Научная статья
УДК 519.2
doi: 10.34822/1999-7604-2022-3-46-56

ПРИМЕНЕНИЕ БАЙЕСОВСКИХ МЕТОДОВ ДЛЯ ОПТИМИЗАЦИИ ОБУЧЕНИЯ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ТЕСТИРОВАНИЯ ПРИЛОЖЕНИЙ

Павел Валерьевич Полухин
Воронежский государственный университет, Воронеж, Россия
alfa_force@bk.ru, https://orcid.org/0000-0001-5645-6312

Аннотация. Нейросетевые модели являются универсальным инструментом для решения вероятностных задач в различных сферах и областях деятельности. Однако при решении ряда задач применение классических нейросетевых моделей и алгоритмов обучения может приводить к переобучению сети, вследствие чего происходит снижение точности апостериорного распределения выходного слоя нейросети и наблюдается слабая адаптивность сети к самообучению на основе ранее непредставленных данных в обучающей выборке. Тем самым снижаются возможности сети по выявлению аномалий. Одним из методов решения данной проблемы является использование байесовских методов для обучения и вероятностного вывода в нейросетях. При использовании байесовского подхода в нейросетях подразумевается задание весов сети в виде вероятностного распределения по всем допустимым значениям, которые могут принимать параметры сети. Тогда для определения весов можно использовать классические механизмы обучения и вероятностного вывода, которые используются в байесовских сетях, в частности алгоритмы на основе метода Монте-Карло и цепей Маркова. В работе рассмотрены вопросы применения байесовских нейросетей для моделирования процесса тестирования веб-приложений.

Ключевые слова: метод Монте-Карло, цепь Маркова, метод максимального правдоподобия, максимум апостериорной вероятности, вариационный вывод, байесовская нейронная сеть, многослойный перцептрон, методы тестирования

Для цитирования: *Полухин П. В. Применение байесовских методов для оптимизации обучения нейросетевых моделей тестирования приложений // Вестник кибернетики. 2022. № 3 (47). С. 46–56. DOI 10.34822/1999-7604-2022-3-46-56.*

Original article

USING BAYESIAN METHODS TO OPTIMIZE TRAINING OF NEURAL NETWORK MODELS FOR APPLICATION TESTING

Pavel V. Polukhin
Voronezh State University, Voronezh, Russia
alfa_force@bk.ru, https://orcid.org/0000-0001-5645-6312

Abstract. Neural network models are a universal tool for solving probabilistic problems in various fields and areas. However, when solving a set of problems, typical neural networks and training algorithms can lead to overfitted networks, resulting in a decrease in the accuracy of the posterior distribution of the output layer of the neural network. In this case, the network fails to adapt fully to self-training due to no data in the training sample, leading to a decrease in the network's ability to detect anomalies. One of the methods to solve the problem is by using Bayesian methods for training of and inference in neural networks. When using the Bayesian approach in neural networks, it is intended to set the weights in the network as a probability distribution over all values that are permissible for the parameters of the network. Then, classical learning mechanisms and probabilistic inference used in Bayesian networks, such as algorithms based on the Monte Carlo methods and Markov chains, can be used to determine weights. The article considers the issues of using Bayesian neural networks for modeling the process of web application testing.

Keywords: Monte Carlo method, Markov chain, maximum likelihood method, maximum posterior probability, variation inference, Bayesian neural network, multilayer perceptron, testing methods

For citation: *Polukhin P. V. Using Bayesian Methods to Optimize Training of Neural Network Models for Application Testing // Proceedings in Cybernetics. 2022. No. 3 (47). P. 46–56. DOI 10.34822/1999-7604-2022-3-46-56.*

ВВЕДЕНИЕ

Применение байесовского подхода для решения задач обучения и вероятностного вывода является общепринятым. Наряду с классическими вероятностными графическими моделями, особый интерес представляет возможность применения байесовского подхода к нейронным сетям и формирование байесовской нейронной сети, позволяющей исключить недостатки классических нейросетей общего назначения. Первое полное описание байесовских нейронных сетей приведено в работах Нила [1] и Маккея [2], рассматривающих возможности оптимизации обучения нейронных сетей на основе применения байесовского подхода. Основное преимущество таких сетей заключается в вероятностном определении весов между слоями нейронной сети. Такой подход позволяет учитывать априорное распределение по всем выходным параметрам в процессе обучения сети и вычисления апостериорного распределения в соответствии с генерируемыми предположениями (выборками). Использование различных подходов формирования выборок, основывающихся на методе Монте-Карло с применением цепей Маркова, позволяет сократить объем данных, необходимых для обеспечения требуемого уровня обученности сети, а также исключить вероятность ложного срабатывания. Процедура обучения параметров происходит за счет вероятностного вывода с использованием байесовского вывода. Для оптимизации процедуры обучения байесовской нейронной сети возникает необходимость исследования новых алгоритмов вероятностного вывода. Хинтоном и Ван Кампом предложен один из таких алгоритмов, который основывается на методе вариационного вывода. Сущность данного метода заключается в определении вероятностного распределения для весов из всех слоев, учитывая факт условной независимости весов каждого из слоев. Применение такого подхода позволяет учитывать веса всех слоев в процессе формирования результирующих выборок байесовских нейронных сетей (далее – БНС) и повысить точность процедуры вариационного вывода. На практике при моделировании БНС могут рассматриваться различные последовательные алгоритмы Монте-Карло с применением цепей Маркова, в частности алгоритм фильтрации частиц. Каждая последующая выборка будет формироваться с учетом предшествующей выборки и пропорционально ее весу. Тогда переход между слоями можно будет представить в виде однородного марковского процесса.

