Автоматическая регуляризации байесовских нейронных сетей

A Preprint

Басов Дмитрий Константинович

Abstract

В байесовском выводе, в отличии 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 вычисляется 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 сэмплируются 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

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

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

$KL(q \mid\mid p) = \int q(\boldsymbol{Z}) \ln \frac{q(\boldsymbol{Z})}{p(\boldsymbol{Z})} \, d\boldsymbol{Z}$ — дивергенция Кульбака–Лейблера (KL–дивергенции)

$\boldsymbol{x}$ — вектор признаков

$\boldsymbol{y}$ — вектор целевой переменной

D — датасет — пары значений $\{\boldsymbol{x_i}, \boldsymbol{y_i}\}$, где $i = 1, \ldots, L$

$\boldsymbol{W}$ — веса модели — случайная величина размерности M

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

$p(D|\boldsymbol{W}) = \prod_{i=1}^{L} p(\boldsymbol{y_i}|\boldsymbol{x_i}, \boldsymbol{W})$ — правдоподобие (likelihood)

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

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

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

$\boldsymbol{\theta}$ — обучаемые параметры байесовской модели

2 Введение

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

$$\boldsymbol{W_{ML}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, p(D|\boldsymbol{W}) \tag{2.1}$$

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

$$p(\boldsymbol{y}|\boldsymbol{x}, D) \approx p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W_{ML}}) \tag{2.2}$$

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 $\boldsymbol{W}$ могут слишком сильно подстроиться под обучающую выборку D, что черевато 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 $p(\boldsymbol{W}|D)$ по теореме Байеса.

$$p(\boldsymbol{W}|D) = \frac{p(D|\boldsymbol{W})\, p(\boldsymbol{W})}{\int p(D|\boldsymbol{W})\, p(\boldsymbol{W})\, d\boldsymbol{W}} \tag{2.3}$$

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

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

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

Неудачный выбор $p(\boldsymbol{W})$ может сильно ухудшить качество модели, а расчёт апостериорного распределения $p(\boldsymbol{W}|D)$ 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 $p(\boldsymbol{W}|D)$ аппроксимируется распределением $q_{\boldsymbol{\theta}}(\boldsymbol{W})$, и задача сводится к максимизации нижней вариационной границы $\mathcal{L}$ по параметрам $\boldsymbol{\theta}$.
2. Распределение $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ задаётся в виде нормального распределения с диагональной матрицей ковариации.
3. Так как распределение $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ является нормальным, применяя трюк с репараметризацией, становится возможным использовать градиентные методы для максимизации $\mathcal{L}$.
4. Априорное распределение весов модели $p(\boldsymbol{W})$ 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 $p(\boldsymbol{W})$ из датасета D известна как эмпирический Байес.
5. Вводятся новые параметры $\boldsymbol{\gamma}$ и $\boldsymbol{\rho}$, через которые выражаются матожидание и дисперсия распределения $q_{\boldsymbol{\theta}}(\boldsymbol{W})$. Это нужно, чтобы избежать неопределенности деления $\frac{0}{0}$, которая может возникнуть из–за определения дисперсии распределения $p(\boldsymbol{W})$ из данных.

3 Постановка задачи

Задача машинного обучения с учителем в вероятностной постановке формулируется следующим образом: получить распределение вероятностей $p(\boldsymbol{y}|\boldsymbol{x}, D)$ целевой переменной $\boldsymbol{y}$ для неразмеченных $\boldsymbol{x}$,

используя информацию из датасета D. В случае параметрических моделей, которыми являются нейронные сети, информация из датасета D кодируется посредством весов модели $\boldsymbol{W}$. Сделаем следующие преобразования:

$$p(\boldsymbol{y}|\boldsymbol{x}, D) = \int p(\boldsymbol{y}, \boldsymbol{W}|\boldsymbol{x}, D)\, d\boldsymbol{W} = \int p(\boldsymbol{y}|\boldsymbol{W}, \boldsymbol{x}, D)\, p(\boldsymbol{W}|\boldsymbol{x}, D)\, d\boldsymbol{W} = \int p(\boldsymbol{y}|\boldsymbol{W}, \boldsymbol{x})\, p(\boldsymbol{W}|D)\, d\boldsymbol{W} \tag{3.1}$$

Пояснения:

