GAN: Meeting #3

Due on May , 07 2019 at 11:00pm

Student Zhaokun Zhou Section A

Zhaokun Zhou

Problem 1

一.给定四个公式的证明
符号说明：
l-1层有k个神经元，l层有j个神经元，设l+1层有m个神经元
${W_{jk}}^{(l)}$ 是第L-1层的第个K神经元到第L层第J个神经元的权值
${b_j}^{(l)}$ 是第L层的第J个神经元的偏置（在此次推导中设每层偏置量相同）
${a_j}^{(l)}$ 是第L层的第J个神经元的激活函数输出值
${\delta_j}^{(l)}$ 定义为第L层的第J个神经元的误差值
${z_j}^{(l)}$ 定义为第L层的第J个神经元的输入
J为代价函数

$$ {a_j}^{(l)} = \sigma\left(\sum_k {W_{jk}}^{(\mathrm{l})} {a_k}^{(\mathrm{l-1})} + {b_j}^{(\mathrm{l})}\right) \tag{1} $$

$$ {z_j}^{(l)} = \sum_k {W_{jk}}^{(l)} {a_j}^{(l-1)} + {b_j}^{(l)} \tag{2} $$

将式1转化成矩阵形式：

$$ a^{(l)} = \sigma\left(\sum_k W^{(l)} a^{(l-1)} + b^{(l)}\right) \tag{3} $$

其中$a^{(l)}$是j行列向量，$W^{(l)}$是j行k列矩阵，$a^{(l-1)}$ 是k 行列向量，$b^{(l)}$是j行列向量。
将式2转化成矩阵形式：

$$ z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)} \tag{4} $$

结合函数链式求导法则与上述公式，得到以下四个公式：
公式一:

$$ {\delta_j}^{(l)} = \frac{\partial J}{\partial {z_j}^{(l)}} = \frac{\partial J}{\partial {a_j}^{(l)}} \cdot \frac{\partial {a_j}^{(l)}}{\partial {z_j}^{(l)}} = \frac{\partial J}{\partial {a_j}^{(l)}} \cdot \sigma'\left({z_j}^{(l)}\right) \tag{5} $$

将式5转化成矩阵形式：

$$ \delta^{(l)} = \nabla_a J \odot \sigma'\left(z^{(l)}\right) \tag{6} $$

其中$\delta^{(l)}$为j行1列，$\nabla_a C$为j行1列，$\sigma'\left(z^{(l)}\right)$为j行1列。

公式二：
已知$\delta^{(l+1)}$求$\delta^{(l)}$：

$$ \delta^{(l)} = \left(W^{l+1}\right)^T \delta^{(l+1)} \odot \sigma'\left(z^{(l)}\right) \tag{7} $$

设l+1层有m个神经元，l层有j个神经元，推导如下：
第l+1层第m个神经元的偏差$\delta_j{}^{(l+1)}$对第l层第j个神经元的偏差$\delta_j^{(l)}$的影响：

$$\delta_j(m)^{(l)} = \frac{\partial J}{\partial z_m^{(l+1)}} \cdot \frac{\partial z_m^{(l+1)}}{\partial z_j^{(l)}} = \delta_j{}^{(l+1)} \cdot W_{mj}^{(l+1)} \cdot \sigma'\left(z_j{}^{(l)}\right) \tag{8}$$

整个第l+1层对第l层第j个神经元的偏差$\delta_j^{(l)}$的影响：

$$\delta_j^{(l)} = \sum_m \frac{\partial J}{\partial z_m^{(l+1)}} \cdot \frac{\partial z_m^{(l+1)}}{\partial z_j^{(l)}} = \delta_j{}^{(l+1)} \cdot W_{mj}^{(l+1)} \cdot \sigma'\left(z_j{}^{(l)}\right) \tag{9}$$

把上式归纳成整个l层的偏差：

$$\delta^{(l)} = \sum_j \sum_m \frac{\partial J}{\partial z_m^{(l+1)}} \cdot \sum_j \frac{\partial z_m^{(l+1)}}{\partial z_j^{(l)}} = \left(W^{l+1}\right)^T \delta^{(l+1)} \cdot \sigma'\left(z^{(l)}\right) \tag{10}$$

