Bayesian Deep Learning

Extending Probabilistic Backpropagation and Transfer Learning

The University of Texas at Austin
Department of Statistics and Data Sciences
College of Natural Sciences

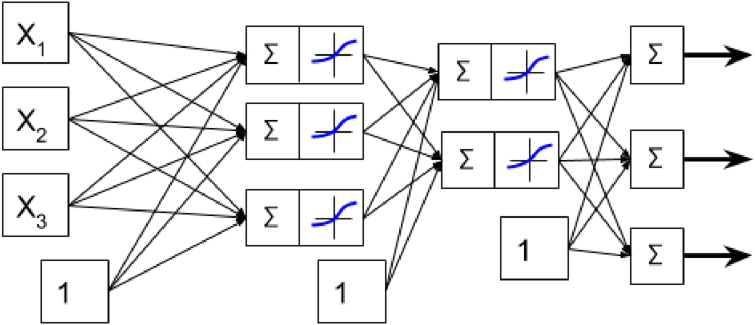
Evan Ott

Advisor: Sinead Williamson
November 2, 2018

Feedforward Neural Networks

For unknown weights $\{W_l\}_{l=1}^{L}$ and biases $\{\mathbf{b}_l\}_{l=1}^{L}$

$$\mathbf{z}_0 = \mathbf{x}$$
$$\mathbf{z}_l = \sigma_l\left(W_l \mathbf{z}_{l-1} + \mathbf{b}_l\right), \quad l = 1 : L$$



Backpropagation (BP)

- BP minimizes a cost function, for example:

$$\min_{\mathcal{W}} \sum_{i=1}^{N} \|y_n - \mathtt{NN}(\mathbf{x}_n; \mathcal{W})\|_2^2$$

- Cleverly applies chain rule of derivatives

$$y = e^{-wx}, \qquad z = \frac{1}{1+e^{-y}}$$

$$\frac{dz}{dy} = z(1-z), \qquad \frac{dz}{dw} = \frac{dz}{dy}\frac{dy}{dw} = -yx\frac{dz}{dy}$$

$$= -\frac{xe^{-\exp[-wx]-wx}}{\left(e^{-\exp[-wx]}+1\right)^2}$$

Deep Neural Networks

Advantages:

- Fast to train (e.g., SGD with backpropagation)
- Can achieve high accuracy/precision/recall
  - Identifying objects in images (Szegedy et al. 2015)
  - Melanoma detection from images (Esteva et al. 2017)
  - Tuberculosis detection from chest x-rays (Lakhani and Sundaram 2017)

Drawbacks:

- Typically, only provides point estimates
- Tendency for overfitting
- Unclear choice of structure

Christian Szegedy et al. Going Deeper with Convolutions. In: *Computer Vision and Pattern Recognition*. 2015.

Andre Esteva et al. Dermatologist-level Classification of Skin Cancer with Deep Neural Networks. In: *Nature* 542.7639 (2017), p. 115.

Paras Lakhani and Baskaran Sundaram. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by using Convolutional Neural Networks. In: *Radiology* 284.2 (2017), pp. 574–582.

Our Bayesian Neural Network Model

Standard Neural Network

$$\begin{aligned}
\hat{y}_n &= \mathtt{NN}\left(\mathbf{x}_n; \mathcal{W}\right) \\
&= \mathbf{z}_L \\
\mathbf{z}_l &= \sigma_l\left(W_l \mathbf{z}_{l-1} + \mathbf{b}_l\right) \\
\mathbf{z}_0 &= \mathbf{x}_n
\end{aligned}$$

Bayesian Neural Network

$$\begin{aligned}
Y_n | \mathcal{W}, \gamma &\sim \mathrm{N}\left(\mathtt{NN}\left(\mathbf{x}_n; \mathcal{W}\right), \gamma^{-1}\right) \\
W_{ij,l} | \lambda &\sim \mathrm{N}\left(0, \lambda^{-1}\right) \\
\gamma &\sim \mathrm{Ga}\left(\alpha_0^\gamma, \beta_0^\gamma\right) \\
\lambda &\sim \mathrm{Ga}\left(\alpha_0^\lambda, \beta_0^\lambda\right)
\end{aligned}$$

