrt{1-\beta(t)\Delta t} \approx 1 - \frac{1}{2}\beta(t)\Delta t$

$x_t \approx x_{t-1} - \frac{\beta(t)\Delta t}{2} x_{t-1} + \sqrt{\beta_t \Delta t}\,N(0,I)$

令 $\Delta t \to 0$ 得到前向 SDE

$dx_t = -\frac{1}{2}\beta(t)\underline{x_t dt} + \underline{\sqrt{\beta(t)}\,dw_t}$

drift term　　diffusion term

Wiener process 布朗 → 连续时间噪声极限

把 $x_t \to 0$ 拉，一方面加小新噪声

2.2 Reverse-Time SDE and score function

SDE

前向：$dx_t = -\frac{1}{2}\beta(t)x_t dt + \sqrt{\beta_t}\,dw_t$

反向：$dx_t = [-\frac{1}{2}\beta(t)x_t - \beta(t)\nabla_{x_t}\log q_t(x_t)]\,dt + \sqrt{\beta(t)}\,d\bar{w}_t$

score：时间边缘分布 q_t 的 $\log$ 对 x_t 的梯度；反向 → 反向布朗

$\nabla_{x_t}\log q_t(x_t)$ 表示当前 x_t 应往哪里走，才能更接近高频数据区

- Score function log-density 对 input 的 ∇　$s(x) = \nabla_x \log q(x)$

若某点在低频区，score 指引移动至高频区域方向

在diffusion中，反向生成的本质是学每个噪声等级下的 score

$s_\theta(x_t,t) \approx \nabla_{x_t}\log q_t(x_t)$

反向 SDE 推导：前向 $dx_t = -f(x_t,t)dt + g(t)\,dw_t$

小时间步转移 $x_{t+\Delta t} = x_t + f(x_t,t)\Delta t + g(t)\sqrt{\Delta t}\,\varepsilon$, $\varepsilon \sim N(0,I)$

$p(x'|x) = N(x'; x + f(x,t)\Delta t, g(t)^2\Delta t I)$

贝叶斯 $p(x|x') \propto p(x'|x)q_t(x)$

一阶展开 $\log q_t(x) \approx \log q_t(x') + (x-x')^T \nabla \log q_t(x')$

Gaussian Completion $E[x_t | x_{t+\Delta t}] = x_{t+\Delta t} - f(x_{t+\Delta t},t)\Delta t + g(t)^2 \nabla \log q_t(x_{t+\Delta t})\Delta t$

$\Rightarrow dx_t = [f(x_t,t) - g(t)^2 \nabla_{x_t}\log q_t(x_t)]\,dt + g(t)\,d\bar{w}_t$，代入 VP-SDE

$g(t) = \sqrt{\beta(t)}$

$f(x_t,t) = -\frac{1}{2}\beta(t)x_t$

$dx_t = [-\frac{1}{2}\beta(t)x_t - \beta(t)\nabla_{x_t}[\log q_t(x_t)]]\,dt + \sqrt{\beta(t)}\,d\bar{w}_t$

3. Score Matching & Denoising Score Matching

希望 $\min_\theta E_{t\sim U(0,T)} E_{x_t\sim q_t(x_t)}[\|S_\theta(x_t,t)-\nabla_{x_t}\log q_t(x_t)\|_2^2]$ 直接训，但 $\nabla_{x_t}\log q_t(x_t)$ 不可算。所有数据点扩散后边缘混合分布，且 $q_0(x_0)$ 未知

但有 $\nabla_{x_t}\log q_t(x_t) = E_{x_0\sim q(x_0|x_t)}[\nabla_{x_t}\log q_t(x_t|x_0)]$ 条件得分在后验分布下期望=边缘得分

Proof: $\log q(x_0|x_t) = \log q_0(x_0) + \log q_t(x_t|x_0) - \log q_t(x_t)$ 贝叶斯，对 x_t 求 ∇

$\nabla_{x_t}\log q(x_0|x_t) = \nabla_{x_t}\log q_0(x_t) + \nabla_{x_t}\log q_t(x_t|x_0) - \nabla_{x_t}\log q_t(x_t)$ （$\log q_0(x_t)$ 无关）

移项 $\nabla_{x_t}\log q_t(x_t|x_0) = \nabla_{x_t}\log q(x_0|x_t) + \nabla_{x_t}\log q_t(x_t) = \int q(x_0|x_t)\nabla_{x_t}\log q(x_0|x_t)\,dx_0$

固定 x_t 取条件期望 $E_{x_0\sim q(x_0|x_t)}[\nabla_{x_t}\log q(x_0|x_t)] = \nabla_{x_t}\log q_t(x_t)$ ↓在 x_t 固定时与 x_0 无关

用 $\nabla_{x_t}\log q(x_0|x_t) = \frac{\nabla_{x_t} q(x_0|x_t)}{q(x_0|x_t)}$ log trick $\Rightarrow \int \nabla_{x_t} q(x_0|x_t)\,dx_0 = \nabla_{x_t}\int q(x_0|x_t)\,dx_0 = \nabla_{x_t}(1) = 0$