公式三：
基于以上公式，对权重求偏导得到：

$$\frac{\partial J(W,b)}{\partial W_{jk}{}^{(\mathrm{l})}} = \delta_j{}^{(\mathrm{l})} a_k{}^{(\mathrm{l}-1)} \tag{11}$$

公式四：
对偏置求偏导得到：

$$\frac{\partial J(W,b)}{\partial b_j{}^{(l)}} = \delta_j{}^{(l)} \tag{12}$$

将上述11式重写为矩阵形式：

$$\nabla_{W^{(l)}} J(W,b) = \delta^{(l)} a^{(l-1)^T} \tag{13}$$

将上述12式重写为矩阵形式：

$$\nabla_{b^{(l)}} J(W,b) = \delta^{(l)} \tag{14}$$

完整的迭代小批量随机梯度下降算法：

1.设定$\triangle W^{(l)} := 0, \triangle b^{(l)} := 0$ 为误差积累项
2.for i = 1:m
2a. 使用BP算法计算$\nabla_{W^{(l)}} J(W,b), \nabla_{b^{(l)}} J(W,b)$
2b. 设置$\nabla W^{(l)} := \triangle W^{(l)} + \nabla_{W^{(l)}} J(W,b)$
2c. 设置$\triangle b^{(l)} := \triangle b^{(l)} + \nabla_{b^{(l)}} J(W,b)$
3.更新参数

$$W^{(l)} := W^{(l)} - \alpha \left[(\frac{1}{m}) \triangle W^{(l)} + \lambda W^{(l)} \right]$$

$$b^{(l)} := b^{(l)} - \alpha \left[(\frac{1}{m}) \triangle b^{(l)} \right]$$

Problem 1 continued on next page...

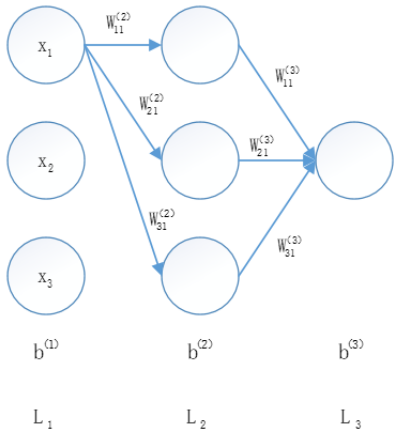

Figure 1: 3-layer MLP

其中α为学习率。

代价函数：

$$J(W,b)=\left[\frac{1}{m}\sum_{1}^{m}(\frac{1}{2}\left\|h_{W,b}(x^{(i)}-y^{(i)})\right\|^{2})\right]+\frac{\lambda}{2}\sum_{l=1}^{2}\sum_{j=1}^{S_l}\sum_{k=1}^{S_{l+1}}{W_{jk}^{(l)}}^2 \tag{15}$$

实例计算：

已知数值的变量有：$a_j^{(l)},x_i,y_i,W_{jk}^{(l)}$
设激活函数为logistic sigmoid函数，用$\sigma(x)$表示

$$\sigma'(z_j^l)=\frac{\exp^x}{(1+\exp^x)^2}=\frac{1}{(1+\exp^x)}\left(1-\frac{1}{(1+\exp^x)}\right)=a_i^l\left(1-a_i^l\right)$$

那么第三层权值参数的梯度计算为：

$$\frac{\partial J(W,b)}{\partial {W_{11}}^{(3)}}=\left[\frac{1}{m}\sum_{1}^{m}\frac{\partial J(W,b;x^i,y^i)}{\partial W_{11}^{(3)}}\right]+\lambda {W_{11}}^{(3)}$$

$$\frac{\partial J(W,b;x^i,y^i)}{\partial W_{11}^{(3)}}=\delta_1^{(3)}\cdot a_1^2=\frac{\partial J}{\partial {a_1}^{(3)}}\cdot\frac{\partial {a_1}^{(3)}}{\partial {z_1}^{(3)}}\cdot a_1^2=\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_1}^{(2)}$$

$$\frac{\partial J(W,b)}{\partial {W_{11}}^{(3)}}=\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_1}^{(2)}+\lambda {W_{11}}^{(3)}$$