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

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

$$p(\boldsymbol{W}|D) = \frac{p(\boldsymbol{W}, D)}{p(D)} = \frac{p(\boldsymbol{W}, D)}{\int p(\boldsymbol{W}, D)\, d\boldsymbol{W}} = \frac{p(D|\boldsymbol{W})\, p(\boldsymbol{W})}{\int p(D|\boldsymbol{W})\, p(\boldsymbol{W})\, d\boldsymbol{W}} \tag{3.2}$$

Предсказательное распределение можно аппроксимировать следующим образом:

$$p(\boldsymbol{y}|\boldsymbol{x}, D) \approx \frac{1}{T} \sum_{t=1}^{T} p(\boldsymbol{y}|\boldsymbol{x}, \hat{\boldsymbol{W}}_{\boldsymbol{t}}) \tag{3.3}$$

где $\hat{\boldsymbol{W}}_{\boldsymbol{t}}$ — сэмпл весов модели из распределения $p(\boldsymbol{W}|D)$.

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

Один из подходов к решению этой проблемы — сэмплирование из $p(\boldsymbol{W}|D)$ 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 $p(\boldsymbol{W}|D)$ распределением $q_{\boldsymbol{\theta}}(\boldsymbol{W})$, из которого сэмплировать намного проще.

4 Вариационный вывод

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

$$\mathcal{L} = \int q_{\boldsymbol{\theta}}(\boldsymbol{W})\, \ln \frac{p(\boldsymbol{W}, D)}{q_{\boldsymbol{\theta}}(\boldsymbol{W})}\, d\boldsymbol{W} \tag{4.1}$$

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

$$\begin{aligned} KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \parallel p(\boldsymbol{W}|D)) = \int q_{\boldsymbol{\theta}}(\boldsymbol{W})\, \ln \frac{q_{\boldsymbol{\theta}}(\boldsymbol{W})}{p(\boldsymbol{W}|D)} d\boldsymbol{W} = \\ \int q_{\boldsymbol{\theta}}(\boldsymbol{W})\, \ln \frac{p(D)\, q_{\boldsymbol{\theta}}(\boldsymbol{W})}{p(\boldsymbol{W}, D)} d\boldsymbol{W} = \\ \ln p(D) \int q_{\boldsymbol{\theta}}(\boldsymbol{W}) d\boldsymbol{W} - \int q_{\boldsymbol{\theta}}(\boldsymbol{W})\, \ln \frac{p(\boldsymbol{W}, D)}{q_{\boldsymbol{\theta}}(\boldsymbol{W})} d\boldsymbol{W} = \\ \ln p(D) - \mathcal{L} \end{aligned} \tag{4.2}$$

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

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

$$\begin{aligned}\mathcal{L} = \int q_{\boldsymbol{\theta}}(\boldsymbol{W}) \ln \frac{p(\boldsymbol{W}, D)}{q_{\boldsymbol{\theta}}(\boldsymbol{W})} d\boldsymbol{W} = \int q_{\boldsymbol{\theta}}(\boldsymbol{W}) \ln \frac{p(D|\boldsymbol{W})\, p(\boldsymbol{W})}{q_{\boldsymbol{\theta}}(\boldsymbol{W})} d\boldsymbol{W} = \\ \int q_{\boldsymbol{\theta}}(\boldsymbol{W}) \ln p(D|\boldsymbol{W}) d\boldsymbol{W} - \int q_{\boldsymbol{\theta}}(\boldsymbol{W}) \ln \frac{q_{\boldsymbol{\theta}}(\boldsymbol{W})}{p(\boldsymbol{W})} d\boldsymbol{W} = \\ \int q_{\boldsymbol{\theta}}(\boldsymbol{W}) \ln p(D|\boldsymbol{W}) d\boldsymbol{W} - KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \,||\, p(\boldsymbol{W})) = \\ \int q_{\boldsymbol{\theta}}(\boldsymbol{W}) \sum_{i=1}^{L} \ln p(\boldsymbol{y_i}|\boldsymbol{x_i}, \boldsymbol{W}) d\boldsymbol{W} - KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \,||\, p(\boldsymbol{W}))\end{aligned} \tag{4.3}$$

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

$$p(\boldsymbol{y}|\boldsymbol{x}, D) \approx \int p(\boldsymbol{y}|\boldsymbol{W}, \boldsymbol{x})\, q_{\boldsymbol{\theta}}(\boldsymbol{W}) d\boldsymbol{W} \approx \frac{1}{T} \sum_{t=1}^{T} p(\boldsymbol{y}|\boldsymbol{x}, \hat{\boldsymbol{W}}_{\boldsymbol{t}}) \tag{4.4}$$

$$\mathcal{L} \approx \frac{1}{S} \sum_{j=1}^{S} \sum_{i=1}^{L} \ln p(\boldsymbol{y_i}|\boldsymbol{x_i}, \hat{\boldsymbol{W}}_{\boldsymbol{ij}}) - KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \,||\, p(\boldsymbol{W})) \tag{4.5}$$

