

Эмпирические байесовские нейронные сети

Басов Дмитрий Константинович

Аннотация

Данная статья посвящена применению техники эмпирического Байеса к байесовским нейронным сетям. Концептуально идея следующая:

1. Мы используем диагональное нормальное распределение для аппроксимации апостериорного распределения весов модели — $q(W)$.
2. Априорное распределение весов модели так же задаётся диагональным распределением с нулевым матожиданием — $p(W)$.
3. Используя вариационный вывод, мы приходим к ситуации, когда ELBO зависит от $KL(q(W) \parallel p(W))$. Так как оба распределения являются нормальными, то KL дивергенция считается аналитически.
4. Мотивация следующего этапа была взята из RVM — взять дисперсию априорного распределения весов модели $p(W)$ из данных. Там несложно берётся производная и всё получается красиво, кроме возможного деления 0/0. Но сделав замену переменных, от этой беды можно уйти.

Пункты 1–3 в принципе были описаны в статье [Weight Uncertainty in Neural Networks](#). А вот четвёртый пункт я ни в книгах, ни в статьях не находил.

1 Обозначения и сокращения

$N(\mu, \sigma^2)$ — нормальное распределение

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

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

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

\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

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

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

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

$p(D)$ — маргинальная вероятность датасета (evidence)

$p(\mathbf{W}, D) = p(D|\mathbf{W}) \cdot p(\mathbf{W}) = p(\mathbf{W}|D) \cdot p(D)$ — совместная вероятность весов модели и данных

$q(\mathbf{W}|\boldsymbol{\theta})$ — аппроксимация апостериорного распределения весов модели

$\boldsymbol{\theta}$ — 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.

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

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

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

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.

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

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

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

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

Байесовские модели машинного обучения можно рассматривать как ансамбль из бесконечного числа моделей, веса которых сэмплируются 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. Вместо распределения $p(\mathbf{W}|D)$ используется его аппроксимация $q(\mathbf{W}|\boldsymbol{\theta})$. Применяя вариационный вывод, задача сводится к максимизации нижней вариационной границы \mathcal{L} по параметрам $\boldsymbol{\theta}$.
2. Распределение $q(\mathbf{W}|\boldsymbol{\theta})$ задаётся в виде нормального распределения с диагональной матрицей ковариации. То есть каждый вес модели определяется двумя числами, которые определяют его математическое ожидание и дисперсию. Таким образом, количество обучаемых параметров относительно классической нейронной сети возрастает в 2 раза.
3. Априорное распределение весов модели $p(\mathbf{W})$ задаётся в виде нормального распределения с нулевым матожиданием и диагональной матрицей ковариации. То есть мы вносим следующее априорное знание: веса модели находятся около нуля.
4. Для определения дисперсии априорного распределения весов модели $p(\mathbf{W})$ применяется техника эмпирического Байеса. То есть значение дисперсии $p(\mathbf{W})$ берётся из датасета D . Так как распределения $q(\mathbf{W}|\boldsymbol{\theta})$ и $p(\mathbf{W})$ являются нормальными, оптимальное значение дисперсии распределения $p(\mathbf{W})$ определяется аналитически. Такой подход обладает большой универсальностью, однако из-за этого теряется теоретическая устойчивость к переобучению.
5. Так как распределение $q(\mathbf{W}|\boldsymbol{\theta})$ является нормальным, применяя трюк с репараметризацией, становится возможным использовать градиентные методы для максимизации \mathcal{L} .
6. Вводятся новые параметры $\boldsymbol{\gamma}$ и $\boldsymbol{\rho}$, через которые выражаются матожидание и дисперсия распределения $q(\mathbf{W}|\boldsymbol{\theta})$. Это нужно, чтобы избежать неопределённости деления $\frac{0}{0}$, которая может возникнуть из-за определения дисперсии распределения $p(\mathbf{W})$ из данных.

