Эмпирические байесовские нейронные сети

Басов Дмитрий Константинович

1 Обозначения и сокращения

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

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

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

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

$\mathbf{x}$ — вектор признаков

$\mathbf{y}$ — вектор целевой переменной

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

$\mathbf{W}$ — параметры 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 параметров модели (prior)

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

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

$q(\mathbf{W})$ — аппроксимация апостериорного распределения параметров модели

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

2 Постановка задачи

Постановка задачи следующая: у нас есть датасет D и наша цель — смоделировать распределение $p(\mathbf{y}|\mathbf{x}, D)$. То есть мы хотим получить распределение вероятностей целевой переменной $\mathbf{y}$ для неразмеченных $\mathbf{x}$, используя датасет D. Сделаем следующие преобразования:

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

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: взять сэмпл весов $\hat{\mathbf{W}}$ из $p(\mathbf{W}|D)$, и используя $\hat{\mathbf{W}}$ и $\mathbf{x}$, получить $\hat{\mathbf{y}}$. Однако для этого необходимо уметь сэмплировать из распределения $p(\mathbf{W}|D)$.

Получить аналитическое решение 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 параметров это практически невозможно. Альтернативный подход к решению таких задач — аппроксимация распределения $p(\mathbf{W}|D)$ распределением $q(\mathbf{W})$, из которого сэмплировать намного проще.

3 Вариационный вывод для нейронной сети

Запишем выражение ELBO для распределения $q(\mathbf{W})$ и преобразуем его, используя тождество $p(\mathbf{W}, D) = p(\mathbf{W}|D) \cdot p(D)$:

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

Из равенства $\mathcal{L}(q(\mathbf{W})) = ln(p(D)) - KL(q(\mathbf{W})||p(\mathbf{W}|D))$ видно, что максимизируя $\mathcal{L}(q(\mathbf{W}))$, мы не только максимизируем $\ln p(D)$, но и минимизируем $KL(q(\mathbf{W})||p(\mathbf{W}|D))$. То есть распределение $q(\mathbf{W})$ будет приближаться к распределению $p(\mathbf{W}|D)$.

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

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

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

$p(\mathbf{W}) = N(\mathbf{W}|\mathbf{0}, \boldsymbol{\sigma_{p(\mathbf{W})}}^2 \cdot \mathbf{I})$, где $\boldsymbol{\sigma_{p(\mathbf{W})}}$ — вектор длины M

$q(\mathbf{W}) = N(\mathbf{W}|\boldsymbol{\mu}, \boldsymbol{\sigma_{q(\mathbf{W})}}^2 \cdot \mathbf{I})$, где $\boldsymbol{\mu}$ и $\boldsymbol{\sigma_{q(\mathbf{W})}}$ — вектора длины M

Так как распределения $p(\mathbf{W})$ и $q(\mathbf{W})$ являются нормальными, то $KL(q(\mathbf{W})||p(\mathbf{W}))$ можно посчитать аналитически:

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

Априорное распределение параметров модели определяется параметром $\boldsymbol{\sigma_{p(\mathbf{W})}}$. Воспользуемся техникой эмпирического Байеса — нахождения параметров априорного распределения из данных. Посчитаем $\frac{d\mathcal{L}(q(\mathbf{W}))}{d(\sigma^{-2}_{p(W)_k})}$:

$$\frac{d\mathcal{L}(q(\mathbf{W}))}{d(\sigma^{-2}_{p(W)_k})} = \frac{d(\int q(\mathbf{W}) \cdot \ln p(D|\mathbf{W}) d\mathbf{W} - KL(q(\mathbf{W})||p(\mathbf{W})))}{d(\sigma^{-2}_{p(W)_k})} = -\frac{d(KL(q(\mathbf{W})||p(\mathbf{W})))}{d(\sigma^{-2}_{p(W)_k})}$$

$$\frac{d\mathcal{L}(q(\mathbf{W}))}{d(\sigma^{-2}_{p(W)_k})} = -\frac{1}{2}(\sigma^2_{q(W)_k} + \mu_k^2 - \sigma^2_{p(W)_k})$$

Приравняв производную к нулю, получим:

$$-\frac{1}{2}(\sigma^2_{q(W)_k} + \mu_k^2 - \sigma^2_{p(W)_k}) = 0$$

$$\sigma^2_{p(W)_k} = \sigma^2_{q(W)_k} + \mu_k^2$$

Подставив полученное выражение в $KL(q(\mathbf{W})||p(\mathbf{W}))$, получим:

$$KL(q(\mathbf{W})||p(\mathbf{W})) = \frac{1}{2} \sum_{k=1}^{M} \ln\left(1 + \frac{\mu_k^2}{\sigma^2_{q(W)_k}}\right)$$

Чтобы избежать неопределенности $\frac{0}{0}$, и чтобы $\boldsymbol{\sigma_{q(\mathbf{W})}}$ была всегда положительна, сделаем следующую замену переменных:

$$\sigma_{q(W)_k} = \ln(1 + e^{\rho_k}) = Softplus(\rho_k)$$

$$\mu_k = \gamma_k \cdot Softplus(\rho_k)$$

Тогда:

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

Таким образом, функция потерь будет иметь следующий вид:

