Slides_Batuhan

January 27, 2019

1 Uncertainty and Deep Learning

1.1 Outline

1. Theoretical Background 1.1. Types of Uncertainty 1.2. Bayesian Inference 1.3. Approximating the Posterior 1.4. Variational Inference 1.5. Monte Carlo Dropout
2. Application
3. Case Study: Lender's Club

2 Uncertainty and Deep Learning

- Accounting for uncertainty is crucial in decision-making systems
  - Health sector, autonomous driving, reinforcement learning, asset management, ...
- Deterministic NN's: Falsely overconfident in predictions
- We need to know whether a model is certain about its output
- Our use case: Loan Allocation Decision on Lender Profitability (Kaggle Lending Club Loan Data)

2.1 1.1. Types of Uncertainty

- Epistemic Uncertainty: Uncertainty a deterministic NN can measure, like a softmax probability
- Aleatoric Uncertainty: F.Knight: 'Out of reach of measurement'?

2.2 1.2. Bayesian Inference

Generative vs. Bayesian Models

```
Generative Models
        Causal Rules:        Cause --> Effect

Inference
        Diagnostic Rules:     Cause <-- Effect
```

2.3 1.2. Bayesian Inference

$$p(\omega|X) = \frac{p(X,\omega)}{P(X)} \implies p(\omega|X) = \frac{p(X|\omega)p(\omega)}{p(X)}$$

* The degree of belief in a model: **the posterior function** $P(\omega|X)$) * The likelihood of data: **the likelihood function** $P(X|\omega)$ * Our knowledge about the data: **the prior** $P(\omega)$ * And the evidence: **the marginal likelihood** $P(X)$\

Having defined the posterior as above, the prediction on new observations x_{new} is made through model criticism on the **posterior predictive distribution**:

$$p(x_{new}|X) = \int p(x_{new}|\omega)p(\omega|X)d\omega$$

$$p(x_{new}|X) = \int p(x_{new}|\omega)p(\omega|X)d\omega$$

The ultimate problem in Bayesian Inference: $p(x_{new}|X)$ is intractable because $P(\omega|X)$ is intractable

$$p(\omega|X) = \frac{p(X|\omega)p(\omega)}{p(X)} \implies p(\omega|X) = \frac{p(X|\omega)p(\omega)}{\int p(X,\omega)d\omega}$$

... and the posterior $p(\omega|X)$ is intractable because $P(X)$ is intractable.

2.4 1.3. Approximating the Posterior

```
In [10]: from IPython.core.display import Image
         Image(filename='images/posterior_predictive.png')
```

```
Out[10]: <IPython.core.display.HTML object>
```

Different Methodologies:

- Maximum a Posteriori (MAP)
- Sampling Based Approximations: MCMC, HMC, Gibbs, Metropolian
- **Variational Inference**
- ...

2.5 1.4. Variational Inference

- Idea: Pick a distribution $q(\omega)$ that is similar to the posterior $p(\omega|x)$
- Minimize the Kullback-Leibler divergence (KL-divergence) of $q(\omega)$ to $p(\omega|x)$
- Assume that $q(\omega)$ is parametrized by 'variational parameters' θ

$$\min_{\theta} KL(q(\omega;\theta)||p(\omega|x)) \iff \min_{\theta} \mathop{\mathbb{E}}_{q(\omega|\theta)}[logq(\omega;\theta) - logp(\omega|x)]$$

- But the minimization above contains the posterior, therefore it is intractable, too

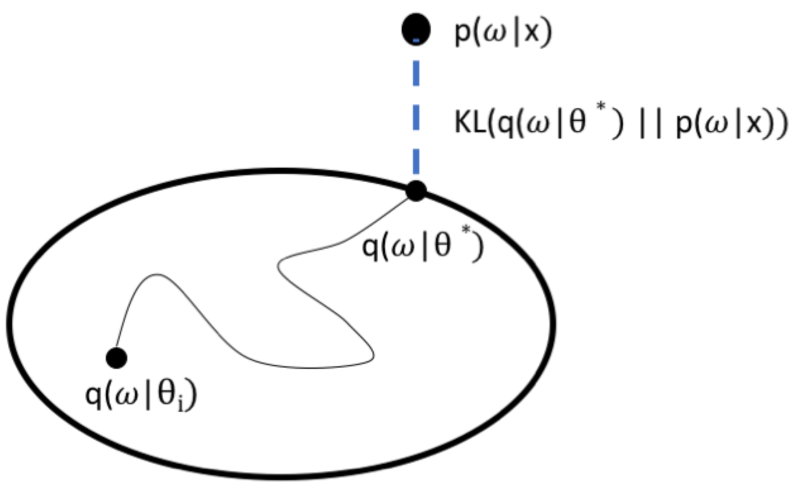
$$KL(q(\omega;\theta)||p(\omega|x)) = -\mathbb{E}[logp(x,\omega) - logq(\omega;\theta)] + \mathbb{E}\, logp(x)$$

