Bayesian neural networks

(and VI in implicit models)

Dmitry Molchanov

Samsung AI Center, Samsung-HSE Laboratory

Lecture outline

- What are Bayesian Neural Networks (BNNs)
- Why go Bayesian
- How to train BNNs
- Variational inference with implicit posteriors

What you already know

- Stochastic optimization
- Bayesian modelling
- Latent variable models
- Variational inference
  - Bayesian inference ↔ (stochastic) optimization
  - (Doubly) Stochastic variational inference
  - Reparameterization Trick

⇒ **Bayesian neural networks**

Regularization by noise

Traditional (1943+) regularization: add some penalty for model complexity

- Norm-based regularization (L2, L1)

$$Objective = DataLoss(X, W) + Regularizer(W)$$

- Max norm constraint

More recent (1990+) approaches: regularization by noise

- Input noise:
    - Denoising autoencoders
    - Data augmentation
- Weight noise:
    - Dropout (reviving noise regularization in 2012)
    - Gaussian weight noise

$$Objective = \mathbb{E}_{p(\Omega)} DataLoss(X, W, \Omega)$$

- Gradient noise

Bayesian framework provides a principled approach to training with noise!

Generative models vs discriminative models

Bayesian Discriminative Model:

Likelihood $$p(\boldsymbol{t}|X, \boldsymbol{w}) = \prod_{i=1}^{N} p(t_i|\boldsymbol{x}_i, \boldsymbol{w})$$

Can be a neural network with weights W!

Prior $$p(\boldsymbol{w})$$

Posterior $$p(\boldsymbol{w}|X, \boldsymbol{t}) = \frac{p(\boldsymbol{t}|X, \boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{t}|X, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}} = ?$$

- No local latent variables; we want the posterior over the weights instead
- Much higher dimensionality
  - 10^2-10^3 for generative models, 10^5-10^8 and more for discriminative models

Why go Bayesian?

A principled framework with many useful applications

- Regularization
- Ensembling
- Uncertainty estimation
- On-line / continual learning
- And more (stay tuned for the next lecture!)

Ensembling

A Bayesian neural network is **an infinite ensemble** of neural networks

$\boldsymbol{w} \sim p(\boldsymbol{w}|X, \boldsymbol{t})$

One sample from the posterior

One element of the ensemble

Predictive distribution $p(t^*|\boldsymbol{x}^*, X, \boldsymbol{t}) = \int p(t^*|\boldsymbol{x}^*, \boldsymbol{w})p(\boldsymbol{w}|X, \boldsymbol{t})d\boldsymbol{w}$