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

5 Задание функциональных форм распределений

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

$$q_{\boldsymbol{\theta}}(\boldsymbol{W}) = N(\boldsymbol{W}|\boldsymbol{\mu}, diag(\boldsymbol{\sigma})^2) \tag{5.1}$$

$$p(\boldsymbol{W}) = N(\boldsymbol{W}|\boldsymbol{0}, diag(\boldsymbol{\delta})^2) \tag{5.2}$$

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

$$\hat{\boldsymbol{W}}_{\boldsymbol{ij}} = \boldsymbol{\varepsilon_{ij}} \odot \boldsymbol{\sigma} + \boldsymbol{\mu} \tag{5.3}$$

$$\boldsymbol{\varepsilon_{ij}} \sim N(\boldsymbol{0}, \boldsymbol{I}) \tag{5.4}$$

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

$$KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \,||\, p(\boldsymbol{W})) = \frac{1}{2} \sum_{k=1}^{M} (\frac{\sigma_k^2}{\delta_k^2} + \frac{\mu_k^2}{\delta_k^2} - \ln \frac{\sigma_k^2}{\delta_k^2} - 1) \tag{5.5}$$

Подставив 5.5 в 4.5, получим:

$$\mathcal{L} \approx \frac{1}{S} \sum_{j=1}^{S} \sum_{i=1}^{L} \ln p(\boldsymbol{y_i}|\boldsymbol{x_i}, \hat{\boldsymbol{W}}_{\boldsymbol{ij}}) - \frac{1}{2} \sum_{k=1}^{M} (\frac{\sigma_k^2}{\delta_k^2} + \frac{\mu_k^2}{\delta_k^2} - \ln \frac{\sigma_k^2}{\delta_k^2} - 1) \tag{5.6}$$

Таким образом $\boldsymbol{\mu}$ и $\boldsymbol{\sigma}$ это обучаемые параметры модели, а параметр $\boldsymbol{\delta}$ — гиперпараметр (так как является параметром априорного распределения $p(\boldsymbol{W})$). 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.

6 Эмпирический Байес

Найдём такое значение $\boldsymbol{\delta}$, при котором $\mathcal{L}$ максимальна. Так как в выражении 4.5 левое слагаемое не зависит от параметров распределения $p(\boldsymbol{W})$, то максимум $\mathcal{L}$ достигается при минимуме $KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \mid\mid p(\boldsymbol{W}))$ по параметру $\boldsymbol{\delta}$.

Пусть $\boldsymbol{\alpha} = diag(\boldsymbol{\delta})^{-2}$. Тогда выражение 5.5 будет иметь следующий вид:

$$KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \mid\mid p(\boldsymbol{W})) = \frac{1}{2}\sum_{k=1}^{M}\left(\alpha_k \cdot \sigma_k^2 + \alpha_k \cdot \mu_k^2 - (\ln\sigma_k^2 + \ln\alpha_k) - 1\right) \tag{6.1}$$

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

$$\frac{\partial(KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \mid\mid p(\boldsymbol{W})))}{\partial\alpha_k} = \frac{1}{2}\left(\sigma_k^2 + \mu_k^2 - \frac{1}{\alpha_k}\right) = \frac{1}{2}\left(\sigma_k^2 + \mu_k^2 - \delta_k^2\right) \tag{6.2}$$

Приравняв 6.2 к нулю, получим выражение для оптимального значения $\boldsymbol{\delta}$:

$$\delta_k^2 = \sigma_k^2 + \mu_k^2 \tag{6.3}$$

Подставив 6.3 в 5.5 и 5.6, получим:

$$KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \mid\mid p(\boldsymbol{W})) = \frac{1}{2}\sum_{k=1}^{M}\ln\left(1 + \frac{\mu_k^2}{\sigma_k^2}\right) \tag{6.4}$$

$$\mathcal{L} \approx \frac{1}{S}\sum_{j=1}^{S}\sum_{i=1}^{L}\ln p(\boldsymbol{y_i}|\boldsymbol{x_i}, \hat{\boldsymbol{W}}_{\boldsymbol{ij}}) - \frac{1}{2}\sum_{k=1}^{M}\ln\left(1 + \frac{\mu_k^2}{\sigma_k^2}\right) \tag{6.5}$$

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 нестабильности.

