MATH5836: Data and Machine Learning

Week 5: Bayesian Neural Networks

Sarat Moka *UNSW, Sydney*

Key Topics

References:

(A) [Book] *Pattern Recognition and Machine Learning* by Christopher Bishop (2006).

(B) [Paper] Gal, Y., and Ghahramani, Z. (2016), *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.*

5.1 Motivation for Bayesian Neural Networks

- Standard neural networks learn a *point estimate* of the weights $\hat{\theta}$ by minimizing the loss function $C(\theta)$. This yields $\hat{y} = f_{\hat{\theta}}(x)$ but no assessment of *confidence* in $\hat{y}$.
- This means, once the neural network is trained, the weights are biases are fixed; see Figure 5.1 for an illustration.

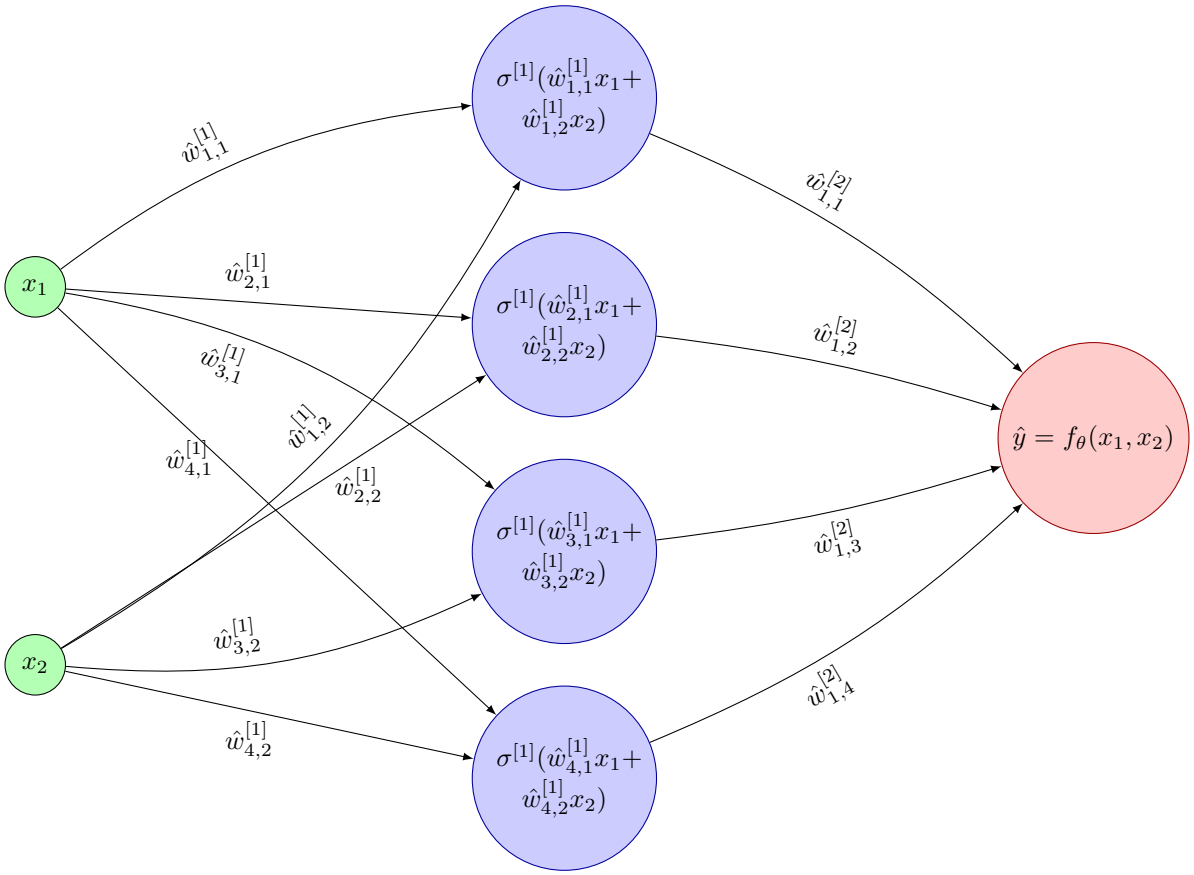


Figure 5.1: A standard small network of two inputs, four hidden units, and one output, without biases. The weights shown are point estimates after the training.

- A *Bayesian neural network* (BNN) places a prior $p(\theta)$ over all weights and biases,

$$p(\theta) \;=\; \prod_{\ell=1}^{L} p\big(W^{[\ell]}\big)\, p\big(b^{[\ell]}\big),$$

  and computes the posterior using

$$p(\theta \mid \mathcal{D}) \;\propto\; p(\mathcal{D} \mid \theta)\, p(\theta), \quad p(\mathcal{D} \mid \theta) = \prod_{i=1}^{n} p\big(y^{(i)} \mid x^{(i)}, \theta\big).$$

- Predictions integrate over parameter uncertainty via the *posterior predictive*:

$$p(y^* \mid x^*, \mathcal{D}) = \int p(y^* \mid x^*, \theta)\, p(\theta \mid \mathcal{D})\, d\theta,$$

  where x^* is an new data point. This yields *credible intervals* and calibrated uncertainty estimates.

