
Автоматическая регуляризации байесовских нейронных сетей

Препринт

Басов Дмитрий Константинович

Аннотация

В байесовском выводе, в отличие от гипотезы максимального правдоподобия, не делается никаких предположений о размере обучающей выборки. Это делает байесовские модели устойчивыми к переобучению.

Однако применение байесовского вывода сопряжено со следующими проблемами: заданием подходящего априорного распределения и вычислением апостериорного распределения весов модели.

В данной работе предлагается следующее решение этих проблем:

1. Применяя вариационный вывод, апостериорное распределение весов модели аппроксимируется нормальным распределением с диагональной матрицей ковариации, и задача сводится к максимизации нижней вариационной границы. В этом случае каждый вес модели определяется двумя обучаемыми параметрами.
2. Априорное распределение весов модели задается в виде нормального распределения с нулевым матожиданием и диагональной матрицей ковариации, элементы которой вычисляются из данных. Этот приём лежит в основе Relevance Vector Machine — байесовского варианта SVM.

Полученную модель можно рассматривать как ансамбль из бесконечного числа нейронных сетей, веса которых сэмпляются из нормального распределения. При этом каждый вес имеет свой индивидуальный коэффициент $L2$ регуляризации, который автоматически определяется из тренировочных данных при обучении.

1 Обозначения и сокращения

$\mathbf{x} \odot \mathbf{y}$ — поэлементное произведение (произведение Адамара) векторов

\mathcal{L} — Evidence Lower Bound (ELBO)

$KL(q \parallel p) = \int q(\mathbf{Z}) \ln \frac{q(\mathbf{Z})}{p(\mathbf{Z})} d\mathbf{Z}$ — дивергенция Кульбака–Лейблера (KL-дивергенции)

\mathbf{x} — вектор признаков

\mathbf{y} — вектор целевой переменной

D — датасет — пары значений $\{\mathbf{x}_i, \mathbf{y}_i\}$, где $i = 1, \dots, L$

\mathbf{W} — веса модели — случайная величина размерности M

$p(\mathbf{y}|\mathbf{x}, D)$ — предсказательное распределение

$p(D|\mathbf{W}) = \prod_{i=1}^L p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W})$ — правдоподобие (likelihood)

$p(\mathbf{W})$ — априорное распределение весов модели (prior)

$p(\mathbf{W}|D)$ — апостериорное распределение весов модели (posterior)

$q_{\theta}(\mathbf{W})$ — аппроксимация апостериорного распределения весов модели

θ — обучаемые параметры байесовской модели

2 Введение

В классическом машинном обучении делается следующее предположение: веса модели \mathbf{W} являются пусть и неизвестной, но фиксированной величиной. В этом случае можно получить точечную оценку весов модели согласно гипотезе максимального правдоподобия:

$$\mathbf{W}_{ML} = \operatorname{argmax}_{\mathbf{W}} p(D|\mathbf{W}) \quad (2.1)$$

Тогда распределение $p(\mathbf{y}|\mathbf{x}, D)$ аппроксимируется следующим образом:

$$p(\mathbf{y}|\mathbf{x}, D) \approx p(\mathbf{y}|\mathbf{x}, \mathbf{W}_{ML}) \quad (2.2)$$

Однако это справедливо при условии, что количество объектов в датасете D сильно больше количества весов модели ($L \gg M$). Если это не так, веса модели \mathbf{W} могут слишком сильно подстроиться под обучающую выборку D , что чревато переобучением.

Для борьбы с переобучением используется ряд приёмов (штрафы на норму весов, early stopping, dropout), однако для их настройки требуются вычислительные ресурсы и отложенные (не участвующие в обучении) выборки данных.

