AE 自编码器

$\phi: \mathcal{X}\to F$, e.g. $h=\sigma(W_1x+b_1)$

$\psi: F\to\mathcal{X}$, e.g. $x'=\sigma(W_2h+b_2)$

$\phi,\psi=\underset{\phi,\psi}{\arg\min}\|x-(\psi\circ\phi)x\|^2$

$=\|x-x'\|^2=\|x-\sigma(W_2\sigma(W_1x+b_1)+b_2)\|^2$

VAE: 变分自编码器

推断网络: $q(z|x;\phi)=N(z;\mu_I,\sigma_I^2I)$

其中 $\begin{pmatrix}\mu_I\\ \sigma_I^2\end{pmatrix}=f_I(x;\phi)$ e.g. $h=\sigma(W^{(1)}x+b^{(1)})$, $\mu_I=W^{(2)}h+b^{(2)}$, $\sigma_I^2=\text{softplus}(W^{(3)}h+b^{(3)})$

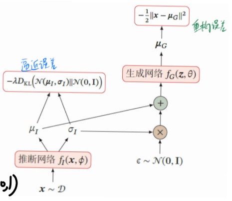
生成网络: $p(x|z;\theta)=N(x;\mu_G,\sigma_G^2I)$

总目标函数: $\underset{\theta,\phi}{\max}\ ELBO(\theta,\phi)$

$$=\max_{\theta,\phi}\ \underset{z\sim q(z;\phi)}{E}\left[\log\frac{p(x|z;\theta)p(z;\theta)}{q(z;\phi)}\right]$$

$$=\max_{\theta,\phi}\ \underbrace{\underset{z\sim q(z;\phi)}{E}[\log p(x|z;\theta)]}_{①}\underbrace{-KL(q(z|x;\phi),\overset{\text{通常是 }N(0,I)}{\boxed{p(z;\theta)}})}_{②}$$

$①\approx\frac{1}{M}\sum_{m=1}^{M}\log p(x|z^{(m)};\theta)$, $z^{(m)}=\mu_I+\sigma_I\odot\epsilon^{(m)}$, 其中 $\epsilon^{(m)}\sim N(0,1)$



重参数化技巧, 使得 $z^{(m)}$ 对 μ_I 和 σ_I 可以求导

$$J(\phi,\theta)=\sum_{n=1}^{N}\left(\frac{1}{M}\sum_{m=1}^{M}\log p(x^{(n)}|z^{(n,m)},\theta)-KL(q(z|x^{(n)};\phi),N(0,1))\right)$$

采样 $\epsilon\sim N(0,1)$, 计算 $z=\mu_I+\sigma_I\epsilon$, $\mu_G=f_G(z;\theta)$

$$J(\phi,\theta)=\underbrace{-\frac{1}{2}\|x-\mu_G\|^2}_{\text{重构误差}}-\underset{\substack{\downarrow\\ \text{超参数}}}{\lambda}\cdot\underbrace{KL(q(z|x^{(n)};\phi),N(0,1))}_{\text{逼近误差}}$$

由 $KL(N(\mu_1,\Sigma_1),N(\mu_2,\Sigma_2))=\frac{1}{2}\left(tr(\Sigma_2^{-1}\Sigma_1)+(\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1)-D+\log\frac{|\Sigma_2|}{|\Sigma_1|}\right)$

得 $KL(q(z|x;\phi),p(z;\theta))=\frac{1}{2}(tr(\sigma_I^2I)+\mu_I^T\mu_I-d-\log(|\sigma_I^2I|))$

GAN

流程图: $x\sim D\longrightarrow$ 判别网络 $D(x,\phi)\to\overset{\text{真实}}{1}/0$ 生成

$z\sim N(0,1)\to$ 生成网络 $G(z,\theta)\nearrow$

判别网络的目标函数: $\min_{\phi}-E_x[y\log p(y=1|x)+(1-y)\log(y=0|x)]$

若真实:生成=1:1, 则等价于 $\max_{\phi}\{E_{x\sim p_r(x)}[\log D(x;\phi)]+E_{z\sim p(z)}[\log(1-D(G(z;\theta);\phi))]\}$

生成网络的目标函数: $\min_{\theta}\underset{z\sim p(z)}{E}[\log(1-D(G(z;\theta);\phi))]\xrightarrow{\text{等价}}\max_{\theta}\underset{z\sim p(z)}{E}[\log(D(G(z;\theta);\phi))]$