- This means, every time we use the neural network in the production, the weights and biases of the network are are generated randomly from learned probability distributions; see Figure 5.2 for an illustration.

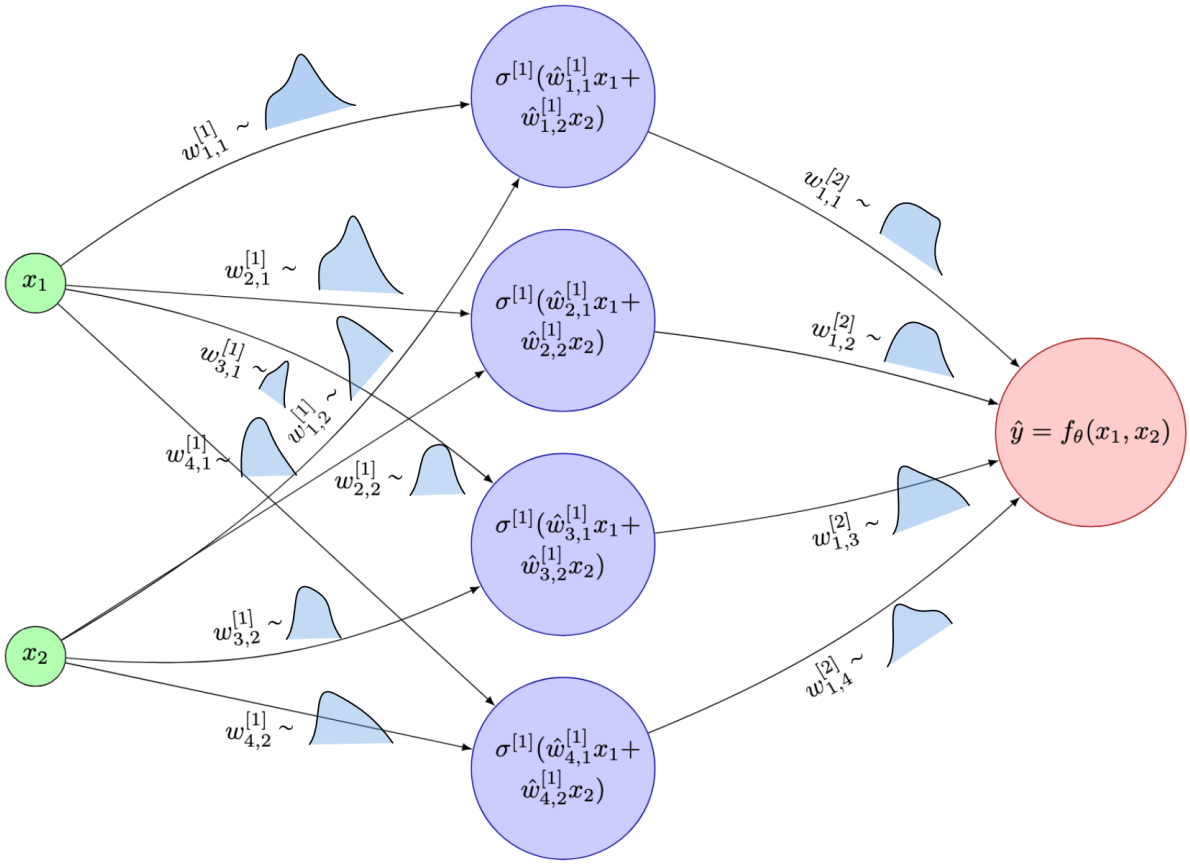


Figure 5.2: The same small network as above. But, after the training, each time we use the network, the weights are generated from learned probability distributions.

- BNNs naturally decompose uncertainty into
  - *Aleatoric uncertainty* (inherent data noise), and
  - *Epistemic uncertainty* (uncertainty in model parameters).
- Exact posterior inference is intractable for realistic BNNs. Common approximations include

  - Monte Carlo Markov Chain (e.g. Hamiltonian Monte Carlo (HMC), No U-Turn Sampler (NUTS).
  - Variational Bayes (mean-field, reparameterization).
  - Laplace's method, Monte Carlo-dropout.

