# Bayesian Inference for Deep Learning

## Inference and modern trends for Bayesian Neural Networks: Sampling with Stochastic Gradient methods

Simone Rossi and Maurizio Filippone

Data Science Department, EURECOM (France)

# Markov chain Monte Carlo

# Motivation

- Predictive distributions can be computed as:

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}) = \int p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|\mathbf{Y}, \mathbf{X})d\mathbf{w}$$

- The integral is analytically intractable but we can approximate it as:

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}) \approx \sum_{i=1}^{\mathrm{MC}} p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{w}^{(i)})$$

  as long as we can obtain samples $\mathbf{w}^{(i)} \sim p(\mathbf{w}|\mathbf{Y}, \mathbf{X})$

# Motivation

- Predictive distributions can be computed as:

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}) = \int p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|\mathbf{Y}, \mathbf{X})d\mathbf{w}$$

- The integral is analytically intractable but we can approximate it as:

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}) \approx \sum_{i=1}^{\mathrm{MC}} p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{w}^{(i)})$$

  as long as we can obtain samples $\mathbf{w}^{(i)} \sim p(\mathbf{w}|\mathbf{Y}, \mathbf{X})$

- Markov chain Monte Carlo (MCMC) allows one to obtain sample from intractable distribution

# MCMC

- The posterior density is known up to a normalization constant

$$p(\mathbf{w}|\mathbf{X},\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})$$

- For many MCMC algorithms that is enough to obtain samples from the posterior

# MCMC

- The posterior density is known up to a normalization constant

$$p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$$

- For many MCMC algorithms that is enough to obtain samples from the posterior
- Stochastic MCMC algorithms relax the need to evaluate the likelihood and rely on stochastic gradients of the log-likelihood

# Metropolis-Hastings (MH)

- Produces a sequence of samples – $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots$
- Imagine we've just produced $\mathbf{w}^{(i-1)}$

# Metropolis-Hastings (MH)

- Produces a sequence of samples – $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots$
- Imagine we've just produced $\mathbf{w}^{(i-1)}$
- MH firsts *proposes* a possible $\mathbf{w}^{(i)}$ (call it $\widetilde{\mathbf{w}^{(i)}}$) based on $\mathbf{w}^{(i-1)}$.

# Metropolis-Hastings (MH)

- Produces a sequence of samples – $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots$
- Imagine we've just produced $\mathbf{w}^{(i-1)}$
- MH firsts *proposes* a possible $\mathbf{w}^{(i)}$ (call it $\widetilde{\mathbf{w}^{(i)}}$) based on $\mathbf{w}^{(i-1)}$.
- MH then decides whether or not to *accept* $\widetilde{\mathbf{w}^{(i)}}$
  - If accepted, $\mathbf{w}_i = \widetilde{\mathbf{w}^{(i)}}$
  - If not, $\mathbf{w}_i = \mathbf{w}^{(i-1)}$

# Metropolis-Hastings (MH)

- Produces a sequence of samples – $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots$
- Imagine we've just produced $\mathbf{w}^{(i-1)}$
- MH firsts *proposes* a possible $\mathbf{w}^{(i)}$ (call it $\widetilde{\mathbf{w}^{(i)}}$) based on $\mathbf{w}^{(i-1)}$.
- MH then decides whether or not to *accept* $\widetilde{\mathbf{w}^{(i)}}$
    - If accepted, $\mathbf{w}_i = \widetilde{\mathbf{w}^{(i)}}$
    - If not, $\mathbf{w}_i = \mathbf{w}^{(i-1)}$
- Two distinct steps – proposal and acceptance.

# Metropolis-Hastings – proposal

- Treat $\widetilde{\mathbf{w}^{(i)}}$ as a random variable conditioned on $\mathbf{w}^{(i-1)}$
- i.e. need to define $p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)})$
    - Note that this does not necessarily have to be similar to posterior we're trying to sample from.
- Can choose *whatever we like*!

