

实战adadelta+weight noise

主讲人：刘勇杰

入职时间：2020.02.28





论文回顾

一般网络损失

$$L^N(\mathbf{w}, \mathcal{D}) = -\ln \Pr(\mathcal{D}|\mathbf{w}) = - \sum_{(\mathbf{x}, y) \in \mathcal{D}} \ln \Pr(y|\mathbf{x}, \mathbf{w})$$

参数解释

- $L^N(\mathbf{w}, \mathcal{D})$ 为损失函数；
- $-\sum_{(\mathbf{x}, y) \in \mathcal{D}} \ln \Pr(y|\mathbf{x}, \mathbf{w})$ 交叉熵，在给定模型参数 \mathbf{w} 与输入变量 \mathbf{x} ，预测 y 的概率；

·
·
·
·
·

论文回顾

在神经网络上进行贝叶斯推理
需要给定数据的网络权值的后验分布

$$P(\mathbf{w}|\alpha) \longrightarrow \Pr(\mathbf{w}|\mathcal{D}, \alpha)$$

如果权重 \mathbf{w} 具有依赖于
某些参数 α 的先验概率

则后验概率可以写成

$$\Pr(\mathbf{w}|\mathcal{D}, \alpha)$$

对于大多数神经网络，不能解析地计算

通过用更易处理的分布 $Q(\mathbf{w}|\beta)$

逼近 $\Pr(\mathbf{w}|\mathcal{D}, \alpha)$ 来解决这个问题

$$\mathcal{F} = - \left\langle \ln \left[\frac{\Pr(\mathcal{D}|\mathbf{w})P(\mathbf{w}|\alpha)}{Q(\mathbf{w}|\beta)} \right] \right\rangle_{\mathbf{w} \sim Q(\beta)}$$

对于满足分布为 $p(x)$ 的随机变量 x 的某些函数 g

$\langle g \rangle_{x \sim p}$ 代表了函数 g 的期望

论文回顾

将上述公式重新排列

$$\mathcal{F} = \langle L^N(\mathbf{w}, \mathcal{D}) \rangle_{\mathbf{w} \sim Q(\beta)} + D_{KL}(Q(\beta) || P(\alpha))$$

因KL散度的非负性，由Shannon 编码定理可知，等式右边第一项即为函数的下限

其意义为将D中的输入，经过神经网络的预测得到结果，神经网络权值符合分布Q (β)

由于该项随网络预测精度的提高而减小，我们将其定义

为误差损失 $L^E(\beta, \mathcal{D}) = \langle L^N(\mathbf{w}, \mathcal{D}) \rangle_{\mathbf{w} \sim Q(\beta)}$

对于等式右边的KL散度，也定义一个复杂损失

$$L^C(\alpha, \beta) = D_{KL}(Q(\beta) || P(\alpha))$$

$$L(\alpha, \beta, \mathcal{D}) = L^E(\beta, \mathcal{D}) + L^C(\alpha, \beta)$$

然后在数据集D上训练网络

使关于α和β的函数 $L(\alpha, \beta, \mathcal{D})$ 最小

·
·
·
·
·

论文回顾

现在分别讨论 α 与 β 的可能分布

先规定 $Q(\beta) = \prod_{i=1}^W q_i(\beta_i)$

即 $L^C(\alpha, \beta) = \sum_{i=1}^W D_{KL}(q_i(\beta_i) || P(\alpha))$

先看 β 的分布

假设 $Q(\beta)$ 满足(Dirac) delta distribution, 即将概率1赋值给一个特殊的权值 w , 而其他的权值赋予概率0

在该情况下, $\alpha=\beta$, 则 $L^E(\beta, \mathcal{D}) = L^N(\mathbf{w}, \mathcal{D})$

且 $L^C(\alpha, \beta) \stackrel{\sim}{=} L^C(\alpha, \mathbf{w}) = -\log P(\mathbf{w}|\alpha) \stackrel{\sim}{=} C$

再看 α 的分布

假设 α 满足Laplace distribution

则 α 取决于两个参数 $\{b, \mu\}$

$$P(\mathbf{w}|\alpha) = \prod_{i=1}^W \frac{1}{2b} \exp\left(-\frac{|w_i - \mu|}{b}\right)$$

那么

$$L^C(\alpha, \mathbf{w}) = W \ln 2b + \frac{1}{b} \sum_{i=1}^W |w_i - \mu| + C$$

$$\Rightarrow \frac{\partial L^C(\alpha, \mathbf{w})}{\partial w_i} = \frac{\text{sgn}(w_i - \mu)}{b}$$