Альтернативным подходом к машинному обучению является нахождение апостериорного распределения весов модели $p(\mathbf{W}|D)$ по теореме Байеса.

$$p(\mathbf{W}|D) = \frac{p(D|\mathbf{W}) p(\mathbf{W})}{\int p(D|\mathbf{W}) p(\mathbf{W}) d\mathbf{W}} \quad (2.3)$$

Тогда предсказательное распределение $p(\mathbf{y}|\mathbf{x}, D)$ рассчитывается следующим образом:

$$p(\mathbf{y}|\mathbf{x}, D) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{W}) p(\mathbf{W}|D) d\mathbf{W} \quad (2.4)$$

Байесовские модели можно рассматривать как ансамбль из бесконечного числа моделей, веса которых сэмпляются из распределения $p(\mathbf{W}|D)$. Такой подход устойчив к переобучению, так как о размере обучающей выборки не делается никаких предположений. Однако возникают следующие проблемы: выбор подходящего априорного распределения $p(\mathbf{W})$ и вычисление апостериорного распределения $p(\mathbf{W}|D)$.

Неудачный выбор $p(\mathbf{W})$ может сильно ухудшить качество модели, а расчёт апостериорного распределения $p(\mathbf{W}|D)$ требует вычисления интеграла по всему пространству весов модели, что для нейронных сетей практически невозможно.

В данной статье предлагается следующий подход к решению этих проблем.

1. Применяя вариационный вывод, распределение $p(\mathbf{W}|D)$ аппроксимируется распределением $q_{\theta}(\mathbf{W})$, и задача сводится к максимизации нижней вариационной границы \mathcal{L} по параметрам θ .
2. Распределение $q_{\theta}(\mathbf{W})$ задаётся в виде нормального распределения с диагональной матрицей ковариации.
3. Так как распределение $q_{\theta}(\mathbf{W})$ является нормальным, применяя трюк с репараметризацией, становится возможным использовать градиентные методы для максимизации \mathcal{L} .
4. Априорное распределение весов модели $p(\mathbf{W})$ задаётся в виде нормального распределения с нулевым матожиданием и диагональной матрицей ковариации, элементы которой определяются при обучении из датасета D . Такой подход обладает большой универсальностью, однако из-за этого теряется теоретическая устойчивость к переобучению. Идея определения некоторых параметров априорного распределения $p(\mathbf{W})$ из датасета D известна как эмпирический Байес.
5. Вводятся новые параметры γ и ρ , через которые выражаются матожидание и дисперсия распределения $q_{\theta}(\mathbf{W})$. Это нужно, чтобы избежать неопределённости деления $\frac{0}{0}$, которая может возникнуть из-за определения дисперсии распределения $p(\mathbf{W})$ из данных.

3 Известные результаты

В [1], [2], [3] рассматривается применение байесовского подхода к нейронным сетям, в том числе использование эмпирического Байеса для нахождения гиперпараметров.

В работах [4], [5], [6] представлена техника Automatic Relevance Determination (ARD), которая легла в основу метода релевантных векторов (RVM) [7], байесовского варианта SVM. В отличие от SVM, RVM не требует подбора коэффициента регуляризации.

В [1], [2], [3], [4], [5] при вычислении апостериорного распределения весов модели использовалась аппроксимация Лапласа, а в [6] для сэмплирования весов из апостериорного распределения применялись методы Монте-Карло для марковских цепей (MCMC). К сожалению, данные подходы не масштабируются на большие модели.

В работе [8] была применена техника вариационного вывода для аппроксимации апостериорного распределения весов модели нормальным распределением с диагональной матрицей ковариации. Параметры распределения были найдены аналитически, что возможно только в случае нейронных сетей с одним скрытым слоем.

В [9] для оценки параметров аппроксимации апостериорного распределения используются градиентные методы, что позволяет масштабировать вариационный вывод на большие модели.

В [10] для обучения байесовской нейронной сети используется трюк с репараметризацией, который до этого был реализован в модели вариационного автокодировщика [11].

Различные способы задания априорных распределений весов байесовских нейронных сетей рассматриваются в [12], [13], [14], [15].

Обзор байесовских методов машинного обучения приведён в книгах [16], [17], [18].