可得到 $\min_\theta E_{t\sim U(0,T)} E_{x_0\sim q_0(x_0)} E_{x_t\sim q_t(x_t|x_0)}[\|S_\theta(x_t,t)-\nabla_{x_t}\log q_t(x_t|x_0)\|_2^2]$

PPT：在期望后 $S_\theta(x_t,t)\approx\nabla_{x_t}\log q_t(x_t)$ 其实应为 $\nabla_{x_t}\log q_t(x_t) = E_{x_0\sim q(x_0|x_t)}[\nabla_{x_t}\log q_t(x_t|x_0)]$

2.1 代入 $x_t = \gamma_t x_0 + \sigma_t\varepsilon$ $\varepsilon\sim N(0,I)$ $\nabla_{x_t}\log q_t(x_t|x_0) = -\nabla_{x_t}\frac{(x_t-\gamma_t x_0)^2}{2\sigma_t^2} = -\frac{\varepsilon}{\sigma_t} \Rightarrow S_\theta(x_t,t) = -\frac{\varepsilon_\theta(x_t,t)}{\sigma_t}$

$\Rightarrow \min_\theta E_{t,x_0,\varepsilon}\left[\frac{1}{\sigma_t^2}\|\varepsilon-\varepsilon_\theta(x_t,t)\|_2^2\right]$ 说明DDPM中预测噪声本质上在学score

4. Probability Flow ODE

反向生成SDE在分布意义上等价于一个确定性ODE：$dx_t = -\frac{1}{2}\beta(t)[x_t+\nabla_{x_t}\log q_t(x_t)]dt$ probability flow ODE

随 = 都在每个时间 t 的边缘分布 $q_t(x_t)$ 相同，可用ODE solver采样

当用 $q_T(x_T)\approx N(x_T;0,I)$ 初始化

5. Flow Matching

Flow Model $\frac{dx}{dt} = u_t^\theta(x_t)$ 不随机，学vector field，解ODE

Diffusion $dx_t = u_t^\theta(x_t)+\sigma_t dw_t$ 随机，学drift/score/noise 解SDE/ODE

Flow Model 生成：① $x_0\sim p_{init}$，$dx_t = u_t^\theta(x_t)dt$ ② 从 $t=0$ 模拟到 $t=1$，得到 $x_t\sim p_{data}$

5.1 Time dependent vector field

瞬时速度 time=t loc=x

$u: R^d\times[0,1]\to R^d$ $(x,t)\mapsto u_t(x)$ 给每点一个运动方向和速度，沿速度场就能把初始分布运过去

轨迹 $\dot{x}_t = u_t(x_t) \Leftrightarrow \frac{dx_t}{dt} = u_t(x_t)$

5.2 Conditional Prob Path

给定目标数据点 $z\in R^d$，定义 cond prob path $p_t(\cdot|z)$ 满足 ← 从noise收缩到目标z的路径。如Gaussian $p_t(\cdot|z) = N(\alpha_t z, \beta_t^2 I_d)$

$p_0(\cdot|z) = p_{init}$，$p_1(\cdot|z) = \delta_z$ 集中在z的Dirac delta

控制从从0到z 控方差从1到0

边界 $\alpha_0=0$，$\beta_0=1 \Rightarrow p_0(\cdot|z) = N(0,I_d) = p_{init}$ $p_1(\cdot|z) = N(z|0) = \delta_z$

5.3 Cond Vector Field

对任意 cond prob path 都存在等价 vector field，使得 $x_0\sim p_{init}$，$\frac{d}{dt}x_t = u_t(x_t|z) \Rightarrow x_t\sim p_t(\cdot|z)$ 对高斯路径 $p_t(\cdot|z) = N(\alpha_t z, \beta_t^2 I_d)$

Cond vector field 为 $u_t(x_t|z) = \left(\dot\alpha_t - \frac{\dot\beta_t}{\beta_t}\alpha_t\right)z + \frac{\dot\beta_t}{\beta_t}x$ $\dot\alpha_t = \frac{d\alpha_t}{dt}$ $\dot\beta_t = \frac{d\beta_t}{dt}$ $u_t(x|z)$ 在目标z，把x推到z的速度

推导：若有 $x_t = \alpha_t z + \beta_t\varepsilon$，$\varepsilon$ fix，则 $\frac{dx_t}{dt} = \dot\alpha_t z + \dot\beta_t\varepsilon$，又因为 $\varepsilon = \frac{x_t-\alpha_t z}{\beta_t}$ 整理 $u_t(x_t|z) = \frac{dx_t}{dt} = \left(\dot\alpha_t - \frac{\dot\beta_t}{\beta_t}\alpha_t\right)z + \frac{\dot\beta_t}{\beta_t}x_t$

与t无关

5.4 Marginal Prob Path and Vector field

生成整个数据分布

令 $z\sim p_{data}$ 定义marginal prob path $p_t(x) = \int p_t(x|z)\,p_{data}(z)\,dz$

