Uncertainty via Stochastic Variational Inference

Paul Ursulean
Department of Informatics
Technische Universität München

I. Introduction

UNCERTAINTY is a central concept within statistical modeling, referring to circumstances involving incomplete information as a result of partial observability, non-determinism, or both.[1] In the context of supervised machine learning, a model's ability to quantify the uncertainty of a prediction runs concurrent to its ability to make correct predictions. However, standard deep neural networks do not capture model uncertainty, but rather attempt to provide a point estimation for the true distribution underlying the prediction.[2] In classification tasks the output class probabilities are sometimes erroneously interpreted as model confidence, though true uncertainty quantification in this case would involve a probability distribution over those predictive probabilities. While simple neural networks are able to produce predictive class probabilities that reflect the true correctness likelihood, modern architectures such as the ResNet[3] exhibit a significant mismatch between the two.[4] This phenomenon defines the concept of uncalibrated confidence, which has received significant attention in research[5,6,7] since deep neural networks have started to take on important responsibilities such as driving.[8]

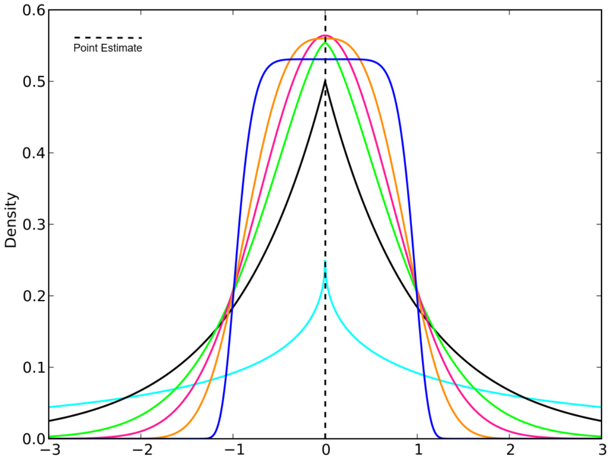


Fig. 1: The same point estimation corresponds to the peak of infinitely many distributions. Using point estimations therefore collapses most of the nuance of the underlying distribution, including the confidence of the estimation, characterized by the general spread and shape of the distribution.

This report will serve as an overview of variational inference as it relates to uncertainty quantification and Bayesian neural networks. Chapter II provides the theoretical foundations that build up to variational inference. Chapter III takes a more practical view on the theoretical foundations laid out previously, and provides implementation tricks necessary for a minimal working example of variational inference at scale. Chapter IV then shows how these building blocks come together with two different model types that employ stochastic variational inference, and their benefits. Lastly, chapters V-VI look to the future by showcasing more recent innovations along with their respective advantages and disadvantages, as well as reflecting on the trajectory of variational inference and the niche it fills in the field of artificial intelligence.

II. Theoretical Background

A. Latent Variable Models

The goal of supervised learning is to learn a function that maps one or more input features to one or more target variables, given an observed set of input-output pairs.[1] In probabilistic terms, this ideally corresponds to obtaining a joint probability distribution of the target variables over the feature space, which models the uncertainty inherent to the general incompleteness of data, in terms of both the finite number of observations, and the lack of perfect causality between features and targets. The problem with explicitly learning this joint distribution is that its dimensionality grows linearly with the number of features and targets, as a result of the arbitrary variable inter-dependencies it models.

$$p(x_{1:N}) = p(x_1)\, p(x_2|x_1)\, p(x_3|x_2, x_1)\, p(x_4|x_3, x_2, x_1) \dots\, p(x_N|x_{1:N-1})$$

In order to reduce the complexity of the joint distribution to the realm of tractability, simplifying assumptions have to be made, such as the Naive Bayes assumption of conditional independence of the features given the target variable. These assumptions are often not realistic, and reduce the expressivity of the resulting model.[9]

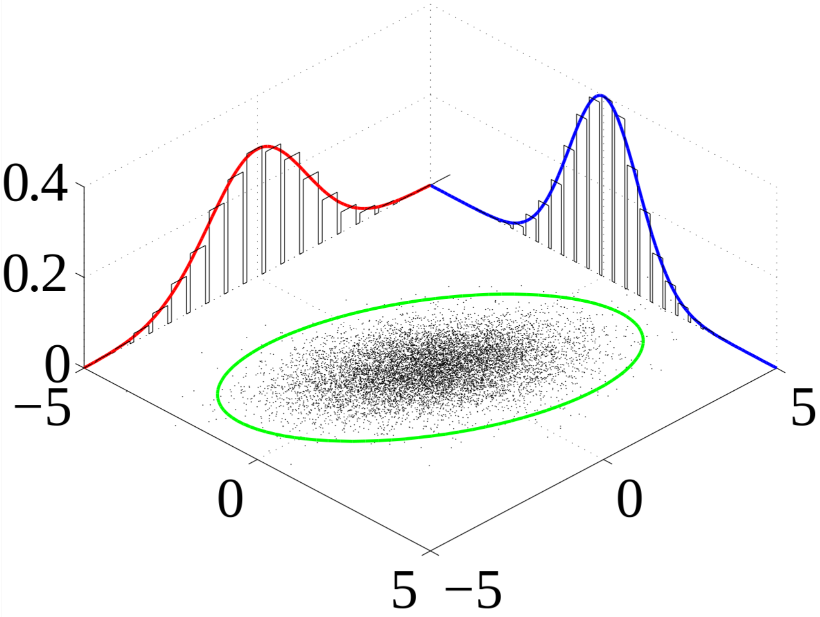


Fig. 2: The joint probability distribution (green) models the correlation between the two variables, while their marginal distributions (red and blue) are perfectly Gaussian. This simplistic example is tractable to compute given just the raw data (black dots), but the complexity of this task quickly explodes with higher dimensionalities and irregular distributions.