最优判别器: $D^*(x)=\frac{p_r(x)}{p_r(x)+p_\theta(x)}$

在最优判别器下, 生成网络的目标函数简化为: $L(G|D^*)=E_{x\sim p_r(x)}[\log D^*(x)]+E_{x\sim p_\theta(x)}[\log(1-D^*(x))]$

$=E_{x\sim p_r(x)}\left[\log\frac{p_r(x)}{p_r(x)+p_\theta(x)}\right]+E_{x\sim p_\theta(x)}\left[\log\frac{p_\theta(x)}{p_r(x)+p_\theta(x)}\right]=2JS(P_r,P_\theta)-2\log2$, 其中 $JS(P_1,P_2)=\frac{1}{2}KL(P_1\|\frac{P_1+P_2}{2})+\frac{1}{2}KL(P_2\|\frac{P_1+P_2}{2})$, 当 P_r 和 P_θ 不重叠时, JS 恒为0, 即 $\frac{\partial L(G|D^*)}{\partial\theta}=0$, 梯度消失

所以判别网络不能训练太好, 但也不能太差, 否则生成网络的梯度是错的

还可以计算 $L'(G|D^*)=E_{x\sim p_\theta(x)}[\log D^*(x)]=E_{x\sim p(\theta)}\left[\log\frac{p_r(x)}{p_r(x)+p_\theta(x)}\frac{p_\theta(x)}{p_\theta(x)}\right]$

$=E_{x\sim p(\theta)}\left[\log\frac{p_r(x)}{p_\theta(x)}\right]+E_{x\sim p_\theta(x)}\left[\log\frac{p_\theta(x)}{p_r(x)+p_\theta(x)}\right]=-KL(P_\theta,P_r)+E_{x\sim p_\theta(x)}[\log(1-D^*(x))]$

$=-KL(P_\theta,P_r)+2JS(P_r,P_\theta)-2\log2-E_{x\sim p_\theta(x)}[\log D^*(x)]$

即 P_θ 会使得 KL 散度尽可能小

WGAN

用 Wasserstein 距离衡量分布, 解决 ①、② 模式崩溃, 即生成图片都很像

各种距离:

① Total Variation: $\delta(P_r,P_g)=\max_{A}|P_r(A)-P_g(A)|$

② KL distance: $KL(P_r,P_g)=\int\log\left(\frac{P_r(x)}{P_g(x)}\right)P_r(x)dx$

③ JS-divergenle: $JS(P_r,P_g)=KL(P_r\|P_m)+KL(P_g\|P_m)$ 其中, $P_m=\frac{P_r+P_g}{2}$

④ Wasserstein-1, 或 Earth-Mover distance: $W(P_r,P_g):=\inf_{\gamma\in\Pi(P_r,P_g)}E_{(x,y)\sim\gamma}[\|x-y\|]$

$=\int_{-\infty}^{+\infty}|F_1(x)-F_2(x)|dx$

Kantorovich-Rubinstein 对偶 $W(P_r,P_g)=\sup_{\|f\|_L\le1}E_{x\sim P_r}[f(x)]-E_{x\sim P_g}[f(x)]$, 其中 $\|f\|_L$ 是 f 的 Lipschtz 常数. 根据对偶的优化目标: $\max_{w}\{E_{x\sim P_r}[f_w(x)]-E_{z\sim p(z)}[f_w(g_\theta(z))]\}$

加入梯度惩罚项: $\min_{w}E_{z\sim p(z)}[f_w(g_\theta(z))]-E_{x\sim P_r}[f_w(x)]+\lambda E_{\hat{x}}[(\|\nabla_x f_w(\hat{x})\|-1)^2]$, 其中 $\hat{x}$ 和 $tX_1+(1-t)X_2$ 同分布, $X_1\sim P_r$, $X_2=g_\theta(z)$ with $z\sim p(z)$, $t\sim U(0,1)$

注意力机制

高斯核函数 $K(u)=\frac{1}{\sqrt{2\pi}}e^{-\frac{u^2}{2}}$

$f(x)=\sum_{i=1}^{n}\alpha(x,x_i)y_i$

$=\sum_{i=1}^{n}\frac{\exp(-\frac{1}{2}(x-x_i)^2)}{\sum_{j=1}^{n}\exp(-\frac{1}{2}(x-x_i)^2)}y_i$