Эксперименты показали, что предлагаемый подход сохраняет гибкость классических нейронных сетей и делает их устойчивыми к переобучению.

3 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:

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

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 $p(\mathbf{a}) = \int p(\mathbf{a}, \mathbf{b}) d\mathbf{b}$
- $p(\mathbf{y}, \mathbf{W}|\mathbf{x}, D) = p(\mathbf{y}|\mathbf{W}, \mathbf{x}, D) \cdot p(\mathbf{W}|\mathbf{x}, D)$, так как для любых случайных величин \mathbf{a} и \mathbf{b} справедливо $p(\mathbf{a}, \mathbf{b}) = p(\mathbf{a}|\mathbf{b}) \cdot p(\mathbf{b})$
- $p(\mathbf{y}|\mathbf{W}, \mathbf{x}, D) = p(\mathbf{y}|\mathbf{W}, \mathbf{x})$, 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:

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

Для аппроксимации распределения ответов модели можно воспользоваться методом Монте-Карло. Идея следующая: сэмплируем конечное количество весов $\hat{\mathbf{W}}_1, \dots, \hat{\mathbf{W}}_T$ из распределения $p(\mathbf{W}|D)$ и аппроксимируем распределение $p(\mathbf{y}|\mathbf{x}, D)$ следующим образом:

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

Однако для этого нужно иметь возможность сэмплировать из распределения $p(\mathbf{W}|D)$. Получить аналитическое решение интеграла $\int p(D|\mathbf{W}) \cdot p(\mathbf{W}) d\mathbf{W}$ можно только в очень ограниченном числе случаев.

Существует возможность сэмплировать из $p(\mathbf{W}|D)$, используя методы Монте-Карло для марковских цепей (MCMC). Однако для больших датасетов и большого числа весов это практически невозможно.

Альтернативным подходом к решению такой задачи является вариационный вывод — аппроксимация распределения $p(\mathbf{W}|D)$ распределением $q(\mathbf{W}|\boldsymbol{\theta})$, 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 $q(\mathbf{W}|\boldsymbol{\theta})$ записывается следующим образом:

$$\mathcal{L} = \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln \frac{p(\mathbf{W}, D)}{q(\mathbf{W}|\boldsymbol{\theta})} d\mathbf{W}$$

Покажем мотивацию максимизации ELBO. Запишем выражение для $KL(q(\mathbf{W}|\boldsymbol{\theta}) || p(\mathbf{W}|D))$ и преобразуем его, используя тождество $p(\mathbf{W}, D) = p(\mathbf{W}|D) \cdot p(D)$:

$$KL(q(\mathbf{W}|\boldsymbol{\theta}) || p(\mathbf{W}|D)) = \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln \frac{q(\mathbf{W}|\boldsymbol{\theta})}{p(\mathbf{W}|D)} d\mathbf{W} = \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln \frac{p(D) \cdot q(\mathbf{W}|\boldsymbol{\theta})}{p(\mathbf{W}, D)} d\mathbf{W} =$$

$$\ln p(D) \cdot \int q(\mathbf{W}|\boldsymbol{\theta}) d\mathbf{W} - \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln \frac{p(\mathbf{W}, D)}{q(\mathbf{W}|\boldsymbol{\theta})} d\mathbf{W} = \ln p(D) - \mathcal{L}$$

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

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

$$\begin{aligned} \mathcal{L} &= \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln \frac{p(\mathbf{W}, D)}{q(\mathbf{W}|\boldsymbol{\theta})} d\mathbf{W} = \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln \frac{p(D|\mathbf{W}) \cdot p(\mathbf{W})}{q(\mathbf{W}|\boldsymbol{\theta})} d\mathbf{W} = \\ &= \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln p(D|\mathbf{W}) d\mathbf{W} - \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln \frac{q(\mathbf{W}|\boldsymbol{\theta})}{p(\mathbf{W})} d\mathbf{W} = \\ &= \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln p(D|\mathbf{W}) d\mathbf{W} - KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W})) = \\ &= \int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \sum_{i=1}^L \ln p(\mathbf{y}_i | \mathbf{x}_i, \mathbf{W}) d\mathbf{W} - KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W})) \end{aligned}$$