МАТЕРИАЛЫ И МЕТОДЫ

Современный подход к обучению БНС основывается на различных алгоритмах, основанных на методе вариационного вывода и генерации выборок. Специфика применимее вариационного вывода (далее – ВВ) предложена Хинтоном для обучения нейронной сети типа «многослойный перцептрон», имеющей неограниченное число скрытых слоев. Типовая нейронная сеть в виде многослойного перцептрона представлена на рис. 1, где видно, что X – входной слой, Y – выходной слой, $H = (H_1, H_2, \ldots, H_n)$ – множество скрытых слоев нейронной сети. Основной подход, используемый при построении нейронной сети, заключается в вычисленных выходных параметрах Y с учетом X и H.

Для установления связи между данными параметрами используется активационная функция.

В качестве примера такой функции рассмотрим гиперболический тангенс:

$$H(X) = \tanh\left(a + \sum_{i=1}^{n} W_i X_i\right), \qquad (1)$$

где W_i – вес связи между входным параметром X_i и одним из параметров скрытого слоя H_{ij};

a – смещение, соответствующее параметрам слоев H.

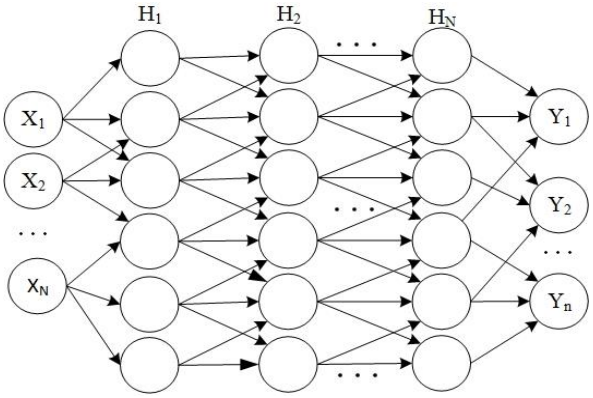

Рис. 1. Пример нейронной сети типа «многослойный перцептрон»
Примечание: составлено по [1].

В соответствии с активационной функцией происходит вычисление значений весов для каждого из параметров скрытого слоя H_{ij} с учетом предыдущего слоя. Модель нейронной сети, представленной на рис. 1, может быть использована для решения задач регрессии и классификации. Обе задачи предусматривают установление связи между входным и выходным набором параметров нейронной сети при условии наличия выборки $D = \left\{\left(X^i, Y^i\right)\right\}_{i=1}^{N}$. Решение задачи регрессии может быть задано в виде поиска условного распределения $P(Y \mid X)$, которое, в случае гауссовского распределения входных параметров X_i, будет иметь следующий вид [3]:

$$P(Y \mid X) = \prod_{k=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{-(f_k(X)-Y_k)^2}{2\sigma_k^2}}, \qquad (2)$$

$$f_k\left(X\right) = b + \sum_{i=1}^{n} W_i H_i\left(X\right), \qquad (3)$$

где b – смещение для всех параметров из H.

Решение задачи классификации для идентификации k-объектов, согласно Нилу, может быть сведено к поиску распределения $P(Y = k \mid X)$. Тогда получим:

$$P\left(Y = k|X\right) = e^{f_k(X)} / \sum_{K} e^{f_K(X)}, K \subseteq Y. \qquad (4)$$

Определение параметров $P(Y \mid X)$ и $P\left(Y = k|X\right)$ реализуется в нейронной сети за счет обучения на основе выборки D. В таком случае для определения весов W можно воспользоваться классическими методами математической статистики: максимального правдоподобия (далее – МП) и максимума апостериорной вероятности (далее – МАВ). Выражение для вычисления весов W^{ML} на основе МП имеет следующий вид:

$$\begin{aligned} W^{ML} &= arg \max_{W} \log P\left(D|W\right) = \\ &= arg \max_{W} \sum_{i=1}^{n} \log P\left(Y_i \mid X_i, W_k\right). \end{aligned} \qquad (5)$$

В таком случае выражение для МАВ W^{MAP} имеет следующий вид:

$$\begin{aligned} W^{MAP} &= arg \max_{W} \log P\left(W|D\right) = \\ &= arg \max_{W} \log P\left(D|W\right) + \log P\left(W\right). \end{aligned} \qquad (6)$$

Для решения уравнений (4) и (5) можно воспользоваться различными численными методами, в частности методом градиентного спуска, а также одним из квазиньютоновских алгоритмов. В общем случае задача сводится к поиску максимального значения $\log\left(f\left(x\right)\right)$. Рассмотрим решение данных уравнений методом Бройдена – Флетчера – Гольдфарба – Шанно (далее – БФГШ). На каждом шаге алгоритма БФГШ используется процедура разложения оценочной функции на полином вида:

$$P\left(Y = k|X\right) = e^{f_k(X)} / \sum_{K} e^{f_K(X)}, K \subseteq Y, \qquad (7)$$

где λ – шаг сдвига алгоритма.