$=\sum_{i=1}^{n}\text{softmax}\left(-\frac{1}{2}(x-x_j)^2\right)y_i$

加性注意力 $q\in R^q$, $k\in R^k$

$a(q,k)=W_v^T\tanh(W_qq+W_kk)\in R$

点积注意力 $q,k\in R^d$

$a(q,k)=q^Tk/\sqrt{d}$ 除以d是为了保持scale不变

批量化的公式: $\text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V\in R^{n\times v}$

其中 $Q\in R^{n\times d}$, $K\in R^{m\times d}$, $V\in R^{m\times v}$

n = m

单头 $(X_1,X_2,\dots X_T)$, $X_i\in R^{512}$

$T\ \overset{512}{\boxed{X}}\Rightarrow$ $Q=X\cdot W_Q$ ($\nearrow R^{T\times512}$), $K=X\cdot W_K$, $V=X\cdot W_V$ $\xRightarrow{\text{Attention}}T\ \overset{512}{\boxed{\text{output}}}$

多头, h=8

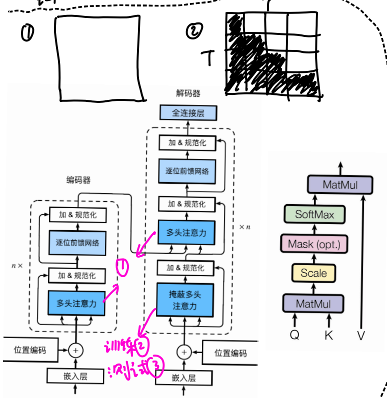
$T\ \overset{512}{\boxed{X}}\Rightarrow$

$\left\{\begin{array}{l}Q_1=X\cdot W_Q^{(1)}\\ K_1=X\cdot W_K^{(1)}\\ V_1=X\cdot W_V^{(1)}\end{array}\right.$ ($\nearrow R^{T\times64}$) $\xRightarrow{\text{Attention}}T\ \overset{64}{\boxed{\ }}$

⋮ ($\nearrow R^{T\times64}$)

$\left\{\begin{array}{l}Q8=X\cdot W_Q^{(8)}\\ K8=X\cdot W_K^{(8)}\\ V8=X\cdot W_V^{(8)}\end{array}\right.$ $\xRightarrow{\text{Attention}}T\ \overset{64}{\boxed{\ }}$

concat $\}\ T\ \overset{512}{\boxed{\ }}\times_{512}\overset{512}{\boxed{W_O}}\to T\ \overset{512}{\boxed{\text{output}}}$



位置编码 第i行第j列的位置编码: i, j 从 0开始

$P_{i,2j}=\sin\left(\frac{i}{10000^{\frac{2j}{d}}}\right)$, $P_{i,2j+1}=\cos\left(\frac{i}{10000^{\frac{2j}{d}}}\right)$

$$\begin{pmatrix}\cos(\delta w_j)&\sin(\delta w_j)\\-\sin(\delta w_j)&\cos(\delta w_j)\end{pmatrix}\begin{pmatrix}P_{i,2j}\\P_{i,2j+1}\end{pmatrix}=\begin{pmatrix}P_{i+\delta,2j}\\P_{i+\delta,2j+1}\end{pmatrix}$$

Bandit

遗憾值 $R_n=\underbrace{n\max_{a}\mu_a}_{\text{最优回报}}-\underbrace{E\left[\sum_{t=1}^{n}X_t\right]}_{\text{当前策略实际回报}}$

遗憾值 ↓ 越小越好

Stochastic bandit

每个臂的真期望: $\mu_a(\nu)=\int_{-\infty}^{\infty}x\,dP_a(x)$

记 $\mu^*(\nu):=\max_{a}\mu_a(\nu)$

对任意策略π, 目标函数为遗憾值最小

$R_n(\pi,\nu):=n\mu^*(\nu)-E\left[\sum_{t=1}^{n}X_t\right]$

希望 $\lim_{n\to\infty}\frac{R_n(\pi,\nu)}{n}=0$, $\forall\nu\in\mathcal{E}$

或 $R_n(\pi,\nu)\le Cn^P$, $\forall\nu\in\mathcal{E}$

其中 $C>0$, $P<1$

这样 $\lim_{n\to\infty}\frac{R_n(\pi,\nu)}{n}\le\lim_{n\to\infty}\frac{C\cdot n^P}{n}\to0$