4 Постановка задачи

Задача машинного обучения с учителем в вероятностной постановке формулируется следующим образом: получить распределение вероятностей $p(\mathbf{y}|\mathbf{x}, D)$ целевой переменной \mathbf{y} для неразмеченных \mathbf{x} , используя информацию из датасета D . В случае параметрических моделей, которыми являются нейронные сети, информация из датасета D кодируется посредством весов модели \mathbf{W} . Сделаем следующие преобразования:

$$p(\mathbf{y}|\mathbf{x}, D) = \int p(\mathbf{y}, \mathbf{W}|\mathbf{x}, D) d\mathbf{W} = \int p(\mathbf{y}|\mathbf{W}, \mathbf{x}, D) p(\mathbf{W}|\mathbf{x}, D) d\mathbf{W} = \int p(\mathbf{y}|\mathbf{W}, \mathbf{x}) p(\mathbf{W}|D) d\mathbf{W} \quad (4.1)$$

Пояснения:

- $p(\mathbf{y}|\mathbf{x}, D) = \int p(\mathbf{y}, \mathbf{W}|\mathbf{x}, D) d\mathbf{W}$, так как для любых случайных величин \mathbf{a} и \mathbf{b} справедливо $p(\mathbf{a}) = \int p(\mathbf{a}, \mathbf{b}) d\mathbf{b}$
- $p(\mathbf{y}, \mathbf{W}|\mathbf{x}, D) = p(\mathbf{y}|\mathbf{W}, \mathbf{x}, D) p(\mathbf{W}|\mathbf{x}, D)$, так как для любых случайных величин \mathbf{a} и \mathbf{b} справедливо $p(\mathbf{a}, \mathbf{b}) = p(\mathbf{a}|\mathbf{b}) p(\mathbf{b})$
- $p(\mathbf{y}|\mathbf{W}, \mathbf{x}, D) = p(\mathbf{y}|\mathbf{W}, \mathbf{x})$, так как вся информация из датасета D отражена в весах \mathbf{W}
- $p(\mathbf{W}|\mathbf{x}, D) = p(\mathbf{W}|D)$, так как веса модели \mathbf{W} не зависят от неразмеченных \mathbf{x} , которых не было в датасете D .

Получим выражение для $p(\mathbf{W}|D)$, используя формулу Байеса:

$$p(\mathbf{W}|D) = \frac{p(\mathbf{W}, D)}{p(D)} = \frac{p(\mathbf{W}, D)}{\int p(\mathbf{W}, D) d\mathbf{W}} = \frac{p(D|\mathbf{W}) p(\mathbf{W})}{\int p(D|\mathbf{W}) p(\mathbf{W}) d\mathbf{W}} \quad (4.2)$$

Предсказательное распределение можно аппроксимировать следующим образом:

$$p(\mathbf{y}|\mathbf{x}, D) \approx \frac{1}{T} \sum_{t=1}^T p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{W}}_t) \quad (4.3)$$

где $\hat{\mathbf{W}}_t$ — сэмпл весов модели из распределения $p(\mathbf{W}|D)$.

Однако для этого нужно иметь возможность сэмплировать из распределения $p(\mathbf{W}|D)$.

Один из подходов к решению этой проблемы — сэмплирование из $p(\mathbf{W}|D)$ используя методы Монте-Карло для марковских цепей (MCMC). Однако для больших датасетов и большого числа весов это практически невозможно.

Другим подходом является вариационный вывод — аппроксимация распределения $p(\mathbf{W}|D)$ распределением $q_{\theta}(\mathbf{W})$, из которого сэмплировать намного проще.

5 Вариационный вывод

Идея вариационного вывода — сведение задачи байесовского вывода к задаче максимизации нижней вариационной границы (ELBO) \mathcal{L} , которая для распределения $q_{\theta}(\mathbf{W})$ записывается следующим образом:

$$\mathcal{L} = \int q_{\theta}(\mathbf{W}) \ln \frac{p(\mathbf{W}, D)}{q_{\theta}(\mathbf{W})} d\mathbf{W} \quad (5.1)$$