最优解

$$\hat{\mu} = \mu_{1/2}(\mathbf{w})$$

$$\hat{b} = \frac{1}{W} \sum_{i=1}^W |w_i - \hat{\mu}|$$

论文回顾

β 仍是delta distribution, 继续讨论 α 的分布

先规定 $Q(\beta) = \prod_{i=1}^W q_i(\beta_i)$

即 $L^C(\alpha, \beta) = \sum_{i=1}^W D_{KL}(q_i(\beta_i) || P(\alpha))$

先看 β 的分布

假设 $Q(\beta)$ 满足(Dirac) delta distribution, 即将概率1赋值给一个特殊的权值 w , 而其他的权值赋予概率0

在该情况下, $\alpha=\beta$, 则 $L^E(\beta, \mathcal{D}) = L^N(\mathbf{w}, \mathcal{D})$

且 $L^C(\alpha, \beta) \approx L^C(\alpha, \mathbf{w}) = -\log P(\mathbf{w}|\alpha) + C$

假设 α 满足Gaussian distribution

则 α 取决于两个参数 $\{\mu, \sigma^2\}$

那么 $P(\mathbf{w}|\alpha) = \prod_{i=1}^W \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(w_i - \mu)^2}{2\sigma^2}\right)$

$$L^C(\alpha, \mathbf{w}) = W \ln(\sqrt{2\pi\sigma^2}) + \frac{1}{2\sigma^2} \sum_{i=1}^W (w_i - \mu)^2 + C$$

$$\Rightarrow \frac{\partial L^C(\alpha, \mathbf{w})}{\partial w_i} = \frac{w_i - \mu}{\sigma^2}$$

最优解

$$\hat{\mu} = \frac{1}{W} \sum_{i=1}^W w_i$$

$$\hat{\sigma}^2 = \frac{1}{W} \sum_{i=1}^W (w_i - \hat{\mu})^2$$



论文回顾

假设 β 是Gaussian distribution $\beta = \{\mu, \sigma^2\}$

那么对于一般的神经网络，不管是 $L^E(\beta, \mathcal{D})$ 还是其导数，都不能准确地计算，因此只能sample

$$L^E(\beta, \mathcal{D}) \approx \frac{1}{S} \sum_{k=1}^S L^N(\mathbf{w}^k, \mathcal{D})$$

$$\frac{\partial L^E(\beta, \mathcal{D})}{\partial \mu_i} = \left\langle \frac{\partial L^N(\mathbf{w}, \mathcal{D})}{\partial w_i} \right\rangle_{\mathbf{w} \sim Q(\beta)} \approx \frac{1}{S} \sum_{k=1}^S \frac{\partial L^N(\mathbf{w}^k, \mathcal{D})}{\partial w_i}$$

$$\frac{\partial L^E(\beta, \mathcal{D})}{\partial \sigma_i^2} = \frac{1}{2} \left\langle \frac{\partial^2 L^N(\mathbf{w}, \mathcal{D})}{\partial w_i^2} \right\rangle_{\mathbf{w} \sim Q(\beta)} \approx \frac{1}{2} \left\langle \left[\frac{\partial L^N(\mathbf{w}, \mathcal{D})}{\partial w_i} \right]^2 \right\rangle_{\mathbf{w} \sim Q(\beta)} \approx \frac{1}{2S} \sum_{k=1}^S \left[\frac{\partial L^N(\mathbf{w}^k, \mathcal{D})}{\partial w_i} \right]^2$$

·
·
·
·
·

论文回顾

假设 β 是Gaussian distribution, α 也为Gaussian distribution

$$\beta = \{\mu, \sigma^2\}$$

$$\alpha = \{\mu, \sigma^2\}$$

那么KL($\alpha \parallel \beta$)就化为如下公式

$$L^C(\alpha, \beta) = \sum_{i=1}^W \ln \frac{\sigma}{\sigma_i} + \frac{1}{2\sigma^2} \left[(\mu_i - \mu)^2 + \sigma_i^2 - \sigma^2 \right]$$
$$\Rightarrow \frac{\partial L^C(\alpha, \beta)}{\partial \mu_i} = \frac{\mu_i - \mu}{\sigma^2}, \quad \frac{\partial L^C(\alpha, \beta)}{\partial \sigma_i^2} = \frac{1}{2} \left[\frac{1}{\sigma^2} - \frac{1}{\sigma_i^2} \right]$$