其中P不可能低于 $\frac{1}{2}$

遗憾值分解 最优Arm与当前Arm的gap

记 $\Delta_a(\nu):=\mu^*(\nu)-\mu_a(\nu)$

记 $T_a(t)=\sum_{s=1}^{t}I(A_s=a)$, 即使用 Arm a 的次数

则 $R_n=\sum_{a\in A}\Delta_aE[T_a(n)]$

ETC: Explore then Commit

m是超参数

Explore: 每个Arm都尝试 m 次

①太小的话会选出次优臂

②太大的话会花费太多时间在探索上

Commit: 选则 Explore 阶段 μ 最大的

$$A_t=\begin{cases}(t\bmod k)+1, & \text{if } t\le mk\\ \underset{i}{\arg\max}\,\hat{\mu}_i(mk), & \text{if } t>mk\end{cases}$$

其中, $\hat{\mu}_i(t):=\frac{1}{T_i(t)}\sum_{s=1}^{t}I(A_s=i)X_s$

$T_i(t)$ 就等于 m

当 bandit 是 1-subgaussian, 且 $1\le m\le\frac{n}{k}$

则有 $R_n\le\underbrace{m\sum_{i=1}^{k}\Delta_i}_{\text{Explore 时的Regret}}+(n-mk)\sum_{i=1}^{k}\Delta_i\underbrace{\exp\left(-\frac{m\Delta_i^2}{4}\right)}_{\text{来自 Subguassian}}$

当 k=2 时, 令 $\Delta_1=0$, $\Delta_2=\Delta$, 则 $R_n\le m\Delta+(n-2m)\Delta\exp(-m\Delta^2/4)$

取 $m=\max\left\{1,\left\lceil\frac{4}{\Delta^2}\log\left(\frac{n\Delta^2}{4}\right)\right\rceil\right\}$, 则可得 $R_n\le\min\left\{n\Delta,\Delta+\frac{4}{\Delta}\left(1+\max\left\{0,\log\left(\frac{n\Delta^2}{4}\right)\right\}\right)\right\}$

进一步得 $R_n\le\Delta+C\sqrt{n}$

$\approx C\cdot n^{\frac{1}{2}}$ 似乎是最小的P

若 Δ 未知, 则最优的界为 $R_n=O(n^{\frac{2}{3}})$

若 ETC 可以不断调整 m, 则 ETC 略微更好

若 ETC 对 Δ 未知, 则 ETC 通常比 UCB 更差

Upper Confidence Bound, UCB

若 X_t 服从 μ=1 的 1-subgaussian 分布

令 $\hat{\mu}=\frac{1}{n}\sum_{t=1}^{n}X_t$, 则 $P\left(\mu\ge\hat{\mu}+\sqrt{\frac{2\log(\frac{1}{\delta})}{n}}\right)\le\delta$, for all $\delta\in(0,1)$

定义 UCB 指标; 其中 $T_i(t-1)$ 是 t 时刻前用 arm i 的次数

$$UCB_i(t-1,\delta)=\begin{cases}\infty & \text{if } T_i(t-1)=0\\ \hat{\mu}_i(t-1)+\sqrt{\frac{2\log(\frac{1}{\delta})}{T_i(t-1)}} & \text{o.w.}\end{cases}$$

修正版, 不带超参数 δ

$\hat{\mu}_i(t-1)+\sqrt{\frac{2\log f(t)}{T_i(t-1)}}$

$f(t)=1+t\log^2(t)$

```
算法: Input: $k,\delta$, for $t=1,\dots,n$,
    Choose $A_t:=\underset{i}{\arg\max}\,UCB_i(t-1,\delta)$
    Observe reward $X_t$ and update $UCB_s$
```

比 ETC 的优势:

a. 不需要事先知道 Suboptimality gap

b. 当超过两个臂时, 表现更好

c. 算法设计可以不依赖于 horizon n.