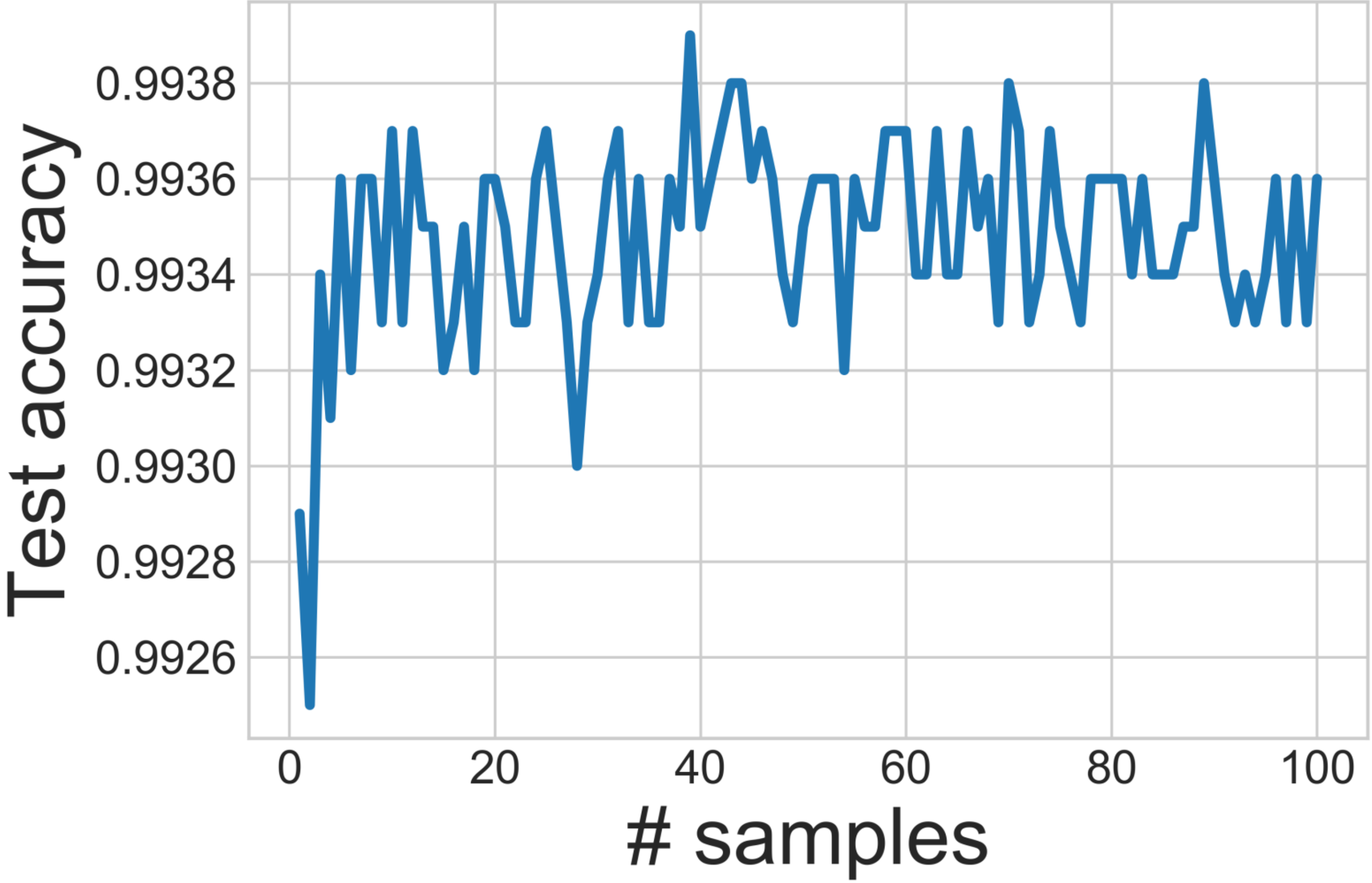
And its unbiased estimate

$$\mathbb{E}_{p(\boldsymbol{w}|X,\boldsymbol{t})}p(t^*|\boldsymbol{x}^*, \boldsymbol{w}) \simeq \frac{1}{K}\sum_{i=1}^{K} p\left(t^*\middle|\boldsymbol{x}^*, \boldsymbol{w}^k\right); \qquad \boldsymbol{w}^k \sim p(\boldsymbol{w}|X, \boldsymbol{t})$$

- Higher accuracy
- More robust

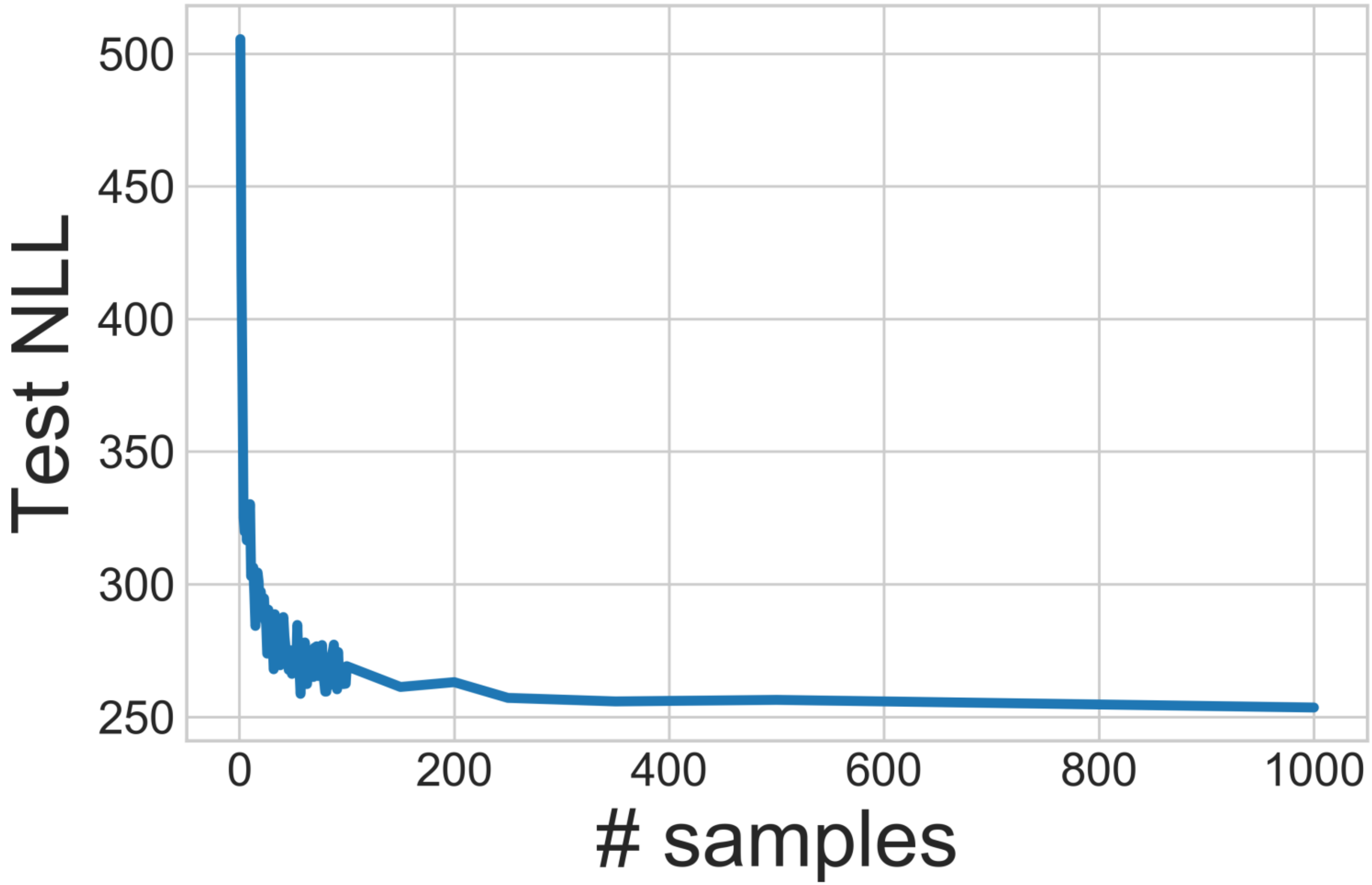
Average SoftMax outputs across several samples