7 Замена переменных

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

Так же при градиентной оптимизации компоненты вектора $\boldsymbol{\sigma}$ могут попасть в отрицательную область, что нежелательно, так как среднеквадратическое отклонение не может быть отрицательным по определению.

Чтобы избежать этих проблем, определим параметры $\boldsymbol{\sigma}$ и $\boldsymbol{\mu}$ через новые параметры $\boldsymbol{\rho}$ и $\boldsymbol{\gamma}$ следующим образом:

$$\boldsymbol{\sigma} = \ln(1 + e^{\boldsymbol{\rho}}) = Softplus(\boldsymbol{\rho}) \tag{7.1}$$

$$\boldsymbol{\mu} = \boldsymbol{\gamma} \odot \boldsymbol{\sigma} = \boldsymbol{\gamma} \odot Softplus(\boldsymbol{\rho}) \tag{7.2}$$

Подставив 7.1 и 7.2 в 5.5 и 5.6, получим:

$$KL(q_{\boldsymbol{\theta}}(\boldsymbol{W}) \mid\mid p(\boldsymbol{W})) = \frac{1}{2}\sum_{k=1}^{M}\ln(1 + \gamma_k^2) \tag{7.3}$$

$$\mathcal{L} \approx \frac{1}{S}\sum_{j=1}^{S}\sum_{i=1}^{L}\ln p(\boldsymbol{y_i}|\boldsymbol{x_i}, \hat{\boldsymbol{W}}_{\boldsymbol{ij}}) - \frac{1}{2}\sum_{k=1}^{M}\ln(1 + \gamma_k^2) \tag{7.4}$$

Таким образом, задача свелась к максимизации 7.4 по параметрам $\boldsymbol{\rho}$ и $\boldsymbol{\gamma}$. Значение $\hat{\boldsymbol{W}}_{\boldsymbol{ij}}$ вычисляется по 5.3, которое в свою очередь через 5.4, 7.1 и 7.2.

8 Алгоритм обучения

Возьмем выражение для $\mathcal{L}$ из 7.4 со знаком минус и разделив на размер обучеющей выборки L, получим следующую функцию потерь:

$$loss(\boldsymbol{\rho}, \boldsymbol{\gamma}) = -\frac{1}{S \cdot L} \sum_{j=1}^{S} \sum_{i=1}^{L} \ln p(\boldsymbol{y_i} | \boldsymbol{x_i}, \hat{\boldsymbol{W}}_{\boldsymbol{ij}}) + \frac{\frac{1}{2} \sum_{k=1}^{M} \ln(1 + \gamma_k^2)}{L} \tag{8.1}$$

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 обучающех выборке, а на случайном подмножестве (батче).

Тогда выражение 8.1 для функции потерь будет выглядить следующие образом:

$$loss(\boldsymbol{\rho}, \boldsymbol{\gamma}) \approx -\frac{1}{B} \sum_{b=1}^{B} \ln p(\boldsymbol{y_b} | \boldsymbol{x_b}, \hat{\boldsymbol{W}}_{\boldsymbol{b}}) + \frac{\frac{1}{2} \sum_{k=1}^{M} \ln(1 + \gamma_k^2)}{L} \tag{8.2}$$

Запишем алгоритм стохастического градиентного спуска для минимизации 8.2.

Задаем шаг градиентного спуска η и инициализируем параметры распределения $\boldsymbol{\rho}$ и $\boldsymbol{\gamma}$. Затем повторяем, пока не достигнем критерия остановки:

1. $\boldsymbol{\sigma} \leftarrow Softplus(\boldsymbol{\rho})$ — расчёт среднеквадратических отклонений весов
2. $\boldsymbol{\mu} \leftarrow \boldsymbol{\gamma} \odot \boldsymbol{\sigma}$ — расчёт математических ожиданий весов
3. $\boldsymbol{\varepsilon} \leftarrow N(0, 1)$ — сэмплирование случайных весов
4. $\hat{\boldsymbol{W}} \leftarrow \boldsymbol{\varepsilon} \odot \boldsymbol{\sigma} + \boldsymbol{\mu}$ — репараметризация
5. $nll \leftarrow -\frac{1}{B} \sum_{b=1}^{B} \ln p(\boldsymbol{y_b} | \boldsymbol{x_b}, \hat{\boldsymbol{W}})$ — расчёт среднего отрицательного логарифма правдоподобия
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. $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} - \eta \frac{\partial l}{\partial \boldsymbol{\rho}}$ — обновление $\boldsymbol{\rho}$
9. $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} - \eta \frac{\partial l}{\partial \boldsymbol{\gamma}}$ — обновление $\boldsymbol{\gamma}$

9 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:

$z = ReLU(matmul(x, W_1))$
$y = Sigmoid(matmul(z, W_2))$

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.