Markov Decision Tree, MDP

状态空间、动作空间、状态转移函数, 奖励函数, 折扣因子

随机策略: $\pi(a|s)=P(A=a|S=s)\in[0,1]$

确定策略: $\pi(a|s)=\begin{cases}1, & \text{if } \mu(s)=a\\ 0, & \text{o.w.}\end{cases}$

回报: 奖励的总和 $U_t=R_t+R_{t+1}+\dots+R_n$

折扣回报: $U_t=R_t+\gamma R_{t+1}+\gamma^2R_{t+2}+\gamma^3R_{t+3}+\cdots$, $\gamma\in[0,1]$

动作价值函数: ⇢ 依赖于 S_t, a_t 和 π

$Q_\pi(S_t,a_t)=E_{S_{t+1},A_{t+1},\dots,S_n,A_n}[U_t|S_t=s_t,A_t=a_t]$

最优动作价值函数: 消除策略π的影响, 只评价 S_t 和 a_t 的好坏

$Q^*(s_t,a_t)=\max_{\pi}Q_\pi(s_t,a_t)\xRightarrow{\text{定义①}}\pi^*(a|s)=\begin{cases}1 & a=\underset{a'}{\arg\max}\,Q^*(s,a')\\ 0 & \text{o.w}\end{cases}$

状态价值函数: 消除动作 A_t 的影响, 只依赖于 π 和 S_t

$V_\pi(S_t):=E_{A_t\sim\pi(\cdot|s_t)}[Q_\pi(s_t,A_t)]\overset{\text{或}}{=}\sum_{a\in A}\pi(a|s_t)Q_\pi(s_t,a)$

最优状态价值函数, 消除π和 A_t 的影响, 只依赖于 S_t

$V^*(s_t)=E_{A\sim\pi^*(\cdot|s)}Q^*(s,A)=\max_{a}Q^*(s,a)$

定义② $\pi^*=\underset{\pi}{\arg\max}\,V_\pi(s)$, $\forall s\in S$

$V_\pi(S_t)$ 也可写成 $E_{A_t,S_{t+1},A_{t+1}\cdots S_n,A_n}[U_t|S_t=s_t]$

贝尔曼方程 Bellman equation

$V_\pi(s_t)=E_\pi[R_t+\gamma V_\pi(S_{t+1})|S_t=s_t]$

$$\left\{\begin{array}{l}V_\pi(s)=\sum_{a}\pi(a|s)\left\{\sum_{s'}P(s'|s,a)\cdot r(s,a,s')+\sum_{s'}P(s'|s,a)\cdot\gamma\cdot V_\pi(s')\right\}\\ V_\pi(s)=\underset{a\sim\pi(\cdot|s)}{E}[Q_\pi(s,a)]\\ Q_\pi(s,a)=\sum_{s'}P(s'|s,a)\cdot r(s,a,s')+\sum_{s'}P(s'|s,a)\cdot\gamma\cdot V_\pi(s')\end{array}\right.$$

最优贝尔曼方程：

$$\pi^*(s)=\arg\max_a Q^*(s,a) \Rightarrow \pi^*(a|s)=\begin{cases}1 & a=\arg\max_{a'} Q^*(s,a') \\ 0 & o.w\end{cases}$$

$$V^*(s)=\max_a Q^*(s,a)$$

$$\begin{cases}Q^*(s,a)=\sum_{s'}P(s'|s,a)r(s,a,s')+\sum_{s'}P(s'|s,a)\cdot\gamma\cdot V^*(s') \\ \quad =\sum_{s'}P(s'|s,a)r(s,a,s')+\sum_{s'}P(s'|s,a)\cdot\gamma\cdot\max_{a'}Q^*(s',a')\end{cases}$$

基础算法：①随机生成策略 π，②计算价值函数 V_π

③更新策略 $\pi(s)=\arg\max_a \sum_{s'}P(s'|s,a)[r(s,a,s')+\gamma V_\pi(s')]$ 状态转移已知

④重复②和③至收敛，返回确定性策略 π

计算状态价值函数 V_π

$$\vec{V}_\pi^{S\times1}=\tilde{R}_\pi^{S\times1}+\gamma\cdot\tilde{P}_\pi^{S\times S}\cdot\vec{V}_\pi^{S\times1} \Rightarrow \vec{V}_\pi=(I-\gamma\cdot\tilde{P}_\pi)^{-1}\tilde{R}_\pi$$