Покажем мотивацию максимизации \mathcal{L} . Запишем выражение для $KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W}|D))$ и преобразуем его, используя тождество $p(\mathbf{W}, D) = p(\mathbf{W}|D)p(D)$:

$$\begin{aligned} KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W}|D)) &= \int q_{\theta}(\mathbf{W}) \ln \frac{q_{\theta}(\mathbf{W})}{p(\mathbf{W}|D)} d\mathbf{W} = \\ &= \int q_{\theta}(\mathbf{W}) \ln \frac{p(D) q_{\theta}(\mathbf{W})}{p(\mathbf{W}, D)} d\mathbf{W} = \\ &= \ln p(D) \int q_{\theta}(\mathbf{W}) d\mathbf{W} - \int q_{\theta}(\mathbf{W}) \ln \frac{p(\mathbf{W}, D)}{q_{\theta}(\mathbf{W})} d\mathbf{W} = \\ &= \ln p(D) - \mathcal{L} \end{aligned} \quad (5.2)$$

Так как $\ln p(D)$ не зависит от θ , максимизация \mathcal{L} по параметрам θ ведёт к минимизации $KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W}|D))$. Тем самым, при максимизации \mathcal{L} распределение $q_{\theta}(\mathbf{W})$ будет приближаться к распределению $p(\mathbf{W}|D)$.

Преобразуем выражение для \mathcal{L} , используя тождество $p(\mathbf{W}, D) = p(D|\mathbf{W}) \cdot p(\mathbf{W})$.

$$\begin{aligned} \mathcal{L} &= \int q_{\theta}(\mathbf{W}) \ln \frac{p(\mathbf{W}, D)}{q_{\theta}(\mathbf{W})} d\mathbf{W} = \int q_{\theta}(\mathbf{W}) \ln \frac{p(D|\mathbf{W}) p(\mathbf{W})}{q_{\theta}(\mathbf{W})} d\mathbf{W} = \\ &= \int q_{\theta}(\mathbf{W}) \ln p(D|\mathbf{W}) d\mathbf{W} - \int q_{\theta}(\mathbf{W}) \ln \frac{q_{\theta}(\mathbf{W})}{p(\mathbf{W})} d\mathbf{W} = \\ &= \int q_{\theta}(\mathbf{W}) \ln p(D|\mathbf{W}) d\mathbf{W} - KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W})) \end{aligned} \quad (5.3)$$

Поставив $p(D|\mathbf{W}) = \prod_{i=1}^L p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W})$ в (5.3), получим:

$$\mathcal{L} = \int q_{\theta}(\mathbf{W}) \sum_{i=1}^L \ln p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W}) d\mathbf{W} - KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W})) \quad (5.4)$$

Так как в случае нейронной сети аналитически посчитать интеграл по всему пространству весов \mathbf{W} не представляется возможным, воспользуемся следующей аппроксимацией для $p(\mathbf{y}|\mathbf{x}, D)$ и \mathcal{L} :

$$p(\mathbf{y}|\mathbf{x}, D) \approx \int p(\mathbf{y}|\mathbf{W}, \mathbf{x}) q_{\theta}(\mathbf{W}) d\mathbf{W} \approx \frac{1}{T} \sum_{t=1}^T p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{W}}_t) \quad (5.5)$$

$$\mathcal{L} \approx \frac{1}{S} \sum_{j=1}^S \sum_{i=1}^L \ln p(\mathbf{y}_i|\mathbf{x}_i, \hat{\mathbf{W}}_{ij}) - KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W})) \quad (5.6)$$

где $\hat{\mathbf{W}}_t$ и $\hat{\mathbf{W}}_{ij}$ — сэмплы весов модели из распределения $q_{\theta}(\mathbf{W})$.