- Fixed element-wise activation functions $(\sigma_l(\mathbf{z}))_i = \mathrm{ReLU}(z_i) = \max(z_i, 0)$
- Final layer's range is the real line $\sigma_L(\mathbf{z}) = \mathbf{z}$

Bayesian Neural Networks (BNNs)

The problem:

- Posterior (or posterior predictive, etc.) is intractable
- MCMC possible for small networks (Neal 1993)

Methods used for BNN inference:

- **Assumed density filtering**
- Dropout as deep GP (Gal and Ghahramani 2016)
- Expectation propagation (Soudry et al. 2014)
- **Laplace approximation**
- **Variational inference**

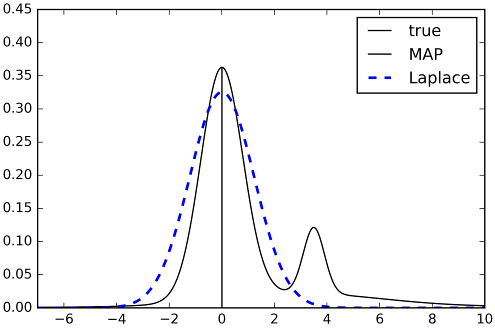
Radford M Neal. Bayesian Learning via Stochastic Dynamics. In: *Advances in Neural Information Processing Systems*. 1993, pp. 475–482.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: *International Conference on Machine Learning*. 2016, pp. 1050–1059.

Daniel Soudry et al. Expectation Backpropagation: Parameter-free Training of Multilayer Neural Networks with Continuous or Discrete Weights. In: *Advances in Neural Information Processing Systems*. 2014, pp. 963–971.

Laplace Approximation

- Approximate posterior introduced by (MacKay 1992)
- Identify MAP estimate by standard backpropagation
- Locally-quadratic approximation to form a Gaussian
  - Requires computing Hessian



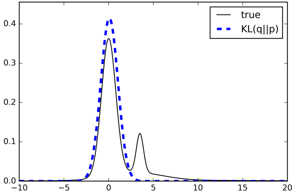
David JC MacKay. A Practical Bayesian Framework for Backpropagation Networks. In: *Neural Computation* 4.3 (1992), pp. 448–472.

Variational Inference

- Posit variational family $\mathcal{Q}$ to approximate $p(W|\mathcal{D})$
- Identify $q(W) \in \mathcal{Q}$ that minimizes

$$KL(q(W)\|p(W|\mathcal{D})) = \int_{\mathcal{W}} q(W) \log \left(\frac{q(W)}{p(W|\mathcal{D})} \right) dW$$

- Applied to BNNs by (Graves 2011) with MCMC likelihood, see also (Blundell et al. 2015)



Alex Graves. Practical Variational Inference for Neural Networks. In: *Advances in Neural Information Processing Systems*. 2011, pp. 2348–2356.

Charles Blundell et al. Weight Uncertainty in Neural Networks. In: *arXiv preprint arXiv:1505.05424* (2015).

Assumed Density Filtering

Approximate Bayesian approach to online learning (Opper and Winther 1998)

- Approximate posterior $q(\theta|\gamma_t)$ at iteration t with parameters γ_t
  - For example, if q is Gaussian, $\gamma_t = (\mu_t, \sigma_t^2)$
- Given new data y_{t+1}:

  Update Update "exact" posterior:

$$p(\theta|y_{t+1}, \gamma_t) = \frac{p(y_{t+1}|\theta)q(\theta|\gamma_t)}{\int p(y_{t+1}|\theta)q(\theta|\gamma_t)d\theta}$$