# Metropolis-Hastings – proposal

- Treat $\widetilde{\mathbf{w}^{(i)}}$ as a random variable conditioned on $\mathbf{w}^{(i-1)}$
- i.e. need to define $p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)})$
    - Note that this does not necessarily have to be similar to posterior we're trying to sample from.
- Can choose *whatever we like*!
- e.g. use a Gaussian centered on $\mathbf{w}^{(i-1)}$ with some covariance:

$$p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)}, \mathbf{\Sigma}_p) = \mathcal{N}(\mathbf{w}^{(i-1)}, \mathbf{\Sigma}_p)$$

# Metropolis-Hastings – acceptance

- Choice of acceptance based on the following ratio:

$$r = \frac{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{Y},\mathbf{X})}{p(\mathbf{w}^{(i-1)}|\mathbf{Y},\mathbf{X})}\frac{p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}},\mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)},\mathbf{\Sigma}_p)}.$$

# Metropolis-Hastings – acceptance

- Choice of acceptance based on the following ratio:

$$r = \frac{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{Y},\mathbf{X})}{p(\mathbf{w}^{(i-1)}|\mathbf{Y},\mathbf{X})}\frac{p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}},\boldsymbol{\Sigma}_p)}{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)},\boldsymbol{\Sigma}_p)}.$$

- Which simplifies to (all of which we can compute):

$$r = \frac{p(\mathbf{Y}|\mathbf{X},\widetilde{\mathbf{w}^{(i)}})p(\widetilde{\mathbf{w}^{(i)}})}{p(\mathbf{Y}|\mathbf{X},\mathbf{w}^{(i-1)})p(\mathbf{w}^{(i-1)})}\frac{p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}},\boldsymbol{\Sigma}_p)}{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)},\boldsymbol{\Sigma}_p)}.$$

# Metropolis-Hastings – acceptance

- Choice of acceptance based on the following ratio:

$$r = \frac{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{Y},\mathbf{X})}{p(\mathbf{w}^{(i-1)}|\mathbf{Y},\mathbf{X})} \frac{p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}},\boldsymbol{\Sigma}_p)}{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)},\boldsymbol{\Sigma}_p)}.$$

- Which simplifies to (all of which we can compute):

$$r = \frac{p(\mathbf{Y}|\mathbf{X},\widetilde{\mathbf{w}^{(i)}})p(\widetilde{\mathbf{w}^{(i)}})}{p(\mathbf{Y}|\mathbf{X},\mathbf{w}^{(i-1)})p(\mathbf{w}^{(i-1)})} \frac{p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}},\boldsymbol{\Sigma}_p)}{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)},\boldsymbol{\Sigma}_p)}.$$

- We now use the following rules:
    - If $r \geq 1$, accept: $\mathbf{w}^{(i)} = \widetilde{\mathbf{w}^{(i)}}$.
    - If $r < 1$, accept with probability $r$.

# Metropolis-Hastings – acceptance

- Choice of acceptance based on the following ratio:

$$r = \frac{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{Y},\mathbf{X})}{p(\mathbf{w}^{(i-1)}|\mathbf{Y},\mathbf{X})} \frac{p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}},\mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)},\mathbf{\Sigma}_p)}.$$

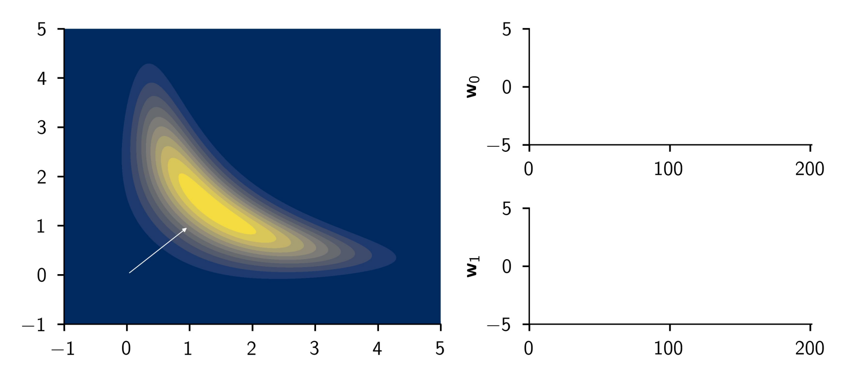
- Which simplifies to (all of which we can compute):

$$r = \frac{p(\mathbf{Y}|\mathbf{X},\widetilde{\mathbf{w}^{(i)}})p(\widetilde{\mathbf{w}^{(i)}})}{p(\mathbf{Y}|\mathbf{X},\mathbf{w}^{(i-1)})p(\mathbf{w}^{(i-1)})} \frac{p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}},\mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)},\mathbf{\Sigma}_p)}.$$