Ensembling



Accuracy quickly saturates

Ensembling

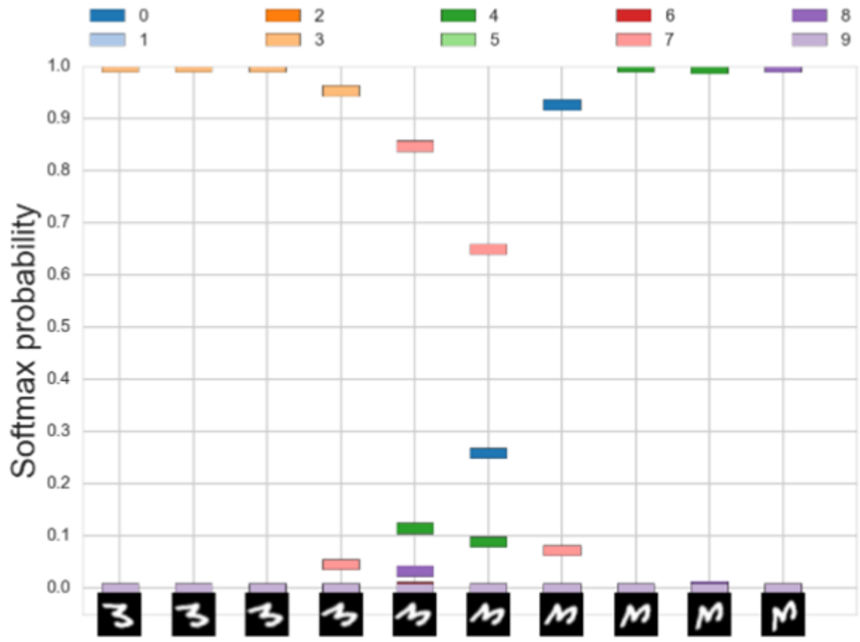


But the NLL keeps improving!
This is a measure of “uncertainty”

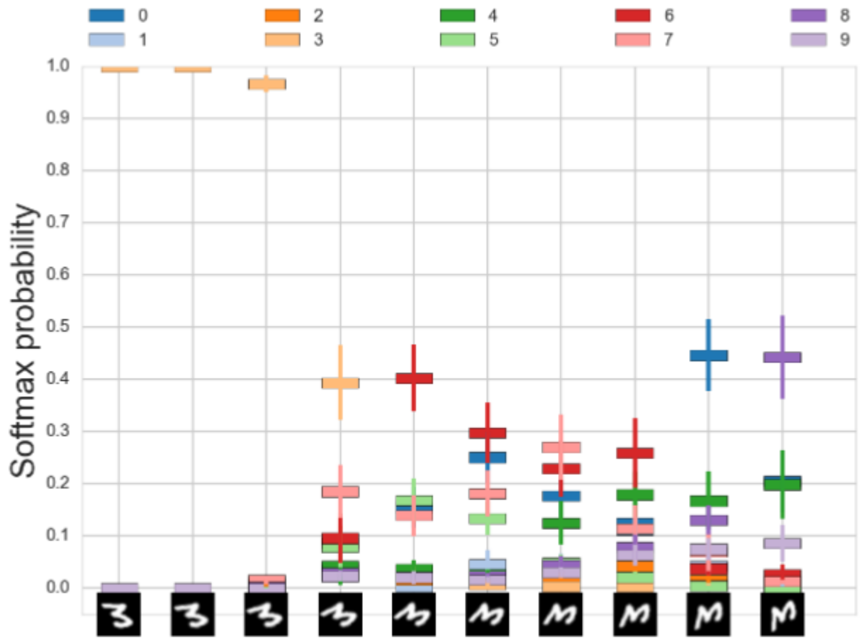
Uncertainty estimation

Deterministic NNs: **a point estimate** of the output, overconfident

Bayesian framework allows us to obtain **a distribution** over the outputs



(a) LeNet with weight decay



(b) LeNet with multiplicative formalizing flows

Louizos, Christos, and Max Welling. "Multiplicative normalizing flows for variational bayesian neural networks." *ICML 2017*

Model selection and compression

- Empirical Bayes (maximum evidence)
  - Choose hyperparameters
  - Model compression
    - Similar to the Relevance Vector Machine
- Special sparsity-inducing priors
  - Stay tuned for the next lecture

On-line / incremental learning

Assume that the dataset arrives in independent parts.

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \cup \mathcal{D}_M$$

We can train on the first dataset as usual…

$$p(\boldsymbol{w}|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\boldsymbol{w})p(\boldsymbol{w})}{\int p(\mathcal{D}_1|\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}$$

… And then use the obtained posterior as the prior for the next step!

$$p(\boldsymbol{w}|\mathcal{D}_2, \mathcal{D}_1) = \frac{p(\mathcal{D}_2|\boldsymbol{w})p(\mathcal{D}_1|\boldsymbol{w})p(\boldsymbol{w})}{\int p(\mathcal{D}_2|\boldsymbol{w})p(\mathcal{D}_1|\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}} =$$

$$= \frac{p(\mathcal{D}_2|\boldsymbol{w})p(\boldsymbol{w}|\mathcal{D}_1)}{\int p(\mathcal{D}_2|\boldsymbol{w})p(\boldsymbol{w}|\mathcal{D}_1)d\boldsymbol{w}}$$

Using these sequential updates, we can find $p(\boldsymbol{w}|\mathcal{D})$!