Latent variable models attempt to break the complexity of the task down into manageable chunks, by making the assumption of an underlying latent structure from which the observations stem. The model then evaluates the conditional distribution of the observations given the latent variables, while the training procedure attempts to learn the distribution over the latent variables given the observations.

B. Bayesian Inference

Bayesian inference is a statistical inference method used in machine learning to obtain the full posterior probability distribution over some latent parameters, given a set of observations. This comes at the cost of computational complexity, which often renders the search for an exact solution infeasible, relegating high-dimensional models to approximative methods such as the maximum likelihood (MLE) or maximum a priori (MAP) estimations.[9]

$$p(\theta|\mathbf{X},\alpha) = \frac{\overbrace{p(\mathbf{X}|\theta,\alpha)}^{\text{likelihood}} \cdot \overbrace{p(\theta|\alpha)}^{\text{prior}}}{\underbrace{p(\mathbf{X}|\alpha)}_{\text{evidence}}} \propto p(\mathbf{X}|\theta,\alpha) \cdot p(\theta|\alpha)$$

$$p(\mathbf{X}\,|\,\alpha) = \int p(\mathbf{X}\,|\,\theta)\, p(\theta\,|\,\alpha)\, d\theta$$

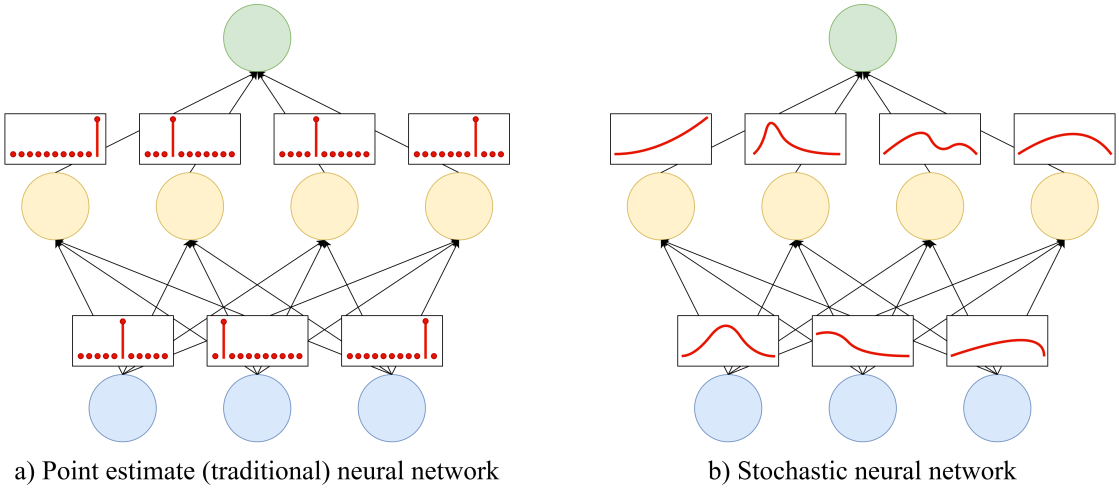
$$\theta_{\text{MLE}} = \arg\max_{\theta}\; p(\mathbf{X}\,|\,\theta,\alpha)$$

$$\theta_{\text{MAP}} = \arg\max_{\theta}\; p(\mathbf{X}\,|\,\theta,\alpha)\, p(\theta\,|\,\alpha)$$

Eq. 3: Given prior α and observations $\mathbf{X} = [\vec{x}_1, \ldots, \vec{x}_n]^\top$, with $\vec{x}_i \sim p(\vec{x}|\theta)$, Bayes' rule gives the posterior probability distribution for latent variable(s) θ. The evidence $p(\mathbf{X}|\alpha)$ demands integration over the entire latent space, rendering the exact computation of the posterior intractable and justifying the widespread avoidance of Bayesian inference in favor of the θ_{MLE} and θ_{MAP} estimations.

C. Bayesian Neural Networks

Stochastic Neural Networks are artificial neural networks with incorporated stochastic components.[10] This definition allows for the variation in techniques found in literature, such as stochastic activation functions,[11] weights,[12] and inference values.[12,13] Bayesian Neural Networks are then defined as stochastic neural networks trained with Bayesian inference.[10]


a) Point estimate (traditional) neural network

b) Stochastic neural network

Fig. 4: Traditional feed-forward networks approximate a point estimate during training, collapsing all of the nuance in the underlying distribution to a single point, thus underestimating the true variance of the weights.[10] Leveraging this additional information, Bayesian neural networks are shown to better calibrate their output to the real probabilities corresponding to the target variable.[14,15,16] Note that this does not hold for stochastic neural networks in general, as a network with random weight distributions would be classified as a stochastic neural network. Bayesian neural networks are by definition trained using Bayesian inference, resulting in meaningful weight distributions.

$$p(\tilde{x} \,|\, \mathbf{X}, \alpha) = \int p(\tilde{x} \,|\, \theta)\, p(\theta \,|\, \mathbf{X}, \alpha)\, d\theta$$

Eq. 5: Given a new data point $\tilde{x}$, marginalizing over the latent space gives the posterior predictive distribution of $\tilde{x}$, consisting of its probability $p(\tilde{x} \,|\, \theta)$ under the latent variables, and the posterior $p(\theta \,|\, \mathbf{X}, \alpha)$. In the context of Bayesian Neural Networks, the latent variables can be defined as the weights of the network, meaning that $p(\tilde{x} \,|\, \theta)$ is simply a forward pass of $\tilde{x}$ through the network. As illustrated in Eq. 3, even a point-wise evaluation of the posterior is intractable due to its dependency on the evidence term $p(\mathbf{X}|\alpha)$, therefore this formulation is still not suitable for training BNNs in practice.