- We now use the following rules:
  - If $r \geq 1$, accept: $\mathbf{w}^{(i)} = \widetilde{\mathbf{w}^{(i)}}$.
  - If $r < 1$, accept with probability $r$.
- If we do this enough, we'll eventually be sampling from $p(\mathbf{w}|\mathbf{Y},\mathbf{X})$, no matter where we started!
  - i.e. for any $\mathbf{w}^{(1)}$

# Metropolis-Hastings (MH) algorithm

$$\text{Acceptance probability}: r = \frac{p(\mathbf{Y}|\mathbf{X}, \widetilde{\mathbf{w}^{(i)}})p(\widetilde{\mathbf{w}^{(i)}})}{p(\mathbf{Y}|\mathbf{X}, \mathbf{w}^{(i-1)})p(\mathbf{w}^{(i-1)})} \frac{p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}}, \mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)}, \mathbf{\Sigma}_p)}.$$




Metropolis et al., *JoCP*, 1953 - Hastings, *Biometrika*, 1970

# Metropolis-Hastings Derivation from Detailed Balance

- Detailed balance

$$p(\mathbf{w}'|\mathbf{Y}, \mathbf{X})p(\mathbf{w}|\mathbf{w}') = p(\mathbf{w}|\mathbf{Y}, \mathbf{X})p(\mathbf{w}'|\mathbf{w})$$

  is a sufficient condition to ensure existence of a stationary distribution $p(\mathbf{w}|\mathbf{Y}, \mathbf{X})$

# Metropolis-Hastings Derivation from Detailed Balance

- Detailed balance
$$p(\mathbf{w}'|\mathbf{Y}, \mathbf{X})p(\mathbf{w}|\mathbf{w}') = p(\mathbf{w}|\mathbf{Y}, \mathbf{X})p(\mathbf{w}'|\mathbf{w})$$
is a sufficient condition to ensure existence of a stationary distribution $p(\mathbf{w}|\mathbf{Y}, \mathbf{X})$
- Ergodicity (Morkov chain being aperiodic and positive recurrent) ensures uniqueness of the stationary distribution $p(\mathbf{w}|\mathbf{Y}, \mathbf{X})$

# Metropolis-Hastings Derivation from Detailed Balance

- Rewrite detailed balance condition:

$$p(\mathbf{w}'|\mathbf{Y},\mathbf{X})p(\mathbf{w}|\mathbf{w}') = p(\mathbf{w}|\mathbf{Y},\mathbf{X})p(\mathbf{w}'|\mathbf{w}) \quad \Rightarrow \quad \frac{p(\mathbf{w}'|\mathbf{Y},\mathbf{X})}{p(\mathbf{w}|\mathbf{Y},\mathbf{X})} = \frac{p(\mathbf{w}'|\mathbf{w})}{p(\mathbf{w}|\mathbf{w}')}$$

# Metropolis-Hastings Derivation from Detailed Balance

- Rewrite detailed balance condition:

$$p(\mathbf{w}'|\mathbf{Y}, \mathbf{X})p(\mathbf{w}|\mathbf{w}') = p(\mathbf{w}|\mathbf{Y}, \mathbf{X})p(\mathbf{w}'|\mathbf{w}) \quad \Rightarrow \quad \frac{p(\mathbf{w}'|\mathbf{Y}, \mathbf{X})}{p(\mathbf{w}|\mathbf{Y}, \mathbf{X})} = \frac{p(\mathbf{w}'|\mathbf{w})}{p(\mathbf{w}|\mathbf{w}')}$$

- Break transition in proposal and acceptance steps:

$$p(\mathbf{w}'|\mathbf{w}) = \mathrm{pro}(\mathbf{w}'|\mathbf{w})\,\mathrm{acc}(\mathbf{w}'|\mathbf{w})$$

- Substitute back and rearrange:

$$\frac{\mathrm{acc}(\mathbf{w}'|\mathbf{w})}{\mathrm{acc}(\mathbf{w}|\mathbf{w}')} = \frac{p(\mathbf{w}'|\mathbf{Y}, \mathbf{X})\mathrm{pro}(\mathbf{w}|\mathbf{w}')}{p(\mathbf{w}|\mathbf{Y}, \mathbf{X})\mathrm{pro}(\mathbf{w}|\mathbf{w}')}$$

- Easy to verify that the MH acceptance rule satisfies this condition

# Beyond Random Walk

- MH can be inefficient due to its random walk nature!