例：

$$P(s'|s,a=1)=\begin{array}{c|ccc} & & s' & \\ \hline s & 0.8 & 0.1 & 0.1 \\ & 0.7 & 0.2 & 0.1 \\ & 0.6 & 0.2 & 0.2\end{array},\quad P(s'|s,a=2)=\begin{array}{c|ccc} & & s' & \\ \hline s & 0.1 & 0.6 & 0.3 \\ & 0.1 & 0.8 & 0.1 \\ & 0.1 & 0.4 & 0.5\end{array}$$

$$T(s,a,s')\equiv r(s)=\begin{pmatrix}1\\0\\-1\end{pmatrix}，则\ \tilde{P}_\pi(s'|s)=\sum_{a\in A}\pi(a|s)\cdot P(s'|s,a)$$

$$若\ \pi(a|s)=\begin{cases}\frac{1}{2} & a=1 \\ \frac{1}{2} & a=2\end{cases},\ \forall s，则\ \tilde{P}_\pi(s'|s)=0.5\times\ \ +0.5\times$$

$$\tilde{R}_\pi(s)=\sum_{a\in A,s'\in S}\pi(a|s)\cdot P(s'|s,a)\cdot T(s,a,s')=\begin{cases}\sum_{a\in A}\pi(a|s)\cdot r(s,a) & 若不依赖于s'，但依赖于a \\ r(s) & 若不依赖于s'和a\end{cases}$$

若状态转移未知：

$$\begin{cases}动作价值函数建模\begin{cases}\text{Q-Learning}\begin{cases}\text{DQN}(连续)\\ 离散\end{cases} & \text{off-policy}异策略 \\ \text{SARSA, on-policy} & 同策略\end{cases} \\ 策略函数建模\end{cases}$$

最优贝尔曼方程：$Q^*(s_t,a_t)=E_{S_{t+1}\sim p(\cdot|s_t,a_t)}[R_t+\gamma\cdot\max_A Q^*(S_{t+1},A)|S_t=s_t,A_t=a_t]$

定义损失函数 $L(w)=\frac{1}{2}[Q(s_t,a_t;w)-\hat{y}_t]^2$，其中 $\hat{y}_t=r_t+\gamma\max_a\cdot Q(s_{t+1},a;w_{now})$

梯度为 $\nabla_w L(w)=(\hat{q}_t-\hat{y}_t)\nabla_w Q(s_t,a_t;w)$，其中 $\hat{q}_t=Q(s_t,a_t;w_{now})$

Q-Learning 行为策略和目标策略不同

①采样：$a_t=\begin{cases}\arg\max_a Q^{(t-1)}(s_t,a) & \text{with } Pr\ 1-\varepsilon \\ \text{uniformly sampling} & \text{with } Pr\ \varepsilon\end{cases}$

②更新：$Q^{(t)}(s_t,a_t)=(1-\alpha)Q^{(t-1)}(s_t,a_t)+\alpha\hat{y}_t$

其中 $\hat{y}_t=r_t+\gamma\cdot\max_a Q^{(t-1)}(s_{t+1},a)$

③返回最后的动作价值函数 $Q^{(T)}$

SARSA On policy，不能使用经验回放

$\hat{y}_t=r_t+\gamma\cdot Q^{(t-1)}(s_{t+1},\tilde{a}_{t+1})$，$\tilde{a}_{t+1}\sim\pi_{now}(\cdot|s^{(t+1)})$ Q-Learning有max，而SARSA没有

梯度：$\nabla Ax=A^T$，$\nabla x^TA=A$，$\nabla x^TAx=(A+A^T)x$，$\nabla\|x\|^2=2x$ 清除梯度：x.grad_zero_()

线性回归解析解：$\hat{w}=(X^TX)^{-1}X^TY$，二元交叉熵损失 $-y_i\cdot\log(p(y_i))-(1-y_i)\cdot\log(1-p(y_i))$

激活函数性质①连续可导（允许少数点不可导）的非线性函数②导函数尽量简单③导函数值域合适

Logistic：$\sigma(x)=\frac{1}{1+e^{-x}}$，$\sigma'(x)=\sigma(x)\cdot(1-\sigma(x))$

Tanh：$\tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$，$\tanh'(x)=1-\tanh^2(x)$

$\tanh(x)=2\sigma(2x)-1$

$\text{ReLU}(x)=\begin{cases}x & x\geq0 \\ 0 & x<0\end{cases}=\max\{0,x\}$ ①计算高效，②单侧抑制，宽兴奋边界 ③稀疏性 ④缓解梯度消失

$\text{LeakyReLU}(x)=\begin{cases}x & x\geq0 \\ \gamma x & x<0\end{cases}$，$\gamma$可取0.01