Variational inference for Bayesian NNs

The posterior distribution $p(\boldsymbol{w}|X, \boldsymbol{t}) = \frac{p(\boldsymbol{t}|X, \boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{t}|X, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}$

How to find it? Use (doubly stochastic) variational inference!

$$q(\boldsymbol{w}|\boldsymbol{\phi}) \approx p(\boldsymbol{w}|X, \boldsymbol{t})$$

$$\mathrm{KL}\big(q(\boldsymbol{w}|\boldsymbol{\phi}) \parallel p(\boldsymbol{w}|X, \boldsymbol{t})\big) \to \min_{\boldsymbol{\phi}}$$

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{q(\boldsymbol{w}|\boldsymbol{\phi})} \log p(\boldsymbol{t}|X, \boldsymbol{w}) - \mathrm{KL}(q(\boldsymbol{w}|\boldsymbol{\phi}) \parallel p(\boldsymbol{w})) \to \max_{\boldsymbol{\phi}}$$

Only two differences from LVMs:

- KL-term is global
- Extremely high-dimensional posterior

Reparameterization trick for Bayesian NNs

Reparameterize $q(\boldsymbol{w}|\boldsymbol{\phi})$ and plug the sample into the ELBO

$$\boldsymbol{w} \sim q(\boldsymbol{w}|\boldsymbol{\phi}) \quad \Leftrightarrow \quad \boldsymbol{w} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi}); \qquad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{p(\boldsymbol{\epsilon})} \log p(\boldsymbol{t}|X, \boldsymbol{w} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi})) - \mathrm{KL}(q \parallel p) \to \max_{\boldsymbol{\phi}}$$

Obtain an unbiased differentiable mini-batch estimator

$$\mathcal{L}(\boldsymbol{\phi}) \simeq \sum_i \log p\left(\boldsymbol{t}_{m_i} \middle| \boldsymbol{x}_{m_i}, \boldsymbol{w} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi})\right) - \mathrm{KL}(q \parallel p); \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

Very similar to conventional loss functions

Basically, using any kind of noise during training is close to being Bayesian

Usually just **1 sample per iteration is enough!**

Ex: dropout training as variational inference

Binary dropout results in a binary dropout posterior

$$\boldsymbol{w} = \boldsymbol{\phi} \cdot \mathrm{diag}(\boldsymbol{\epsilon}); \qquad \epsilon_i \sim \mathrm{Bernoulli}(p)$$

It can be shown that a Gaussian prior leads to L2 regularization here

ELBO for binary dropout training:

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{p(\boldsymbol{\epsilon})} \log p(\boldsymbol{t}|X, \boldsymbol{w} = \boldsymbol{\phi} \cdot \mathrm{diag}(\boldsymbol{\epsilon})) - \lambda \|\phi\|_2^2 \to \max_{\phi}$$

- Using binary dropout means being Bayesian ☺
- There are other uses beyond regularization!
  - Ensembling
  - Uncertainty estimation
  - We can tune the dropout rate p using REINFORCE and extensions

Gal, Yarin, and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning." *ICML* 2016.

Ex: Fully-Factorized Gaussians

Approximate posterior

$$q(\boldsymbol{w}) = \prod_i \mathcal{N}\left(w_i \middle| \mu_i, \sigma_i^2\right)$$

Reparameterization

$$\boldsymbol{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}; \qquad \epsilon_i \sim \mathcal{N}(0, 1)$$

The prior here is, e.g. a zero-centered FF Gaussian prior with variance σ_{prior}^2

ELBO:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \mathbb{E}_{p(\boldsymbol{\epsilon})} \log p(\boldsymbol{t}|X, \boldsymbol{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}) - \underbrace{\frac{\|\mu\|_2^2 + \|\sigma\|_2^2}{2\sigma_{prior}^2} + \sum_i \log \frac{\sigma_i^2}{\sigma_{prior}^2}}_{\text{KL between two } \mathcal{N}} \to \max_{\mu,\sigma}$$

- More tractable
- Richer approximation
- Twice as many parameters
- Start with small σ, optimize w.r.t. $\log \sigma$ to avoid constrained optimization

Titsias, Michalis, and Miguel Lázaro-Gredilla. "Doubly stochastic variational Bayes for non-conjugate inference." *ICML* 2014.

The local reparameterization trick

ELBO estimator may have high variance:

Shared noise sample!

$$\mathcal{L}(\phi) \simeq \hat{\mathcal{L}}(\phi) = \frac{N}{M}\sum_{i=1}^{M} L_i(\phi, \epsilon)$$

$$\mathrm{Var}\left[\hat{\mathcal{L}}\right] = \frac{N^2}{M^2}\left(\sum_{i=1}^{M} \mathrm{Var}[L_i] + 2\sum_{i}^{M}\sum_{i}^{M} \mathrm{Cov}\left[L_i, L_j\right]\right)$$

$$= N^2\left(\frac{1}{M}\mathrm{Var}[L_i] + \frac{M-1}{M}\mathrm{Cov}\left[L_i, L_j\right]\right)$$

Kingma, Diederik P., Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick." *Advances in NIPS* 2015.