10 Выводы

По результатам работы можно сделать следующие выводы:

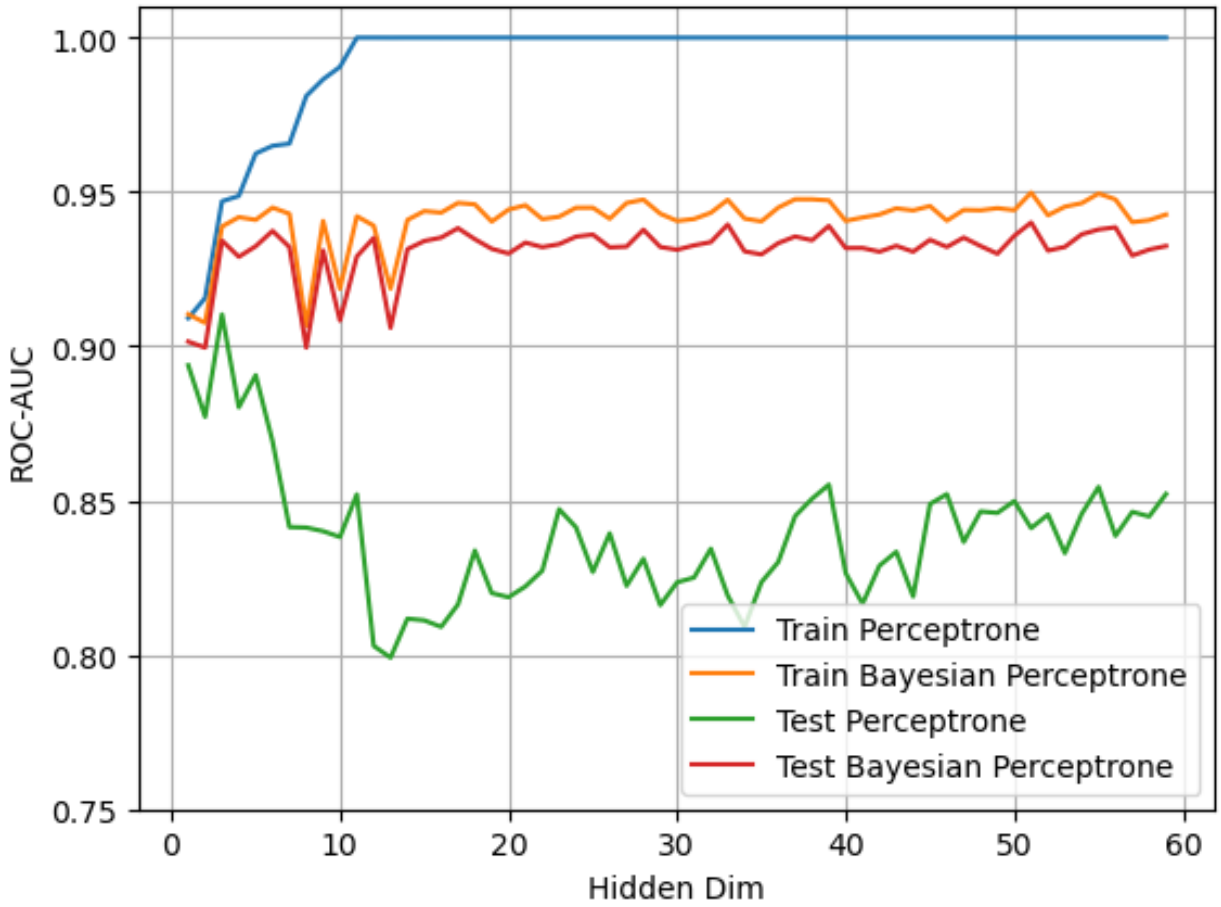

Рис. 1: Зависимость ROC–AUC от размерности скрытого состояния на тренировочных и тестовых данных

- 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`.