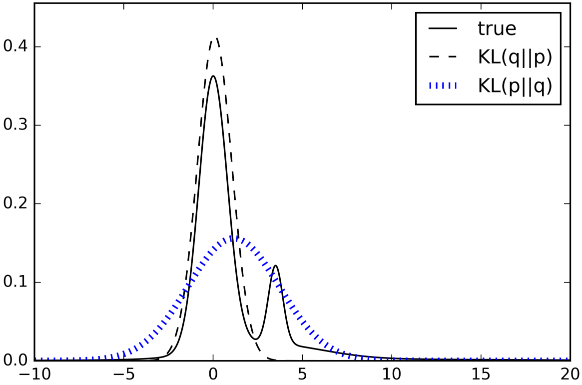
  Projection $\gamma_{t+1} := \arg\min_{\gamma} KL\left(p(\cdot|y_{t+1}, \gamma_t) \parallel q(\cdot|\gamma)\right)$

Manfred Opper and Ole Winther. A Bayesian Approach to On-line Learning. In: *On-line Learning in Neural Networks, ed. D. Saad* (1998), pp. 363–378.

Assumed Density Filtering

For $q(\theta) = \mathcal{N}(\theta | m, v)$:

$$\mathbb{E}_{q^*}[\Theta] = \mathbb{E}_p[\Theta] \qquad \mathbb{V}_{q^*}[\Theta] = \mathbb{V}_p[\Theta]$$



Probabilistic Backpropagation (PBP)

ADF algorithm for BNNs (Hernández-Lobato and Adams 2015)

- Indep. Gaussian approximation for online posterior: $q(w_{ij,l}) = \mathcal{N}(w_{ij,l} | m_{ij,l}, v_{ij,l})$
- Need to compute moments of “exact” posterior to find $\tilde{q}(w) = \mathcal{N}(w | \tilde{m}, \tilde{v})$
- Use Gaussian properties (Minka 2001) in ADF “exact” step:

$$p(w|y) = Z^{-1} f(y|w) \mathcal{N}(w|m,v)$$

$$\mathbb{E}_{\tilde{q}}[W] = \tilde{m} = \mathbb{E}_p[W] = m + v \frac{\partial \log Z}{\partial m}$$

$$\mathbb{V}_{\tilde{q}}[W] = \tilde{v} = \mathbb{V}_p[W] = v - v^2 \left[\left(\frac{\partial \log Z}{\partial m} \right)^2 - 2 \frac{\partial \log Z}{\partial v} \right]$$

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In: *International Conference on Machine Learning*. 2015, pp. 1861–1869.

Thomas Peter Minka. A family of algorithms for approximate Bayesian inference. PhD thesis. Massachusetts Institute of Technology, 2001.

Probabilistic Backpropagation (PBP)

Closed-form approximation for normalization constant:

$$
\begin{aligned}
Z &= \int \mathcal{N}(y_n | \mathtt{NN}(\mathbf{x}_n; \mathcal{W}), \gamma^{-1}) q(\mathcal{W}, \gamma, \lambda) d\mathcal{W} d\gamma d\lambda \\
&\approx \int \mathcal{N}(y_n | z_L, \gamma^{-1}) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) \mathrm{Ga}(\gamma | \alpha^\gamma, \beta^\gamma) dz_L d\gamma \\
&= \int \mathcal{T}(y_n | z_L, \beta^\gamma / \alpha^\gamma, 2\alpha^\gamma) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) dz_L \\
&\approx \int \mathcal{N}(y_n | z_L, \beta^\gamma / (\alpha^\gamma - 1)) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) dz_L \\
&= \mathcal{N}(y_n | m^{z_L}, \beta^\gamma / (\alpha^\gamma - 1) + v^{z_L})
\end{aligned}
$$

Probabilistic Backpropagation (PBP)

Closed-form approximation for normalization constant:

$$
\begin{aligned}
Z &= \int \mathcal{N}(y_n | \mathtt{NN}(\mathbf{x}_n; \mathcal{W}), \gamma^{-1}) q(\mathcal{W}, \gamma, \lambda) d\mathcal{W} d\gamma d\lambda \\
&\approx \int \mathcal{N}(y_n | z_L, \gamma^{-1}) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) \mathrm{Ga}(\gamma | \alpha^\gamma, \beta^\gamma) dz_L d\gamma \\
&= \int \mathcal{T}(y_n | z_L, \beta^\gamma / \alpha^\gamma, 2\alpha^\gamma) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) dz_L \\
&\approx \int \mathcal{N}(y_n | z_L, \beta^\gamma / (\alpha^\gamma - 1)) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) dz_L \\
&= \mathcal{N}(y_n | m^{z_L}, \beta^\gamma / (\alpha^\gamma - 1) + v^{z_L})
\end{aligned}
$$