- **Benefits of BNNs:**
  - ***Uncertainty quantification* for safer decision-making:** Bayesian models don't just give a single "best" prediction; they provide a measure of how confident they are in that prediction (i.e., the uncertainty). This is crucial in high-stakes scenarios (e.g., medical diagnosis, autonomous driving) because it allows the system to know when it's unsure and perhaps defer to a human expert or request more information, leading to safer and more reliable decisions.
  - ***Regularization* via the prior, reducing overfitting:** In Bayesian models, a "prior" distribution is set on the model parameters (e.g., neural network weights and biases) before seeing the data. This prior expresses initial beliefs about what plausible parameter values are (e.g., preferring smaller weights). During training, this prior acts as a natural form of regularization, pulling the parameters towards simpler configurations and preventing them from fitting the training data too closely (overfitting), thus improving generalization to new, unseen data. This is analogous to L1 or L2 regularization in classical machine learning.
  - ***Principled model comparison* (via marginal likelihoods or Bayes factors):** Bayesian frameworks offer a theoretically sound way to compare different models (e.g., different architectures or complexities). The "marginal likelihood" (or model evidence) quantifies how well a model explains the observed data, naturally penalizing overly complex models (a concept known as Bayesian Occam's Razor). Bayes factors are ratios of these marginal likelihoods, providing a direct measure of evidence in favor of one model over another. This is often more robust than relying solely on validation set performance.
  - **Improved *calibration* of predictive probabilities:** "Calibration" means that if a model predicts an event with, say, 70% probability, then that event should actually occur about 70% of the time for all instances where the model made such a prediction. Standard neural networks can often be overconfident (e.g., predicting 99% probability for things that are only 80% likely). Bayesian methods, by averaging over many possible parameter settings consistent with the data, tend to produce predictive probabilities that are better calibrated, meaning they more accurately reflect the true underlying likelihoods.

5.2 Bayesian Foundations

In this section we review the univariate Gaussian distribution, its density and distribution functions, and key summary measures such as quantiles and the interquartile range (IQR).

Recap: Univariate Gaussian (Normal) Distribution

- A random variable X is (univariate) Gaussian with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$, denoted

$$X \sim \mathcal{N}(\mu, \sigma^2),$$

if its probability density function (PDF) is

$$f_X(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\Big(- \frac{(x-\mu)^2}{2\sigma^2} \Big), \quad x \in \mathbb{R}.$$

- The cumulative distribution function (CDF) is

$$F_X(x) = \mathbb{P}(X \le x) = \Phi\Big(\frac{x-\mu}{\sigma}\Big),$$

where

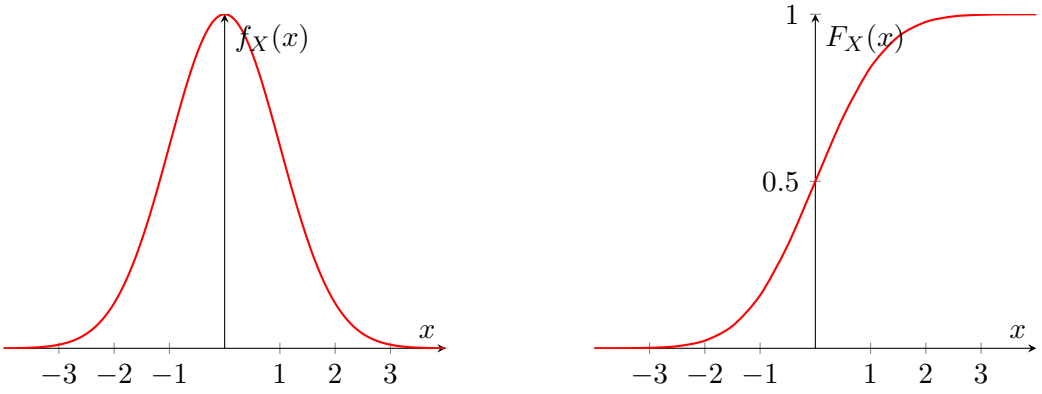
$$\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} e^{-t^2/2}\, dt$$

is the standard normal CDF.

- The mean and the variance are, respectively, given by

$$\mathbb{E}[X] = \mu, \qquad \mathrm{Var}(X) = \sigma^2,$$

and the PDF is symmetric about $x = \mu$.



(a) Standard normal PDF (b) Standard normal CDF

Figure 5.3: PDF and CDF of the standard Gaussian $X \sim \mathcal{N}(0, 1)$.

Quantiles and Interquartile Range

- The p-th quantile $Q(p)$ satisfies $F_X(Q(p)) = p$. Equivalently,

$$Q(p) = \mu + \sigma\, \Phi^{-1}(p), \quad p \in (0, 1).$$

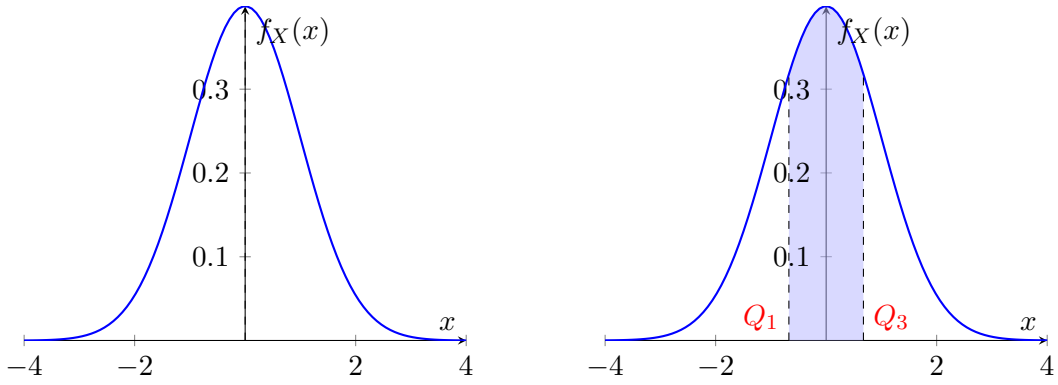
- Special cases:

  median: $Q(0.5) = \mu$, first quartile: $Q_1 = Q(0.25)$, third quartile: $Q_3 = Q(0.75)$.

