Лекция 12

Методы глобальной оптимизации

Как уже было сказано ранее, функционал ошибки $D_r(g)$ в большинстве случаев имеет много локальных минимумов, что ставит под сомнение оптимальность решения найденного градиентным методом. В связи с этим, при стремлении найти более точное решение, необходимо применить некоторую процедуру, при помощи который мы можем найти глобальный минимум или хотя бы приблизиться к нему. Существует несколько подходов к такому глобальному поиску. Однако все они, так или иначе, используют, как один из своих этапов, какой либо градиентный или стохастический метод локального поиска. Общей идеей для таких методов является либо последовательное повторение процесса обучения, либо параллельное обучение какого то числа копий нейронных сетей с отбором «лучших» из них на очередном этапе.

Рассмотрим некоторые из таких методов.

Метод рестартов

Метод рестартов использует один из локальных градиентных или стохастических алгоритмов.

1. Выбираем начальные значения настраиваемых параметров сети в соответствии с выбранным локальным алгоритмом.
2. Нейронная сеть обучается локальным алгоритмом оптимизации до условия останова (таковым может быть любой из рассмотренных ранее критериев останова алгоритма).
3. Если требуемая величина ошибки не достигнута, тогда фиксируем конечное значение настраиваемых параметров и соответствующее значение функционала ошибки.
4. Повторяем пункт 1, выбирая другой вектор начальных параметров и, возможно, другие параметры алгоритма (величину шага).
5. Повторяем пункт 2, добавляя в качестве условия останова близость к достигнутым в предыдущих стартах значениям.
6. Повторяем пункт 3, фиксируя только одну точку из близких в смысле пункта 5 точек.
7. Повторяем пункты 4-6 до достижения требуемой величины ошибки, истечения времени расчётов или реализации заданного числа рестартов.

В этом варианте алгоритма рассматривается последовательная динамика поиска решения. Однако можно модифицировать алгоритм так, что пункты 4-5-6 будут выполняться параллельно для нескольких одновременно обучаемых сетей.

Эволюционные алгоритмы

Эволюционные алгоритмы используют такие понятия как *ген, хромосома, мутация, поколение.* Эти термины пришли в теорию оптимизации из биологии вместе с работами Дж. Холланда, в начале 1970-х годов [3]. В основе оптимизации некоторой системы этими алгоритмами лежит имитация эволюционных процессов происходящих в популяциях живых организмов. Результатом такого эволюционного моделирования может быть определение решения задачи оптимизации. Возможности использования генетических алгоритмов для решения задач оптимизации подробно описаны в [4].

Определим основные понятия. Пусть необходимо подобрать оптимальный набор параметров W некоторой нейронной сети. Этими параметрами могут быть и матрицы весов связей между слоями и матрица связей сети и вектора параметров функции активации. Для упрощения изложения будем считать что W - это матрица весов связей, но результаты справедливы и для более общего случая. Эту матрицу мы можем представить линейным вектором как

$$W = (w_{11}, w_{12}, ..., w_{1n}, w_{21}, w_{22}, ..., w_{2n}, ..., w_{nn}) = (w_1, w_2, ..., w_{n^2=N}). \qquad (12.1)$$

Каждый из элементов этого вектора будем называть *геном*, а сам вектор W - *хромосомой*. В случае если таких хромосом для каждой сети несколько (матрица связей, матрица весов, вектор параметров активационных функций), то все они являются составляющими отдельной *особи*. В нашем случае, единственная хромосома полностью определяет вид нейронной сети, то есть одна хромосома соответствует одной особи. Каждой особи можно сопоставить так называемую *функцию приспособленности (fitness function)*, которая, по сути, является инвертированной целевой функцией этой нейронной сети: $F(W) = -D(W)$. Чем выше функция приспособленности – тем больше шансов выживания этой особи. Некоторое множество $\{W^i\}_{i=1}^M$ таких особей называется *популяцией.* Популяция и есть тот составной объект, над которым в процессе исполнения алгоритма проводятся некоторые операции, приводящие к максимизации функции приспособленности особей.

Опишем основные операторы над особями и популяцией в целом.

Мутация. Мутация определяется как изменение значений хромосомы в соответствии с некоторым случайным законом:

$$M(W) \ \mathbf{a}\ W'. \qquad (12.2)$$

Например, в случае хромосомы состоящей из битовых значений, мутация может выглядеть как замена случайно выбранного гена на противоположное значение. В случае действительных значений генов, может быть выбран более сложный закон изменения.

Скрещивание. Скрещивание определяется как операция над двумя (или более) однотипными хромосомами разных особей, в результате которой появляется новая хромосома по следующему закону:

$$C(W^1, W^2) \ \mathbf{а} \ W^3, \quad W^3 = (w_1^1, w_2^1, \ldots w_k^1, w_{k+1}^2, w_{k+2}^2, \ldots, w_N^2). \tag{12.3}$$

где $1 < k < N$ - некоторое случайно выбранное число. Большие затруднения представляет способ выбора двух скрещиваемых хромосом. Как правило, выбираются наиболее приспособленные хромосомы, хотя в выборе может присутствовать и фактор случайности. Например, если $F(W^i)$ - функция приспособленности i-й особи, то условие выбора этой особи для скрещивания может определяться законом

$$P(i) = \begin{cases} 1, \ rand(0,1) \geq F(W^i) / \sum_{j=1}^{M} F(W^j), \\ 0, \ rand(0,1) < F(W^i) / \sum_{j=1}^{M} F(W^j). \end{cases} \tag{12.4}$$

где $rand(0,1)$ - некоторое случайное число из диапазона [0,1]. Могут применяться и более сложные законы отбора для скрещивания.

