INTRODUCTION TO BAYESIAN NEURAL NETS

Nicholas Orriols

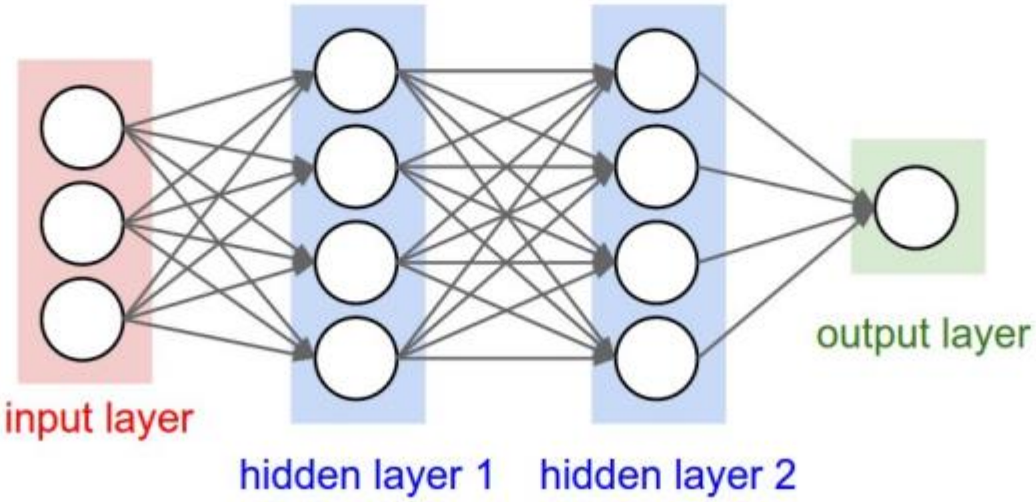
Feedforward Neural Nets

- Suppose we have data set D with inputs $D_x \subset P(R^p)$ and labels $D_y \subset P(R^q)$
- Let the n layers of a feedforward net output

$$F_o(x) = x$$

$$F_i(x) = \sigma(W_i F_{i-1}(x) + b_i)$$

- We find $\theta = \{W_i\} \cup \{b_i\}$ by backpropagation minimizing cost function (standard method)
- What if we are interested in the distribution of possible values θ?
- What about distribution of $F(x) = y$?



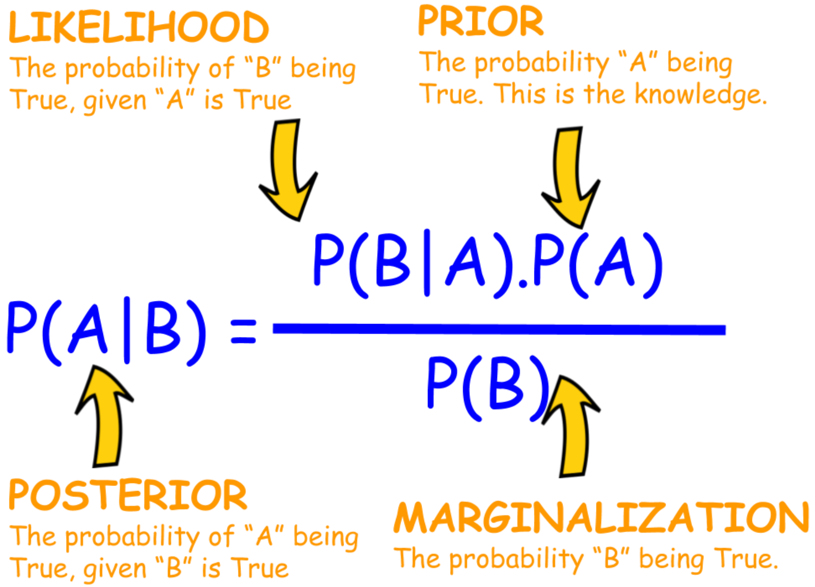
Two hidden layer neural net with single output

Formulating Posterior

- Assume we have a chosen network architecture (i.e. functional model)
- Let $p(\theta)$ be distribution of θ
- Let $p(y|x,\theta)$ be distribution of F(x)
- Assume inputs (and outputs) are independent
- By Bayes Formula, posterior is

$$\mathrm{p}(\theta|\mathrm{D}) = \frac{p\left(\mathrm{D_y}\middle|\mathrm{D_x},\theta\right)p(\theta)}{\int_{\Theta} p\left(\mathrm{D_y}\middle|\mathrm{D_x},\theta_t\right)p(\theta_t)d\theta_t}$$

- Left term in numerator is likelihood, product of distributions of F(x)
- Neither distribution is computed directly



Bayes Formula with labels

Prediction Confidence / Ensembling

- If we have $p(\theta|D)$, we can calculate

$$p(\boldsymbol{y}|\boldsymbol{x}, D) = \int_{\Theta} p(\boldsymbol{y}|\boldsymbol{x}, \theta_t) p(\theta_t|D)\, d\theta_t$$

- This gives distribution of outputs for x based on data
- Can quantify how much we trust our net's prediction(s), and the variance of predictions given by our net
- If we sample $\{\theta_i\}$ from $p(\theta)$, for any input x, we can estimate

$$\hat{\boldsymbol{y}} = \frac{1}{N} \sum_{i=1}^{N} F_{\theta_i}(\boldsymbol{x})$$

$$\hat{\Sigma}_{\boldsymbol{y}|\boldsymbol{x},D} = \frac{1}{N-1} \sum_{i=1}^{N} \left(F_{\theta_i}(\boldsymbol{x}) - \hat{\boldsymbol{y}}\right)\left(F_{\theta_i}(\boldsymbol{x}) - \hat{\boldsymbol{y}}\right)^T$$

