Reading Papers

Kangcheng Hou

Zhejiang University

May 11, 2018

Deep Probabilistic Programming[?]

- You can take a DL model and make it probabilistic. (Section 3 assign prior probability on the parameter θ and infer the posterior.)
- You can integrate the DL module in probabilistic models. (Section 4 introduced variational autoencoders.)
- Advantages from probabilistic models(Section 5.1)
- Advantages from deep models(Section 5.2)

Understanding Black-box Predictions via Influence Functions [?]

Goal: answer the question: "Why did the system make this prediction". One way to answer it is "How would the model's predictions change if we remove the training point $z_i = (x_i, y_i)$. We first assume the empirical risk is twice-differentiable and strictly convex.

- I do a summary of the paper.
- I run an experiment with MNIST dataset.

Upweighting a training point I

Instead of answer question of $\hat{\theta}_{-z} - \hat{\theta}$. We can ask question like $\hat{\theta}_{\epsilon,z} = \text{argmin}_\theta \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$. It can be proved that

$$\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})$$

This can be calculated **without retraining the model**. Note that $\hat{\theta}_{\frac{1}{n},z} = \hat{\theta}_{-z}$, we can use Taylor expansion of $\hat{\theta}$ to approximate

$$\hat{\theta}_{-z} - \hat{\theta} = -\frac{1}{n} \mathcal{I}_{\text{up, params}}(z)$$

Upweighting a training point II

. Now we know the influence of training point z on the parameters of the model θ. Next we investigate how upweighting z changes function of $\hat{\theta}$. Applying the chain rule,

$$\begin{aligned}
\mathcal{I}_{\text{up, loss}}(z, z_{\text{test}}) &= \frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon,z})}{d\epsilon}|_{\epsilon=0} \\
&= \nabla_\theta(z_{\text{test}}, \hat{\theta})^\top \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}|_{\epsilon=0} \\
&= -\nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})
\end{aligned}$$

Perturing a training input I

We want to know how pertubing a data point $(x, y) \to (x + \delta, y)$ will influence the estimated parameter. We define

$$\hat{\theta}_{\epsilon, z_\delta, -z} = \text{argmin}$$

Convergence analysis of two-layer neural networks with relu activation[?]

Questions:

- Why simple networks works very well?
- How does SGD relate to Bayesian inference?

Some references:

- https://zhuanlan.zhihu.com/p/36624193
- https://zhuanlan.zhihu.com/p/27609238

References I