D. Variational Inference

Variational methods were adopted in the late 1980s to serve as an alternative to Monte-Carlo sampling techniques as a solution for the intractable posterior problem.[17,18,19] Variational inference, the resulting technique, formulates the approximation of the posterior distribution as an optimization problem aiming to find the closest distribution to the real posterior, from a restricted family of comparatively simple distributions.[19]

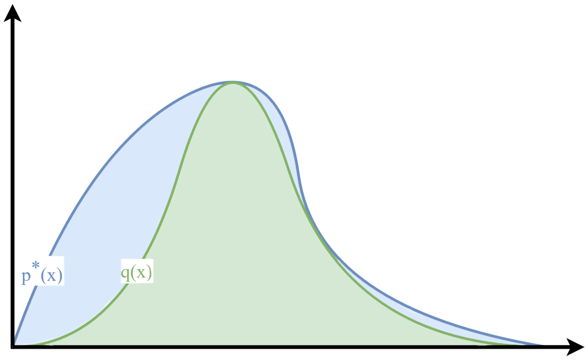


Fig. 6: Minimizing the Kullback-Leibler divergence to the true posterior p^* gives its closest distribution q from a tractable family of parametric distributions, which can then be used as an approximation for p^*.

$$\begin{aligned}\mathbb{KL}(p^*||q) &= \int p^*(\theta)\log\frac{p^*(\theta)}{q(\theta|\lambda)}\,d\theta\\ \neq\ \mathbb{KL}(q||p^*) &= \int q(\theta|\lambda)\log\frac{q(\theta|\lambda)}{p^*(\theta)}\,d\theta\\ &= \mathbb{E}_q\left[\log\frac{q(\theta|\lambda)}{p^*(\theta)}\right]\end{aligned}$$

Eq. 7: The Kullback-Leibler divergence[20] is used as a measure of distance between the true intractable distribution $p^*(\theta)$ and an approximate distribution $q(\theta|\lambda)$ chosen from a parametric family of distributions. Intuitively, the Kullback-Leibler divergence represents the expectation of the log difference between the two probabilities, with respect to the first probability distribution. This makes it favorable to use $\mathbb{KL}(q||p^*)$, since the expectation with respect to q is tractable, while the opposite is not.[9] Even so, the formulation remains problematic due to the dependency of $p^*(\theta) = p(\theta|\mathbf{X}, \alpha)$ on the evidence term $p(\mathbf{X}|\alpha)$. [Eq. 3] A discussion on the asymmetry of the $\mathbb{KL}$ operator can be found in Appendix A.

$$\begin{aligned}J(q) &= \mathbb{KL}(q||\tilde{p})\\ &= \int q(\theta|\lambda)\log\frac{q(\theta|\lambda)}{\tilde{p}(\theta)}\,d\theta\\ &= \int q(\theta|\lambda)\log\frac{q(\theta|\lambda)}{p^*(\theta)\cdot Z}\,d\theta\\ &= \mathbb{KL}(q||p^*) - \log Z\end{aligned}$$

Fig. 8: The variational objective $J(q)$ is instead defined with respect to the unnormalized distribution $\tilde{p}(\theta) = p(\mathbf{X}|\theta,\alpha)\cdot p(\theta|\alpha) = p^*(\theta)\cdot Z$. Unlike $p^*(\theta)$, $\tilde{p}(\theta)$ is tractable, and this formulation enables the minimization of the distance between q and p^*, without requiring the explicit evaluation of $p^*(\theta)$.

$$\begin{gathered}J(q) = \mathbb{KL}(q||p^*) - \log Z \geq -\log Z\\ \Rightarrow J(q) \geq -\log p(\mathbf{X}|\alpha)\\ \Rightarrow \underset{\lambda\in\mathcal{F}(q)}{\arg\min}\ \mathbb{KL}(q||\tilde{p}) = \underset{\lambda\in\mathcal{F}(q)}{\arg\min}\ \mathbb{KL}(q||p^*)\end{gathered}$$

Eq. 9: Since the Kullback-Leibler divergence is always non-negative, the variational objective $J(q)$ provides an upper bound on $-\log p(\mathbf{X}|\alpha)$, the negative log likelihood of the observed data. Minimizing $J(q)$ necessarily minimizes $\mathbb{KL}(q||p^*)$, as the evidence remains constant with respect to the choice of q.

$$\begin{aligned}\underset{\lambda\in\mathcal{F}(q)}{\arg\max}\ ELBO(\lambda) &= \underset{\lambda\in\mathcal{F}(q)}{\arg\max}\ -J(q)\\ &= \underset{\lambda\in\mathcal{F}(q)}{\arg\max}\ \int q(\theta|\lambda)\log\frac{\tilde{p}(\theta)}{q(\theta|\lambda)}\,d\theta\end{aligned}$$

Eq. 10: The Evidence Lower Bound (ELBO) reformulation of the variational objective is most commonly used in applications of variational inference. The optimization task becomes one of gradient ascent, since the integral objective does not admit a closed-form solution.