# Beyond Random Walk

- MH can be inefficient due to its random walk nature!
- Improve efficiency by using gradient information

# Beyond Random Walk

- MH can be inefficient due to its random walk nature!
- Improve efficiency by using gradient information
- Hamiltonian Monte Carlo (HMC):
  - The proposal mechanism uses the gradient of the unnormalized log-density:

$$\nabla_{\mathbf{w}} \log \left[p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})\right]$$

    to simulate trajectories in the space of parameters.
  - Thanks to this, proposals $\widetilde{\mathbf{w}^{(i)}}$ can be far away from the starting point $\mathbf{w}^{(i-1)}$!

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- Introduce momentum variables $\mathbf{p}$ and introduce the kinetic energy

$$V = \frac{1}{2}\mathbf{p}^\top \mathbf{M}^{-1}\mathbf{p}$$

where $\mathbf{M}$ is referred to as the mass matrix

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- Introduce momentum variables $\mathbf{p}$ and introduce the kinetic energy

$$V = \frac{1}{2}\mathbf{p}^\top \mathbf{M}^{-1}\mathbf{p}$$

  where $\mathbf{M}$ is referred to as the mass matrix

- Then interpret the negative of the log-density as the potential energy:

$$U = -\log\left[p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})\right]$$

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- Introduce momentum variables $\mathbf{p}$ and introduce the kinetic energy

$$V = \frac{1}{2}\mathbf{p}^\top \mathbf{M}^{-1}\mathbf{p}$$

  where $\mathbf{M}$ is referred to as the mass matrix
- Then interpret the negative of the log-density as the potential energy:

$$U = -\log\left[p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})\right]$$

- Now simulate the Hamiltonian system with energy $H = U + V$ with a random $\mathbf{p} \sim \mathcal{N}(\mathbf{0}, \mathbf{M})$ and for a random duration $T$.
- This means solving Hamilton-Jacobi equations:

$$\frac{d\mathbf{w}}{dt} = \frac{dH}{d\mathbf{p}} = \frac{dV}{d\mathbf{p}}$$

$$\frac{d\mathbf{p}}{dt} = -\frac{dH}{d\mathbf{w}} = -\frac{dU}{d\mathbf{w}}$$

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- Solving Hamilton-Jacobi equations for a given $T$ is generally intractable

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- Solving Hamilton-Jacobi equations for a given $T$ is generally intractable
- We need discretization of the differential equations but ...
- ... the choice of discretization method matters in making HMC correct

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- Solving Hamilton-Jacobi equations for a given $T$ is generally intractable
- We need discretization of the differential equations but ...
- ... the choice of discretization method matters in making HMC correct
- The discretization needs to preserve reversibility so that:

$$p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}}, \mathbf{\Sigma}_p) = p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)}, \mathbf{\Sigma}_p)$$

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- Solving Hamilton-Jacobi equations for a given $T$ is generally intractable
- We need discretization of the differential equations but ...
- ... the choice of discretization method matters in making HMC correct
- The discretization needs to preserve reversibility so that:

$$p(\mathbf{w}^{(i-1)}|\widetilde{\mathbf{w}^{(i)}}, \mathbf{\Sigma}_p) = p(\widetilde{\mathbf{w}^{(i)}}|\mathbf{w}^{(i-1)}, \mathbf{\Sigma}_p)$$

- Leapfrog Integrator ensures reversibility – sketch of the integration scheme:

$$\mathbf{p}^{(i-1)}_{t+\Delta t/2} = \mathbf{p}^{(i-1)}_{t} - \frac{\Delta t}{2}(\nabla_{\mathbf{w}} U)(\mathbf{w}^{(i-1)}_{t})$$

$$\mathbf{w}^{(i-1)}_{t+\Delta t} = \mathbf{w}^{(i-1)}_{t} + \Delta t \mathbf{M}^{-1}\mathbf{p}^{(i-1)}_{t+\Delta t/2}$$

$$\mathbf{p}^{(i-1)}_{t+\Delta t} = \mathbf{p}^{(i-1)}_{t+\Delta t/2} - \frac{\Delta t}{2}(\nabla_{\mathbf{w}} U)(\mathbf{w}^{(i-1)}_{t+\Delta t})$$