同理：

$$\frac{\partial J(W,b)}{\partial {W_{12}}^{(3)}}=\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_2}^{(2)}+\lambda {W_{12}}^{(3)}$$

$$\frac{\partial J(W,b)}{\partial {W_{13}}^{(3)}}=({a_1}^{(3)}-y^i)\cdot {a_1}^{(3)}(1-{a_1}^{(3)})\cdot {a_3}^{(2)}+\lambda {W_{13}}^{(3)}$$

第二层权值参数：

$$\begin{aligned}\frac{\partial J(W,b)}{\partial {W_{11}}^{(2)}} &= \left[\frac{1}{m}\sum_{1}^{m}\frac{\partial J(W,b;x^i,y^i)}{W_{11}^{(2)}}\right] + \lambda {W_{11}}^{(3)} = {W_{11}}^{(3)}\delta_1^{(3)}\cdot {a_1}^{(2)}\left(1-{a_1}^{(2)}\right)\cdot x_1 \\ &= {W_{11}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_1}^{(2)}\left(1-{a_1}^{(2)}\right)\cdot x_1 + \lambda {W_{11}}^{(2)}\end{aligned}$$

同理：

$$\frac{\partial J(W,b)}{\partial {W_{21}}^{(2)}} = {W_{12}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_2}^{(2)}\left(1-{a_2}^{(2)}\right)\cdot x_1 + \lambda {W_{21}}^{(2)}$$

$$\frac{\partial J(W,b)}{\partial {W_{31}}^{(2)}} = {W_{13}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_3}^{(2)}\left(1-{a_3}^{(2)}\right)\cdot x_1 + \lambda {W_{31}}^{(2)}$$

$$\frac{\partial J(W,b)}{\partial {W_{12}}^{(2)}} = {W_{11}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_1}^{(2)}\left(1-{a_1}^{(2)}\right)\cdot x_2 + \lambda {W_{12}}^{(2)}$$

$$\frac{\partial J(W,b)}{\partial {W_{22}}^{(2)}} = {W_{12}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_2}^{(2)}\left(1-{a_2}^{(2)}\right)\cdot x_2 + \lambda {W_{22}}^{(2)}$$

$$\frac{\partial J(W,b)}{\partial {W_{32}}^{(2)}} = {W_{13}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_3}^{(2)}\left(1-{a_3}^{(2)}\right)\cdot x_2 + \lambda {W_{32}}^{(2)}$$

$$\frac{\partial J(W,b)}{\partial {W_{13}}^{(2)}} = {W_{11}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_1}^{(2)}\left(1-{a_1}^{(2)}\right)\cdot x_3 + \lambda {W_{13}}^{(2)}$$

$$\frac{\partial J(W,b)}{\partial {W_{23}}^{(2)}} = {W_{12}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_2}^{(2)}\left(1-{a_2}^{(2)}\right)\cdot x_3 + \lambda {W_{23}}^{(2)}$$

$$\frac{\partial J(W,b)}{\partial {W_{33}}^{(2)}} = {W_{13}}^{(3)}\left({a_1}^{(3)}-y^i\right)\cdot {a_1}^{(3)}\left(1-{a_1}^{(3)}\right)\cdot {a_3}^{(2)}\left(1-{a_3}^{(2)}\right)\cdot x_3 + \lambda {W_{33}}^{(2)}$$

Problem 2

WGAN论文理解:

此文章与作者第一篇文章《Towards Principled Methods for Training Generative Adversarial Networks》内容联系紧密。相比于原始GAN改进了四处:
1.判别器最后一层去掉sigmoid
2.生成器和判别器的loss不取log
3.每次更新判别器的参数之后把它们的绝对值截断到不超过一个固定常数c
4.不用基于动量的优化算法（包括momentum和Adam），推荐RMSProp，SGD亦可
原始GAN存在的问题:
根据GAN论文中的判别器的loss，求得最有判别器的形式；而在最有判别器的条件下，将生成器loss等式变换为P_r 与P_g 之间的JS散度，在训练Discrimator接近最有的过程中，最小化Generator的loss也会近似于最小化P_r与P_g之间的JS散度。