WARNING: NOTATION ABUSE

Green means element-wise

The local reparameterization trick

Consider a linear layer with weight matrix W, input A and output B.

$$w_{ij} \sim \mathcal{N}\left(\mu_{ij}, \sigma_{ij}^2\right)$$
$$B = AW$$

Predictions have **high** correlation because there is **one weight sample per batch**

$$\mathbb{E}B = A\mu \qquad \mathrm{Var}B = A^2\sigma^2$$
$$B \sim \mathcal{N}(A\mu, A^2\sigma^2)$$
$$B = A\mu + \sqrt{A^2\sigma^2} \odot \epsilon$$

Predictions have **zero** correlation because there is **one weight sample per object**

Kingma, Diederik P., Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick." *Advances in NIPS* 2015.

The local reparameterization trick

LRT also reduces the variance of the stochastic gradient for **one** object

$\frac{\partial L}{\partial \mu_i}$ is the same for both RT and LRT, but

$$\frac{\partial L}{\partial \sigma_i} = \frac{\partial L}{\partial b} \cdot \frac{\partial b}{\partial \sigma_i} = \frac{\partial L}{\partial b} \cdot a_i \epsilon_i$$

RT, 1 sample per weight
A lot of redundant stochasticity

$$\frac{\partial L}{\partial \sigma_i} = \frac{\partial L}{\partial b} \cdot \frac{\partial b}{\partial \sigma_i} = \frac{\partial L}{\partial b} \cdot \frac{a_i^2 \sigma_i \epsilon}{\sqrt{a^{2\top} \sigma^2}}$$

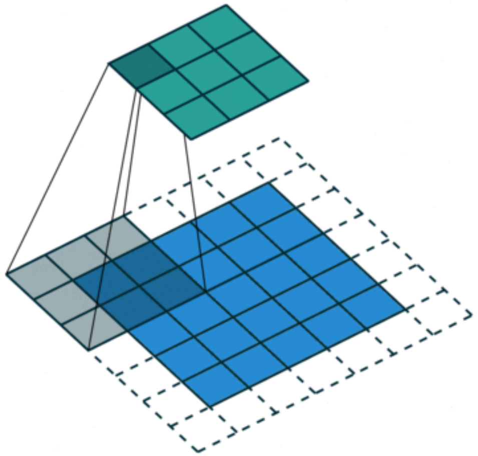
LRT, 1 sample per neuron
No redundant stochasticity

Kingma, Diederik P., Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick." *Advances in NIPS* 2015.

WARNING: NOTATION ABUSE

Green means element-wise

LRT for convolutions



- B no longer factorizes in convolutional layers
  - Same weight samples should be used for different spatial positions
- Exact local reparameterization is too complex
  - We need to calculate the full covariance matrix for each activation
- We can use the mean-field local reparameterization as an approximation
  - Not justified *(yet)*
  - Performs much better than plain reparameterization

$$\mathbb{E}B = A \star \mu \qquad \mathrm{Var}B = A^2 \star \sigma^2$$

$$B \sim \mathcal{N}(A \star \mu, A^2 \star \sigma^2)$$

$$B = A \star \mu + \sqrt{A^2 \star \sigma^2} \odot \epsilon$$

What next?

- How to choose prior
- Faster test-time averaging
- Better posterior approximations
  - Implicit models

A good time take a 5 minute break?

Treating deterministic parameters

What about other parameters, not just weight matrices?

- Biases
- Linear transformation in BatchNorm
- Any other “non-expressive” parameters

1) We can put priors over them, and treat them as random variables
2) We can treat them as deterministic parameters
   - Essentially it assumes a flat prior and a delta-peak posterior
   - Or we could see it as bounding the marginal likelihood of the data given these parameters

$$\log p(\boldsymbol{t}|X,\boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\phi},\boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{w}|\boldsymbol{\phi})} \log p(\boldsymbol{t}|X,\boldsymbol{w},\boldsymbol{\theta}) - \mathrm{KL}(q(\boldsymbol{w}|\boldsymbol{\phi}) \parallel p(\boldsymbol{w})) \to \max_{\boldsymbol{\phi},\boldsymbol{\theta}}$$

Empirical Bayes for Bayesian NNs

- How to choose the prior distribution?
- Type-II maximum likelihood (maximum evidence):

$$\log p(\boldsymbol{t}|X,\boldsymbol{\theta}) \to \max_{\boldsymbol{\theta}}$$

$$\log p(\boldsymbol{t}|X,\boldsymbol{\theta}) \geq$$

$$\geq \mathcal{L}(\boldsymbol{\phi},\boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{w}|\boldsymbol{\phi})} \log p(\boldsymbol{t}|X,\boldsymbol{w}) - \mathrm{KL}\big(q(\boldsymbol{w}|\boldsymbol{\phi}) \,\|\, p(\boldsymbol{w}|\boldsymbol{\theta})\big) \to \max_{\boldsymbol{\phi},\boldsymbol{\theta}}$$

- It is okay when $\dim \boldsymbol{\theta}$ is small
- May overfit if $\dim \boldsymbol{\theta}$ is large
  - Ideally we would have $p(w|\boldsymbol{\theta}) = q(w|\boldsymbol{\theta}) = \delta(\boldsymbol{w}_{ML})$
  - You never know whether you can overfit with a particular parameterization
- Usually used to induce sparsity (RVM, SWS, …)

Distillation

Test-time averaging is expensive

Imaging we have a good sampler $q_t(w_t)$ for $\boldsymbol{w}$

- SG MCMC
- Variational approximate posterior

We can train a separate deterministic neural network (student) to "mimic" the ensemble (teacher):

$$\mathcal{L}(\boldsymbol{w}_{st}) = \mathbb{E}_{q_t(\boldsymbol{w}_t)} \mathcal{H}\left(p(\boldsymbol{t}|X, \boldsymbol{w}_{st}), p(\boldsymbol{t}|X, \boldsymbol{w}_t)\right)$$

- Worse than the ensemble
- Better than a single network

Balan, Anoop Korattikara, et al. "Bayesian dark knowledge." *NIPS* 2015.