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- We started integrating from a pair $(\mathbf{w}^{(i-1)}, \mathbf{p}^{(i-1)})$
- Acceptance of a new pair $(\tilde{\mathbf{w}}^{(i)}, \tilde{\mathbf{p}}^{(i)})$ after a few integration steps requires evaluating

$$H = -\log\left[p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})\right] + \frac{1}{2}\mathbf{p}^{\top}\mathbf{M}^{-1}\mathbf{p}$$

at $(\tilde{\mathbf{w}}^{(i)}, \tilde{\mathbf{p}}^{(i)})$ and $(\mathbf{w}^{(i-1)}, \mathbf{p}^{(i-1)})$

Duane et al., *Physics Lett. B*, 1987 – Neal, 1993

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

- We started integrating from a pair $(\mathbf{w}^{(i-1)}, \mathbf{p}^{(i-1)})$
- Acceptance of a new pair $(\tilde{\mathbf{w}}^{(i)}, \tilde{\mathbf{p}}^{(i)})$ after a few integration steps requires evaluating

$$H = -\log\left[p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})\right] + \frac{1}{2}\mathbf{p}^{\top}\mathbf{M}^{-1}\mathbf{p}$$

at $(\tilde{\mathbf{w}}^{(i)}, \tilde{\mathbf{p}}^{(i)})$ and $(\mathbf{w}^{(i-1)}, \mathbf{p}^{(i-1)})$

- The system has no friction so in theory all proposals should be accepted!

---

Duane et al., *Physics Lett. B*, 1987 – Neal, 1993

# Hamiltonian Monte Carlo (or Hybrid Monte Carlo)

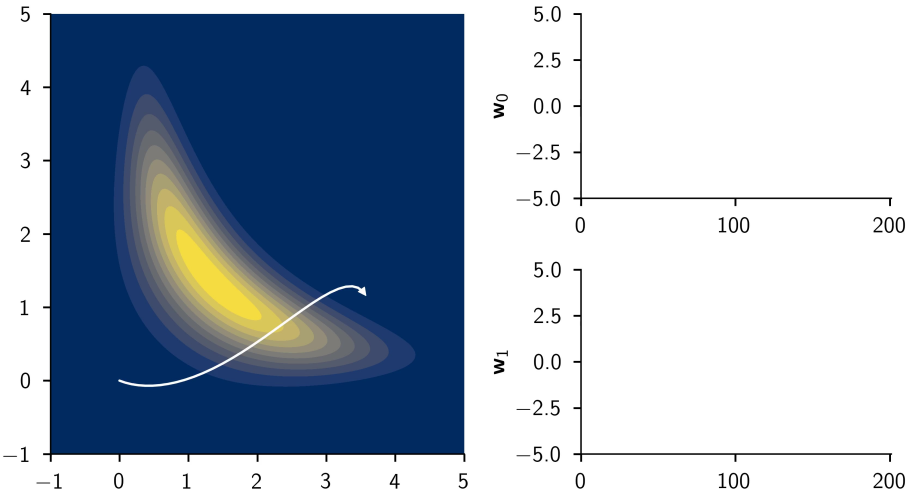
- We started integrating from a pair $(\mathbf{w}^{(i-1)}, \mathbf{p}^{(i-1)})$
- Acceptance of a new pair $(\tilde{\mathbf{w}}^{(i)}, \tilde{\mathbf{p}}^{(i)})$ after a few integration steps requires evaluating

$$H = -\log\left[p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})\right] + \frac{1}{2}\mathbf{p}^{\top}\mathbf{M}^{-1}\mathbf{p}$$

  at $(\tilde{\mathbf{w}}^{(i)}, \tilde{\mathbf{p}}^{(i)})$ and $(\mathbf{w}^{(i-1)}, \mathbf{p}^{(i-1)})$
- The system has no friction so in theory all proposals should be accepted!
- However, the integrator introduces errors which require the acceptance based on the ratio of $H$ at time $T$ and 0.