III. Implementation and Scaling

Although the ELBO formulation in Equation 10 avoids even point-wise evaluations of the true posterior p^*, optimizing over it remains a challenge. Without any guarantees of convexity, a closed-form solution for its global maximum is hopeless, relegating any optimization effort to iterative gradient-based methods. Gradient ascent is not itself a walk in the park, as the gradient with respect to the variational parameters λ does not magically cancel out with the integral over the latent variables θ. In order to find a closed-form expression of the ELBO gradient, a restricted variational family of distributions must be chosen for which the gradient can be analytically computed, such as the Conjugate-Exponential family.[21] One disadvantage of this approach is that tighter restrictions upon the variational family result in lessened expressive power for estimating the true posterior. Furthermore, the lack of generality in the analytical computations required for each new model greatly hinders wide adoption of the variational inference approach, especially for models for which a closed-form solution does not exist. To increase generality and lighten the computational load, various stochastic approximation approaches have been proposed in the past decade, providing a closed-form approximation of the ELBO gradient in its general form.[22,23,24]

A. Mean Field Approximation

The mean field assumption restricts the variational distribution over the latent variables to be fully factorizable into independent distributions over each variable.[25]

$$q(\theta|\lambda) = \prod_{i=1}^{N} q_i(\theta_i|\lambda_i)$$

This simplifies the computation of a closed-form solution for gradient ascent on the ELBO at the cost of reduced expressivity of the variational distribution. This assumption can also be combined with stochastic approximation methods to reduce the computational load of the gradient ascent procedure by reducing the number of variational parameters $|\lambda|$, thus reducing the dimensionality of the search space.

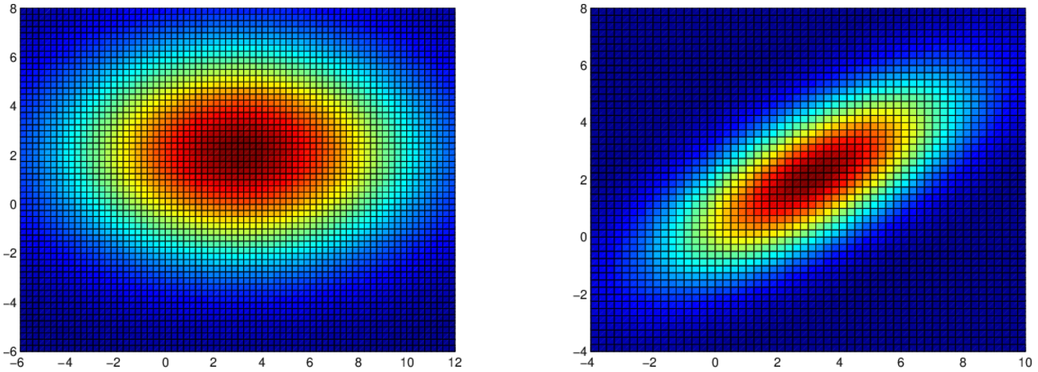


Fig. 11: Multivariate Gaussian distributions depicting factorizability. The distribution on the left has each dimension independent of the other, while the distribution on the right illustrates an obvious positive correlation. The multivariate Gaussian can be factorized if its covariance matrix is diagonal.[26] Restricting the multivariate Gaussian family $\mathcal{N}(\vec{\mu}, \mathbf{\Sigma})$ to a diagonal covariance matrix $\mathcal{N}(\vec{\mu}, \text{diag}(\vec{\sigma}))$ reduces the number of variational parameters $|\lambda|$ from $\mathcal{O}(|\theta|^2)$ to $\mathcal{O}(|\theta|)$. Further restricting the family to $\mathcal{N}(\vec{0}, \sigma\mathbf{I})$ reduces $|\lambda|$ to $\mathcal{O}(1)$, as exemplified in the Variational Auto-Encoder.[24]

B. Score function stochastic gradient estimation

The score function of the variational distribution is defined as $\nabla_\lambda \log q(\theta_s|\lambda)$. This method reformulates the gradient of the ELBO such that the only gradient that needs to be computed is that of the score function.[23]

$$\nabla_\lambda ELBO(\lambda) = \nabla_\lambda \int q(\theta|\lambda) \log \frac{\tilde{p}(\theta)}{q(\theta|\lambda)}\, d\theta$$

$$= \mathbb{E}_q \left[\nabla_\lambda \left[\log q(\theta|\lambda)\right] \left(\log \tilde{p}(\theta) - \log q(\theta|\lambda)\right)\right]$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \log q(\theta_s|\lambda)(\log \tilde{p}(\theta_s) - \log q(\theta_s|\lambda), \text{ with } \theta_s \sim q(\theta|\lambda)$$

Eq. 12: The above reformulates the ELBO gradient in terms of the score function gradient, the model forward pass $\tilde{p}$, and the variational distribution q. The proof for this reformulation can be found in Appendix B-C. The expectation over the variational distribution can then be estimated with Monte Carlo sampling, giving a noisy approximation of the ELBO gradient and enabling coordinate ascent.

The score function estimator for the ELBO gradient has the clear advantage of side-stepping the gradient of the model function $\tilde{p}$ entirely, enabling variational inference for arbitrary models without even the common constraint of differentiability. This comes at the cost of variance, which has been shown to be very high in the score function estimator both experimentally and theoretically.[27] Intuitively, the high variance can be explained as a result of missing out on the local information of the model function that its gradient would provide. Thus, the score function estimator turns out to be a double-edged sword. Nevertheless, proponents of the score function estimator have showcased various variance reduction techniques that improve its viability.