6 Задание функциональных форм распределений

Для дальнейшего вывода положим, что распределения $p(\mathbf{W})$ и $q_{\theta}(\mathbf{W})$ являются нормальными с диагональными матрицами ковариации:

$$q_{\theta}(\mathbf{W}) = N(\mathbf{W}|\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma})^2) = \prod_{k=1}^M N(W_k|\mu_k, \sigma_k^2) \quad (6.1)$$

$$p(\mathbf{W}) = N(\mathbf{W}|\mathbf{0}, \text{diag}(\boldsymbol{\delta})^2) = \prod_{k=1}^M N(W_k|0, \delta_k^2) \quad (6.2)$$

Так как распределение $q_{\theta}(\mathbf{W})$ нормальное, мы можем использовать трюк с репараметризацией при сэмплировании весов, что позволяет использовать градиентные методы для оптимизации:

$$\hat{\mathbf{W}}_{ij} = \boldsymbol{\varepsilon}_{ij} \odot \boldsymbol{\sigma} + \boldsymbol{\mu} \quad (6.3)$$

$$\boldsymbol{\varepsilon}_{ij} \sim N(\mathbf{0}, \mathbf{I}) \quad (6.4)$$

Так как распределения $p(\mathbf{W})$ и $q_{\theta}(\mathbf{W})$ нормальные, $KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W}))$ считается аналитически:

$$KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^M \left(\frac{\sigma_k^2}{\delta_k^2} + \frac{\mu_k^2}{\delta_k^2} - \ln \frac{\sigma_k^2}{\delta_k^2} - 1 \right) \quad (6.5)$$

Подставив (6.5) в (5.6), получим:

$$\mathcal{L} \approx \frac{1}{S} \sum_{j=1}^S \sum_{i=1}^L \ln p(\mathbf{y}_i | \mathbf{x}_i, \hat{\mathbf{W}}_{ij}) - \frac{1}{2} \sum_{k=1}^M \left(\frac{\sigma_k^2}{\delta_k^2} + \frac{\mu_k^2}{\delta_k^2} - \ln \frac{\sigma_k^2}{\delta_k^2} - 1 \right) \quad (6.6)$$

Таким образом $\boldsymbol{\mu}$ и $\boldsymbol{\sigma}$ это обучаемые параметры модели, а параметр $\boldsymbol{\delta}$ — гиперпараметр (так как является параметром априорного распределения $p(\mathbf{W})$). В байесовском выводе все параметры априорного распределения должны задаваться до начала обучения. Однако мы можем определить параметр $\boldsymbol{\delta}$ из данных. Такой прием называется эмпирический Байес.

7 Эмпирический Байес

Найдём такое значение $\boldsymbol{\delta}$, при котором \mathcal{L} максимальна. Так как в выражении (5.6) левое слагаемое не зависит от параметров распределения $p(\mathbf{W})$, то максимум \mathcal{L} достигается при минимуме $KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W}))$ по параметру $\boldsymbol{\delta}$.

Пусть $\boldsymbol{\alpha} = \text{diag}(\boldsymbol{\delta})^{-2}$. Тогда выражение (6.5) будет иметь следующий вид:

$$KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^M (\alpha_k \cdot \sigma_k^2 + \alpha_k \cdot \mu_k^2 - (\ln \sigma_k^2 + \ln \alpha_k) - 1) \quad (7.1)$$

Найдём производную $KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W}))$ по параметру $\boldsymbol{\alpha}$:

$$\frac{\partial(KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W})))}{\partial \alpha_k} = \frac{1}{2} \left(\sigma_k^2 + \mu_k^2 - \frac{1}{\alpha_k} \right) = \frac{1}{2} (\sigma_k^2 + \mu_k^2 - \delta_k^2) \quad (7.2)$$

Приравняв (7.2) к нулю, получим выражение для оптимального значения $\boldsymbol{\delta}$:

$$\delta_k^2 = \sigma_k^2 + \mu_k^2 \quad (7.3)$$

Подставив (7.3) в (6.5) и (6.6), получим:

$$KL(q_{\theta}(\mathbf{W}) \parallel p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^M \ln \left(1 + \frac{\mu_k^2}{\sigma_k^2} \right) \quad (7.4)$$

$$\mathcal{L} \approx \frac{1}{S} \sum_{j=1}^S \sum_{i=1}^L \ln p(\mathbf{y}_i | \mathbf{x}_i, \hat{\mathbf{W}}_{ij}) - \frac{1}{2} \sum_{k=1}^M \ln \left(1 + \frac{\mu_k^2}{\sigma_k^2} \right) \quad (7.5)$$

Таким образом задача свелась к максимизации \mathcal{L} по параметрам $\boldsymbol{\mu}$ и $\boldsymbol{\sigma}$. Однако градиентная оптимизация по параметрам $\boldsymbol{\mu}$ и $\boldsymbol{\sigma}$ может привести к численной неустойчивости.