$$G_{loss} = 2JS(P_r||P_g) - 2log2 \tag{12}$$

当P_r与P_g的支撑集是高维空间中的低维流形时，P_r与P_g重叠部分测度为0 的概率为1。
在近似最优D下，优化G的loss等价于最小化P_r与P_g的JS散度，几乎不可能存在不可忽略的重叠，因此$DCGAN$与$MLPGAN$大部分时间的JS散度都是log2，不仅无法指示训练过程，最终还会导致生成器的梯度近似为0，梯度消失。
判别器训练得太好，生成器梯度消失，生成器loss降不下去；判别器训练得不好，生成器梯度不准，无法收敛。只有判别器训练得不好不坏才行，但是这个火候又很难把握，甚至在同一轮训练的前后不同阶段的火候都可能不一样，所以GAN难训练。
对生成样本和真实样本加噪声，让其产生重叠，可以解决训练不稳定的问题。
EM距离的性质：平滑，本质是一个路径规划的最低成本问题。

弱对偶、强对偶解决Wasserstein的求解问题

W距离的定义：

$$W(P_r, P_g) = inf_{\gamma \in \prod(P_r, P_g)} E_{(x,y)\in\gamma}[\|x - y\|] \tag{13}$$

分析上式，在联合分布函数连续的情况下将公式转变为下式:

$$W(P_r, P_g) = inf_{\gamma \in \prod(P_r, P_g)} E_{(x,y)\in\gamma} \int\int \gamma(x,y)d(x,y)dxdy \tag{14}$$

分析上式，$d(x,y)$式成本函数，形式为范数，在此论文中选取1范数，根据范数等价相容性知范数形式可变。
联合分布函数$\gamma(x,y)$,根据联合分布性质有$\int \gamma(x,y)dy = P_r(x)$且$\int \gamma(x,y)dx = P_g(y)$
原始优化为找出总成本最低的W距离即可。即MIN13式，在满足如下约束条件下：

$$\int \gamma(x,y)dy = p(x), \int \gamma(x,y)dx = q(y), \gamma(x,y) \geq 0$$

将上式离散化变成内积形式：

$$\boldsymbol{\Gamma} = \begin{pmatrix} \gamma(\boldsymbol{x}_1, \boldsymbol{y}_1) \\ \gamma(\boldsymbol{x}_1, \boldsymbol{y}_2) \\ \vdots \\ \hline \gamma(\boldsymbol{x}_2, \boldsymbol{y}_1) \\ \gamma(\boldsymbol{x}_2, \boldsymbol{y}_2) \\ \vdots \\ \hline \vdots \\ \hline \gamma(\boldsymbol{x}_n, \boldsymbol{y}_1) \\ \gamma(\boldsymbol{x}_n, \boldsymbol{y}_2) \\ \vdots \\ \hline \vdots \end{pmatrix}, \quad \boldsymbol{D} = \begin{pmatrix} d(\boldsymbol{x}_1, \boldsymbol{y}_1) \\ d(\boldsymbol{x}_1, \boldsymbol{y}_2) \\ \vdots \\ \hline d(\boldsymbol{x}_2, \boldsymbol{y}_1) \\ d(\boldsymbol{x}_2, \boldsymbol{y}_2) \\ \vdots \\ \hline \vdots \\ \hline d(\boldsymbol{x}_n, \boldsymbol{y}_1) \\ d(\boldsymbol{x}_n, \boldsymbol{y}_2) \\ \vdots \\ \hline \vdots \end{pmatrix}$$

Figure 2: 离散化内积

$$\underbrace{\left(\begin{array}{ccc|ccc|c|ccc|c} 1 & 1 & \dots & 0 & 0 & \dots & \dots & 0 & 0 & \dots & \dots \\ 0 & 0 & \dots & 1 & 1 & \dots & \dots & 0 & 0 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & \dots & 0 & 0 & \dots & \dots & 1 & 1 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\ \hline 1 & 0 & \dots & 1 & 0 & \dots & \dots & 1 & 0 & \dots & \dots \\ 0 & 1 & \dots & 0 & 1 & \dots & \dots & 0 & 1 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & \dots & 0 & 0 & \dots & \dots & 0 & 0 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \end{array}\right)}_{\boldsymbol{A}} \underbrace{\begin{pmatrix} \gamma(\boldsymbol{x}_1, \boldsymbol{y}_1) \\ \gamma(\boldsymbol{x}_1, \boldsymbol{y}_2) \\ \vdots \\ \hline \gamma(\boldsymbol{x}_2, \boldsymbol{y}_1) \\ \gamma(\boldsymbol{x}_2, \boldsymbol{y}_2) \\ \vdots \\ \hline \vdots \\ \hline \gamma(\boldsymbol{x}_n, \boldsymbol{y}_1) \\ \gamma(\boldsymbol{x}_n, \boldsymbol{y}_2) \\ \vdots \\ \hline \vdots \end{pmatrix}}_{\boldsymbol{\Gamma}} = \underbrace{\begin{pmatrix} p(\boldsymbol{x}_1) \\ p(\boldsymbol{x}_2) \\ \vdots \\ p(\boldsymbol{x}_n) \\ \vdots \\ \hline q(\boldsymbol{y}_1) \\ q(\boldsymbol{y}_2) \\ \vdots \\ q(\boldsymbol{y}_n) \\ \vdots \end{pmatrix}}_{\boldsymbol{b}}$$

