MCMC for Bayesian Neural Networks

Slides

This document provides a high-level overview of the topics covered during the Wednesday afternoon lecture, focusing specifically on the aspects the TAs should be aware of. Some version of the actual slides will be ready by roughly June 10th, but it is not needed for supporting the hands-on work.

Hands-on material

The practical hands-on material is provided in the notebook, with clearly outlined tasks for the students. The notebook shows a draft of the 'model solutions', but I will later prepare a version that will be shared for the students where some parts are replaced with instructions on what they should do.

The lecture will have two breaks for hands-on work, both of which will be relatively short and not very involving.

Contents

Topic 0: Contents, learning objectives etc

Topic 1: Very quick summary of NN basics, as iterative optimization with SGD.

- Each iteration is a separate model, but in the end we just keep the last one as it is the best one.

Topic 2: Bayesian inference for NNs

- We do not care about $p(w|D)$ as it means nothing
- We do care about $p(y|x, D) = \int p(y|x, w)p(w|D)dw$, the predictive distribution
- Monte Carlo integration: Replace integral with a sum over finite collection of alternative models
  - Relate to VI: We get $w \sim q(w)$
- MCMC means algorithms that give sequences of $w \sim p(w|D)$ by construction

Topic 3: Towards MCMC for BNNs

- Fake algorithm: What if we just keep multiple weights during SGD, instead of just the last one
- Predictions then by passing the inputs through *multiple* models
- The only missing piece is that SGD is not a MCMC algorithm

Topic 4: SGD as posterior sampler

- Except it actually is, as long as we are really careful
- https://www.jmlr.org/papers/volume18/17-214/17-214.pdf shows how we can choose the step-length of SGD such that it produces samples from the correct distribution, around one particular mode
- Basic derivation and the practical algorithm
  1. First find the mode
  2. Estimate the noise covariance caused by minibatching the gradient
  3. Solve for step-length for which this noise matches the posterior
  4. Sample with that step-length

Programming break: Implement and try out SGD as sampler

Topic 5: SGLD

- A bit more realistic sampler is stochastic gradient Langevin dynamics
- https://www.stats.ox.ac.uk/~teh/research/compstats/WelTeh2011a.pdf
- Despite a bit involved derivation, the algorithm is really simple:
  1. Again first find the mode
  2. Otherwise standard SGD, but we add explicit Gaussian noise for each step

3. If the noise variance is set correctly, this samples from the posterior at the limit of extremely small step-length

Topic 6: Fixing the bias

- For finite step-lengths SLGD is biased
- Metropolis-Hastings is a standard protocol for resolving this
- Naive random-walk MH algorithms do not work, but the acceptance check fixes also SGLD and other gradient-based methods
- MH for BNNs:
  - Make a proposal as in SGD and compute also the proposal distribution
  - Temporarily update the parameters and compute the proposal from the new configuration to the old one (yep, doubles computation)
  - Accept with the MH rule
- Metropolis-adjusted Langevin Algorithm (MALA) is the fixed version of SGLD
- https://en.wikipedia.org/wiki/Metropolis-adjusted_Langevin_algorithm

*** Programming break: Implement and try out SGLD and perhaps MALA ***

Topic 7: Limitations and what is missing

- Explore around a mode, even the theory breaks if not there
- Inefficient when the parameters correlates (as they always do)
  - Direct generalisations as in optimization:
    - Parameter-specific learning rates <=> diagonal preconditioners
    - Position-dependent preconditioners (Riemannian methods)
    - Low-rank/kronecker/last layer for modelling dependencies
  - Improved integrators and dynamic; Hamiltonian MC instead of Langeving etc

Topic 8: Briefly on other similar methods: Bayesian dropout and SWAG

- Dropout: Instead of MCMC, we interpret dropout as mechanism for giving as the set of w to average over
- SWAG forms explicit distribution from samples during training, then samples new weights at prediction time
- May skip this if not enough time

Topic 9: Bayesian NNs in practice

- Never implement by yourself, but use libraries
- Should not use SGLD or SGD as such, but they help understanding the theory
- MALA is often used as component (e.g. in diffusion models), but is not the best sampler as such
- Does it work? Linearised NNs, cold posteriors, ...