Probabilistic Backpropagation (PBP)

Closed-form approximation for normalization constant:

$$
\begin{aligned}
Z &= \int \mathcal{N}(y_n|\mathtt{NN}(\mathbf{x}_n;\mathcal{W}),\gamma^{-1})q(\mathcal{W},\gamma,\lambda)d\mathcal{W}d\gamma d\lambda \\
&\approx \int \mathcal{N}(y_n|z_L,\gamma^{-1})\mathcal{N}(z_L|m^{z_L},v^{z_L})\mathrm{Ga}(\gamma|\alpha^\gamma,\beta^\gamma)dz_L d\gamma \\
&= \int \mathcal{T}(y_n|z_L,\beta^\gamma/\alpha^\gamma,2\alpha^\gamma)\mathcal{N}(z_L|m^{z_L},v^{z_L})dz_L \\
&\approx \int \mathcal{N}(y_n|z_L,\beta^\gamma/(\alpha^\gamma-1))\mathcal{N}(z_L|m^{z_L},v^{z_L})dz_L \\
&= \mathcal{N}(y_n|m^{z_L},\beta^\gamma/(\alpha^\gamma-1)+v^{z_L})
\end{aligned}
$$

Probabilistic Backpropagation (PBP)

Closed-form approximation for normalization constant:

$$
\begin{aligned}
Z &= \int \mathcal{N}(y_n | \mathtt{NN}(\mathbf{x}_n; \mathcal{W}), \gamma^{-1}) q(\mathcal{W}, \gamma, \lambda) d\mathcal{W} d\gamma d\lambda \\
&\approx \int \mathcal{N}(y_n | z_L, \gamma^{-1}) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) \mathrm{Ga}(\gamma | \alpha^\gamma, \beta^\gamma) dz_L d\gamma \\
&= \int \mathcal{T}(y_n | z_L, \beta^\gamma / \alpha^\gamma, 2\alpha^\gamma) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) dz_L \\
&\approx \int \mathcal{N}(y_n | z_L, \beta^\gamma / (\alpha^\gamma - 1)) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) dz_L \\
&= \mathcal{N}(y_n | m^{z_L}, \beta^\gamma / (\alpha^\gamma - 1) + v^{z_L})
\end{aligned}
$$

Probabilistic Backpropagation (PBP)

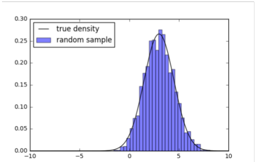
Closed-form approximation for normalization constant:

$$
\begin{aligned}
Z &= \int \mathcal{N}(y_n | \mathtt{NN}(\mathbf{x}_n; \mathcal{W}), \gamma^{-1}) q(\mathcal{W}, \gamma, \lambda) d\mathcal{W} d\gamma d\lambda \\
&\approx \int \mathcal{N}(y_n | z_L, \gamma^{-1}) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) \mathrm{Ga}(\gamma | \alpha^\gamma, \beta^\gamma) dz_L d\gamma \\
&= \int \mathcal{T}(y_n | z_L, \beta^\gamma / \alpha^\gamma, 2\alpha^\gamma) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) dz_L \\
&\approx \int \mathcal{N}(y_n | z_L, \beta^\gamma / (\alpha^\gamma - 1)) \mathcal{N}(z_L | m^{z_L}, v^{z_L}) dz_L \\
&= \mathcal{N}(y_n | m^{z_L}, \beta^\gamma / (\alpha^\gamma - 1) + v^{z_L})
\end{aligned}
$$

Probabilistic Backpropagation (PBP)