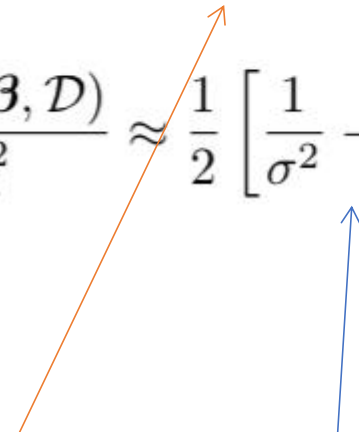
最优解

$$\hat{\mu} = \frac{1}{W} \sum_{i=1}^W \mu_i, \quad \hat{\sigma}^2 = \frac{1}{W} \sum_{i=1}^W \left[\sigma_i^2 + (\mu_i - \hat{\mu})^2 \right]$$

·
·
·

论文回顾

优化函数 以 α 和 β 都为Gaussian distribution为例

$$\frac{\partial L(\alpha, \beta, \mathcal{D})}{\partial \mu_i} \approx \frac{\mu_i - \mu}{\sigma^2} + \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \frac{1}{S} \sum_{k=1}^S \frac{\partial L^N(\mathbf{w}^k, \mathbf{x}, \mathbf{y})}{\partial w_i}$$
$$\frac{\partial L(\alpha, \beta, \mathcal{D})}{\partial \sigma_i^2} \approx \frac{1}{2} \left[\frac{1}{\sigma^2} - \frac{1}{\sigma_i^2} \right] + \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \frac{1}{2S} \sum_{k=1}^S \left[\frac{\partial L^N(\mathbf{w}^k, \mathbf{x}, \mathbf{y})}{\partial w_i} \right]^2$$


需要加入到网络训练中的weight noise

·
·
·
·

算法实现

■ 利用当前权重产生tparam_p_u与

tparam_p_ls2, beta, prior_u,

prior_s2

■ 回传损失得到梯度后，利用Beta, prior_u,prior_s2产生新的梯度

new_grads_miu, new_grads_sigma

■ adadelta算法利用new_grads_miu, new_grads_sigma对

tparam_p_u,tparam_p_ls2更新

■ 将tparam_p_u回传当前权值

tparam_p_u保存梯度对U的更新权值

tparam_p_ls2保存梯度对σ2的更新权值

prior_u

$$\frac{\partial L(\alpha, \beta, \mathcal{D})}{\partial \mu_i} \approx \frac{\mu_i - \mu}{\sigma^2} + \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \frac{1}{S} \sum_{k=1}^S \frac{\partial L^N(\mathbf{w}^k, \mathbf{x}, \mathbf{y})}{\partial w_i}$$

prior_s2

$$\frac{\partial L(\alpha, \beta, \mathcal{D})}{\partial \sigma_i^2} \approx \frac{1}{2} \left[\frac{1}{\sigma^2} - \frac{1}{\sigma_i^2} \right] + \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \frac{1}{2S} \sum_{k=1}^S \left[\frac{\partial L^N(\mathbf{w}^k, \mathbf{x}, \mathbf{y})}{\partial w_i} \right]^2$$

代码地址 <https://github.com/JianshuZhang/TAP>



算法实现

| Name | Posterior | Prior | Error | Epochs | Ratio |
|-----------------------------|--------------------------|---------------------------------|-------|--------|-------|
| Adaptive L1 | Delta | Laplace | 49.0 | 7 | — |
| Adaptive L2 | Delta | Gauss | 35.1 | 421 | — |
| Adaptive mean L2 | Delta | Gauss $\sigma^2 = 0.1$ | 28.0 | 53 | — |
| L2 | Delta | Gauss $\mu = 0, \sigma^2 = 0.1$ | 27.4 | 59 | — |
| Maximum likelihood | Delta | Uniform | 27.1 | 44 | — |
| L1 | Delta | Laplace $\mu = 0, b = 1/12$ | 26.0 | 545 | — |
| Adaptive mean L1 | Delta | Laplace $b = 1/12$ | 25.4 | 765 | — |
| Weight noise | Gauss $\sigma_i = 0.075$ | Uniform | 25.4 | 220 | — |
| Adaptive prior weight noise | Gauss $\sigma_i = 0.075$ | Gauss | 24.7 | 260 | 0.542 |
| Adaptive weight noise | Gauss | Gauss | 23.8 | 384 | 0.286 |

Q & A

有道 youdao