Distillation: examples

VI distillation:

1) Train the “teacher” network, obtain approx. posterior $q(\boldsymbol{w})$
2) For each iteration of learning of the “student” network:
   1) Sample a minibatch of data
   2) Sample predictions of the “teacher”
   3) Use SoftMax output of the teacher as “soft” labels

MCMC distillation:

1) Warm-up a Markov Chain for the “teacher”
2) For each iteration of learning of the “student” network:
   1) Sample a minibatch of data
   2) Make one SG MCMC update of the teacher
   3) Use SoftMax output of the teacher as “soft” labels

Variational inference with implicit posteriors

Implicit posterior: $\boldsymbol{w} = f(\boldsymbol{\epsilon}, \boldsymbol{\phi})$, $q_\phi(\boldsymbol{w})$ is intractable

Example: $f(\epsilon, \boldsymbol{\phi})$ is an arbitrary neural network

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{q_\phi(\boldsymbol{w})} \log p(\boldsymbol{t}, \boldsymbol{w}|X) - \mathbb{E}_{q_\phi(\boldsymbol{w})} \log q_\phi(\boldsymbol{w})$$

Semi-implicit formulation:

$$q_\phi(\boldsymbol{w}) = \int q_\phi(\boldsymbol{w}|\boldsymbol{z}) q_\phi(\boldsymbol{z}) d\boldsymbol{z}$$

$$\boldsymbol{z} \sim q_\phi(\boldsymbol{z}), \qquad \boldsymbol{w} \sim q_\phi(\boldsymbol{w}|\boldsymbol{z}) \quad \Leftrightarrow \quad \boldsymbol{w} \sim q_\phi(\boldsymbol{w})$$

- Any implicit distribution has a semi-implicit formulation $q(\boldsymbol{w}|\boldsymbol{z}) = \delta(\boldsymbol{w} - \boldsymbol{z})$
- Any semi-implicit distribution is implicit

Variational inference with implicit posteriors

	MCMC	IPM	Variational Bayes
Bias	No	???	Strong
Sampling/ Ensembling	Inefficient	???	Efficient
Density	No	???	Yes
Likelihood	Needed	???	Needed

Variational inference with implicit posteriors

	MCMC	IPM	Variational Bayes
Bias	No	Weak	Strong
Sampling/ Ensembling	Inefficient	Efficient	Efficient
Density	No	Can be estimated?	Yes
Likelihood	Needed	Can be avoided	Needed

Hierarchical variational inference

How to perform approximate inference with a semi-implicit posterior?

You should already know this:

- VI with auxiliary variables
- VI in RL with options

Assumptions:

- Can compute densities of both $q(w|z)$ and $q(z)$
- Can reparameterize both $q(w|z)$ and $q(z)$
- Can approximate inverse model $r(z|w) \approx \frac{q(w|z)q(z)}{q(w)}$

$$\mathcal{L} \geq \mathbb{E}_{q(w)} \log p(t, w|X) - \mathbb{E}_{q(z)q(w|z)}[\log q(w|z)q(z) - \log r(z|w)]$$

(Prove it if you didn't do it yet!)

Ranganath, Rajesh, Dustin Tran, and David Blei. "Hierarchical variational models." ICML 2016

Hierarchical variational inference

$$\mathcal{L}(\boldsymbol{\phi}) \geq \underline{\mathcal{L}}(\phi, \theta) = \mathbb{E}_{q_\phi(\boldsymbol{w})} \log p(\boldsymbol{t}, \boldsymbol{w}|X) - \mathbb{E}_{q_\phi(\boldsymbol{w}, \boldsymbol{z})} \log \frac{q_\phi(w|z) q_\phi(z)}{r_\theta(z|w)} \to \max_{\phi, \theta}$$

Additional inference gap:

$$\mathcal{L}(\boldsymbol{\phi}) - \underline{\mathcal{L}}(\phi, \theta) = \mathbb{E}_{q_\phi(w)} \mathrm{KL}\left(r_\theta(z|w) \parallel q_\phi(z|w) \right)$$

Multiplicative normalizing flows

Consider the following semi-implicit posterior:

$$\left.\begin{aligned} q_\phi(\boldsymbol{w}) &= \int q_\phi(\boldsymbol{w}|\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z} \\ q_\phi(w_{ij}|z_i) &= \mathcal{N}\left(w_{ij}|\mu_{ij}z_i, \sigma_{ij}^2\right) \\ q_\phi(\boldsymbol{z}) &= \mathrm{NF}(\boldsymbol{\epsilon}, \boldsymbol{\phi});\;\; \boldsymbol{\epsilon} \sim \mathcal{N}(0, I) \\ r_\theta(z|w) &= \mathrm{NF}(f_\theta(w), \theta) \end{aligned}\right\}$$

Can reparameterize and compute log-density

- Learn non-trivial correlations between neurons
- Not clear how to evaluate the inference gap of HVI
- Normalizing flows are limited
  - Need a lot of depth for complex distributions
  - The width of each layer is the same
- One of the most complex posteriors for Bayesian NNs *yet*

Louizos, Christos, and Max Welling. "Multiplicative normalizing flows for variational bayesian neural networks." *ICML 2017*

Semi-implicit variational inference

A new challenger approaches!