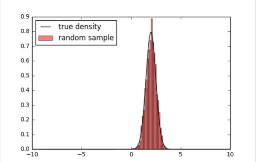
Compute m^{z_l}, v^{z_l} by minimizing KL between true distribution and a Gaussian for each step in network

Illustration with simulated data:



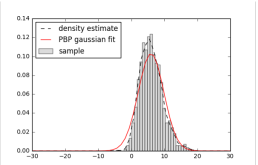
X~N(3, 2.25)

X

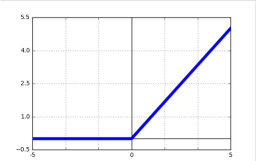


W~N(2, 0.25)

=

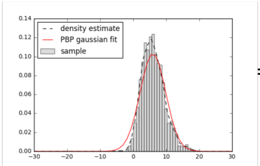


p(XW)



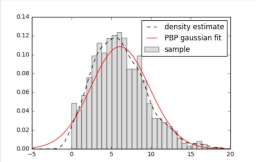
f(z)=max(0, z)

O



p(XW)

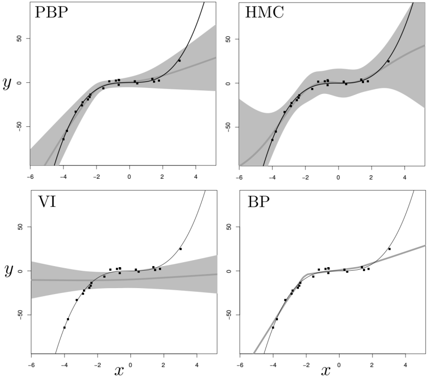
=



p(f(XW))

PBP Properties

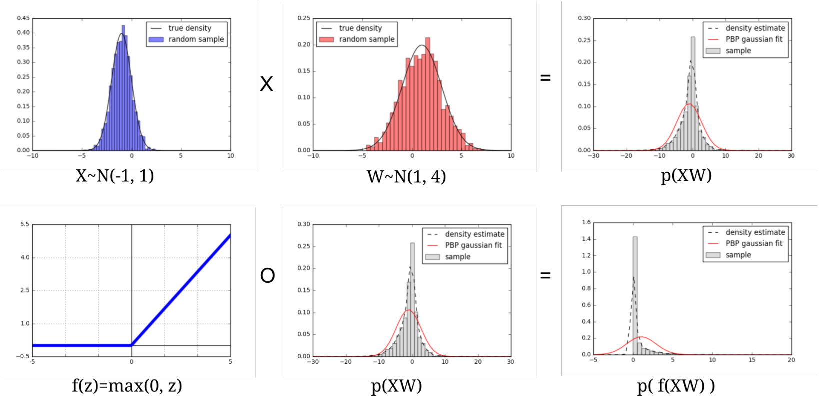
- Trains like standard DNN, so it's fast
- Extended by (Ghosh et al. 2016) to binary classification (probit) and multiclass via MCMC step



Soumya Ghosh et al. Assumed Density Filtering Methods for Learning Bayesian Neural Networks. In: *AAAI Conference on Artificial Intelligence*. 2016, pp. 1589–1595.

Hernández-Lobato and Adams 2015, Figure 1.

Current Work



Led to questions about Gaussian approximation - replace with spike and slab?

$$Z_l \sim (1 - \pi^{z_l})\delta_0 + \pi^{z_l}\mathcal{N}(m^{z_l}, v^{z_l})$$

Spike-and-slab PBP

In order to sequentially compute the parameters of $q(z_l)$, need:

- Conditions to minimize $KL(p||q)$ for $q = (1-\pi)\delta_0 + \pi\mathcal{N}(m, v)$
  - Spoiler alert, need $\mathbb{P}_p[Z=0], \mathbb{E}_p[Z], \mathbb{V}_p[Z]$
- Compute moments of p with closed-form expressions for linear combination and activation steps
- Compute normalization constant Z