$Loss(\boldsymbol{\rho},\boldsymbol{\gamma}) = -\frac{\mathcal{L}(q(\mathbf{W}))}{L} = \int N(\mathbf{W}|\boldsymbol{\mu},{\boldsymbol{\sigma}_{\boldsymbol{q}(\mathbf{W})}}^2\cdot\mathbf{I})\cdot NLL\cdot d\mathbf{W} + \frac{KL}{L}$, где:

$\boldsymbol{\sigma}_{\boldsymbol{q}(\mathbf{W})} = Softplus(\boldsymbol{\rho})$

$\boldsymbol{\mu} = \boldsymbol{\gamma}\cdot Softplus(\boldsymbol{\rho})$

$NLL = -\frac{1}{L}\sum_{i=1}^{L}\ln p(\mathbf{y_i}|\mathbf{x_i},\mathbf{W})$

$KL = \frac{1}{2}\sum_{k=1}^{M}\ln(1+\gamma_k^2)$

4 Алгоритм обучения

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

1. $\boldsymbol{\sigma} \leftarrow Softplus(\boldsymbol{\rho})$
2. $\boldsymbol{\mu} \leftarrow \boldsymbol{\gamma}\cdot\boldsymbol{\sigma}$
3. $\hat{\mathbf{W}} \leftarrow N(0,1)$ — сэмплируем случайные веса
4. $\hat{\mathbf{W}} \leftarrow \hat{\mathbf{W}}\cdot\boldsymbol{\sigma} + \boldsymbol{\mu}$ — репараметризация
5. $nll \leftarrow -\frac{1}{L}\sum_{i=1}^{L}\ln p(\mathbf{y_i}|\mathbf{x_i},\hat{\mathbf{W}})$
6. $kl \leftarrow \frac{1}{2}\sum_{k=1}^{M}\ln(1+\gamma_k^2)$
7. $l \leftarrow nll + \frac{kl}{L}$ — считаем функцию потерь
8. $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} - \alpha\frac{dl}{d\boldsymbol{\rho}}$
9. $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} - \alpha\frac{dl}{d\boldsymbol{\gamma}}$

5 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

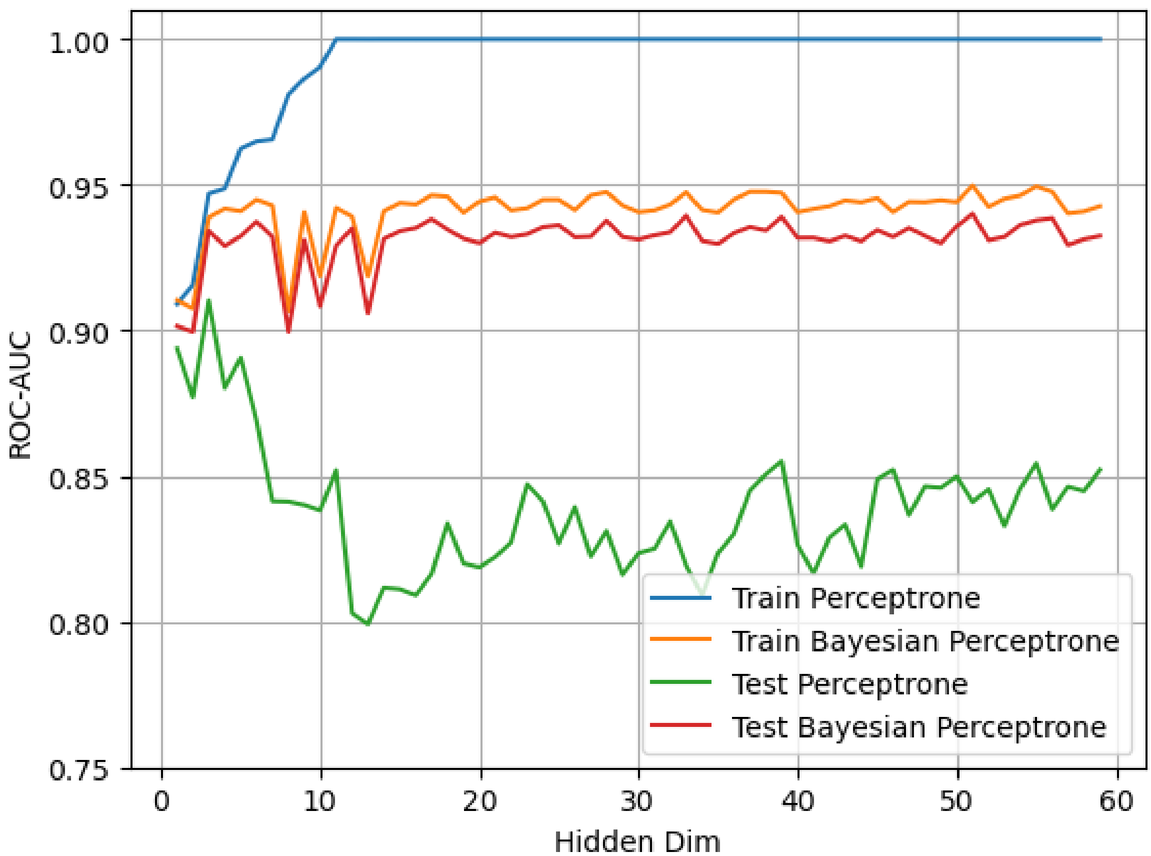

Рис. 1: Зависимость $ROC - AUC$ от размерности скрытого состояния на тренировочных и тестовых данных

6 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.