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} :

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, D) &\approx \int p(\mathbf{y}|\mathbf{W}, \mathbf{x}) \cdot q(\mathbf{W}|\boldsymbol{\theta}) d\mathbf{W} \approx \frac{1}{T} \sum_{t=1}^T p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{W}}_t) \\ \mathcal{L} &\approx \frac{1}{S} \sum_{j=1}^S \sum_{i=1}^L \ln p(\mathbf{y}_i | \mathbf{x}_i, \hat{\mathbf{W}}_{ij}) - KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W})) \end{aligned}$$

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

5 Задание функциональных форм распределений

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

$$\begin{aligned} q(\mathbf{W}|\boldsymbol{\theta}) &= N(\mathbf{W} | \boldsymbol{\mu}, \text{diag}(\sigma_{q(\mathbf{W})}^2)) \\ p(\mathbf{W}) &= N(\mathbf{W} | \mathbf{0}, \text{diag}(\sigma_{p(\mathbf{W})}^2)) \end{aligned}$$

Так как распределение $q(\mathbf{W}|\boldsymbol{\theta})$ нормальное, мы можем использовать трюк с репараметризацией при сэмплировании весов, что позволяет использовать градиентные методы для оптимизации: $\hat{\mathbf{W}}_{ij} = \boldsymbol{\epsilon}_{ij} \odot \sigma_{q(\mathbf{W})} + \boldsymbol{\mu}$, где $\boldsymbol{\epsilon}_{ij}$ — сэмпл из $N(\mathbf{0}, \mathbf{I})$.

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

$$KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^M \left(\frac{\sigma_{q(\mathbf{W})_k}^2}{\sigma_{p(\mathbf{W})_k}^2} + \frac{\mu_k^2}{\sigma_{p(\mathbf{W})_k}^2} - \ln \frac{\sigma_{q(\mathbf{W})_k}^2}{\sigma_{p(\mathbf{W})_k}^2} - 1 \right)$$

Априорное распределение весов модели $p(\mathbf{W})$ имеет нулевое математическое ожидание (из соображений симметрии), и среднеквадратическое отклонение $\sigma_{p(\mathbf{W})}$. В классическом байесовском выводе параметр $\sigma_{p(\mathbf{W})}$ должен задаваться до начала обучения, то есть являться гиперпараметром. Однако мы можем воспользоваться техникой эмпирического Байеса, то есть определить параметр априорного распределения $\sigma_{p(\mathbf{W})}$ из данных.

6 Эмпирический Байес

Пусть $\alpha = \text{diag}(\sigma_{\mathbf{p}(\mathbf{W})})^{-2}$. Тогда выражение для $KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W}))$ будет иметь следующий вид:

$$KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^M (\alpha_k \cdot \sigma_{q(W)_k}^2 + \alpha_k \cdot \mu_k^2 - (\ln \sigma_{q(W)_k}^2 + \ln \alpha_k) - 1)$$

Так как в выражении \mathcal{L} интеграл $\int q(\mathbf{W}|\boldsymbol{\theta}) \cdot \ln p(D|\mathbf{W}) d\mathbf{W}$ не зависит от параметров распределения $p(\mathbf{W})$, то:

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = -\frac{\partial (KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W})))}{\partial \alpha_k} = -\frac{1}{2} (\sigma_{q(W)_k}^2 + \mu_k^2 - \frac{1}{\alpha_k}) = -\frac{1}{2} (\sigma_{q(W)_k}^2 + \mu_k^2 - \sigma_{p(W)_k}^2)$$