KL Minimization

For $q(z;\pi,m,v) = (1-\pi)\delta_0(z) + \pi\mathcal{N}(z;m,v)$, need to minimize KL:

$$
\begin{aligned}
0 &= \frac{d}{dm} KL(p\|q) = -\frac{d}{dm}\int_{\mathbb{R}} p(z)\log(q(z))dz \\
&= -\int_{\mathbb{R}} p(z)\frac{\frac{\partial}{\partial m}\left((1-\pi)\delta_0(z) + \pi\mathcal{N}(z;m,v)\right)}{(1-\pi)\delta_0(z) + \pi\mathcal{N}(z;m,v)}dz \\
&= -\frac{1}{v}\left(\mathbb{E}_p\left[Z\right] - m\right) + \int_{\mathbb{R}} p(z)\frac{(1-\pi)\delta_0(z)\left(\frac{z-m}{v}\right)}{(1-\pi)\delta_0(z) + \pi\mathcal{N}(z;m,v)}dz
\end{aligned}
$$

If all mass on slab, recover $m = \mathbb{E}_p[Z]$

KL Minimization

Approximate δ_0 as limiting distribution of $u_a = \mathrm{Unif}([-1/(2a), 1/(2a)])$ as $a \to \infty$:

$$\begin{aligned}
I &= \lim_{a\to\infty} \int_{\mathbb{R}} p(z) \frac{(1-\pi)u_a(z)\left(\frac{z-m}{v}\right)}{(1-\pi)u_a(z) + \pi\mathcal{N}(z; m, v)} dz \\
&= \lim_{a\to\infty} \int_{-1/2a}^{1/2a} p(z) \frac{(1-\pi)a\left(\frac{z-m}{v}\right)}{(1-\pi)a + \pi\mathcal{N}(z; m, v)} dz
\end{aligned}$$

Introduce approximation for $(1-\pi)a \gg \pi\mathcal{N}(z; m, v)$:

$$\begin{aligned}
I &\approx \lim_{a\to\infty} \int_{-1/2a}^{1/2a} p(z) \frac{(1-\pi)a\left(\frac{z-m}{v}\right)}{(1-\pi)a} dz \\
&= \lim_{\epsilon\to 0^+} \int_{-\epsilon}^{\epsilon} p(z) \left(\frac{z-m}{v}\right) dz = -\frac{m}{v}\mathbb{P}_p[Z = 0]
\end{aligned}$$

KL Minimization

Recover moment-matching along with spike-probability matching:

$$\mathbb{P}_q[Z=0] = \mathbb{P}_p[Z=0] \qquad \tilde{\pi} = 1 - \mathbb{P}_p[Z=0]$$

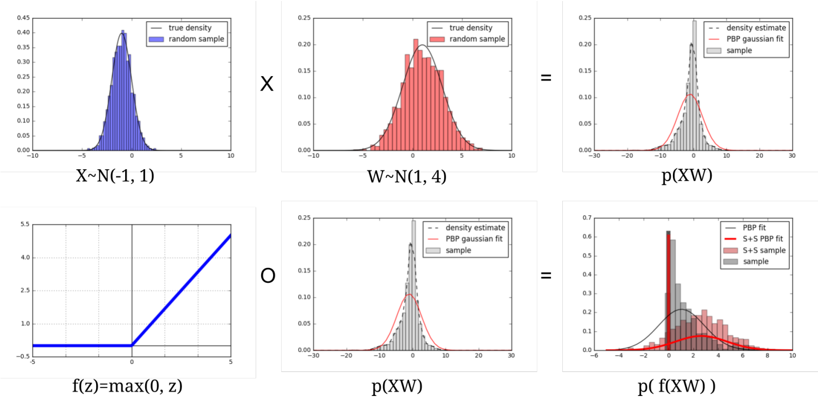
$$\mathbb{E}_q[Z] = \mathbb{E}_p[Z] \qquad \tilde{m} = \frac{1}{\tilde{\pi}}\mathbb{E}_p[Z]$$

$$\mathbb{V}_q[Z] = \mathbb{V}_p[Z] \qquad \tilde{v} = \frac{\mathbb{V}_p[Z] - \tilde{\pi}(1-\tilde{\pi})\tilde{m}^2}{\tilde{\pi}}$$