- The rough idea: obtain a "MC" estimate of $q(w)$

$$q(w) = \int q(w|z)q(z)dz = \mathbb{E}_{q(z)}q(w|z) \approx$$

$$\approx q_{1..K}(w|z^1, \dots, z^K) = \frac{1}{K}\sum_{k=1}^{K} q\left(w\middle|z^k\right)$$

- Note that it is asymptotically exact:

$$q_{1..K}(w|z^1, \dots, z^K) \xrightarrow[K\to\infty]{} q(w)$$

- This idea can be used to obtain a sandwich bound for the entropy of $q(w)$:

$$-\mathbb{E}_{z^0,\dots,z^K}\mathbb{E}_{w|z^0}\log\frac{1}{K}\sum_{k=1}^{K} q\left(w\middle|z^k\right) \geq$$

$$\geq -\mathbb{E}_{q(w)}\log q(w) \geq$$

$$\geq -\mathbb{E}_{z^0,\dots,z^K}\mathbb{E}_{w|z^0}\log\frac{1}{K+1}\sum_{k=0}^{K} q(w|z^k)$$

Yin, Mingzhang, and Mingyuan Zhou. "Semi-Implicit Variational Inference." *ICML 2018*

SIVI: upper bound

$$-\mathbb{E}_{z^0,\dots,z^K}\mathbb{E}_{w|z^0}\log\frac{1}{K}\sum_{k=1}^{K} q\left(w\middle|z^k\right) \geq -\mathbb{E}_{q(w)}\log q(w)$$

This is just plain Jensen's inequality ($\log \mathbb{E} \geq \mathbb{E}\log$)

$$-\mathbb{E}_{z^0,\dots,z^K}\mathbb{E}_{w|z^0}\log\frac{1}{K}\sum_{k=1}^{K} q\left(w\middle|z^k\right) = -\mathbb{E}_{z^0,w|z^0}\mathbb{E}_{z^1\dots z^K}\log\frac{1}{K}\sum_{k=1}^{K} q\left(w\middle|z^k\right) \geq$$

$$\geq -\mathbb{E}_{z^0,w|z^0}\log\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{z^k} q\left(w\middle|z^k\right) = -\mathbb{E}_{q(w)}\log q(w)$$

SIVI: lower bound

$$-\mathbb{E}_{q(w)} \log q(w) \geq -\mathbb{E}_{z^0,\ldots,z^K} \mathbb{E}_{w|z^0} \log \frac{1}{K+1} \sum_{k=0}^{K} q(w|z^k)$$

Symmetrize the right side:

$$-\mathbb{E}_{z^0,\ldots,z^K} \mathbb{E}_{w|z^0} \log \frac{1}{K+1} \sum_{k=0}^{K} q\left(w|z^k\right) = -\mathbb{E}_{z^0,\ldots,z^K} \mathbb{E}_{w|z^j} \log \frac{1}{K+1} \sum_{k=0}^{K} q\left(w|z^k\right) =$$

$$= -\frac{1}{K+1} \sum_{j=0}^{K} \mathbb{E}_{z^0,\ldots,z^K} \mathbb{E}_{w|z^j} \log \frac{1}{K+1} \sum_{k=0}^{K} q\left(w|z^k\right) =$$

$$= -\mathbb{E}_{z^0,\ldots,z^K} \int \frac{1}{K+1} \sum_{j=0}^{K} q\left(w|z^j\right) \log \frac{1}{K+1} \sum_{k=0}^{K} q\left(w|z^k\right) dw =$$

$$= -\mathbb{E}_{z^0,\ldots,z^K} \mathbb{E}_{q_{0..K}\left(w|z^0,\ldots,z^K\right)} \log q_{0..K}\left(w|z^0,\ldots,z^K\right)$$

SIVI: lower bound

$$-\mathbb{E}_{q(w)} \log q(w) \geq -\mathbb{E}_{z^0,\dots,z^K} \mathbb{E}_{q_{0..K}\left(w \middle| z^0,\dots,z^K\right)} \log q_{0..K}(w|z^0,\dots,z^K)$$

Rewrite the left side in the same expectations:

$$-\mathbb{E}_{q(w)} \log q(w) = -\mathbb{E}_{z^0,\dots,z^K} \mathbb{E}_{q_{0..K}\left(w \middle| z^0,\dots,z^K\right)} \log q(w)$$

And subtract the right side:

$$-\mathbb{E}_{z^0,\dots,z^K} \mathbb{E}_{q_{0..K}\left(w \middle| z^0,\dots,z^K\right)} \log q(w) +$$

$$+\mathbb{E}_{z^0,\dots,z^K} \mathbb{E}_{q_{0..K}\left(w \middle| z^0,\dots,z^K\right)} \log q_{0..K}(w|z^0,\dots,z^K) =$$

$$= \mathbb{E}_{z^0,\dots,z^K} \mathbb{E}_{q_{0..K}\left(w \middle| z^0,\dots,z^K\right)} \log \frac{q_{0..K}(w|z^0,\dots,z^K)}{q(w)} =$$

$$= \mathbb{E}_{z^0,\dots,z^K} \mathrm{KL}\left(q_{0..K}(w|z^0,\dots,z^K) \parallel q(w)\right) \geq 0$$