Rao-Blackwellisation is a form of variance reduction that involves probabilistically conditioning the estimator on a subset of dimensions and integrating out the remaining dimensions analytically. Concretely, this means that the dimensions $\{1, \dots, D\}$ of θ are partitioned into two disjoint sets $\mathcal{S}$ and $\bar{\mathcal{S}}$, and expectation can be estimated as by performing Monte-Carlo integration over the smaller parameter space $\theta_{\mathcal{S}}$, conditioned on $\theta_{\bar{\mathcal{S}}}$ which is kept constant. This conditional estimator has provably smaller variance, and is advantageous as long as the conditional expectations can be computed efficiently.[27,28]

Control variates comprise a generic technique for reducing the variance of any Monte-Carlo method. Given the task of computing $\mathbb{E}_{q(\theta|\lambda)}[f(\theta)]$, a control variate is defined as a function $h(\theta)$ with known expectation, which can be used to construct a substitute $\tilde{f}(\theta)$ for $f(\theta)$ with identical expectation and smaller variance as follows.[23,27]

$$\tilde{f}(\theta) = f(\theta) - \beta(h(\theta) - \mathbb{E}_{q(\theta|\lambda)}[h(\theta)])$$

C. Reparametrization Trick

An alternative to the score function estimator is the reparametrization trick,[24] which additionally leverages gradient information from the model itself, at the cost of requiring differentiable latent variables. The fundamental problem that the score function estimator avoids is that sampling is not a differentiable operation, therefore the gradient is not able to propagate backward past these sampling operations. The reparametrization trick reformulates the sampling procedure of θ into a deterministic transformation of a sample that is drawn independently of λ from a base distribution.

$$\theta_s \sim q(\theta|\lambda) \quad \equiv \quad \theta_s = g(\epsilon_s, \lambda), \quad \epsilon_s \sim p(\epsilon)$$

Intuitively, this equivalent formulation removes the sampling from the path of the gradient by removing the dependency of $p(\epsilon)$ on λ. The deterministic function $g(\epsilon, \lambda)$ stores these dependencies, but remains differentiable. Thus, the gradient is able to flow backward through g, without having to touch the actual source of nondeterminism. Monte-Carlo integration is again applied to the resulting reformulation to obtain a noisy but unbiased estimator of the ELBO gradient which leverages the model's gradient information and the score function gradient to achieve significantly lower variance than the score function estimator.[27]

$$\nabla_\lambda ELBO(\lambda) = \mathbb{E}_q [\nabla_\lambda [\log \tilde{p}(g(\epsilon, \lambda)) - \log q(g(\epsilon, \lambda)|\lambda)]]$$
$$\approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \log \tilde{p}(g(\epsilon_s, \lambda)) - \nabla_\lambda \log q(g(\epsilon_s, \lambda)|\lambda), \text{ with } \epsilon_s \sim p(\epsilon)$$

IV. Examples

A. Variational Auto-Encoder

The Variational Auto-Encoder[24] introduced the reparameterization trick to neural networks for the purpose of applying variational inference to the low-dimensional latent space of traditional auto-encoders trained with reconstruction loss.

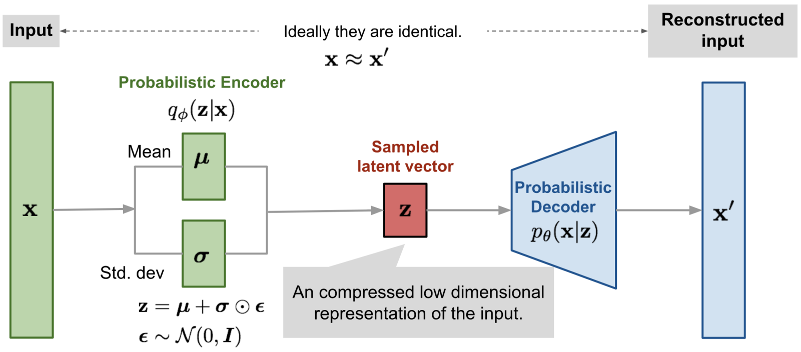
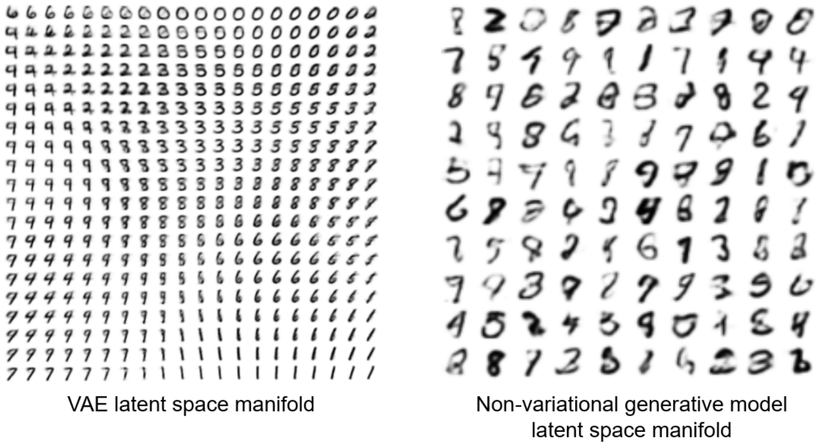

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
$\mathbf{x}$
$\boldsymbol{\sigma}$
Std. dev
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
Sampled latent vector
$\mathbf{z}$
An compressed low dimensional representation of the input.
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$

Fig. 13: The Variational Auto-Encoder has the same bottleneck structure as the traditional auto-encoder, however the prbabilistic encoder now outputs variational parameters $\lambda = [\mu, \sigma]$ from which the latent vector is then sampled. Using the reparameterization trick, the source of nondeterminism is $\epsilon \sim \mathcal{N}(0, I)$, and the deterministic transformation is $g(\epsilon, \lambda) = \mu + \sigma \odot \epsilon$. The probabilistic decoder then evaluates the likelihood of the original input given the sampled latent vector.