$\text{Softplus}(x)=\log(1+e^x)$，求导后为$\sigma(x)$

$\text{Maxout}(x)=\max_k z_k$，其中 $z_k=W_k^Tx+b_k$，

前馈神经网络

$z^{(l)}=W^{(l)}a^{(l-1)}+b^{(l)}$

$a^{(l)}=f_l(z^{(l)})$

反向传播

$$\delta^{(l)}=\frac{\partial L(y,\hat{y})}{\partial z^{(l)}}=\frac{\partial a^{(l)}}{\partial(z^{(l)})}\cdot\frac{\partial z^{(l+1)}}{\partial a^{(l)}}\cdot\frac{\partial L(y,\hat{y})}{\partial z^{(l+1)}}$$

$$=\text{diag}(f_l'(z^{(l)}))\cdot(W^{(l+1)})^T\cdot\delta^{(l+1)}=f_l'(z^{(l)})\odot((W^{(l+1)})^T\delta^{(l+1)})$$

$$\frac{\partial L(y,\hat{y})}{\partial W^{(l)}}=\delta^{(l)}(a^{(l-1)})^T,\quad\frac{\partial L(y,\hat{y})}{\partial b^{(l)}}=\delta^{(l)}$$

MSE：$L(y,\hat{y})=\frac{1}{2}(y-\hat{y})^2$ Logit：$L(y,\hat{y})=-y\cdot\hat{y}+\log(1+e^{\hat{y}})$

AdaGrad $G_t=\sum_{s=1}^t g_s\odot g_s$

$\theta_{t+1}=\theta_t-\frac{\alpha}{\sqrt{G_t+\varepsilon}}\odot g_t$

RMSprop $G_t=\beta G_{t-1}+(1-\beta)g_t\odot g_t$

$=(1-\beta)\sum_{s=1}^t\beta^{t-s}g_s\odot g_s$，$\beta$取0.9

动量法 $\Delta\theta_t=\rho\Delta\theta_{t-1}-\alpha g_t=-\alpha\sum_{s=1}^t\rho^{t-s}g_s$

Adam $M_t=\beta_1M_{t-1}+(1-\beta_1)g_t$，$G_t=\beta_2G_{t-1}+(1-\beta_2)g_t\odot g_t$

动量法 + RMSprop

$$\Delta\theta_t=-\frac{\alpha}{\sqrt{\hat{G}_t+\varepsilon}}\hat{M}_t$$

随机初始化 $N(0,\sigma^2)$或$[-r,r]$，$r=\sqrt{3\sigma^2}$

$a^{(l)}=f(\sum_{i=1}^{M_{l-1}}w_i^{(l)}a_i^{(l-1)})$，若w和a均值为0，且i.i.d. 则 $\text{Var}(a^{(l)})=M_{l-1}\text{Var}(w_i^{(l)})\text{Var}(a_i^{(l-1)})$，因此 $\text{Var}(w_i^{(l)})=\frac{1}{M_{l-1}}$

考虑反向传播，则 $\text{Var}(w_i^{(l)})=\frac{1}{M_l}$，综合为 $\text{Var}(w_i^{(l)})=\frac{2}{M_l+M_{l+1}}$. 因此Xavier对Logistic：$r=4\sqrt{\frac{6}{M_{l-1}+M_l}}$，$\sigma^2=16\times\frac{2}{M_{l-1}+M_l}$. 对Tanh：$r=\sqrt{\frac{6}{M_{l-1}+M_l}}$ $\sigma^2=\frac{2}{M_{l-1}+M_l}$

He初始化 ReLU激活时，$r=\sqrt{\frac{6}{M_{l-1}}}$，$\sigma^2=\frac{2}{M_{l-1}}$ 最小最大规一化：$\hat{x}_n=\frac{x_n-\min_ix_i}{\max_ix_i-\min_ix_i}$ 标准化 $\hat{x}_n=\frac{x_n-\mu}{\sigma}$

BN $\hat{z}^{(l)}=\frac{z^{(l)}-E[z^{(l)}]}{\sqrt{\text{Var}(z^{(l)})+\varepsilon}}\cdot\gamma+\beta$ LN $\hat{z}^{(l)}=\frac{z^{(l)}-\mu^{(l)}}{\sqrt{\sigma^{(l)2}+\varepsilon}}\cdot\gamma+\beta$，$\mu^{(l)}=\frac{1}{M_l}\sum_{i=1}^{M_l}z_i^{(l)}$，$\sigma^{(l)2}=\frac{1}{M_l}\sum_{i=1}^{M_l}(z_i^{(l)}-\mu^{(l)})^2$