Duane et al., *Physics Lett. B*, 1987 – Neal, 1993

# Sampling trajectories with HMC



## Stochastic Hamiltonian Monte Carlo

- HMC is expensive for two reasons:
  - Simulating the dynamics requires calculating the gradient:

$$\sum_{i=1}^{N} \nabla_{\mathbf{w}} \log \left[ p(\mathbf{y}_i | \mathbf{x}_i, \mathbf{w}) p(\mathbf{w}) \right] \qquad \mathcal{O}(N)$$

  - Accepting proposals requires calculating part of the Hamiltonian:

$$\log \left[ p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) p(\mathbf{w}) \right] \qquad \mathcal{O}(N)$$

Chen et al., *ICML*, 2014

# Stochastic Hamiltonian Monte Carlo

- HMC is expensive for two reasons:
  - Simulating the dynamics requires calculating the gradient:

$$\sum_{i=1}^{N} \nabla_{\mathbf{w}} \log \left[ p(\mathbf{y}_i | \mathbf{x}_i, \mathbf{w}) p(\mathbf{w}) \right] \qquad \mathcal{O}(N)$$

  - Accepting proposals requires calculating part of the Hamiltonian:

$$\log \left[ p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) p(\mathbf{w}) \right] \qquad \mathcal{O}(N)$$

- Stochastic-Gradient HMC:
  - Mini-batch unbiased estimate of the gradient based on indices set $\mathcal{I}_M$:

$$\frac{N}{M} \sum_{i \in \mathcal{I}_M} \nabla_{\mathbf{w}} \log \left[ p(\mathbf{y}_i | \mathbf{x}_i, \mathbf{w}) p(\mathbf{w}) \right] \qquad \mathcal{O}(M)$$

  - Always accept!
  - Always accepting would introduce bias in the sampling
  - In SG-HMC, the dynamics is modified to ensure that the bias is negligible

Chen et al., *ICML*, 2014

# Stochastic-Gradient Hamiltonian Monte Carlo

- The main result follows from assuming that:

$$\frac{N}{M}\sum_{i\in\mathcal{I}_M}\nabla_{\mathbf{w}}\log\left[p(\mathbf{y}_i|\mathbf{x}_i,\mathbf{w})p(\mathbf{w})\right] =: \widetilde{\nabla U}(\mathbf{w}) \approx \nabla U(\mathbf{w}) + \mathcal{N}(\mathbf{0},\mathbf{Q}(\mathbf{w}))$$

  by the central limit theorem

- The dynamics now can be seen as a discretization of the following SDE:

$$d\mathbf{w} = \frac{dV}{d\mathbf{p}}dt = \mathbf{M}^{-1}\mathbf{p}$$

$$d\mathbf{p} = -\frac{d\tilde{U}}{d\mathbf{w}}dt = -\nabla U(\mathbf{w})dt + \mathcal{N}(\mathbf{0},\epsilon\mathbf{Q}(\mathbf{w})dt)$$

  where $\epsilon$ is the step-size.

- The stationary distribution is no longer the posterior of interest

# Stochastic-Gradient Hamiltonian Monte Carlo

- Stochastic-Gradient HMC modifies the dynamics by introducing a friction term:

$$d\mathbf{w} = \mathbf{M}^{-1}\mathbf{p}$$

$$d\mathbf{p} = -\nabla U(\mathbf{w})dt + \mathcal{N}(\mathbf{0}, \epsilon\mathbf{Q}(\mathbf{w})dt) - \frac{1}{2}\epsilon\mathbf{Q}\mathbf{M}^{-1}\mathbf{p}dt$$

- The stationary distribution is now the posterior of interest!

Chen et al. (2014). *Stochastic Gradient Hamiltonian Monte Carlo*, ICML

# Stochastic-Gradient Hamiltonian Monte Carlo

- Stochastic-Gradient HMC modifies the dynamics by introducing a friction term:

$$d\mathbf{w} = \mathbf{M}^{-1}\mathbf{p}$$

$$d\mathbf{p} = -\nabla U(\mathbf{w})dt + \mathcal{N}(\mathbf{0}, \epsilon\mathbf{Q}(\mathbf{w})dt) - \frac{1}{2}\epsilon\mathbf{Q}\mathbf{M}^{-1}\mathbf{p}dt$$

- The stationary distribution is now the posterior of interest!
- In practice we need to estimate $\mathbf{Q}$.