Основная идея алгоритма БФГШ заключается в расчете приближенной матрицы Гессе $B(x_k)$ вместо прямого вычисления $H(x_k)$. Тогда значение параметра λ можно выразить через соответствующую приближенную матрицу $B(x_k)$:

$$\lambda_k = -B^{-1}(x_k)\nabla f(x_k). \quad (8)$$

Далее рассмотрим вычисление приближенной матрицы Гессе. Для этого воспользуемся условием Вольфе:

$$\begin{aligned} f(x_k + \alpha_k\lambda_k) &\le f(x_k) + c_1\alpha_k\nabla f_k^T(x)\lambda_k, \\ \nabla f(x_k + \alpha_k\lambda_k)^T &\ge c_2\nabla f_k^T(x)\lambda_k, \end{aligned} \quad (9)$$

где c_1 и c_2 – константы, соответствующие условиям $0 < c_1 < c_2$.

Если приближение функции $f(x)$ будет находиться в пределах локального экстремума, то будет справедливо строгое условие Вольфе:

$$\begin{aligned} f(x_k + \alpha_k\gamma_k) &\le f(x_k) + c_1\alpha_k\nabla f_k^T(x)\lambda_k, \\ \left|\nabla f(x_k + \alpha_k\lambda_k)^T\right| &\le c_2\left|\nabla f_k^T(x)\lambda_k\right|. \end{aligned} \quad (10)$$

Следовательно, каждый следующий шаг для x_k будет удовлетворять условию (10).

$$x_{k+n} = x_k + \alpha_k\lambda_k. \quad (11)$$

Тогда для определения окончательного алгоритма БФГШ рассмотрим метод Дэвидсона – Флетчера – Пауэлла для расчета приближенной матрицы Гессе $B_{X_{k+n}}$, являющейся частью алгоритма БФГШ:

$$\begin{aligned} B_{x_{k+n}} &= \left(I - \delta_{k+n-1}s_{k+n-1}y_{k+n-1}^T\right)\times M_{x_{k+n-1}} \times \\ &\times\left(I - \delta_{k+n-1}y_{k+n-1}s_{k+n-1}^T\right) + \delta_{k+n-1}s_{k+n-1}s_{k+n-1}^T, \end{aligned} \quad (12)$$

где $\delta_{k+n-1} = 1/\, y_{k+n-1}s_{k+n-1}^T$;

$s_{k+n-1} = x_{k+n} - x_{k+n-1}$;

$y_{k+n-1} = \nabla f(x_{k+n}) - \nabla f(x_{k+n-1})$.

Обозначим обратную матрицу Гессе как $H_{x_{k+n}} = B_{x_{k+n}}^{-1}$. Тогда, воспользовавшись тождеством Шермана – Моррисона – Вудбери, можно получить приближенное значение инверсной матрицы $H_{x_{k+n}}$:

$$\begin{aligned} H_{k+n} &= H_{k+n-1} - H'_{k+n-1} + \frac{y_{k+n-1}y_{k+n-1}^T}{y_{k+n-1}^T s_{k+n-1}}, \\ H'_{k+n-1} &= \frac{H_{k+n-1}s_{k+n-1}s_{k+n-1}^T H_{k+n-1}}{s_{k+n-1}^T H_{k+n-1}s_{k+n-1}}. \end{aligned} \quad (13)$$

Отметим, что обратная матрица Гессе H_{k+n-1} может быть выражена через соответствующую матрицу Якоби. Это позволяет исключить необходимость расчета вторых производных для функции $f(x)$ на каждом шаге:

$$H_{k+n} = J_{k+n-1}^T J_{k+n-1} + H_{k+n-1}. \quad (14)$$

Из выражения (13) получим важное следствие, заключающееся в том, что матрица H_{k+n} будет положительно определенной только в том случае, если H_{k+n-1} также является положительно определенной. Данное следствие – достаточно важное, так как является одним из необходимых условий для определения направленности поиска.

Для определения искомого выражения расчета H_{k+n} на основе алгоритма БФГШ введем ограничения относительно матрицы H_{k+n}: матрица должна быть симметричной и положительно определенной, должна удовлетворять уравнению секущих $M_{k+n}y_{k+n-1} = s_{k+n-1}$. Тогда алгоритм БФГШ, с учетом выражения (13), описывается следующим выражением:

$$\begin{aligned} H_{x_{k+n}} &= \left(I - \delta_{k+n-1}s_{k+n-1}y_{k+n-1}^T\right)\times H_{x_{k+n-1}} \times \\ &\times\left(I - \delta_{k+n-1}y_{k+n-1}s_{k+n-1}^T\right) + \delta_{k+n-1}s_{k+n-1}s_{k+n-1}^T, \end{aligned} \quad (15)$$

где $\delta_{k+n-1} = 1/\, y_{k+n-1}s_{k+n-1}^T$.

БФГШ является достаточно эффективным алгоритмом для решения задач поиска экстремума функции $f(x)$. Однако на практике применение для методов максимального правдоподобия и максимума апостериорной вероятности при обучении нейронной сети часто приводит к ее переобучению и снижает в целом эффективность решения задач ре-

грессии и классификации, несмотря на достаточно точное решение задач поиска максимального значения с использованием методов МП и МАВ.