Приравняв производную к нулю, получим:

$$\begin{aligned} -\frac{1}{2} (\sigma_{q(W)_k}^2 + \mu_k^2 - \sigma_{p(W)_k}^2) &= 0 \\ \sigma_{p(W)_k}^2 &= \sigma_{q(W)_k}^2 + \mu_k^2 \end{aligned}$$

Подставив полученное выражение для априорной дисперсии в $KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W}))$, получим:

$$KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^M \ln(1 + \frac{\mu_k^2}{\sigma_{q(W)_k}^2})$$

Таким образом, мы свели задачу к максимизации \mathcal{L} по параметрам $\boldsymbol{\mu}$ и $\sigma_{\mathbf{q}(\mathbf{W})}$.

7 Замена переменных

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

Так же при градиентной оптимизации компоненты вектора $\sigma_{\mathbf{q}(\mathbf{W})}$ могут попасть в отрицательную область, что нежелательно, так как среднеквадратическое отклонение не может быть отрицательным по определению.

Чтобы избежать этих проблем, сделаем следующую замену переменных:

$$\begin{aligned} \sigma_{\mathbf{q}(\mathbf{W})} &= \ln(1 + e^{\boldsymbol{\rho}}) = \text{Softplus}(\boldsymbol{\rho}) \\ \boldsymbol{\mu} &= \boldsymbol{\gamma} \odot \sigma_{\mathbf{q}(\mathbf{W})} = \boldsymbol{\gamma} \odot \text{Softplus}(\boldsymbol{\rho}) \end{aligned}$$

Тогда:

$$KL(q(\mathbf{W}|\boldsymbol{\theta}) \parallel p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^M \ln(1 + \frac{\mu_k^2}{\sigma_{q(W)_k}^2}) = \frac{1}{2} \sum_{k=1}^M \ln(1 + \gamma_k^2)$$

Таким образом, задача сводится к минимизации следующей функции потерь по параметрам $\boldsymbol{\rho}$ и $\boldsymbol{\gamma}$:

$$\text{loss}(\boldsymbol{\rho}, \boldsymbol{\gamma}) = -\frac{\mathcal{L}}{L} \approx -\frac{1}{S \cdot L} \sum_{j=1}^S \sum_{i=1}^L \ln p(\mathbf{y}_i | \mathbf{x}_i, \hat{\mathbf{W}}_{ij}) - \frac{KL}{L}$$

где:

$$\begin{aligned} \hat{\mathbf{W}}_{ij} &= \boldsymbol{\epsilon} \odot \boldsymbol{\sigma} + \boldsymbol{\rho} \\ \boldsymbol{\epsilon} &\sim N(\mathbf{0}, \mathbf{I}) \\ \boldsymbol{\sigma} &= \text{Softplus}(\boldsymbol{\rho}) \\ \boldsymbol{\mu} &= \boldsymbol{\gamma} \odot \boldsymbol{\sigma} \\ KL &= \frac{1}{2} \sum_{k=1}^M \ln(1 + \gamma_k^2) \end{aligned}$$

8 Алгоритм обучения

Задаем шаг градиентного спуска α и инициализируем параметры распределения $q(\mathbf{W}|\boldsymbol{\theta})$ — $\boldsymbol{\rho}$ и $\boldsymbol{\gamma}$. Затем повторяем, пока не достигнем критерия остановки:

1. $\boldsymbol{\sigma} \leftarrow \text{Softplus}(\boldsymbol{\rho})$ — расчёт среднеквадратических отклонений весов
2. $\boldsymbol{\mu} \leftarrow \boldsymbol{\gamma} \odot \boldsymbol{\sigma}$ — расчёт математических ожиданий весов
3. $\boldsymbol{\epsilon} \leftarrow N(0, 1)$ — сэмплирование случайных весов
4. $\hat{\mathbf{W}} \leftarrow \boldsymbol{\epsilon} \odot \boldsymbol{\sigma} + \boldsymbol{\mu}$ — репараметризация
5. $nll \leftarrow -\frac{1}{L} \sum_{i=1}^L \ln p(\mathbf{y}_i | \mathbf{x}_i, \hat{\mathbf{W}})$ — расчёт среднего отрицательного логарифма правдоподобия (возможна аппроксимация по батчам)
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. $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} - \alpha \frac{\partial l}{\partial \boldsymbol{\rho}}$ — обновление $\boldsymbol{\rho}$
9. $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} - \alpha \frac{\partial l}{\partial \boldsymbol{\gamma}}$ — 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:

$$z = \text{ReLU}(\text{matmul}(x, W_1))$$
$$y = \text{Sigmoid}(\text{matmul}(z, W_2))$$

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

10 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>.

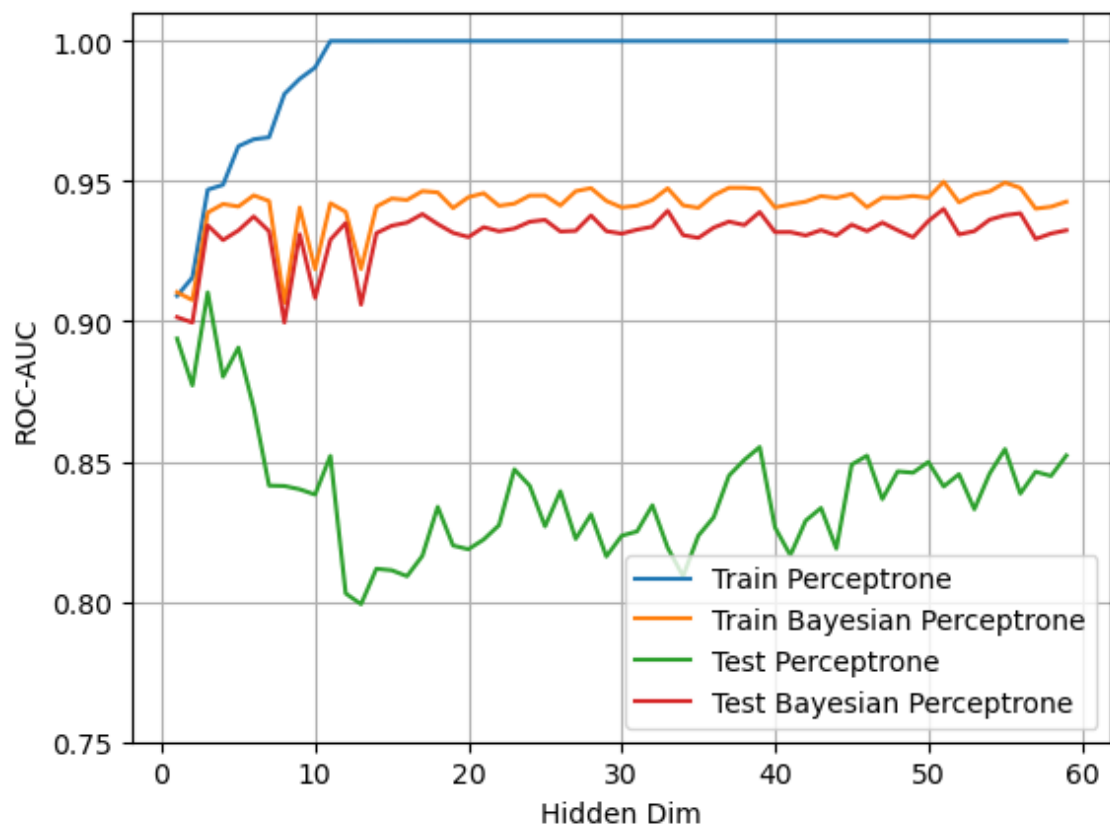


Рис. 1: Зависимость ROC-AUC от размерности скрытого состояния на тренировочных и тестовых данных