8 Замена переменных

При максимизации \mathcal{L} могут возникнуть ситуации, когда какой-либо вес модели перестает быть случайной величиной и вырождается в ноль ($\sigma_k \rightarrow 0$ и $\mu_k \rightarrow 0$). Это приведет к неопределенности деления 0 на 0 при вычислении KL-дивергенции.

Так же при градиентной оптимизации компоненты вектора σ могут попасть в отрицательную область, что нежелательно, так как среднеквадратическое отклонение не может быть отрицательным по определению.

Чтобы избежать этих проблем, определим параметры σ и μ через новые параметры ρ и γ следующим образом:

$$\sigma = \ln(1 + e^\rho) = \text{Softplus}(\rho) \quad (8.1)$$

$$\mu = \gamma \odot \sigma = \gamma \odot \text{Softplus}(\rho) \quad (8.2)$$

Подставив (8.1) и (8.2) в (7.4) и (7.5), получим:

$$KL(q_\theta(\mathbf{W}) \parallel p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^M \ln(1 + \gamma_k^2) \quad (8.3)$$

$$\mathcal{L} \approx \frac{1}{S} \sum_{j=1}^S \sum_{i=1}^L \ln p(\mathbf{y}_i | \mathbf{x}_i, \hat{\mathbf{W}}_{ij}) - \frac{1}{2} \sum_{k=1}^M \ln(1 + \gamma_k^2) \quad (8.4)$$

Таким образом, задача свелась к максимизации (8.4) по параметрам ρ и γ . Значение $\hat{\mathbf{W}}_{ij}$ вычисляется по (6.3), которое в свою очередь через (6.4), (8.1) и (8.2).

9 Алгоритм обучения

Возьмем выражение для \mathcal{L} из (8.4) со знаком минус и разделив на размер обучающей выборки L , получим следующую функцию потерь:

$$\text{loss}(\rho, \gamma) = -\frac{1}{S \cdot L} \sum_{j=1}^S \sum_{i=1}^L \ln p(\mathbf{y}_i | \mathbf{x}_i, \hat{\mathbf{W}}_{ij}) + \frac{\frac{1}{2} \sum_{k=1}^M \ln(1 + \gamma_k^2)}{L} \quad (9.1)$$

Введём следующие упрощения расчёта функции потерь на каждом градиентном шаге:

- на каждый объект делать только один сэмпл весов (то есть задать $S = 1$);
- средний отрицательный логарифм правдоподобия считать не по всей обучающей выборке, а на случайном подмножестве (батче).

Тогда выражение (9.1) для функции потерь будет выглядеть следующим образом:

$$\text{loss}(\rho, \gamma) \approx -\frac{1}{B} \sum_{b=1}^B \ln p(\mathbf{y}_b | \mathbf{x}_b, \hat{\mathbf{W}}_b) + \frac{\frac{1}{2} \sum_{k=1}^M \ln(1 + \gamma_k^2)}{L} \quad (9.2)$$

Для минимизации 9.2 можно использовать любые методы градиентной оптимизации. Для наглядности рассмотрим алгоритм стохастического градиентного спуска.

Задаем шаг градиентного спуска η и инициализируем параметры распределения ρ и γ . Затем повторяем, пока не достигнем критерия остановки:

1. $\sigma \leftarrow \text{Softplus}(\rho)$ — расчёт среднеквадратических отклонений весов
2. $\mu \leftarrow \gamma \odot \sigma$ — расчёт математических ожиданий весов
3. $\varepsilon \leftarrow N(0, 1)$ — сэмплирование случайных весов
4. $\hat{\mathbf{W}} \leftarrow \varepsilon \odot \sigma + \mu$ — репараметризация
5. $nll \leftarrow -\frac{1}{B} \sum_{b=1}^B \ln p(\mathbf{y}_b | \mathbf{x}_b, \hat{\mathbf{W}})$ — расчёт среднего отрицательного логарифма правдоподобия