Альтернативным для преодоления проблем методов МП и МАВ является байесовский подход в процессе обучения нейронной сети. Нейронные сети с вероятностным распределением весов получили название «байесовские нейронные сети». Для данного типа нейронных сетей весам соответствует распределение вероятностей $P(W_{n+1}|D)$, формируемое на основе обучающей выборки D. В таком случае апостериорное распределение по всем весам W при наличии свидетельства относительно X и Y задается в виде условного распределения $\mathrm{P}(W \mid X,Y)$. Полное совместно апостериорное распределение для байесовской нейронной сети будет иметь следующий вид:

$$\begin{aligned} &P\left(Y_{n+1}|X_{n+1},\left(X_{1:n+1},Y_{1:n+1}\right)\right)= \\ &=\int P(X,Y,W)dW_{n+1},\, P(X,Y,W)= \\ &=P\left(Y_{n+1}|X_{n+1},W_{n+1}\right)P(W_{n+1}\mid X_{1:n},Y_{1:n}). \end{aligned} \tag{16}$$

Тогда распределение вероятностей $P(W_{n+1} \mid X_{1:n},Y_{1:n})$ может быть найдено в соответствии с формулой Байеса:

$$\begin{aligned} &P\left(W_{n+1}|D\right)= \\ &=\frac{P\left(D_n|W_{n+1}\right)P\left(W_{n+1}\right)}{P\left(D_n\right)}=\prod_{i=1}^{n}P\left(Y_i\mid X_i,W_k\right). \end{aligned} \tag{17}$$

Тогда в соответствии с выражением (15) процедура нормализации распределения $P\left(Y_{n+1}|X_{n+1}\right)$ имеет следующий вид:

$$\begin{aligned} &P\left(Y_{n+1}|X_{n+1}\right)= \\ &=\int P\left(Y_{n+1}|X_{n+1},W_{n+1}\right)P\left(W_{n+1}\right)dW_{n+1}. \end{aligned} \tag{18}$$

В таком случае оценку параметров выходного слоя БНС, соответствующего шагу, можно определить согласно следующему интегралу:

$$\hat{Y}_{n+1}=\int f\left(x,W\right)P\left(W,D\right)dW, \tag{19}$$

где $f\left(x,W\right)$ – функция выходов Y_{n+1}, имеющая непосредственную зависимость от весов W.

Для нахождения $P(W \mid X,Y)$ воспользуемся методом вариационного вывода (далее – ВВ), имеющим сходство с методом МП, однако вместо логарифма используется дистанция Кульбака – Лейблера (далее – КЛ). Для формулировки метода ВВ требуется задать приближенное вариационное распределение по всем весам $Q\left(W\right)$. Степень близости распределения $Q\left(W\right)$ к $P(W \mid X,Y)$ будет обеспечиваться за счет расчета дистанции КЛ. Определим выражение дистанции КЛ в следующем виде [4]:

$$\begin{aligned} &D_{KL}\left(Q(W),P(W\mid X,Y)\right)= \\ &=\int Q(W)\log\frac{Q(W)}{P(W\mid X,Y)}dW= \\ &=\int Q(W)logQ(W)dW-\int Q(W)P(W|X,Y)dW= \\ &=E_W\left(log\frac{Q(W)}{P(W)}\right)-E_W\left(P(W|X,Y)\right)+logP(W). \end{aligned} \tag{20}$$

С учетом того, что дистанция D_{KL} является строго неотрицательной, можно получить следующее неравенство:

$$\begin{aligned} &D_{KL}\left(Q(W),P(W|X,Y)\right)- \\ &-E_W\left(P(W|X,Y)\right)+logP(W)\geq 0, \\ &logP(W)\geq E_W\left(log\frac{Q(W\mid X,Y)}{P(W)}\right). \end{aligned} \tag{21}$$

Введем следующие обозначения для вариационно-независимой функции:

$$\begin{aligned} F\left(D,Q\right)=D_{KL}\left(Q(W),P(W\mid X,Y)\right)- \\ -E_W\left(logP(D\mid W)\right). \end{aligned} \tag{22}$$

Тогда с учетом дистанции КЛ и функции (19) можно получить оценку весов $\hat{W}$ на основе вариационного распределения $Q(W)$:

$$\hat{W}=arg\min_W F\left(D,Q\right). \tag{23}$$

В таком случае оценка $\hat{W}$ для $F\left(D,Q\right)$ будет являться оценкой максимального правдоподобия, ее расчет можно выполнить

по аналогии с классической нейронной сетью. Для этого достаточно воспользоваться описанным ранее квазиньютоновским алгоритмом БФГШ и оценочной функцией $F(D,Q)$.

Алгоритм ВВ является точным алгоритмом байесовского вывода, используемого для обучения БНС. Основным его недостатком является повышение временной сложности алгоритма в случае увеличения слоев нейронной сети, в частности общего числа скрытых слоев. В связи с этим наибольший интерес представляют стохастические алгоритмы на основе байесовского вывода и на основе метода Монте-Карло. В основе метода Монте-Карло используется порождение случайных значений переменных БНС в соответствии с заданной функцией распределения. Особый интерес представляет возможность формирования выборок для весов W в соответствии с моделью перехода между скрытыми слоями БНС. Обозначим $Q(W)$ как апостериорное распределение вероятности (распределение по значимости) в момент перехода между слоями БНС. Тогда математическую формулировку метода Монте-Карло запишем в следующем виде:

$$Ef(x)=\int_0^1 f(x,W)Q(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i,W_i). \quad (24)$$