  The median is also sometimes called the second quartile, Q_2.

- The interquartile range (IQR) is

$$\text{IQR} = Q_3 - Q_1 = \sigma\left[\Phi^{-1}(0.75) - \Phi^{-1}(0.25)\right] \approx 1.34898\,\sigma.$$



(a) PDF of $\mathcal{N}(\mu, \sigma^2)$.

(b) PDF with IQR shading between Q_1 and Q_3.

Figure 5.4: Illustrations for the univariate Gaussian distribution: (a) density with mean μ; (b) interquartile range highlighted.

Remark 5.2.1

The definitions of quantiles and interquartile range can be easily extended to any random variable with CDF $F_X(x)$.

Bayes's Theorem

We now state Bayes's theorem in both event- and density-form, introduce the notions of prior, likelihood, evidence and posterior, and work through a concrete numerical example.

- Let A and B be events with $\mathbb{P}(B) > 0$. Then

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(B \mid A)\,\mathbb{P}(A)}{\mathbb{P}(B)},$$

  where $\mathbb{P}(B \mid A)$ denotes the probability event B occurring given that A has occurred already.

Example 5.2.1

Disease-Screening Test: Suppose a population has a rare disease with prevalence $\mathbb{P}(\texttt{Disease}) = 0.01$, and a diagnostic test with $\mathbb{P}(\texttt{Positive} \mid \texttt{Disease}) = 0.99$ and $\mathbb{P}(\texttt{Negative} \mid \texttt{No Disease}) = 0.95$. We wish to compute the posterior probability of disease given a positive test:

$$\mathbb{P}(\texttt{Disease} \mid \texttt{Positive}) = \frac{\mathbb{P}(\texttt{Positive} \mid \texttt{Disease})\,\mathbb{P}(\texttt{Disease})}{\mathbb{P}(\texttt{Positive})} = \frac{0.99 \times 0.01}{\mathbb{P}(\texttt{Positive})}.$$

Here,

$$\begin{aligned}\mathbb{P}(\texttt{Positive}) &= \mathbb{P}(\texttt{Positive} \mid \texttt{Disease})\mathbb{P}(\texttt{Disease}) \\ &\quad + \mathbb{P}(\texttt{Positive} \mid \texttt{No Disease})\mathbb{P}(\texttt{No Disease}) \\ &= 0.99 \cdot 0.01 + 0.05 \cdot 0.99 = 0.0594.\end{aligned}$$

Hence,

$$\mathbb{P}(\texttt{Disease} \mid \texttt{Positive}) = \frac{0.0099}{0.0594} \approx 0.167.$$

	Disease	No Disease
Test Positive	$0.99 \cdot 0.01 = 0.0099$	$0.05 \cdot 0.99 = 0.0495$
Test Negative	$0.01 \cdot 0.01 = 0.0001$	$0.95 \cdot 0.99 = 0.9405$
Total	0.01	0.99

Table 5.1: Joint probabilities for disease and test outcomes.

- In parametric inference, let θ be a parameter (or hypothesis) and let x be observed data. We write

$$\underbrace{p(\theta)}_{\text{prior}}, \quad \underbrace{p(x \mid \theta)}_{\text{likelihood}}, \quad \underbrace{p(\theta \mid x)}_{\text{posterior}}, \quad \underbrace{p(x)}_{\text{evidence}}.$$

  Then the probability of θ given x (i.e., posterior) is

$$p(\theta \mid x) = \frac{p(x \mid \theta)\, p(\theta)}{p(x)},$$

  where

$$p(x) = \int p(x \mid \theta)\, p(\theta)\, d\theta.$$

- The evidence $p(x)$ normalizes the posterior and ensures

$$\int p(\theta \mid x)\, d\theta = 1.$$

Example 5.2.2

Beta–Bernoulli Updating: This model demonstrates updating beliefs about a probability θ (e.g., coin bias) given observed binary data (e.g., coin flips).

- **Data Model (Likelihood of one observation):** Coin flips $x_i \in \{0, 1\}$ (1 for heads, 0 for tails). The probability of observing x_i given θ follows a Bernoulli distribution:

$$p(x_i \mid \theta) = \theta^{x_i}(1-\theta)^{1-x_i}$$

- **Prior Belief about θ:** We model our initial belief about θ using a Beta distribution, which is a conjugate prior for the Bernoulli likelihood.

$$\theta \sim \text{Beta}(\alpha, \beta) \implies p(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

  Hyperparameters $\alpha, \beta > 0$ shape the prior (e.g., $\alpha = \beta = 1$ is uniform).