- Minimization of the KL-divergence is the same as maximizing $ELBO(\theta) = \mathbb{E}[logp(x,\omega) - logq(\omega;\theta)]$, which is indeed a lower bound to the evidence:

$$logp(x) = log(\mathbb{E}_q[\frac{p(x,\omega)}{q(\omega;\theta)}]) \geq \mathbb{E}_q[log(\frac{p(x,\omega)}{q(\omega;\theta)})] = ELBO(\theta)$$

```
In [7]: from IPython.core.display import Image
        Image(filename='images/variational_inference.png')
```

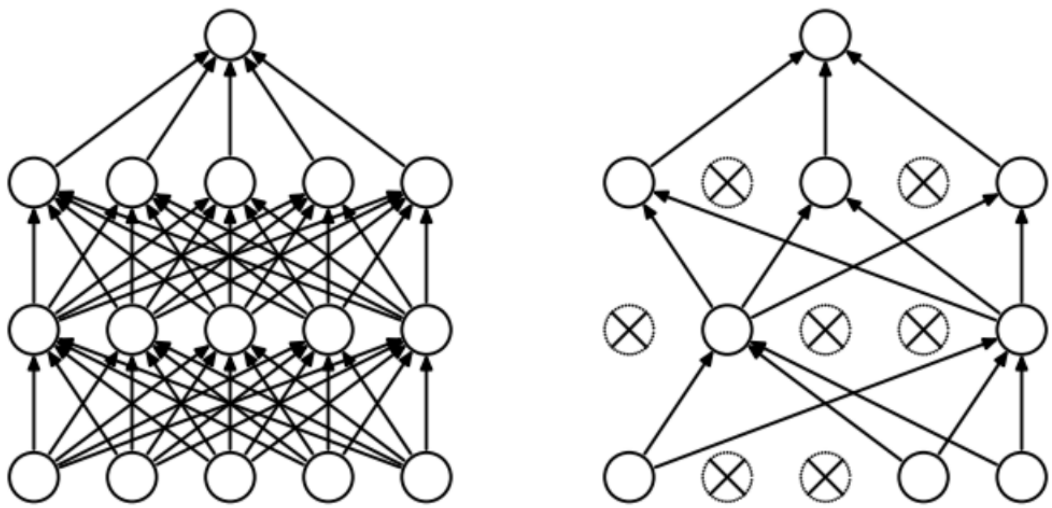
Out[7]:



2.6 1.5. Monte Carlo Dropout

```
In [2]: from IPython.core.display import Image
        Image(filename='images/dropout.png')
```

Out[2]:



(a) Standard Neural Net (b) After applying dropout.

2.7 1.5. Monte Carlo Dropout

- A computationally cheap way of obtaining Variational Inference in the setting of Gaussian Processes
- How: Applying dropout regularization not only during training time, but also during test time
- Result: A distribution of predictions for each observation. They are approximate samples from the posterior predictive distribution.
- with mean

$$\hat{\mathbb{E}}(y) = \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\omega}_t}(x)$$

- and variance

$$\hat{\mathbb{E}}(y^T y) = \tau^{-1} I + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\omega}_t}(x)^T f^{\hat{\omega}_t}(x) - \hat{\mathbb{E}}(y)^T \hat{\mathbb{E}}(y)$$

where

$$\tau = \frac{(1-p)l^2}{2N\lambda}$$

```
* 1 - p: Dropout probability
* N: number of data points
* $\lambda$: weight decay regularization term
* $l$: prior length-scale
```