Так, из (23) следует, что W_i формируется из распределения по значимости $Q(W)$. На практике особую значимость представляют методы, основанные на применении метода Монте-Карло совместно с цепью Маркова (далее – МКМЦ). Такой подход обладает одним важным свойством по сравнению с методом Монте-Карло – позволяет накапливать выборки, то есть каждая следующая генерация выполняется путем случайного изменения уже существующей. Серия таких случайных событий образует цепь Маркова в соответствии со стационарным распределением $Q(W)$. Тогда переходная вероятность будет задаваться в виде распределения $P(W_{i+1}|W_i)$. Алгоритм вычисления апостериорного распределения на основе метода МКМЦ выполняется путем развертывания цепи Маркова на n шагов, характеризующихся состояниями W_{i+1} и W_i с соответствующими вероятностями $P(W_{i+1})$ и $P(W_i)$. Определим цепь Маркова для случайного распределения $P(W_{i+1}|W_i)$, которая будет иметь следующий вид [5]:

$$\begin{aligned} P(\xi(t_{i+1})=W_{i+1}|\xi(t_1)=W_1,W_1,\ldots,\xi(t_i)=W_i)= \\ =P(\xi(t_{i+1})=W_{n+1}|\xi(t_i)=W_i), \end{aligned} \quad (25)$$

где распределение $Q(W)$ будет сходиться к исходному распределению $P(W)$ в силу закона больших чисел при $N\to\infty$. В таком случае точность алгоритма будет зависеть от N. Условие стационарности для распределения $P(W)$ будет иметь следующий вид:

$$P(W_{i+1})=\int P(W_{i+1}|W_i)P(W_i)dW_{i+1}\forall W_{i+1}\in W. \quad (26)$$

Достаточное условие стационарности для распределения $P(W_i)$ будет справедливо только в том случае, если выполняется принцип детального равновесия [6].

$$P(W_{i+1})=\int P(W_{i+1}|W_i)P(W_i)dW_{i+1}\forall W_{i+1}\in W. \quad (27)$$

На методе МКМЦ основаны алгоритмы Метрополиса – Гастингса (далее – МГ), Гиббса. Наряду с этими общераспространенными, рассмотрим гибридные алгоритмы на основе динамики Ланжевена и Гамильтона. Приведем алгоритм МГ, который также используется в гибридных алгоритмах в качестве условия принятия или отклонения выборки. Запишем условие, соответствующее оценке принятия (отбрасывания) выборки на основе алгоритма МГ [7]:

$$\begin{aligned} &E_\varphi \min\{1,\varphi\}=\int dW_i dW_{i+1}\min(1,\Omega), \\ &\Omega=\frac{P(W_{i+1})P(W_i|W_{i+1})}{P(W_i)P(W_{i+1}|W_i)}, W_i\neq W_{i+1}, \end{aligned} \quad (28)$$

$$\begin{aligned} &\varphi(W_{i+1}|W_i)=\frac{P(W_{i+1})P(W_i|W_{i+1})}{P(W_i)P(W_{i+1}|W_i)}, \\ &W_i\sim P(W_i), W_{i+1}\sim P(W_{i+1}|W_i). \end{aligned} \quad (29)$$

Тогда в соответствии с (26) докажем условие стационарности $P(\mathrm{W}_i, \mathrm{W}_{i+1})$:

$$\begin{aligned}\min\begin{pmatrix}P(W_i)P(W_{i+1}|W_i),\\ P(W_{i+1})P(W_{i+1}|W_i)\end{pmatrix}\varphi(W_{i+1}|W_i)=\\ =\min\begin{pmatrix}P(W_{i+1})P(W_{i+1}|W_i),\\ P(W_i)P(W_{i+1}|W_i)\end{pmatrix}=\\ =P(W_{i+1})P(X_{i+1},W_i)\varphi(W_{i+1}|W_i).\end{aligned} \quad (30)$$

Тогда, с учетом того, что распределение $P(W_i)$ является стационарным, выражение (28) перепишем в следующем виде [8, 9]:

$$\varphi(W_{i+1}|W_i)=\frac{P_0(W_{i+1})\prod_{i=1}^{n}P(X_i,Y_i|W_{i+1})P(W_i|W_{i+1})}{P_0(W_i)\prod_{i=1}^{n}P(X_i,Y_i|W_i)P(W_{i+1}|W_i)}. \quad (31)$$

В соответствии с выражением (30) определим условие принятия (отбрасывания) выборки на каждом шаге алгоритма МГ с учетом равномерного распределения $\mathrm{U}(0,1)$:

$$W_i=\begin{cases}W_{i+1}, \varphi(W_{i+1}|W_i)>U(0,1)\\ W_i, \varphi(W_{i+1}|W_i)\le U(0,1)\end{cases}. \quad (32)$$

Из выражения (31) видно, что при формировании условия с учетом распределения $\mathrm{U}(0,1)$ на практике будем наблюдать достаточно большой разброс выборок. Как следствие, для достижения требуемого уровня точности требуется генерация достаточно большого числа выборок при $N\to\infty$. Для решения данной проблемы в работах Барденета [10] были сделаны попытки уменьшения области разброса выборок за счет использования оценки правдоподобия с нормализацией. Запишем оценку правдоподобия в соответствии с выражением (30):

$$L(W_{i+1},W_i)=\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{P(X_i,Y_i|W_{i+1})}{P(X_i,Y_i|W_i)}\right). \quad (33)$$