- **Posterior Belief after Data:** After observing n flips $x_{1:n} = (x_1, \ldots, x_n)$, with $s = \sum_{i=1}^{n} x_i$ being the total number of heads (successes) and $(n - s)$ the number of tails (failures). By Bayes' theorem (Posterior $\propto$ Likelihood $\times$ Prior):

$$\begin{aligned} p(\theta \mid x_{1:n}) &\propto \left(\prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i} \right) \cdot \theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &\propto \theta^{s+\alpha-1}(1-\theta)^{(n-s)+\beta-1} \end{aligned}$$

  This is another Beta distribution:

$$p(\theta \mid x_{1:n}) = \text{Beta}(\alpha_{\text{post}}, \beta_{\text{post}}) = \text{Beta}(\alpha + s, \beta + n - s)$$

  The prior parameters are updated by the counts of successes and failures.

- **Posterior Predictive Distribution (for the next flip):** The probability of the next flip x_{n+1} being a head, given the observed data $x_{1:n}$, is found by averaging θ over its posterior distribution:

$$\begin{aligned} \mathbb{P}(x_{n+1} = 1 \mid x_{1:n}) &= \int_0^1 \theta \cdot p(\theta \mid x_{1:n})\, d\theta = \mathbb{E}[\theta \mid x_{1:n}] \\ &= \frac{\alpha_{\text{post}}}{\alpha_{\text{post}} + \beta_{\text{post}}} \\ &= \frac{\alpha + s}{\alpha + s + \beta + n - s} \\ &= \frac{\alpha + s}{\alpha + \beta + n} \end{aligned}$$

  This prediction blends prior knowledge (α, β) with observed data (s, n).

5.3 Bayesian Linear Regression

In this section, we develop Bayesian inference for the simple linear regression model. We derive the conjugate posterior for the regression weights, obtain the posterior predictive distribution, and illustrate 95% credible bands.

- **Model Specification:** Consider the linear model for a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$:

$$y_i = x_i^\top \beta + \varepsilon_i, \qquad \varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2),$$

  where $\sigma^2 > 0$ is the variance of the noise.

- **Prior and Conjugacy:** We place a Gaussian prior on β:

$$\beta \sim \mathcal{N}(\mu_0, \Sigma_0),$$

  with

$$p(\beta) = \frac{1}{(2\pi)^{p/2}\,|\Sigma_0|^{1/2}} \exp\!\big(-\tfrac{1}{2}(\beta - \mu_0)^\top \Sigma_0^{-1}(\beta - \mu_0)\big),$$

  for mean vector μ_0 and covariance matrix Σ_0. This is conjugate to the Gaussian likelihood.

- **Posterior Distribution:** Then the posterior is

$$p(\beta \mid \mathcal{D}) = \mathcal{N}(\mu_n, \Sigma_n),$$

  with

$$\mu_n = \Sigma_n\Big(\Sigma_0^{-1}\mu_0 + \tfrac{1}{\sigma^2} X^\top y\Big),$$

  and

$$\Sigma_n = \big(\Sigma_0^{-1} + \tfrac{1}{\sigma^2} X^\top X\big)^{-1}.$$

  See Appendix A for a proof if you are interested.

- **Posterior Predictive Distribution:** For a new input x^*, the posterior predictive for the corresponding respose y^* is

$$\begin{aligned} p(y^* \mid x^*, \mathcal{D}) &= \int p(y^* \mid x^*, \beta)\, p(\beta \mid \mathcal{D})\, d\beta \\ &= \mathcal{N}\big(x^{*\top}\mu_n,\ \sigma^2 + x^{*\top}\Sigma_n x^*\big). \end{aligned}$$

  Hence the 95% *pointwise* credible interval at x^* is

$$x^{*\top}\mu_n \ \pm\ 1.96\ \sqrt{\sigma^2 + x^{*\top}\Sigma_n x^*}.$$

> Remark 5.3.1
>
> **Credible vs. Confidence Intervals:** The credible band shown is Bayesian: it is the 95% interval for y^* given the observed data and prior. A frequentist 95% confidence band has a different interpretation ("over repeated sampling, 95% of such bands will cover the true regression function") and generally differs numerically.

- **Illustration: Regression Line with Credible Bands:**

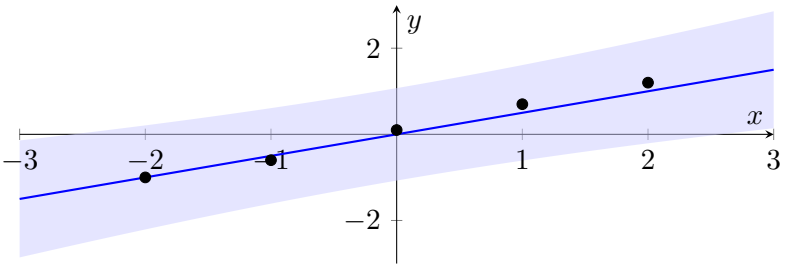


Figure 5.5: Posterior mean regression line (blue) with 95% credible bands (shaded).