VAE latent space manifold

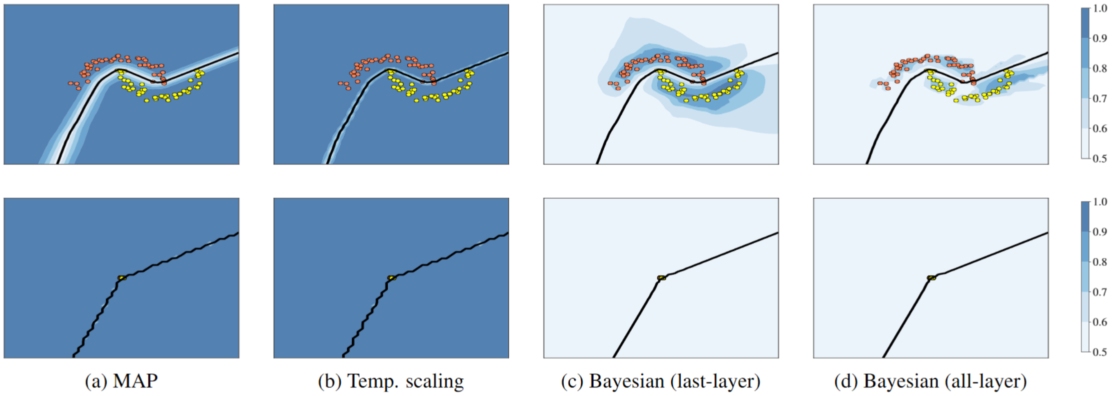
Non-variational generative model latent space manifold

Fig. 14: The advantages of the Variational Auto-Encoder lie in its generative properties. It has been shown empirically that the variational constraints placed on the latent space result in more interpretable dimensions of variation, allowing for smooth interpolation within the latent space.[24]

B. Bayesian Neural Network

As discussed in Chapter II, the Bayesian neural network maintains probability distributions over its parameters instead of a single point estimation. Training can be done via stochastic variational inference using the building blocks laid out in Chapter III, though alternatives certainly exist, such as the Laplace approximation of the posterior.[29]

Using the distribution over its latent parameter space, the Bayesian neural network is able to provide probability distributions over its predictions using the posterior predictive distribution shown in Equation 5. While integration over the entire parameter space remains infeasible, Monte-Carlo sampling can be used here as well, with an arbitrary number of samples.



(a) MAP (b) Temp. scaling (c) Bayesian (last-layer) (d) Bayesian (all-layer)

Fig. 15: Bayesian neural networks are shown empirically to drastically improve calibration in modern neural networks, which typically suffer from over-confidence when leaving the manifold of the training data set. Even when only the final layer of the network is Bayesian, the calibrating effects are obvious, suggesting that the choice does not have to be a binary one between Bayesian and non-Bayesian.[14]

V. Extensions

A. Flipout

In order to apply the reparameterization trick to Bayesian neural networks, the stochasticity of the weights is typically modeled as a sampled weight perturbation from a symmetrical distribution centered around 0. During training, mini-batches are typically used, with a single sampled perturbation being used for each sample in the batch for computational reasons. The Flipout method[30] proposes a random sign matrix to augment the sampled perturbation for each training sample in the mini-batch, maintaining the same sampling distribution as long as it is symmetrical around 0, but decorrelating the gradients of each training sample, leading to much faster convergence.

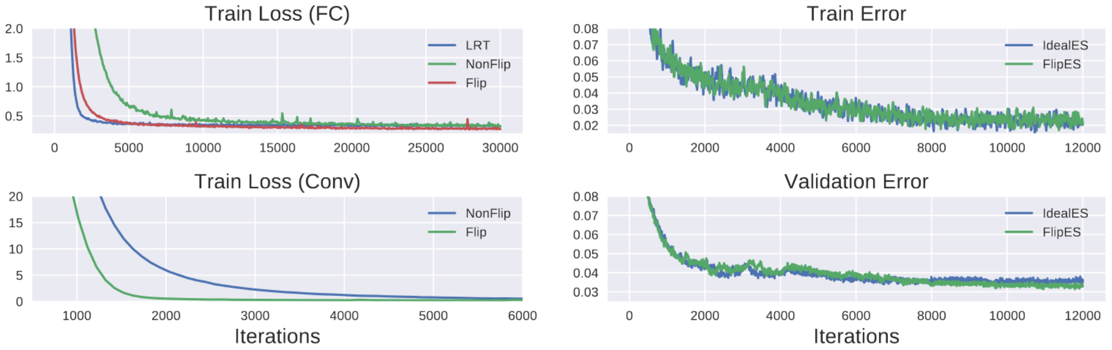


Fig. 16: Empirical results show that Flipout leads to a dramatic increase in convergence speed, without having a significant effect on the training and validation error.

B. *Normalizing Flows*

A significant drawback of variational inference approaches is that a restrictive family of distributions is usually chosen for computational reasons, whose limited flexibility is generally not able to match the true posterior distribution, even given infinite time. Normalizing flows are proposed as an alternative variational objective, which optimizes over a family of invertible transformations, starting from some initial seed distribution.[31] One drawback to this method is reduced scalability, in addition to the multiple hyper-parameters which have to be tuned.

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^\top \mathbf{z} + b)$$

$$\lambda = \{\mathbf{w} \in \mathbb{R}^D, \mathbf{u} \in \mathbb{R}^D, b \in \mathbb{R}\}$$