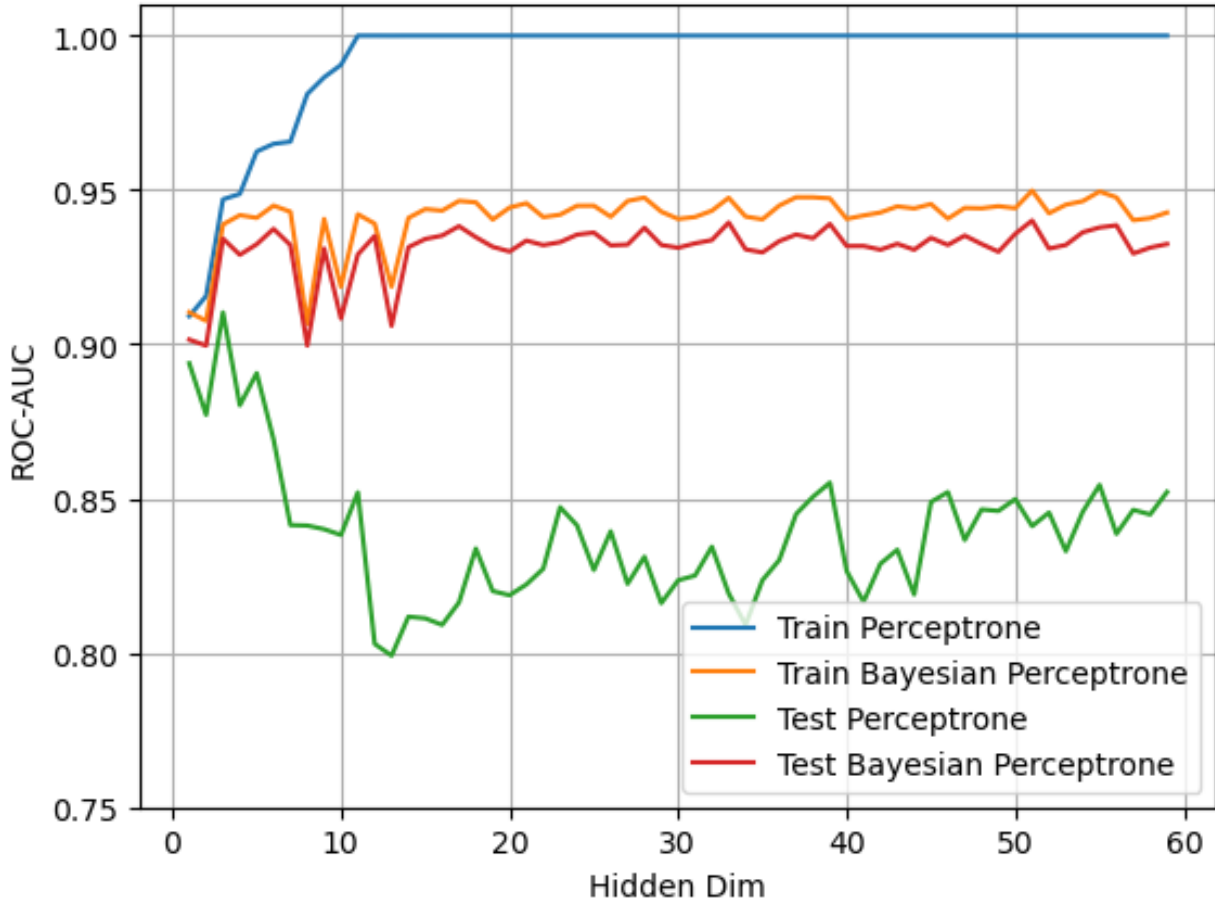


Рис. 1: Зависимость ROC-AUC от размерности скрытого состояния на тренировочных и тестовых данных

6. $kl \leftarrow \frac{1}{2} \sum_{k=1}^M \ln(1 + \gamma_k^2)$ — расчёт KL-дивергенции