5.4 From Linear Models to Neural Networks

To bridge Bayesian linear regression and neural networks, we first recall deterministic feedforward nets, then show how a one–hidden–layer network is a nonlinear basis expansion, and finally introduce priors on all weights to define a Bayesian neural network.

Deterministic Feedforward Networks

- Recall that a feedforward network with L layers defines

$$f_\theta(x) = f^{[L]}_{\theta^{[L]}}\big(\cdots f^{[2]}_{\theta^{[2]}}(f^{[1]}_{\theta^{[1]}}(x))\cdots\big),$$

  with $\theta = \{\theta^{[\ell]}\}_{\ell=1}^{L}$ and $\theta^{[\ell]} = \{W^{[\ell]}, b^{[\ell]}\}$.

- Layer ℓ acts by

$$z^{[\ell]} = W^{[\ell]}a^{[\ell-1]} + b^{[\ell]}, \quad a^{[\ell]} = S^{[\ell]}(z^{[\ell]}),$$

  with $a^{[0]} = x$, and $a^{[L]} = f_\theta(x)$.

- Also, recall that training finds the point estimate $\hat{\theta}$ by minimizing a loss $C(\theta)$ (e.g. mean-squared or log-loss) using backpropagation to compute the gradient $\nabla C(\theta)$.

Priors on Weights and Biases

- To make the network Bayesian, place a prior on every parameter:

$$p(\theta) = \prod_{\ell=1}^{L} p\big(W^{[\ell]}\big)\, p\big(b^{[\ell]}\big).$$

- A common choice is independent Gaussians,

$$W_{ij}^{[\ell]} \sim \mathcal{N}(0, \lambda^2), \quad b_i^{[\ell]} \sim \mathcal{N}(0, \lambda^2).$$

  for $\lambda^2 > 0$.

Definition of a Bayesian Neural Network

- Given data $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$, define

$$\underbrace{p(\theta)}_{\text{prior}}, \qquad \underbrace{p(\mathcal{D} \mid \theta)}_{\substack{\text{likelihood} \\ =\prod_i p(y^{(i)} \mid x^{(i)}, \theta)}},$$

  so the posterior is

$$p(\theta \mid \mathcal{D}) \;\propto\; p(\mathcal{D} \mid \theta)\, p(\theta).$$

- The joint model over (x, y, θ) is

$$p(y, x, \theta) = p(y \mid x, \theta)\, p(\theta)\, p(x).$$

5.5 Approximate Inference in BNNs

Exact Bayesian inference in neural networks requires computing the posterior

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)\, p(\theta)}{p(\mathcal{D})}, \quad \text{where } p(\mathcal{D}) = \int p(\mathcal{D} \mid \theta)\, p(\theta)\, d\theta,$$

which is intractable due to the high-dimensional, non-linear integrals over θ. We now survey three popular approximate inference methods: MC-droupout, VB, and MCMC.

MC-Dropout

- *Key insight.* Training with dropout is equivalent to approximate variational inference with a mixture-of-Diracs posterior over network weights.
- *Procedure.*
  - During training, apply dropout masks to activations (or weights).
  - At test time, perform T stochastic forward passes with dropout enabled.
  - Approximate the posterior predictive by the sample mean and variance of the T outputs.
- *Benefits.* Easy to implement on top of a deterministic network; yields uncertainty estimates with minimal overhead.

- *Limitations.* The uncertainty is approximate and depends on dropout rate; less flexible than full VB or MCMC.

Remark 5.5.1

Standard Dropout vs. MC-Dropout In the last week, we studied dropout for standard neural network training. The key difference between that dropout and the MC-dropout here is as follows.

- **Standard Dropout**
  - During *training*: apply independent masks $M_{ij}^{[\ell]} \sim \text{Bernoulli}(p_{\text{keep}}^{[\ell]})$ and replace each $w_{ij}^{[\ell]}$ with $M_{ij}^{[\ell]} \odot w_{ij}^{[\ell]}$.
  - During *test/inference*: disable masking and use a single deterministic network with *weight scaling*
$$\tilde{w}_{ij}^{[\ell]} = p_{\text{keep}}^{[\ell]}\, w_{ij}^{[\ell]}.$$
- **MC-Dropout**
  - During *training*: identical to standard dropout (random masks).
  - During *test*: To make a prediction for a new data point x^*, sample each mask T times: $M_{ij}^{(t)} \sim \text{Bernoulli}(p_{\text{keep}}^{[\ell]})$, $t = 1, \ldots, T$, and compute stochastic forward passes $y^{*(t)} = f_\theta(x^*; M^{(t)})$.
  - Get prediction by Monte Carlo approximation:
$$y^* = \frac{1}{T}\sum_{t=1}^{T} y^{*(t)}.$$

Variational Bayes (VB)