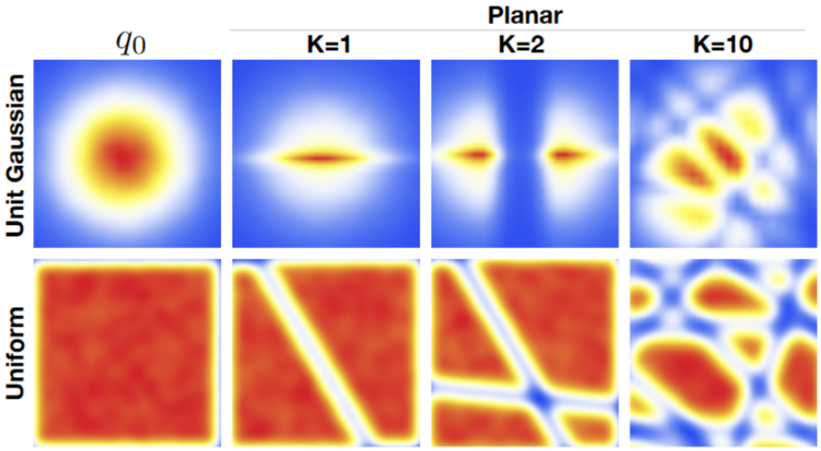
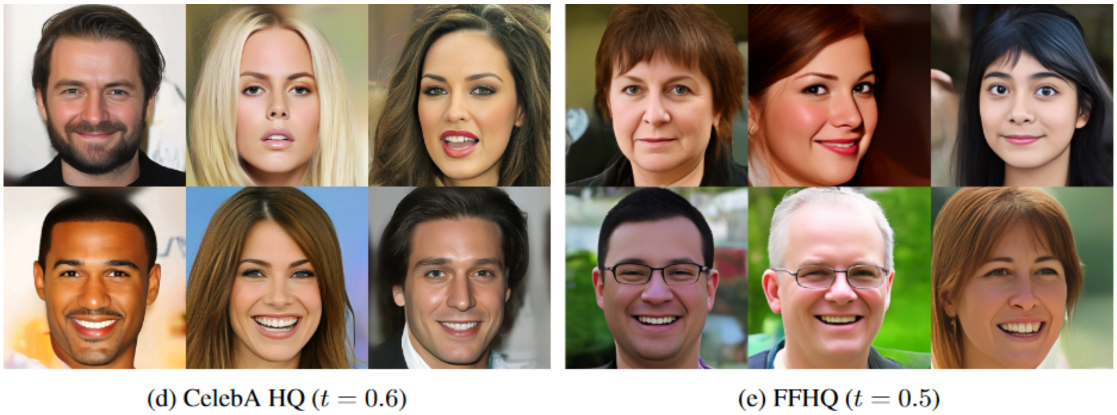


Fig. 17: Starting from different seed distributions q_0, the normalizing flow method arrives at similar results as more transformations are composed together, denoted by the hyper-parameter K.

C. *Hierarchy*

A common criticism of the Variational Auto-Encoder as a generative model relates to its tendency to produce blurry results when scaled up to generate detailed human faces. The Deep Hierarchical Variational Auto-Encoder[32] overcomes this problem by partitioning the latent space into L disjoint groups $z = \{z_1, z_2, \ldots, z_L\}$, with each group of latent variables being assigned its own variational parameters. Intuitively, the goal is for each group of latent parameters to govern a different level of abstraction in the details which are generated by the model. This goal is supported by the different scales at which the partitions are grouped, with lower-dimensional groups governing broader aspects of the image and higher-dimensional groups having the capacity to affect richer and more subtle details.



(d) CelebA HQ ($t = 0.6$)

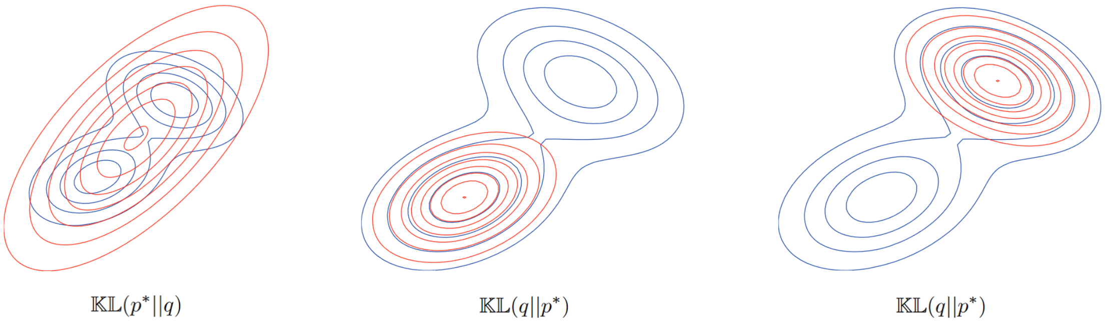
(e) FFHQ ($t = 0.5$)

Fig. 18: The NVAE[32] is able to generate realistic images with minimal blurring of details, due to the assignment of different latent groups to different levels of detail, thereby increasing the expressivity of the joint latent distribution. Previously, this level of quality was only achievable using the Generative Adversarial class of models.[33]

VI. Conclusion

Although there is much mathematical rigour involved not only in the theory but also in the implementation tricks required to make variational inference work well at scale, models such as the Variational Auto-Encoder and Bayesian neural network are uniquely positioned in the amount of probabilistic information they are able to expose to the user. Generative Adversarial Networks are widely considered to be better at generating realistic samples, but the final example in Chapter V shows comparable or even better results with the variational approach, with the added benefit of a direct view of the latent encoding of the underlying distribution, which GANs only model implicitly. These results show that it is impossible to say that one approach is objectively better than the other, and that they should each be pursued further.

Appendix A
Kullback-Leibler divergence asymmetry



$\mathbb{KL}(p^*||q)$

$\mathbb{KL}(q||p^*)$

$\mathbb{KL}(q||p^*)$

Fig. 19: Since the Kullback-Leibler divergence is not a symmetric operator, it is important to first note the consequences of reversing the terms to construct the ELBO, as shown in Equation 7.[9] The blue curves represent the true bimodal distribution p^*, while the red curves represent the approximate distribution q, which is modeled as unimodal. (a) Minimizing the forward divergence $\mathbb{KL}(p^*||q)$ in q typically over-estimates the support of p^*. (b-c) Conversely, minimizing the reverse divergence $\mathbb{KL}(q||p^*)$ tends to under-estimate the support of p^*, however the mode of the true posterior is more accurately matched in this particular bimodal posterior example.

