

Использование специальных методов обучения в нейросетевом классификаторе дефектов.

Мартынов Н. С.

Москва, Московский энергетический институт (технический университет), Россия

В нашей стране, богатой природными ресурсами, существует разветвленная сеть нефте- и газопроводов. Однако эксплуатируемые сети трубопроводов в значительной мере изношены, что может приводить к масштабным катастрофам, имеющим значительные экономические и экологические последствия. Для их предотвращения необходимо проводить ремонт трубопроводов. Целесообразной является выборочная замена участков, на которых располагаются дефекты, которые могут стать причиной аварии. Первым шагом в решении этой задачи является поиск дефектов труб и определение их параметров для оценки степени опасности. Широко применяется магнитный метод контроля, используемый в проходных внутритрубных дефектоскопах. При этом встает задача обработки данных, собранных дефектоскопом.

Одним из возможных способов классификации и параметризации дефектов труб нефте- и газопроводов является использование искусственной нейронной сети. При этом на вход сети подается вектор признаков (features) области интереса (region of interest), например амплитуда сигнала и спектральные составляющие, а с выхода снимается признак принадлежности дефекта к заданному классу, например трещина-коррозия, внутренняя-внешняя, и/или значение параметра, например глубина и протяженность.

В работе исследуется возможность и особенности применения нейронной сети типа многослойный персептрон для определения типа дефектов труб нефте- и газопроводов по вектору признаков и определения их параметров. Оценена возможность применения специальных методов ускорения процесса обучения и улучшения способности к обобщению, таких как алгоритмы RPROP и SARPROP.

Для изменения параметров сети в процессе обучения может применяться алгоритм RPROP. Resilient error backpropagation - алгоритм, предложенный в 1992 году (последняя модификация в 1994) [1, 2]. Это прямой адаптивный алгоритм (он адаптирует изменение весов на основе градиента ошибки), не зависящий от величины производной функции активации. Таким образом, в этом алгоритме вычисляется адаптивное изменение для каждого параметра:

$$\Delta_{ij}(t) = \begin{cases} \Delta_{ij}(t-1) * \eta^+, & \text{если } \partial E / \partial w_{ij}(t-1) * \partial E / \partial w_{ij}(t) > 0, \\ \Delta_{ij}(t-1) * \eta^-, & \text{если } \partial E / \partial w_{ij}(t-1) * \partial E / \partial w_{ij}(t) < 0, \\ \Delta_{ij}(t-1), & \text{иначе} \end{cases}$$

где $0 < \eta^- < 1 < \eta^+$.

Алгоритм RPROP можно описать следующим образом:

1) для каждого слоя сети, начиная с первого скрытого

а) для каждого нейрона слоя

i) Если $\partial E / \partial w_{m,n,ni}^t * \partial E / \partial w_{m,n,ni}^{t-1} > 0$

$$\Delta \Delta w_{m,n,ni} = \min(\Delta \Delta_+ * \Delta \Delta w_{m,n,ni}, \Delta \Delta_{\max})$$

$$\Delta w_{m,n,ni} = -\text{SGN}(\partial E / \partial w_{m,n,ni}) * \Delta \Delta w_{m,n,ni}$$

$$w_{m,n,ni} = w_{m,n,ni} + \Delta w_{m,n,ni}$$

$$\partial E / \partial w_{m,n,ni}^{t-1} = \partial E / \partial w_{m,n,ni}^t$$

ii) Если $\partial E / \partial w_{m,n,ni}^t * \partial E / \partial w_{m,n,ni}^{t-1} < 0$

$$\Delta \Delta w_{m,n,ni} = \max(\Delta \Delta_- * \Delta \Delta w_{m,n,ni}, \Delta \Delta_{\min})$$

$$\partial E / \partial w_{m,n,ni}^{t-1} = 0$$

iii) Если $\partial E / \partial w_{m,n,ni}^t * \partial E / \partial w_{m,n,ni}^{t-1} = 0$

$$\Delta w_{m,n,ni} = -\text{SGN}(\partial E / \partial w_{m,n,ni}) * \Delta \Delta w_{m,n,ni}$$

$$w_{m,n,ni} = w_{m,n,ni} + \Delta w_{m,n,ni}$$

$$\partial E / \partial w_{m,n,ni}^{t-1} = \partial E / \partial w_{m,n,ni}^t$$

iv) Аналогично для порога

Таким образом, если алгоритм проскакивает локальный минимум, то изменение параметра не происходит, а шаг изменения уменьшается, если знак производной ошибки не изменяется, то происходит увеличение шага изменения параметра.

$\Delta \Delta_{\max}$ задает максимальное значение изменения параметров.

$\Delta \Delta_{\min}$ задает минимальное значение изменения параметров.

$\Delta \Delta_+$ задает скорость увеличения изменения параметров.

$\Delta \Delta_-$ задает скорость уменьшения изменения параметров.