- *Idea.* Introduce a tractable family of distributions $\{q_\rho(\theta)\}$ (e.g. Gaussian family) and choose ρ to make $q_\rho \approx p(\theta \mid \mathcal{D})$.
- *ELBO.* Maximize the evidence lower bound
$$\mathcal{L}(\rho) = \mathbb{E}_{q_\rho}\big[\log p(\mathcal{D}, \theta)\big] - \mathbb{E}_{q_\rho}\big[\log q_\rho(\theta)\big] \;=\; \log p(\mathcal{D}) \;-\; \text{KL}(q_\rho \,\|\, p(\theta \mid \mathcal{D})).$$
- *Mean-field approximation.* Assume $q_\rho(\theta) = \prod_j q_{\rho_j}(\theta_j)$. For example, $\theta_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$, and then the variational parameters are $\rho = \{(\mu_j, \sigma_j)\}_j$.
- *Reparameterization trick.* Write $\theta = \mu + \sigma \odot \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$. Then
$$\nabla_\rho\, \mathbb{E}_{q_\rho}\big[f(\theta)\big] = \mathbb{E}_\varepsilon\Big[\nabla_\rho\, f\big(\mu + \sigma\varepsilon\big)\Big],$$
enabling low-variance gradient estimates via Monte Carlo.

- *Optimization.* Use stochastic gradient ascent on $\mathcal{L}(\rho)$ ("Bayes by Backprop").

Markov Chain Monte Carlo (MCMC)

- *Goal.* Generate samples $\{\theta^{(s)}\}$ whose stationary distribution is $p(\theta \mid \mathcal{D})$.
- *Hamiltonian Monte Carlo (HMC).* Introduce momentum r, simulate Hamiltonian dynamics on (θ, r) to reduce random walk behavior, then accept/reject via Metropolis step.
- *No-U-Turn Sampler (NUTS).* An adaptive variant of HMC that automates the choice of trajectory length.
- *Trade-offs.* $\begin{cases} \text{MCMC: asymptotically exact, but often slow to converge in high dimensions.} \\ \text{VB: fast and scalable by optimization, but approximates the true posterior.} \end{cases}$

Remark 5.5.2

Choice among VB, MCMC, and MC-dropout depends on the desired balance between computational budget and fidelity of the posterior approximation.

Quantiles and Interquartile Range in Bayesian Neural Networks

Once we have T samples from the posterior predictive distribution

$$\{\, y^{*(t)} \}_{t=1}^{T} \quad \text{with} \quad y^{*(t)} \sim p(y^* \mid x^*, \mathcal{D}),$$

we can summarise *uncertainty* via empirical quantiles and the interquartile range (IQR).

- Sort the samples:
$$y^*_{(1)} \;\le\; y^*_{(2)} \;\le\; \cdots \le\; y^*_{(T)}.$$
- The empirical p-th quantile is
$$\widehat{Q}(p \mid x^*, \mathcal{D}) = y^*_{\left(\lceil pT \rceil\right)}, \qquad p \in (0,1).$$
- Special cases:
$$\text{median: } \widehat{Q}(0.5) = y^*_{(\lceil 0.5T \rceil)}, \quad \text{first quartile: } \widehat{Q}(0.25), \quad \text{third quartile: } \widehat{Q}(0.75).$$
- A $\mathbf{100(1-\alpha)\%}$ credible interval is given by
$$\big[\, \widehat{Q}(\tfrac{\alpha}{2}),\ \widehat{Q}(1-\tfrac{\alpha}{2}) \big].$$
For $\alpha = 0.05$, this is the 95% interval $\big[\widehat{Q}(0.025),\ \widehat{Q}(0.975)\big]$.

- The *interquartile range (IQR)* measures the spread of the middle 50%:

$$\mathrm{IQR}(x^*) = \widehat{Q}(0.75) - \widehat{Q}(0.25).$$

- In practice, one plots the median curve $\widehat{Q}(0.5 \mid x, \mathcal{D})$ and shades the region between $\widehat{Q}(0.025)$ and $\widehat{Q}(0.975)$, or between $\widehat{Q}(0.25)$ and $\widehat{Q}(0.75)$ for a narrower band.

Remark 5.5.3

Quantile-based summaries are nonparametric (do not assume normality of the predictive) and easily computed from posterior samples. The IQR gives a robust measure of predictive spread, while the full set of quantiles describes the shape of $p(y^* \mid x^*, \mathcal{D})$.

5.6 Detailed description of MC-Dropout

MC-Dropout can be understood as approximate variational inference in which each weight (or activation) is randomly "dropped" with probability $1 - p$. Concretely:

- Let $f_\theta(x)$ be a deterministic neural network with parameters $\theta = \{W^{[\ell]}, b^{[\ell]}\}_{\ell=1}^{L}$.
- Similar to the standard dropout from the previous lecture, introduce independent Bernoulli dropout masks $M^{[\ell]} \in \{0,1\}^{\dim(W^{[\ell]})}$ with

$$M_{ij}^{[\ell]} \sim \mathrm{Bernoulli}(p_{\mathrm{keep}}^{[\ell]}), \qquad \text{with} \quad \mathbb{E}[M_{ij}^{[\ell]}] = p_{\mathrm{keep}}^{[\ell]}.$$

- A single stochastic forward pass applies elementwise masking to compute $f_{\tilde{\theta}}(x)$ where

$$\tilde{\theta} = \{\tilde{W}^{[\ell]}, b^{[\ell]}\}_{\ell=1}^{L},$$

  with

$$\tilde{W}^{[\ell]} \;=\; M^{[\ell]} \odot W^{[\ell]},$$

  and $\odot$ denoting the elementwise product of matrices.