Semi-implicit variational inference

- This idea can be used to obtain a sandwich for ELBO
  - $\underline{\mathcal{L}}_K \leq \mathcal{L} \leq \overline{\mathcal{L}}^K$
  - Both $\underline{\mathcal{L}}_K$ and $\overline{\mathcal{L}}^K$ monotonically converge to $\mathcal{L}$
  - We can now estimate ELBO in **any** semi-implicit model!
  - $q(z)$ can be fully implicit (**any** reparameterizable distribution)

Optimize w.r.t. ϕ

$$\underline{\mathcal{L}}_K = \mathbb{E}_w \log p(t, w|X) - \mathbb{E}_{z^0, z^1, \ldots, z^K} \mathbb{E}_{w|z^0} \log \frac{1}{K+1} \sum_{k=0}^{K} q\left(w \middle| z^k\right)$$

$$\overline{\mathcal{L}}^K = \mathbb{E}_w \log p(t, w|X) - \mathbb{E}_{z^0, z^1, \ldots, z^K} \mathbb{E}_{w|z^0} \log \frac{1}{K} \sum_{k=1}^{K} q\left(w \middle| z^k\right)$$

Bound the inference gap

Implicit models: takeaways

- We can now model arbitrarily complex posteriors!
- Applicable to both Bayesian NNs and VAEs

- The next big thing in Bayesian learning!

Bayesian neural networks: takeaways

- Regularization by noise
- Reparameterization
- Local reparameterization
- Deterministic parameters
- Empirical Bayes
- Use implicit model if you want better approximation

Unbiased implicit variational inference

Main idea: we can consider the reparameterization of $q_\phi(z)$ as a part of the conditional $q_\phi(w|z)$:

$$\int q_\phi(w|z)q_\phi(z)dz = \int q_\phi(w|\epsilon)p(\epsilon)d\epsilon$$

$$q_\phi(w|\epsilon) = q_\phi(w|z)\Big|_{z=f_\phi(\epsilon)}$$

The model stays the same, but the joint distribution is now explicit!

$$q_\phi(w,\epsilon) = q_\phi(w|\epsilon)p(\epsilon)$$

We now might be able to run HVI with a fully implicit $q_\phi(z)$…

… Or go even further

Unbiased implicit variational inference

Provides an unbiased gradient estimate for the ELBO

$$\nabla_\phi \mathbb{E}_{q_\phi(w)} \log q_\phi(w) = \nabla_\phi \mathbb{E}_{p(\epsilon)} \log q_\phi\left(f_\phi(\epsilon)\right) =$$

$$= \mathbb{E}_{p(\epsilon)} \nabla_\phi \log q_\phi(w) \Big|_{w=f_\phi(\epsilon)} + \mathbb{E}_{p(\epsilon)} \nabla_w \log q_\phi(w) \Big|_{w=f_\phi(\epsilon)} \cdot \nabla_\phi f_\phi(\epsilon) =$$

$$= \underbrace{\mathbb{E}_{q_\phi(w)} \nabla_\phi \log \mathrm{q}_\phi(\mathrm{w})}_{=0} + \mathbb{E}_{p(\epsilon)} \nabla_w \log q_\phi(w) \Big|_{w=f_\phi(\epsilon)} \cdot \nabla_\phi f_\phi(\epsilon)$$

$$\nabla_w \log q_\phi(w) = \mathbb{E}_{q(\epsilon'|w)} \nabla_\phi \log q_\phi(w|\epsilon')$$

Can be sampled using plain HMC!
We can start HMC using pair (eps, w) to avoid warm-up

Unbiased implicit variational inference: proof?

$$\nabla_w \log q_\phi(w) = \mathbb{E}_{q(\epsilon'|w)} \nabla_w \log q_\phi(w|\epsilon')$$

$$\nabla_w \log q_\phi(w) = \frac{1}{q_\phi(\mathrm{w})} \nabla_{\mathrm{w}} \int q_\phi(w|\epsilon') q(\epsilon') d\epsilon' = \frac{1}{q_\phi(\mathrm{w})} \int \nabla_w q_\phi(w|\epsilon') q(\epsilon')\, d\epsilon' =$$

$$= \frac{1}{q_\phi(\mathrm{w})} \int q_\phi(w|\epsilon') q(\epsilon') \nabla_w \log q_\phi(w|\epsilon')\, d\epsilon' =$$

$$= \int \frac{q_\phi(w|\epsilon') q(\epsilon')}{q_\phi(\mathrm{w})} \nabla_w \log q_\phi(w|\epsilon')\, d\epsilon' =$$

$$= \int q(\epsilon'|w) \nabla_w \log q_\phi(w|\epsilon')\, d\epsilon'$$

Unbiased implicit variational inference

Provides an unbiased gradient estimate for the ELBO

$$\nabla_\phi \mathbb{E}_{q_\phi(w)} \log q_\phi(w) = \mathbb{E}_{p(\epsilon)} \nabla_w \log q_\phi(w) \Big|_{w=f_\phi(\epsilon)} \cdot \nabla_\phi f_\phi(\epsilon)$$

$$\nabla_w \log q_\phi(w) = \mathbb{E}_{q(\epsilon|w)} \nabla_w \log q_\phi(w|\epsilon)$$

- Still cannot estimate ELBO directly (only gradients)
- Can use SIVI bounds to monitor / compare ELBO
- Need sufficiently “nice” $q(\epsilon)$
  - $\mathcal{N}(0, I)$ works fine and is used almost always
  - Discrete $q(\epsilon)$ may be problematic (e.g. no dropout in the implicit generator f_ϕ)