Авторы алгоритма рекомендуют следующие значения для приведенных выше величин:

$$\Delta \Delta_{\max} = 50$$

$$\Delta \Delta_{\min} = 10^{-6}$$

$$\Delta \Delta_+ = 1,2$$

$$\Delta \Delta_- = 0,5$$

Одним из немногочисленных недостатков алгоритма RPROP является то, что он, как и классический алгоритм обратного распространения ошибки может остановить процесс обучения в ближайшем минимуме целевой функции. Для преодоления этого недостатка разработан алгоритм SARPROP [3, 4].

Simulated Annealing (имитация отжига) RPROP - модификация алгоритма RPROP, предложенная в 1998 году, сочетает добавление в веса нейронов шума, а также их угасание.

Количество добавляемого шума ассоциируют с температурой системы, которая убывает по мере увеличения количества пройденных эпох обучения. Это позволяет на начальных этапах провести более широкий поиск минимума целевой функции, а затем по мере уменьшения температуры уточнить найденное решение.

Добавление шума происходит при выполнении двух условий:

- изменение знака градиента ошибки;
- величина изменения шага обучения меньше величины пропорциональной текущему значению среднеквадратической ошибки (RMS).

Величина добавляемого шума зависит как от значения среднеквадратической ошибки, так и от температуры.

Добавление шума только при изменении знака градиента ошибки необходимо для избежания нарушений в процессе адаптирования скорости изменения весов. При этом добавление шума происходит при малых значениях скорости изменения шага обучения. Это позволяет покидать локальные минимумы, в тоже время не нарушая процесс адаптации.

Второе изменение алгоритма RPROP - угасание весов (weights decay), основано на модификации целевой функции. Степень угасания весов также зависит от температуры, благодаря этому степень угасания уменьшается в процессе обучения. Новый градиент ошибки вычисляется как

$$\partial E / \partial w_{ij}^{\text{SARPROP}} = \partial E / \partial w_{ij} - k_1 * w_{ij} * 2^{-T * \text{epoch}}$$

Угасание весов вводится для того, чтобы сделать их значения по возможности меньшими на начальном этапе обучения. Это позволяет, увеличить эффективность поиска решения до того, как веса нейронов начнут расти (уточнение решения).

Псевдокод алгоритма SARPROP:

ДЛЯ ВСЕХ ВЕСОВ И ПОРОГОВ

ЕСЛИ $\partial E / \partial w_{ij}(t-1) * \partial E / \partial w_{ij}(t) > 0$ ТОГДА

$$\Delta_{ij}(t) = \min(\Delta_{ij}(t-1) * \eta^+, \Delta_{\max})$$

$$\Delta w_{ij}(t) = w_{ij}(t) + \Delta w_{ij}(t)$$

$$\partial E / \partial w_{ij}(t-1) = \partial E / \partial w_{ij}(t)$$

ИНАЧЕ ЕСЛИ $\partial E / \partial w_{ij}(t-1) * \partial E / \partial w_{ij}(t) < 0$ ТОГДА

ЕСЛИ $\Delta_{ij}(t-1) < k_2 * \text{error}^2$ ТОГДА

$$\Delta_{ij}(t) = \Delta_{ij}(t-1) * \eta^- + k_3 * r * \text{error} * 2^{-T * \text{epoch}}$$

ИНАЧЕ

$$\Delta_{ij}(t) = \Delta_{ij}(t-1) * \eta^-$$

$$\Delta_{ij}(t) = \max(\Delta_{ij}(t-1) * \eta^-, \Delta_{\min})$$

$$\partial E / \partial w_{ij}(t-1) = 0$$

ИНАЧЕ

$$\Delta w_{ij}(t) = - \text{sign}(\partial E / \partial w_{ij}(t)) * \Delta_{ij}(t)$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$$

$$\partial E / \partial w_{ij}(t-1) = \partial E / \partial w_{ij}(t)$$

Здесь error - значение среднеквадратической ошибки, T - температура, k1, k2, k3 - константы, r - случайное число из диапазона [0; 1].

По результатам практических исследований установлено, что для большинства практических задач подходят следующие значения параметров алгоритма SARPROP [3]:

- k1 = 0.01

- k2 = 0.1

- k3 = 3

- T выбирается из диапазона [0.01; 0.07].

По результатам тестирования установлено, что оба алгоритма значительно превосходят по скорости схождения классический алгоритм обратного распространения ошибки и, что более важно, нейронная сеть обученная по данным алгоритмам показывает значительно большую точность результатов. Алгоритм SARPROP в среднем показывает несколько более высокую либо аналогичную точность результатов по сравнению с RPROP.

ЛИТЕРАТУРА.

1. Martin Riedmiller, RPROP - A fast adaptive learning algorithm. - Karlsruhe, 1992, 12 с.
2. Martin Riedmiller, RPROP - Description and implementation details. - Karlsruhe, 1994, 16 с.
3. N.K. Treadgold, T.D. Gedeon, THE SARPROP ALGORITHM: A SIMULATED ANNEALING ENHANCEMENT TO RESILIENT BACK PROPAGATION. - Sydney, 1998, 7 с.
4. P. A. Castilo, SA-Prop: Optimization of multilayer perceptron parameters using simulated annealing. - Granada, 1998, 10 с.