Тогда вместо распределения $U(0,1)$ в качестве критерия правдоподобия соответствия выборки будем использовать функцию распределения $\xi(U,W_{i+1},W_i)$:

$$\xi(W_{i+1},W_i)=\frac{1}{N}log\left(U(0,1)\frac{P(W_i)P(W_{i+1}|W_i)}{P(W_{i+1})P(W_i|W_{i+1})}\right). \quad (34)$$

Используя выражения (32) и (33), перепишем условие (31) в следующем виде:

$$W_i=\begin{cases}W_{i+1}, L(W_{i+1},W_i)>\xi(W_{i+1},W_i)\\ W_i, \varphi(W_{i+1}|W_i)\le\xi(W_{i+1},W_i)\end{cases}. \quad (35)$$

Использование критерия на основе оценки правдоподобия позволяет сузить область разброса выборок, однако окончательно решить задачу генерации согласованных выборок не может. В связи с этим особую значимость приобретают алгоритмы на основе динамики Ланжевена, довольно часто используемой для описания различных физических процессов. Тогда запишем классическое уравнение динамики Ланжевена для Броуновского движения в дифференциальном виде [11]:

$$\begin{aligned}dP(W_i)=\frac{1}{2}\nabla L\left(P(W_i)\right)dW_i+d\xi(W_i),\\ i\in[0,\infty),\end{aligned} \quad (36)$$

где $\xi(W_i)$ – производная случайного Броуновского процесса, $L\left(P(W_i)\right)=\log P(W_i)$.

Тогда, решая уравнение (35) методом Эйлера, получим:

$$\begin{aligned}P(W_{i+1})=P(W_i)+\frac{1}{2}\in\nabla L\left(P(W_i)\right)+\\ +\sqrt{\in}\pi_i, \pi_i\sim N(\mu, I),\end{aligned} \quad (37)$$

где I – единичная ковариационная матрица, соответствующая многомерному нормальному распределению $N(0,\mathrm{I})$.

В таком случае, подставляя выражения динамики Ланжевена, соответствующие распределениям $P(W_{i+1})$ и $P(W_i)$, в выражение (34), можно получить обновленное условие получения согласованной выборки по методу Метрополиса – Гастингса. В свою очередь распределение $P_n(W_{i+1}|W_i)$ для всех переходов N запишем в виде следующего произведения [12]:

$$P_N\left(W_{i+1}|W_i\right)=\prod_{i=1}^{N}P(W_{i+1}\,|\,W_i). \qquad (38)$$

Далее рассмотрим применение БНС, а также описанных выше алгоритмов обучения для моделирования процессов тестирования веб-приложений. Для этого исследуем общую схему тестирования веб-приложений на основе БНС. Процедура тестирования на этапе обучения реализуется методом «черного ящика». В данном случае структура веб-приложения неизвестна, а каждый последующий тест генерируется на основе случайного изменения предыдущего. Опишем основные особенности тестирования ошибок межсайтового скриптинга (далее – МСС). МСС представляет собой ошибку в веб-приложении, связанную с некорректной фильтрацией входных данных. В таком случае в веб-приложении могут передаваться конструкции языка JavaScript, которые затем передаются в веб-браузер клиента с целью его компрометации. Среди МСС-ошибок выделяют три основных типа: отраженная, хранимая и dom. Рассмотрим типовую схему тестирования межсайтового скриптинга в процессе анализа веб-приложений на предмет наличия ошибок на основе сети БНС.

Из рис. 2 видно, что БНС в процессе обучения производит анализ успешных МСС-тестов, показавших факт присутствия МСС, а также возможность выполнения эксплуатации данных ошибок.

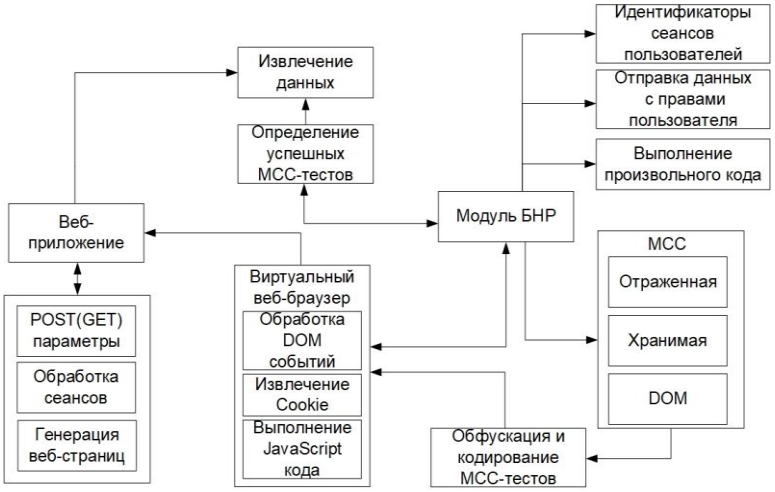

Рис. 2. Модель тестирования МСС на основе БНС
Примечание: составлено автором на основании данных, полученных в исследовании.