Building BNNs

- Choose whatever network architecture (i.e. functional model) you think is best
  - As long as it has weights or activations to simulate
- Stochastic model is usually

$$\theta \sim \mathrm{p}(\theta) = \mathcal{N}(\mu, \Sigma)$$
$$y \sim \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}, \theta) = \mathcal{N}(F_\theta(\boldsymbol{x}), \Sigma)$$

- Unfortunately, no procedural method to determine priors (mu and Sigma)
- Default choices for priors are $\mu = \vec{0}, \Sigma = \sigma I$, although no theoretical justification for this
  - Does bias weight matrices toward 0, reducing scaling symmetry problem
- Choosing priors is a process of testing alternatives to find the best one(s)

Markov Chain Monte Carlo

- Goal is to sample posterior $p(\theta|D)$
- Need only numerator in

$$\mathrm{p}(\theta|\mathrm{D}) = \frac{p(\mathrm{D_y}|\mathrm{D_x}, \theta)p(\theta)}{\int_{\Theta} p(\mathrm{D_y}|\mathrm{D_x}, \theta_t)p(\theta_t)d\theta_t}$$

- Beginning with initial guess, sample from numerator
- If new point more likely wrt posterior accept; otherwise, reject with probability *p*
- Q is best chosen centered around previous point, so normal and uniform are common choices for Q
- If V(Q) is too big, rejection rate is too large
- If V(Q) is too small, highly autocorrelated samples

Algorithm 1 Metropolis-Hasting

Draw $x_0 \sim Initial$
while $n = 0$ **to** N **do**
  Draw $x' \sim Q(x|x_n)$
  $p = min\left(1, \frac{Q(x'|x_n)}{Q(x_n|x')}\frac{f(x')}{f(x_n)}\right)$
  Draw $k \sim Bernoulli(p)$
  if k **then**
    $x_{n+1} = x'$
    $n = n + 1$
  end if
end while

Variational Inference

- Approximate posterior by $q_\phi(\theta)$ with ϕ chosen to maximize closeness (minimize KL divergence)
- KL divergence is information lost by using a distribution to approximate another distribution
- Minimize KL divergence by maximizing elbow using 2nd formula
- Can use many methods to maximize ELBO
  - Most popular is stochastic variational inference
- $p(y|x,D)$ is sampled by sampling θ from $q_\phi(\theta)$ or uniform distribution

- KL divergence

$$\mathrm{D}_{KL}(\mathrm{q}_\phi||P) = \int_\Theta q_\phi(\theta_t)\, log\left(\frac{q_\phi(\theta_t)}{P(\theta_t|D)}\right) d\theta_t$$

- ELBO formula

$$\mathrm{ELBO} = \int_\Theta q_\phi(\theta_t)\, log\left(\frac{P(\theta_t, D)}{q_\phi(\theta_t)}\right) d\theta_t$$
$$= log\big(P(D)\big) - D_{KL}(q_\phi||P)$$

$$p(\boldsymbol{y}|\boldsymbol{x}, D) = \int_\Theta p(\boldsymbol{y}|\boldsymbol{x}, \theta_t)p(\theta_t|D)\, d\theta_t$$

BNN evaluation

- BNNs output a distribution of response estimates $p(y|x, D)$ for a given input, *not* a predicted response for the input
  - Most intuitive estimate from distribution is expected value
- Can compare expected values and observed responses with many metrics (eg MSE)
- What about $p(y|x, D)$? Is the BNN overconfident? Underconfident?
- Calibration curves express observed probability p^* as function of predicted probability $\hat{p}$
  - If $p^* > \hat{p}$, BNN is underconfident
  - If $p^* < \hat{p}$, BNN is overconfident

Calibration Curves: continuous

- For continuous responses, assuming test outputs are independent, we can assume

$$(\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i)^T \Sigma_{\boldsymbol{y}_i}^{-1} (\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i) \sim \chi^2_{Dim(\boldsymbol{y}_i)}$$

- To estimate Σ, use sample covariance matrix given in slide 4
- For every test input, predicted probability is chance of getting error less than or equal to the observed error
  - Observed error distance between expectation of BNN output and obs

$$\hat{p}_i = X^2_{Dim(y)}\left(\left(E_{p(y|x_i,D)}[y] - y_i\right)^{\mathsf{T}} \Sigma^{-1}_{y|x_i,D} \left(E_{p(y|x_i,D)}[y] - y_i\right) \right)$$

Formula for predicted probability associated with i[th] test input

$$p_i^* = \frac{1}{|T|} \sum_{j=1}^{|T|} I(\hat{p}_j \geq \hat{p}_i)$$

Formula for observed probability of predicted probabilities greater than or equal to i[th] predicted probability

References

- [1] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. 2020. Hands-on Bayesian Neural Networks - a Tutorial for Deep Learning Users. ACM Comput. Surv. 1, 1 (July 2020), 35 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn
- [2] Daniel Silvestro, Tobias Andermann. 2005. Prior Choice Affects Ability of Bayesian Neural Networks to Identify Unknowns. https://arxiv.org/abs/2005.04987