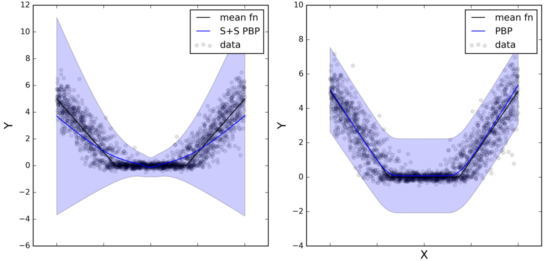
Obtain modified normalization constant, but same posterior update rules:

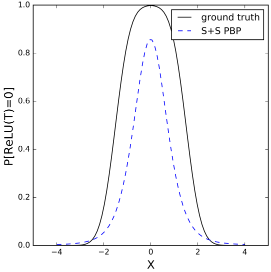
$$\begin{aligned} Z = (1-\pi^{z_L})&\mathcal{N}(y_n|0, \beta^\gamma/(\alpha^\gamma - 1)) \\ &+ \pi^{z_L}\mathcal{N}(y_n|m^{z_L}, \beta^\gamma/(\alpha^\gamma - 1) + v^{z_L}) \end{aligned}$$

Forward Pass Approximation



Toy Data





PBP Training Average LL: -1.1353

S+S PBP Training Average LL: -0.9745

$$Y|T = \mathcal{N}(\mathrm{ReLU}(T), 0.02)$$
$$T \sim \mathcal{N}(2|x| - 3, 1)$$

- PBP matches mean function more closely
- S+S PBP leverages sparsity for higher log-likelihood

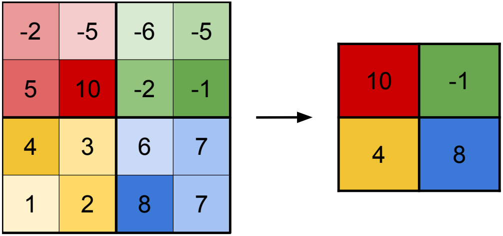
Future Work

Classification Alternative to softmax without MCMC?

$$\hat{p}_i = \mathrm{Softmax}_i(\mathbf{x}) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

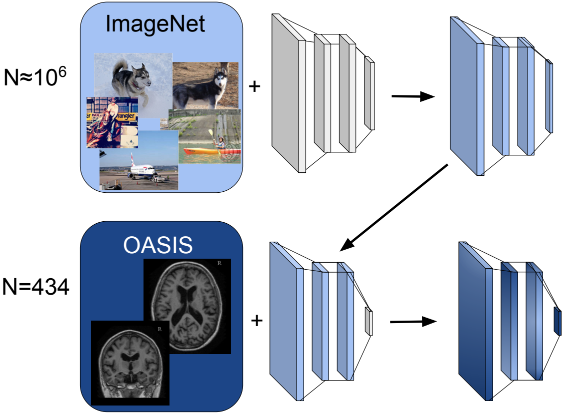
$$\tilde{p}_i = p(\mathtt{NN}(\mathbf{x}; \mathcal{W}) \in \mathcal{A}_i)$$

Pooling Need approximation for $p\left(\max\left(X_1, X_2, \ldots, X_k\right)\right)$



Future Work

Bayesian version of transfer learning



Olga Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

Daniel S Marcus et al. Open Access Series of Imaging Studies: Longitudinal MRI Data in Nondemented and Demented Older Adults. In: *Journal of Cognitive Neuroscience* 22.12 (2010), pp. 2677–2684.

Thanks

Questions?

This presentation:
https://www.evanott.com/research/sds_seminar_2018.pdf

References I

- Blundell, Charles, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. In: *arXiv preprint arXiv:1505.05424* (2015).
- Esteva, Andre, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level Classification of Skin Cancer with Deep Neural Networks. In: *Nature* 542.7639 (2017), p. 115.
- Gal, Yarin and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: *International Conference on Machine Learning*. 2016, pp. 1050–1059.
- Ghosh, Soumya, Francesco Maria Delle Fave, and Jonathan S Yedidia. Assumed Density Filtering Methods for Learning Bayesian Neural Networks. In: *AAAI Conference on Artificial Intelligence*. 2016, pp. 1589–1595.