Appendix B
Score function gradient estimator

$$\nabla_\lambda ELBO(\lambda) = \nabla_\lambda \int q(\theta|\lambda) \log \frac{\tilde{p}(\theta)}{q(\theta|\lambda)} \, d\theta$$

$$= \int \nabla_\lambda \left[q(\theta|\lambda) \left(\log \tilde{p}(\theta) - \log q(\theta|\lambda) \right] \, d\theta \right.$$

$$= \int \nabla_\lambda \left[(\log \tilde{p}(\theta) - \log q(\theta|\lambda) \right] q(\theta|\lambda) \, d\theta$$

$$+ \int \nabla_\lambda \left[q(\theta|\lambda) \right] (\log \tilde{p}(\theta) - \log q(\theta|\lambda) \, d\theta$$

$$= \underbrace{-\mathbb{E}_q \left[\nabla_\lambda \log q(\theta|\lambda) \right]}_{0} + \int \nabla_\lambda \left[q(\theta|\lambda) \right] (\log \tilde{p}(\theta) - \log q(\theta|\lambda) \, d\theta$$

$$= \int q(\theta|\lambda) \nabla_\lambda \left[\log q(\theta|\lambda) \right] (\log \tilde{p}(\theta) - \log q(\theta|\lambda)) \, d\theta$$

$$= \mathbb{E}_q \left[\nabla_\lambda \left[\log q(\theta|\lambda) \right] (\log \tilde{p}(\theta) - \log q(\theta|\lambda)) \right]$$

Appendix C
Score function gradient expectation

$$\mathbb{E}_q \left[\nabla_\lambda \log q(\theta|\lambda) \right] = \mathbb{E}_q \left[\frac{\nabla_\lambda q(\theta|\lambda)}{q(\theta|\lambda)} \right]$$

$$= \int q(\theta|\lambda) \frac{\nabla_\lambda q(\theta|\lambda)}{q(\theta|\lambda)} \, d\theta$$

$$= \nabla_\lambda \int q(\theta|\lambda) \, d\theta$$

$$= \nabla_\lambda 1 = 0$$

References

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. USA: Prentice Hall Press, 2009.

[2] M. Krzywinski and N. Altman, “Points of significance: Importance of being uncertain,” *Nature methods*, vol. 10, pp. 809–10, 09 2013.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” 2017.

[5] A. Kumar, P. Liang, and T. Ma, “Verified uncertainty calibration,” 2020.

[6] J. Zhang, B. Kailkhura, and T. Y.-J. Han, “Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning,” 2020.

[7] S. Thulasidasan, G. Chennupati, J. Bilmes, T. Bhattacharya, and S. Michalak, “On mixup training: Improved calibration and predictive uncertainty for deep neural networks,” 2020.

[8] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, “End to end learning for self-driving cars,” 2016.

[9] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[10] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, “Hands-on bayesian neural networks – a tutorial for deep learning users,” 2020.

[11] I. Yeo, S. Gi, B. Lee, and M. Chu, “Stochastic implementation of the activation function for artificial neural networks,” in *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2016, pp. 440–443.

[12] J. M. Hernndez-Lobato and R. P. Adams, “Probabilistic backpropagation for scalable learning of bayesian neural networks,” 2015.

[13] P.-K. Wu, T.-C. Hsiao, and M. Xiao, “A stochastic artificial neural network model for investigating street vendor behavior in a night market,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 10, p. 1550147716673371, 2016. [Online]. Available: https://doi.org/10.1177/1550147716673371

[14] A. Kristiadi, M. Hein, and P. Hennig, “Being bayesian, even just a bit, fixes overconfidence in relu networks,” 2020.

[15] J. Mitros and B. M. Namee, “On the validity of bayesian neural networks for uncertainty estimation,” 2019.

[16] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, “Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift,” 2019.

[17] C. Peterson and J. Anderson, “A mean field theory learning algorithm for neural networks,” *Complex Systems*, vol. 1, no. 5, pp. 995–1019, 1987.

[18] C. Peterson and E. Hartman, “Explorations of the mean field theory learning algorithm,” *Neural Networks*, vol. 2, no. 6, pp. 475 – 494, 1989.

[19] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, “An introduction to variational methods for graphical models,” *Machine Learning*, vol. 37, pp. 183–233, 1999.

[20] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951. [Online]. Available: https://doi.org/10.1214/aoms/1177729694

[21] Z. Ghahramani and M. Beal, “Propagation algorithms for variational bayesian learning,” in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2001, pp. 507–513. [Online]. Available: https://proceedings.neurips.cc/paper/2000/file/77369e37b2aa1404f416275183ab055f-Paper.pdf

[22] M. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” 2013.

[23] R. Ranganath, S. Gerrish, and D. M. Blei, “Black box variational inference,” 2013.

[24] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2014.

[25] M. Opper and D. Saad, “Advanced mean field methods theory and practice,” 01 2001.

[26] D. Chuong, “The multivariate gaussian distribution,” 2004.

[27] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, “Monte carlo gradient estimation in machine learning,” 2020.

[28] G. Casella and C. P. Robert, “Rao-blackwellisation of sampling schemes,” *Biometrika*, vol. 83, no. 1, pp. 81–94, 1996. [Online]. Available: http://www.jstor.org/stable/2337434

[29] H. Ritter, A. Botev, and D. Barber, “A scalable laplace approximation for neural networks,” in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=Skdvd2xAZ

[30] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. Grosse, “Flipout: Efficient pseudo-independent weight perturbations on mini-batches,” 2018.

[31] D. J. Rezende and S. Mohamed, “Variational inference with normalizing flows,” 2016.

[32] A. Vahdat and J. Kautz, “Nvae: A deep hierarchical variational autoencoder,” 2020.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014.