Chen et al. (2014). *Stochastic Gradient Hamiltonian Monte Carlo*, ICML

# Sampling trajectories of SG-HMC

The discretized dynamics become

$$\Delta \mathbf{w} = \epsilon \mathbf{M}^{-1} \mathbf{p}$$
$$\Delta \mathbf{p} = -\epsilon \nabla \widetilde{U}(\mathbf{w}) + \mathcal{N}(0, 2\epsilon(\mathbf{C} - \widetilde{\mathbf{B}})) - \epsilon \mathbf{C} \mathbf{M}^{-1}$$

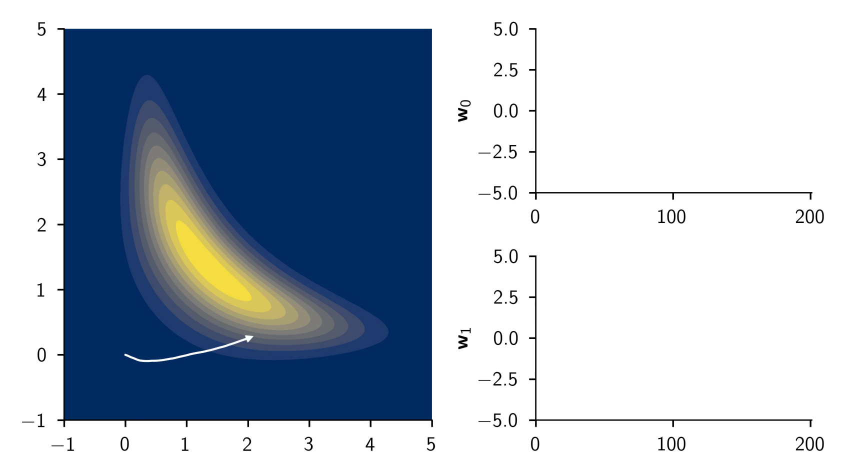
with

- $\widetilde{U}(\mathbf{w})$ is the mini-batch estimation of the log-joint
- $\epsilon$ is the step size
- $\mathbf{C}$ is the friction matrix
- $\widetilde{\mathbf{B}}$ is the estimation of the stochastic gradient noise covariance

# Sampling trajectories of SG-HMC

$$\Delta \mathbf{w} = \epsilon \mathbf{M}^{-1} \mathbf{p}$$

$$\Delta \mathbf{p} = -\epsilon \nabla \widetilde{U}(\mathbf{w}) + \mathcal{N}(0, 2\epsilon(\mathbf{C} - \widetilde{\mathbf{B}})) - \epsilon \mathbf{C} \mathbf{M}^{-1}$$



# Preconditioning SG-HMC

Naive SG-HMC introduces some additional quantities to be estimated:

- **Gradient variance** $\widehat{V}$

$$\widehat{V} \approx \mathbb{E}(\nabla \widetilde{U}(\mathbf{w}))^2 \quad \text{estimated with exponential moving average}$$