Figure 3: 离散化约束

其中$\Gamma \geq 0$ W距离可以用下式描述：

Problem 2 continued on next page...

$$\min_{\Gamma} \{\langle \Gamma, D\rangle \,|A\Gamma = b, \Gamma > 0\}$$

重写上式

$$\min_{x}\{c^T x|Ax = b, x \geq 0\}$$

设存在最小值x^*使得上式成立，则$Ax^* = b$也成立，在等式两边乘$y^T \in R^m$得到$y^T Ax^* = y^T b$,此时设$y^T A \leq c^T$则$y^T Ax^* \leq c^T x^*$，也有$y^T b \leq c^T x^*$。整理上述公式变为语言描述：在条件$y^T A \leq c^T$ 下，任意$y^T b$ 总是不大于$\min_x\{c^T x|Ax = b, x \geq 0\}$。
所以有：

$$\max_{y} \left\{b^T y|A^T y \leq c\right\} \leq \min_{x} \left\{c^T x|Ax = b, x \geq 0\right\}$$

再利用Farkas引理证明上式的强对偶形式，取得等号。

$$\max_{y} \left\{b^T y|A^T y \leq c\right\} = min_x \left\{c^T x|Ax = b, x \geq 0\right\}$$

通过强对偶为W距离找对偶表达式：

$$\min_{\Gamma} \{< \Gamma, D > |A\Gamma = b, \Gamma > 0\} = max_F \left\{< b, F > |A^T F \geq D\right\}$$

处理问题是依然将其转化为离散形式：

$$\langle b, F\rangle = \sum_n p(x_n) f(x_n) + \sum_n q(x_n) g(x_n)$$

约束条件是$A^T F \leq D$

由约束条件得到:

$$\forall i, j, f(x_i) + g(y_i) \leq d(x_i, y_i)$$

$$\boldsymbol{F} = \begin{pmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_n) \\ \vdots \\ \hline g(\boldsymbol{y}_1) \\ g(\boldsymbol{y}_2) \\ \vdots \\ g(\boldsymbol{y}_n) \\ \vdots \end{pmatrix}$$

Figure 4: 神经网络拟合函数

进而得到：
$W\{p,q\} = \max_{f,g} \left\{ \int [p(x)f(x) + q(x)g(x)]dx | f(x) + g(y) \right\} \leq d(x,y)$

由$d(x,x) = 0$得到$f(x) + g(x) \leq d(x,x) + 0$,即$g(x) \leq f(x)$,所以有：

$$p(x)f(x) + q(x)g(x) \leq p(x)f(x) + q(x)(-f(x)) = p(x)f(x) - q(x)f(x)$$

再次重写W距离：

$$W\{p,q\} = \max_{f} \left[\int [p(x)f(x) + q(x)(-f(x))]dx | f(x) - f(y) \leq d(x,y) \right]$$

由上得到W的对偶形式，约束条件重写为$\|f\|_L \leq 1$,称为利普希茨约束。将上式重写为采样形式就得到WGAN论文中的具体优化公式，1则为文章中假设的值。同样K作为利普希茨常数亦可行。
算法流程图中关于梯度迭代的D为加，G迭代为减。
WGAN：先最大化去拟合Wasserstein距离，再去最小化L，来优化生成器。D也就是Critice不是为了判别，正如其意思，是为了评判，所以拿掉了二分类，变成了回归问题。目标是优化G。