В режиме обучения БНС производится случайная генерация тестовых данных с учетом особенностей анализа МСС-ошибок, включая синтаксические правила языка JavaScript. Тестовые генерации подаются на вход группы разнородных приложений $A=(A^1, A^2, ..., A^n)$, после чего анализируются выходные данные приложения. Под выходными данными подразумевается факт присутствия МСС, а также выполнение полезной нагрузки. В этом случае входная последовательность тестов, а также результаты тестирования используются для обучения БНС. Для процедуры обучения входной информацией является выборка $\mathrm{D}=\left\{\left(X^i,Y^i\right)\right\}_{i=1}^{N}$, $X=(X^1,X^2,\ldots,X^N)$ – множество тестовых генераций, $Y=(Y^1,$ $X^2,...,Y^N)$ – информация относительно генераций, успешно прошедших тестирование. Из этого следует, что БНС накапливает информацию исключительно относительно успешно завершившихся тестовых генераций. Для более точного обучения БНС для тестирования МСС-ошибок производится настройка блоков генерации тестов с учетом различных платформ веб-браузеров: Gecko, Chromium.WebKit. Такой подход позволяет производить обучение БНС с учетом особен-

ностей обработки параметров каждой из таких платформ, позволяя выполнять процедуру более осмысленно и точно, по аналогии с тем, как это делает тестировщик в процессе анализа веб-приложения. В результате разработанная модель БНС может работать как в режиме самообучения, так и обучения с учителем. Такой подход дает возможность непрерывной коррекции тестовых генераций и оптимизации процедуры обучения модели тестирования, а также адаптации сети к возникновению аномальных ошибок в процессе реализации процедуры тестирования.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Таким образом, в результате исследования разработаны гибридные алгоритмы обучения БНС на основе динамики Ланжевена, рассмотрены возможные пути их адаптации для решения задач тестирования прикладного программного обеспечения, представленного в виде веб-приложений. Для построения среды тестирования модели, приведенной на рис. 1, а также для реализации процедур обучения БНС, была развернута группа веб-приложений (1С Битрикс 16.5, 16.5.4, WordPress 4.0, 4.1, Joomla 1.5.15, 2.5), включающих в себя средства хранения, обработки поступающей информации, а также имитирующих работу реальных веб-приложений. В качестве платформы для запуска виртуальных веб-браузеров используется Selenium GRID, представляющий собой распределенную платформу, предназначенную для решения задачи оптимизации тестирования веб-приложений динамической обработкой и отображением данных. Отличительной особенностью Selenium является возможность программного доступа к DOM-модели документа, а также к адресному пространству переменных языка JavaScript и его функциональным возможностям. В качества среды развертывания Selenium GRID используем среду Docker, запускающую отдельные экземпляры веб-браузеров Firefox, Chromium и Edge. Взаимодействие с Selenium производится на основе вызова удаленных процедур, позволяющих производить поэтапное выполнение тестов для обнаружения ошибок межсайтового скриптинга. В процессе выполнения эксперимента приведем оценки вычислительной эффективности классических алгоритмов обучения БНС на основе вариационного вывода, Метрополиса – Гастингса, а также разработанного алгоритма на основе динамики Ланжевена.

Из сравнительных характеристик алгоритмов табл. 1 видно, что алгоритм, разработанный на основе динамики Ланжевена, обладает наилучшими показателями. Это, в первую очередь, связано с тем, что динамика Ланжевена позволяет ограничить область согласованных выборок, то есть в процессе каждого последующего шага в соответствии с выражением (35) производится постепенное смещение формируемых выборок к области истинных значений. Следовательно, апостериорное распределению вероятностей по всем выборкам $P(W \mid X, Y)$, формируемое по результатам реализации процедуры тестирования, будет обладать наибольшей точностью.

Таблица 1

Сравнение производительности алгоритмов обучения БНС

№	Объем выборки	Алгоритм ВВ	Алгоритм МГ	Алгоритм ДЛ
1	1000	11,114263 сек.	14,742326 сек.	10,761308 сек.
2	100000	42,800446 сек.	56,590105 сек.	40,128704 сек.
3	1000000	114,214165 сек.	147,883367 сек.	92,710642 сек.
4	100000000	1151,117164 сек.	1484,762449 сек.	855,414459 сек.

Примечание: составлено автором на основании данных, полученных в исследовании.

Отличительной особенностью разработанного алгоритма является применение критерия (34), сочетающего в себе критерий согласованности на основе логарифма правдоподобия и подхода, предложенного Метрополисом. Это позволяет использовать распределение $L(W_{i+1}, W_i)$ взамен классического равномерного распределения $U(0,1)$, что повышает степень согласованности формируемых выборок.

ЗАКЛЮЧЕНИЕ

Моделирование процессов тестирования является сложной и многогранной задачей, требующей создания различного инструментария тестирования, средств накопления статистической информации и интеллектуального анализа данных. Байесовские нейронные сети являются универсальным механизмом моделирования сложных процессов, протекающих в условиях определенности, расширяющим классические модели нейронных сетей за счет использования байесовских методов. Особый научно-практический интерес представляют расширение и оптимизация алгоритмов на основе метода Монте-Карло с применением цепей Маркова. Рассмотренный подход на основе использования динамики Ланжевена в сочетании с логарифмическим правдоподобием позволяют использовать преимущества обоих подходов для повышения степени согласования выборок, формируемых в соответствии с распределениями $P(W_i \mid W_{i+1})$. Предложенный алгоритм обладает хорошей масштабируемостью. Кроме того, разработан параллельный алгоритм, позволяющий оптимизировать генерацию выборок и снизить временные затраты на реализацию процедур обучения. Использование БНС в процессе тестирования веб-приложений позволяет оптимизировать процедуру обучения сети за счет повышения точности определения весов тестовых выборок для каждого скрытого слоя, адаптировать модель нейронной сети для обнаружения аномального поведения приложений. Вычислительный эксперимент и полученные результаты подтверждают правильность разработанных алгоритмов для моделирования тестирования веб-приложений.