对应 marginal vector field $u_t(x) = \int u_t(x|z)\frac{p_t(x|z)\,p_{data}(z)}{p_t(x)}dz = E_{z\sim p(z|x_t=x)}[u_t(x|z)]$ 在位置x上，有多目标数据点z能解释中间态。marginal是后验平均。但 p_{data} 不知，$u_t(x)$ 无法显式写出 用 $\hat u_t(x)$ 逼近

5.5 FM Loss

$L_{CFM} = E_{t\sim U(0,1),z\sim p_{data},x\sim p(\cdot|z)}[\|\hat u_t(x)-u_t(x|z)\|^2]$ 最小化 L_{CFM} 等价最小化 L_{FM}。训练用条件，但期望的 $u_t(x)$

loss推导：有 $u_t(x|z) = (\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t)z + \frac{\dot{\beta}_t}{\beta_t}(\alpha_t z + \beta_t \epsilon)$ 代入 $x = \alpha_t z + \beta_t \epsilon$ $= \dot{\alpha}_t z + \dot{\beta}_t \epsilon$

$L_{CFM} = E_{t,z,\epsilon}[\| \hat{u}_t(\alpha_t z + \beta_t \epsilon) - (\dot{\alpha}_t z + \dot{\beta}_t \epsilon)\|^2]$

5.6 最简单optimal transport path 令 $\alpha_t = t$ $\beta_t = 1-t$ $\dot{\alpha}_t = 1$ $\dot{\beta}_t = -1$ 采样公式 $x = tz + (1-t)\epsilon$ $\epsilon \sim N(0, Id)$

$L_{CFM} = E_{t\sim U(0,1), z\sim p_{data}, \epsilon\sim N(0,Id)}[\| \hat{u}_t(tz + (1-t)\epsilon) - (z-\epsilon)\|^2]$ u_t（副看）$= \dot{\alpha}_t z + \dot{\beta}_t \epsilon = z - \epsilon$

预测 x 真

① $z \sim p_{data}$ ② $t \sim U(0,1)$ ③ $\epsilon \sim N(0,I)$ ④构造中间点 $x = tz + (1-t)\epsilon$ ⑤ 用network预测 $\hat{u}_t(x)$ ⑥ MSE监督

6. 加速采样 DDIM：少步采样，允许右角定性采样

传统：$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1-\bar{\alpha}_{t-1}}\, \bar{\epsilon}_{t-1}$

DDIM：$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1} - \sigma_t^2}\, \hat{\epsilon}_t + \sigma_t z$

从 $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon_t$ 推出

其中 $\hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\, \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}$ $\hat{\epsilon}_t = \frac{x_t - \sqrt{\bar{\alpha}_t}\hat{x}_0}{\sqrt{1-\bar{\alpha}_t}}$

若 $\sigma_t = 0$，则DDIM右角定，可只在子序列时间点采样 $\tau_1 = 1 < \tau_2 \dots < \tau_K \le T$ 用来估 $\hat{\epsilon}_{t-1}$

走K步

7. DPM Solver

8. Stable Diffusion

$(\log f)' = \frac{f'}{f}$

$f' = f \cdot$

9. Classifier Guidance 与 Classifier-free guidance

9.1 Vanilla：给定条件 y，直接训 $u_t^\theta(x|y)$ ①选prompt y ② $x_0 \sim p_{init}$ ③模拟 $dX_t = u_t^\theta(X_t|y)dt$ 从 $t=0$ 到 1

prob：不够强，image和prompt未对齐

9.2 classifier guidance：$\nabla_x \log p_t(x|y) = \nabla_x \log p_t(x) + \nabla_x \log p_t(y|x)$ 分解为C

不能正常预测y分类器

$u_t^{target}(x|y) = u_t^{target}(x) + \alpha_t \nabla_x \log p_t(y|x)$

unconditional dir + prompt dependent

进一步增强 $\tilde{u}_t^w(x|y) = u_t^{target}(x) + w\, a_t \nabla_x \log p_t(y|x)$ w 是 guidance scale，$w\uparrow \to$ 约束↑ → 饱和，失真，多样性↓

9.3 CFG：无需外部 classifier $u_t^{target}(x|y) - u_t^{target}(x)$ 可近似看作 prompt-dependent direction

干提放大差值 $\tilde{u}_t^w(x|y) = u_t^{target}(x) + w(u_t^{target}(x|y) - u_t^{target}(x)) = (1-w)u_t^{target}(x) + w u_t^{target}(x|y)$

CFG采样向量差项为 $u_t^{\theta,w}(x) = (1-w)u_t^\theta(x|\phi) + w u_t^\theta(x|y)$

非严格数据分布建模

$w=1$ 近似正常条件生成 ↑空文本 模拟无条件

$w>1$ $u_t^{\theta,w}(x) = u_t^\theta(x|y) + (w-1)[u_t^\theta(x|y) - u_t^\theta(x|\phi)]$ 沿 prompt-dependent direction 走得比真实分布更远