卷积：$y_t=\sum_{k=1}^K w_kx_{t-k+1}$，$y_{ij}=\sum_{u=1}^U\sum_{v=1}^V w_{uv}x_{i-u+1,j-v+1}$ 互相关 $y_{ij}=\sum_{u=1}^U\sum_{v=1}^V w_{uv}x_{i+u-1,j+v-1}$ $Y_{M-U+1,N-V+1}=\underset{卷积}{W\otimes X}=\underset{互相关}{\text{rot}180(W_{U\times V})*X_{M\times N}}$

卷积后的神经元数量 $\frac{M-K+2P}{S}+1$

交换性 对 X两端各补 U-1和V-1个零，得到 $\tilde{X}_{M+2U-2,N+2V-2}$，则 $W\otimes\tilde{X}=X\otimes\tilde{W}$

$Y=W_{U\times V}\otimes X_{M,N}$，$f(Y)\in R$，则 $\frac{\partial f(Y)}{\partial w_{uv}}=\sum_{i=1}^{M-U+1}\sum_{j=1}^{N-V+1}\frac{\partial y_{ij}}{\partial w_{uv}}\frac{\partial f(Y)}{\partial y_{ij}}=\sum_{i=1}^{M-U+1}\sum_{j=1}^{N-V+1}\frac{\partial f(Y)}{\partial y_{ij}}x_{u+i-1,v+j-1}$ 即 $\frac{\partial f(Y)}{\partial W}=\frac{\partial f(Y)}{\partial Y}\otimes X$

$Z^p=\sum_{d=1}^D W^{p,d}\otimes X^d+b^p$，$Y^p=f(Z^p)$，f用ReLU，参数个数：$P\times D\times U\times V+P$

反卷积

$CX=Y$，$X=C^TY$

stride=1 时，$4\xrightarrow{3}2$

$4=o'=i'+(k-1)=2+(3-1)$

stride >1 时，$5\xrightarrow{3}2$

$5=o'=s(i'-1)+k=2(2-1)+3$

RNN $h_t=f(\overbrace{Uh_{t-1}+Wx_t+b}^{z_t})$，f为Logistic，Tanh或ReLU

$$\delta_{t,k}\triangleq\frac{\partial L_t}{\partial z_k}=\frac{\partial h_k}{\partial z_k}\frac{\partial z_{k+1}}{\partial h_k}\frac{\partial L_t}{\partial z_{k+1}}=\text{diag}(f'(z_k))U^T\delta_{t,k+1}$$

梯度消失

$$\delta_{t,k}=\prod_{\tau=k}^{t-1}(\text{diag}(f'(z_\tau))U^T)\delta_{t,t}$$

$$\approx\gamma^{t-k}\delta_{t,t}\begin{cases}\gamma>1，则\ \gamma^{t-k}\to\infty & 方案①权重衰减，为参数加惩罚②梯度截断 \\ \gamma<1，则\ \gamma^{t-k}\to0 & 方案：h_t=h_{t-1}+g(x_t,h_{t-1};\theta)\end{cases}$$

$$\frac{\partial L_t}{\partial U}=\sum_{k=1}^t\delta_{t,k}h_{k-1}^T \qquad \frac{\partial L_{total}}{\partial U}=\sum_{t=1}^T\sum_{k=1}^t\delta_{t,k}h_{k-1}^T$$

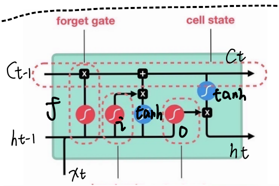
LSTM

$\tilde{C}_t=\tanh(W_cx_t+U_ch_{t-1}+b_c)$

$i_t=\sigma(W_ix_t+U_ih_{t-1}+b_i)$

$C_t=f_t\odot C_{t-1}+i_t\odot\tilde{C}_t$

$h_t=o_t\odot\tanh(C_t)$

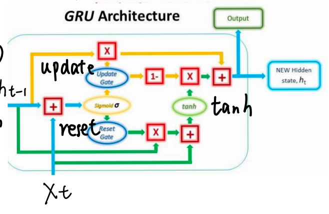


GRU $h_t=z_t\odot h_{t-1}+(1-z_t)\odot\tilde{h}_t$

update: $z_t=\sigma(W_zx_t+U_zh_{t-1}+b_z)$

reset: $r_t=\sigma(W_rx_t+U_rh_{t-1}+b_r)$

$\tilde{h}_t=\tanh(W_hx_t+U_h(r_t\odot h_{t-1}))+b_h$



loss.backward()