Список источников

1. Skansi S. Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence. 2018. 191 p.
2. Зуев В. Н., Кемайкин В. Л. Модифицированные алгоритмы обучения нейронных сетей // Программные продукты и системы. 2019. Т. 4, № 2. С. 258–262.
3. Вершков Н. А., Кучуков В. А., Кучукова Н. Н. Теоретический подход к поиску глобального экстремума при обучении нейронных сетей // Тр. ИСП РАН. 2019. Т. 31, № 2. С. 41–52.
4. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. Hoboken : Pearson, 2020. 1023 p.
5. Bunch P., Godsill S. Improved Particle Approximations to the Joint Smoothing Distribution Using Markov Chain Monte Carlo // IEEE Transactions Signal Processing. 2013. Vol. 61, Is. 4. P. 946–953.
6. Mohan K., Pearl J. Graphical Models for Processing Missing Data // Journal of American Statistical Association. 2018. Vol. 116, Is. 534. P. 1023–1037.
7. Азарнова Т. В., Полухин П. В., Аснина Н. Г., Проскурин Д. К. Формирование структуры байесовской сети процесса тестирования надежности информационных систем // Вестн. Воронеж. гос. техн. ун-та. 2017. Т. 13, № 6. 156 с.
8. Moral P. D., Doucet A. Particle Methods: An Introduction with Application // ESAIM. 2014. Vol. 44. P. 1–46.
9. Мельникова И. В., Сметанников Д. И. Исследование уравнений для вероятностных характеристик случайных процессов, заданных стохастическими уравнениями // Тр. ИММ УрО РАН. 2018. Т. 24, № 2. С. 185–193.
10. Bardenet R., Doucet A., Holmes C., Bardenet R. Towards Scaling up Markov Chain Monte Carlo:

References

1. Skansi S. Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence. 2018. 191 p.
2. Zuev V. N., Kemaikin V. L. An Improved Neural Network Training Algorithm // Software & Systems. 2019. Vol. 4, No. 2. P. 258–262. (In Russian).
3. Vershkov N. A., Kuchukov V. A., Kuchukova N. N. The Theoretical Approach to the Search for a Global Extremum in the Training of Neural Networks // Proceedings of the ISP RAS. 2019. Vol. 31, No. 2. P. 41–52. (In Russian).
4. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. Hoboken : Pearson, 2020. 1023 p.
5. Bunch P., Godsill S. Improved Particle Approximations to the Joint Smoothing Distribution Using Markov Chain Monte Carlo // IEEE Transactions Signal Processing. 2013. Vol. 61, Is. 4. P. 946–953.
6. Mohan K., Pearl J. Graphical Models for Processing Missing Data // Journal of American Statistical Association. 2018. Vol. 116, Is. 534. P. 1023–1037.
7. Azarnova T. V., Polukhin P. V., Asnina N. G., Proskurin D. K. Bayesian Network Structure Formation of Information Systems Reliability Testing Process // Bulletin of Voronezh State Technical University. 2017. Vol. 13, No. 6. 156 p. (In Russian).
8. Moral P. D., Doucet A. Particle Methods: An Introduction with Application // ESAIM. 2014. Vol. 44. P. 1–46.
9. Melnikova I. V., Smetannikov D. I. The Study of Equations for Probability Characteristics of Random Processes Described by Stochastic Equations // Tr. IMM UrO RAN. 2018. Vol. 24, No. 2. P. 185–193. (In Russian).
10. Bardenet R., Doucet A., Holmes C., Bardenet R. Towards Scaling up Markov Chain Monte Carlo:

An Adaptive Subsampling Approach // Proceedings of the 31st International Conference on Machine Learning, June 22–24, 2014, Beijing, China. 2014. Vol. 32, Is. 1. P. 405–413.
11. LeCun Y., Bengio G., Hinton G. Deep Learning // Nature. 2015. Vol. 521. P. 436–444.
12. Durmus A., Majewski S., Miasojedow B. Analysis of Langevin Monte Carlo via Convex Optimization // Journal of Machine Learning Research. 2019. Vol. 20. P. 1–46.

Информация об авторе

П. В. Полухин – кандидат технических наук.

An Adaptive Subsampling Approach // Proceedings of the 31st International Conference on Machine Learning, June 22–24, 2014, Beijing, China. 2014. Vol. 32, Is. 1. P. 405–413.
11. LeCun Y., Bengio G., Hinton G. Deep Learning // Nature. 2015. Vol. 521. P. 436–444.
12. Durmus A., Majewski S., Miasojedow B. Analysis of Langevin Monte Carlo via Convex Optimization // Journal of Machine Learning Research. 2019. Vol. 20. P. 1–46.

Information about the author

P. V. Polukhin – Candidate of Sciences (Engineering).