- **Mass** $\mathbf{M}$

$$\mathbf{M}^{-1} = \text{diag}\left(\widehat{V}^{-\frac{1}{2}}\right)$$

- **Matrix** $\widetilde{\mathbf{B}}$

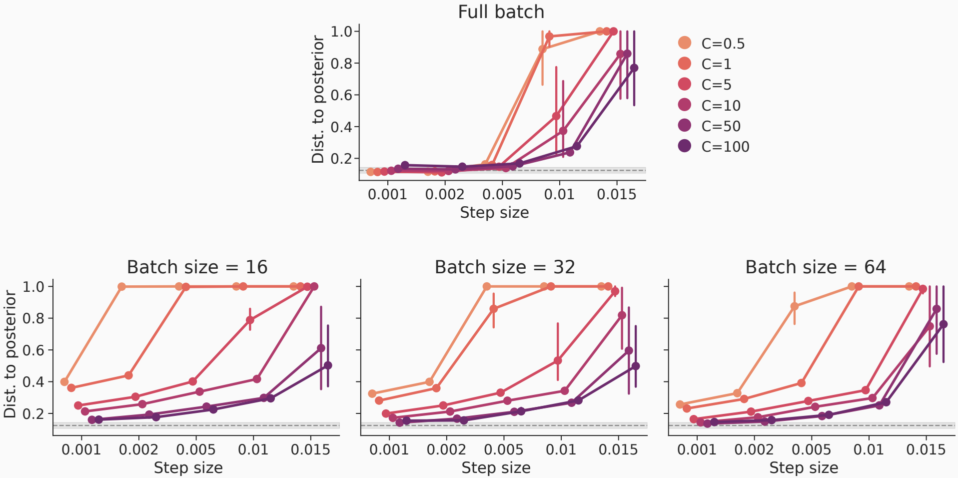
$$\widetilde{\mathbf{B}} = \frac{1}{2}\epsilon\widehat{V}$$

- **Friction**

$$\mathbf{C} = C\mathbf{I}$$

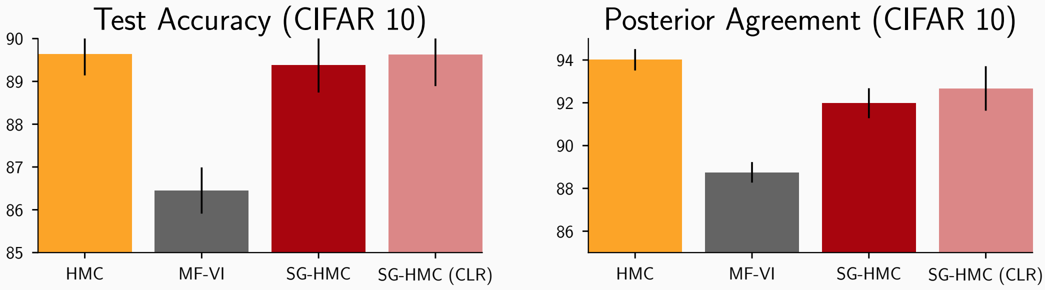
Springenberg et al. (2016). *Bayesian Optimization with Robust Bayesian Neural Networks*. NeurIPS

# Choosing a good friction term is important to achieve convergence




Franzese et al. (2021). *A Unified View of Stochastic Hamiltonian Sampling*. arXiv

# How good are stochastic gradient MCMC methods in practice? (ResNet20)




HMC results obtained using 512 TPUs for 60 milions epochs (v2-512 instance retails at 384 $/hour).

Izmailov et al. (2021). *What Are Bayesian Neural Network Posteriors Really Like?* ICML

# References i


- MacKay (1992). *A Practical Bayesian Framework for Backpropagation Networks*. Neural computation.
- Neal (1996). *Bayesian Learning for Neural Networks*. Springer
- Neal (2011). *MCMC using Hamiltonian Dynamics*. Hand-book of Markov Chain Monte Carlo
- Ahn et al. (2012). *Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring*. ICML
- Chen et al. (2014). *Stochastic gradient Hamiltonian Monte Carlo*. ICML
- Betancourt (2015). *The Fundamental Incompatibility of Scalable Hamiltonian Monte Carlo and Naive Data Subsampling*. ICML
- Chen et al. (2015). *On the Convergence of Stochastic Gradient MCMC Algorithms with High-Order Integrators*. NeurIPS
- Springenberg et al. (2016). *Bayesian Optimization with Robust Bayesian Neural Networks*. NeurIPS
- Mandt et al. (2017). *Stochastic Gradient Descent as Approximate Bayesian Inference*. JMLR

# References ii


- Zhang et al. (2020). *Amagold: Amortized Metropolis Adjustment for Efficient Stochastic Gradient MCMC*. AISTATS
- Zhang et al. (2020). *Cyclical stochastic gradient MCMC for Bayesian deep learning*. ICLR
- Cobb et al. (2021). *Scaling Hamiltonian Monte Carlo Inference for Bayesian Neural Networks with Symmetric Splitting*. UAI
- Franzese et al. (2021). *A Unified View of Stochastic Hamiltonian Sampling*. arXiv
- Izmailov et al. (2021). *What Are Bayesian Neural Network Posteriors Really Like?* ICML