References II

- Graves, Alex. Practical Variational Inference for Neural Networks. In: *Advances in Neural Information Processing Systems*. 2011, pp. 2348–2356.
- Hernández-Lobato, José Miguel and Ryan Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In: *International Conference on Machine Learning*. 2015, pp. 1861–1869.
- Lakhani, Paras and Baskaran Sundaram. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by using Convolutional Neural Networks. In: *Radiology* 284.2 (2017), pp. 574–582.
- MacKay, David JC. A Practical Bayesian Framework for Backpropagation Networks. In: *Neural Computation* 4.3 (1992), pp. 448–472.

References III

- Marcus, Daniel S, Anthony F Fotenos, John G Csernansky, John C Morris, and Randy L Buckner. Open Access Series of Imaging Studies: Longitudinal MRI Data in Nondemented and Demented Older Adults. In: *Journal of Cognitive Neuroscience* 22.12 (2010), pp. 2677–2684.
- Minka, Thomas Peter. A family of algorithms for approximate Bayesian inference. PhD thesis. Massachusetts Institute of Technology, 2001.
- Neal, Radford M. Bayesian Learning via Stochastic Dynamics. In: *Advances in Neural Information Processing Systems*. 1993, pp. 475–482.
- Opper, Manfred and Ole Winther. A Bayesian Approach to On-line Learning. In: *On-line Learning in Neural Networks, ed. D. Saad* (1998), pp. 363–378.

References IV

- Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.
- Soudry, Daniel, Itay Hubara, and Ron Meir. Expectation Backpropagation: Parameter-free Training of Multilayer Neural Networks with Continuous or Discrete Weights. In: *Advances in Neural Information Processing Systems*. 2014, pp. 963–971.

References V

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In: *Computer Vision and Pattern Recognition*. 2015.

Hamiltonian Monte Carlo

- Let q be the parameters of our distribution $P(q)$
- Define potential energy $E(q)$ as $P(q) \propto \exp(-E(q))$
- Augment space to include momentum vector p, same dimension as q
- Define Hamiltonian $H(q,p) = E(q) + \frac{1}{2}\|p\|_2^2$
- Use Hamiltonian dynamics for equal-energy trajectories:

$$\frac{dq}{dt} = \frac{\partial H}{\partial p} = p \qquad\qquad \frac{dp}{dt} = -\frac{\partial H}{\partial q} = -\nabla E(q)$$

- Use log posterior $-\log P(q) = -\log f(X|q) - \log \pi(q) + \log p(X)$
- Find valid state, give it a kick, follow trajectory, move via Metropolis-Hastings.

Dropout as Variational Inference

(Gal and Ghahramani 2016)

- Stochastically set nodes in network to 0
- Connection to deep Gaussian process
- Really, dropout is a regularizer
- Matt Taddy and others: variational dropout provides poor variance estimates

Gal and Ghahramani 2016.

Gaussian Properties

(Minka 2001) showed for $q(\theta) = \mathcal{N}(\theta|m, v)$

$$\begin{aligned}
p(\theta|x) &= Z^{-1} f(x|\theta) q(\theta) \\
Z &= \int_\theta f(x|\theta) q(\theta) d\theta \\
\frac{d}{dm} \log Z &= \frac{1}{Z} \int_\theta \frac{(\theta - m)}{v} f(x|\theta) \frac{1}{\sqrt{2\pi v}} \exp\left[-\frac{(\theta - m)^2}{2v} \right] d\theta \\
&= \int_\theta \frac{(\theta - m)}{v} \frac{f(x|\theta) q(\theta)}{Z} d\theta \\
&= \int_\theta \frac{(\theta - m)}{v} p(\theta|x) d\theta = \frac{1}{v} (\mathbb{E}_p[\Theta] - m) \\
\mathbb{E}_p[\Theta] &= m + v \frac{d}{dm} \log Z
\end{aligned}$$

Minka 2001.