7. $l \leftarrow nll + \frac{kl}{L}$ — расчёт функции потерь

8. $\rho \leftarrow \rho - \eta \frac{\partial l}{\partial \rho}$ — обновление ρ

9. $\gamma \leftarrow \gamma - \eta \frac{\partial l}{\partial \gamma}$ — обновление γ

10 Эксперименты

Для проверки своей гипотезы я выбрал Alzheimer's Disease Dataset. Данные были разбиты на тренировочную и тестовую часть в пропорции 80 на 20. В качестве архитектуры была выбрана полносвязная нейронная сеть с одним скрытым слоем и функцией активации ReLU. То есть:

$$z = \text{ReLU}(\text{matmul}(x, W_1))$$

$$y = \text{Sigmoid}(\text{matmul}(z, W_2))$$

Размерность скрытого состояния z варьировалась от 1 до 60. Для каждой размерности обучались 2 модели — классическая (без регуляризации) и байесовская. Для каждой модели производилась оценка ROC-AUC на тренировочной и тестовой выборках.

На рисунке 1 представлены результаты экспериментов.

11 Выводы

По результатам работы можно сделать следующие выводы:

- с ростом сложности модели байесовская нейронная сеть не переобучилась;
- значение ROC-AUC на тестовой выборке имеет очень высокую корреляцию со значением ROC-AUC на тренировочной выборке (0.97 по Пирсону). Следовательно, для подбора гиперпараметров можно ориентироваться на метрики, полученные по тренировочной выборке. Это даёт нам возможность отказаться от деления на тренировочную и валидационную выборки для подбора гиперпараметров.

Так же стоит отметить, что данный подход переносится на другие архитектуры нейронных сетей (рекуррентные, свёрточные, трансформеры).

Имплементация данного подхода была выполнена с использованием PyTorch. Весь исходный код для проведения экспериментов размещён по адресу <https://github.com/dimabasow/bayesian-neural-networks>.

Список литературы

- [1] MacKay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation* 4(3), 415–447.
- [2] MacKay, D. J. C. (1992b). The evidence framework applied to classification networks. *Neural Computation* 4(5), 720–736.
- [3] MacKay, D. J. C. (1992c). A practical Bayesian framework for back-propagation networks. *Neural Computation* 4(3), 448–472.
- [4] MacKay, D. J. C. (1994). Bayesian methods for backprop networks. In E. Domany, J. L. van Hemmen, and K. Schulten (Eds.), *Models of Neural Networks, III*, Chapter 6, pp. 211–254. Springer.
- [5] MacKay, D. J. C. (1995). Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks”. In: *Network: Computation in Neural Systems* 6.3, pp. 469–505.
- [6] R. Neal. (1996). *Bayesian learning for neural networks*. Springer.
- [7] Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* 1, 211–244.
- [8] Hinton, G. E. and D. van Camp (1993). Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pp. 5–13. ACM.
- [9] A. Graves (2011). *Practical Variational Inference for Neural Networks*. NIPS.
- [10] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. (2015). *Weight Uncertainty in Neural Networks*. ICML.
- [11] Kingma D. P., Welling M. (2013). Welling M. *Auto-Encoding Variational Bayes*. ArXiv e-prints.
- [12] E. T. Nalisnick. (2018). *On Priors for Bayesian Neural Networks*. PhD thesis. UC Irvine.
- [13] A. G. Wilson and P. Izmailov. (2020). *Bayesian Deep Learning and a Probabilistic Perspective of Generalization*. NIPS.
- [14] T. Hoeffler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste. (2021). *Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks*. In arXiv: 2102.00554.
- [15] V. Fortuin. (2022). *Priors in Bayesian Deep Learning: A Review*. In: *Intl. Statistical Review*.
- [16] C. Bishop (2006). *Pattern recognition and machine learning*. Springer.
- [17] Kevin P. Murphy. (2022). *Probabilistic Machine Learning: An introduction*. MIT Press.
- [18] Kevin P. Murphy. (2023). *Probabilistic Machine Learning: Advanced Topics*. MIT Press.