Селекция. Селекция – операция над всеми особями популяции, в результате которой появляется новая популяция особей, более приспособленная, чем предыдущая. Например, в простейшем случае, мы можем выбрать из популяции $m < M$ особей с наибольшими значениями функции приспособленности и заменить ими оставшиеся $(M - m)$ особей. В более сложном случае используется вероятностный закон перехода особи в следующее поколение:

$$S(\{W^i\}) \ \mathbf{а} \ \{W^i\}. \tag{12.5}$$

В общем случае могут применяться и другие операторы над особями для улучшения скорости сходимости или увеличения точности полученного результата.

Типичный эволюционный алгоритм выглядит следующим образом.

1. Инициализация популяции особей в соответствии с некоторым законом (например случайным образом).
2. С некоторой вероятностью P_S выполняется оператор селекции.
3. С некоторой вероятностью P_C выполняется оператор скрещивания.
4. С некоторой вероятностью P_M выполняется оператор мутации.
5. Регулирование численности популяции для удовлетворения неравенства $M < M_{\max}$ (удаление самых неприспособленных особей).

6. Если НС с лучшей функцией приспособленности не удовлетворяет критерию останова (например, недостаточно мало значение ошибки этой НС на обучающей выборке), то переход на шаг 2, иначе – конец.

Хорошие результаты обучения приносит объединение алгоритмов глобальной оптимизации с градиентными алгоритмами. Например, на первом этапе обучения применяется выбранный алгоритм глобальной оптимизации, а после достижения целевой функцией определенного уровня включается градиентная оптимизация лучшей из нейросетей в популяции. Другим вариантом применения градиентных методов в глобальном поиске является добавление в алгоритм еще одного оператора – оператора градиентного спуска для некоторой особи.

Метод имитации отжига металла

Еще один способ добавить «глобальность» в тот или иной метод оптимизации – применение *метода отжига* на некотором этапе оптимизации. Сам термин заимствован из статической механики и отражает поведение металлического тела при отвердевании в процессе охлаждения температуры до нуля. В процессе медленного управляемого охлаждения, называемого отжигом, кристаллизация тела сопровождается уменьшением его энергии. Энергия тела может быть интерпретирована как один из параметров в алгоритме оптимизации.

Опишем алгоритм имитации отжига [1].

1. Запустить процесс из начальной точки W при заданной начальной температуре $T = T_{\max}$. $t = 0$.
2. Увеличим номер итерации: $t = t + 1$. Пока $T > T_{\min}$ повторить L раз следующие действия:
   a. выбрать новое решение W' из окрестности $W(t-1)$;
   b. рассчитать изменение целевой функции $\Delta = D(W) - D(W')$;
   c. если $\Delta \le 0$, принять $W(t) = W'$; в противном случае принять, что $W(t) = W'$ с вероятностью $\exp\left(-\frac{\Delta}{T}\right)$.
3. Уменьшить температуру $(rT \to T)$ с использованием коэффициента уменьшения $r \in [0,1]$ и вернуться к пункту 2.
4. После снижения температуры до нуля провести обучение одним из градиентных методов.

Из описания алгоритма видно, что большое значения на скорость сходимости и точность решения полученного этим алгоритмом влияет выбор начальных параметров, таких как $r, T_{\max}, L$. Эти параметры должны быть определены предварительно каким либо эвристическим методом.

Выше был описан так называемый последовательный вариант, но применение имитации отжига возможно и параллельно или взаимозависимо с другими методами оптимизации. Например, выбор нового решения на шаге 2.а может быть осуществлён градиентным методом:

$$W' = W(t-1) + h(t)\,p(t), \qquad (12.6)$$

где $p(t)$ - направление движения, а $h(t)$ - величина шага, которая может зависеть от значения $\exp\left(-\frac{\Delta}{T}\right)$ следующим образом:

$$h(t) = \frac{h_{\max}}{1 + s\ \exp(-\Delta / T)}. \qquad (12.7)$$

В общем случае, для многоэкстремальной сложной целевой функции применение алгоритма имитации отжига в чистом виде малоэффективно. Хорошие результаты получаются при его комбинировании с другими методами.

Литература

1. Осовский С. Нейронные сети для обработки информации. – М.: «Финансы и статистика», 2004.
2. Тархов Д. А. Нейронные сети. Модели и алгоритмы. Кн. 18. – М.: Радиотехника, 2005.
3. Холланд Дж. Генетические алгоритмы // В мире науки. 1992. №9. с. 32-40.
4. Емельянов В. В., Курейчик В. В., Курейчик В. М. Теория и практика эволюционного моделирования. – М.: ФИЗМАТЛИТ, 2003. – 432 с.
5.