- At test time we draw T independent dropout masks $\{M^{(t)}\}_{t=1}^{T}$ and form Monte Carlo estimates of the posterior predictive:

$$p(y^* \mid x^*, \mathcal{D}) \approx \frac{1}{T} \sum_{t=1}^{T} p\big(y^* \mid \tilde{f}(x^*; W, M^{(t)})\big).$$

- In regression with Gaussian likelihood $y \sim \mathcal{N}\big(\tilde{f}(x; W, M), \sigma^2\big)$, the predictive mean and variance are estimated by

$$y^* \approx \hat{\mu}(x^*) = \frac{1}{T} \sum_{t=1}^{T} \tilde{f}(x^*; W, M^{(t)}), \quad \widehat{\mathrm{Var}}(y^*) = \underbrace{\frac{1}{T} \sum_{t=1}^{T} \big[\tilde{f}(x^*; W, M^{(t)})\big]^2 - \hat{\mu}(x^*)^2}_{\text{epistemic}} + \sigma^2.$$

Thus MC-dropout provides both a fast approximate posterior $q(W)$ and sample-based uncertainty estimates—epistemic via the Monte Carlo variance and aleatoric via σ^2.

Remark 5.6.1

Note that raining with dropout corresponds to minimizing a variational objective—an *ELBO*—where the variational distribution over W is implicitly given by the mixture

$$q(W) = \prod_{\ell,i,j} \left[p_{\text{keep}}^{[\ell]} \, \delta_{W_{ij}^{[\ell]}} + (1 - p_{\text{keep}}^{[\ell]}) \, \delta_0 \right],$$

where δ_u is the usual delta function taking value 1 at u and zero everywhere else.

A Proof of the Gaussian Posterior

If you are interested to learn, here is a proof analysis to establish the Gaussian posterior stated in Section 5.3. We start from the prior and likelihood:

$$p(\beta) \;=\; \mathcal{N}(\beta \mid \mu_0, \Sigma_0), \quad p(y \mid X, \beta) \;\propto\; \exp\Big[-\tfrac{1}{2\sigma^2}(y - X\beta)^\top (y - X\beta)\Big].$$

Hence the un-normalized posterior is

$$p(\beta \mid \mathcal{D}) \;\propto\; \exp\Big[-\tfrac{1}{2}(\beta - \mu_0)^\top \Sigma_0^{-1} (\beta - \mu_0) \;-\; \tfrac{1}{2\sigma^2}(y - X\beta)^\top (y - X\beta)\Big].$$

Expand the two quadratic forms:

$$\begin{aligned}
(\beta - \mu_0)^\top \Sigma_0^{-1} (\beta - \mu_0) \;&=\; \beta^\top \Sigma_0^{-1} \beta - 2\,\beta^\top \Sigma_0^{-1} \mu_0 + \mu_0^\top \Sigma_0^{-1} \mu_0, \quad \text{and} \\
(y - X\beta)^\top (y - X\beta) \;&=\; \beta^\top X^\top X\, \beta - 2\,\beta^\top X^\top y + y^\top y.
\end{aligned}$$

Discarding terms that do not depend on β, the exponent becomes

$$-\tfrac{1}{2}\Big[\beta^\top \Big(\Sigma_0^{-1} + \tfrac{1}{\sigma^2} X^\top X\Big)\beta \;-\; 2\,\beta^\top \Big(\Sigma_0^{-1}\mu_0 + \tfrac{1}{\sigma^2} X^\top y\Big)\Big].$$

Define

$$\Sigma_n^{-1} = \Sigma_0^{-1} + \tfrac{1}{\sigma^2}\, X^\top X, \qquad \mu_n = \Sigma_n \Big(\Sigma_0^{-1}\mu_0 + \tfrac{1}{\sigma^2} X^\top y\Big).$$

Then complete the square:

$$\beta^\top \Sigma_n^{-1} \beta - 2\,\beta^\top \Sigma_n^{-1} \mu_n \;=\; (\beta - \mu_n)^\top \Sigma_n^{-1} (\beta - \mu_n) \;-\; \mu_n^\top \Sigma_n^{-1} \mu_n.$$

Hence

$$p(\beta \mid \mathcal{D}) \;\propto\; \exp\Big[-\tfrac{1}{2}(\beta - \mu_n)^\top \Sigma_n^{-1} (\beta - \mu_n)\Big] \;\Longrightarrow\; p(\beta \mid \mathcal{D}) = \mathcal{N}(\beta \mid \mu_n, \Sigma_n).$$

This completes the derivation.