Технічні науки

УДК 51-74

Мулява Галина Ярославівна
студент
«Інститут Прикладного Системного Аналізу»
Національний Технічний Університет України
«КПІ ім. Ігоря Сікорського»
Мулява Галина Ярославовна
студент
«Институт Прикладного Системного Анализа»
Национальный Технический Университет Украины
«КПИ им. Игоря Сикорского»
Halyna Muliava
student
«Institute for Applied Systems Analysis»
National Technical University of Ukraine
«Igor Sikorsky KPI»

СИСТЕМНИЙ ПІДХІД У ПРИЙНЯТТІ РІШЕННЯ ДЛЯ ЗАДАЧ КЛАСИФІКАЦІЇ
СИСТЕМНЫЙ ПОДХОД В ПРИНЯТИИ РЕШЕНИЯ ДЛЯ ЗАДАЧ КЛАССИФИКАЦИИ
SYSTEM APPROACH IN DECISION-MAKING A CLASSIFICATION ISSUE

Анотація: У роботі наведено підхід до побудови системи прийняття рішення для класифікації, що ґрунтується на методах системного аналізу. Зокрема використані методи керованого навчання для знаходження параметрів моделі нейронної мережі.

Ключові слова: система прийняття рішення, класифікація, кероване навчання, нейрона мережа.

Аннотация: В работе приведен подход к построению системы принятия решения для классификации, основанной на методах системного анализа. В частности использованы методы управляемого обучения для нахождения параметров модели нейронной сети.

Ключевые слова: система принятия решения, классификация, управляемое обучение, нейронная сеть.

Summary: The paper presents an approach to constructing a decision making system for classification based on methods of system analysis. In particular, used methods of controlled learning to find the parameters of the model of the neural network.

Key words: decision making system, classification, controlled learning, neural network.

Загальна постановка задачі класифікації за допомогою керованого навчання

Розглянемо формальну постановку задачі. X – множина об'єктів. Y – множина допустимих відповідей. Існує цільова функція $y^*: X \to Y$, значення якої $y_i = y^*(x_i)$ відомі тільки на скінченній підмножині об'єктів: $X^\ell = (x_i, y_i)_{i=1}^\ell$ (навчальна вибірка).

Задача навчання полягає в тому, щоб за вибіркою X^ℓ відновити залежність y^*. Тобто побудувати вирішальну функцію $a: X \to Y$, яка наближала б цільову функцію, причому не тільки на об'єктах навчальної вибірки, а й на всій множині X. Саме тому задача цієї роботи відноситься до категорії керованного навчання (англ. supervised learning).

Ознака f_j об'єкта x - це результат вимірювання деякої характеристики об'єкта $j = 1, \ldots, n$. Формально ознака називається відображення $f : X \to D f$, де $D f$ - множина допустимих значень ознаки. Зокрема, будь-який алгоритм $a: X \to Y$, також можна розглядати як ознаку.

Маємо матрицю об'єктів-ознак:

$$F = \|f_j(x_i)\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix}, \qquad (2.1)$$

де f_j - ознака об'єкта x_i, $i = 1, \ldots l$

В залежності від природи множини Df ознаки діляться на кілька типів. Якщо $Df = \{0, 1\}$, то f - бінарна ознака;

Якщо Df -скінченна множина, то f - номінальний ознака;

Якщо Df - скінченна впорядкована множина, то f - порядкова ознака;

Якщо $Df = R$, то f - кількісний ознака.

Залежно від природи множини допустимих відповідей Y задачі навчання по прецедентах діляться на наступні типи. Якщо $Y = \{1,\ldots, M\}$, то це завдання класифікації (classification) на M непересетинаючих класів. У цьому випадку вся множина об'єктів X розбивається на класи $Ky = \{x \in X: y^*(x) = y\}$, і алгоритм $a(x)$ повинен давати відповідь на питання, якому класу належить х. Сформуємо задачу класифікації на 2 класи: $Y = \{-1, +1\}$

Функціонал якості $\mathscr{L}(a, x)$ - функція втрат, величина помилки алгоритму a на об'єкті x.

Наприклад (для задач класифікації) $\mathscr{L}(a, x) = [a(x) \neq y(x)]$

Емпіричний ризик – функціонал якості алгоритму a на X^ℓ:

$$Q(a, X^\ell) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathscr{L}(a, x_i), \qquad (2.2)$$

де $\mathscr{L}(a, x)$ - функція втрат алгоритму a на об'єкті x.

Функція втрат, приймаюча лише значення 0 і 1, називається бінарною. В цьому $\mathscr{L}(a, x) = 1$ означає, що алгоритм припускається помилки на об'єкті х, а функціонал Q називається частотою помилок алгоритму на вибірці X^ℓ.

Найбільш часто використовуються наступні функції втрат, при $Y \subseteq R$:

$\mathscr{L}(a, x) = [a(x) \neq y(x)]$ - індикатор помилки, зазвичай застосовується в задачах класифікації;

$\mathscr{L}(a, x) = | a(x) - y^*(x) |$ - відхилення від правильної відповіді; Q функціонал називається середньою помилкою алгоритму на вибірці X^ℓ;

$\mathscr{L}(a,x) = (a(x) - y^*(x))^2$ - квадратична функція втрат; Q функціонал називається середньою квадратичної помилкою алгоритму на вибірці X^ℓ; зазвичай використовується в задачах регресії.

Класичний метод керованого навчання, так звана мінімізація емпіричного ризику (empirical risk minimization), полягає в тому, щоб знайти в заданому моделі A алгоритму a, що доставляє мінімальне значення функціоналу якості Q на заданому навчальній вибірці X^ℓ:

$$\mu(X^\ell) = \arg\min_{a \in A} Q(a, X^\ell) \qquad (2.3)$$

де $X^\ell = (x_i, y_i)_{i=1}^\ell$ - навчальна вибірка, a – алгоритм навчання.

У завданнях навчання по прецедентах елементи множини X - це не реальні об'єкти, а лише доступні дані про них. Дані можуть бути неточними, оскільки вимірювання значень ознак f_j об'єкта x і цільової залежності $y^*(x)$ зазвичай виконуються з похибками. Дані можуть бути неповними, оскільки виміряють не ознаки, а лише фізично доступні для вимірювання. У такому випадку $y^*(x)$, строго кажучи, не є функцією. Усунути цю некоректність дозволяє імовірнісна постановка задачі.

Замість існування невідомої цільової залежності $y^*(x)$ припустимо існування невідомого ймовірного розподілу на множини $X \times Y$ з щільністю $p(x,y)$, з якого випадково і незалежно вибираються обмежена кількість спостережень . Такі вибірки називається прості або випадковими однаково розподіленими (independent identically distributed).

Розглянемо функція втрат для класифікації. З огляду на X як векторний простір всіх можливих входів, і $Y = \{-1, +1\}$ як векторний простір всіх можливих результатів, ми хочемо знайти функцію $F: X \mapsto R$, який найкраще відображає x в y. Проте, через неповної інформації, шуму в

вимірі, або імовірнісних компонентів в основний процес, можна за те ж саме x, щоб генерувати інший y.

Логістична функція втрат визначаються як сигмоїда з t - це параметр функції, що визначає її крутизну. Коли t прямує до нескінченності, функція вироджується в порогову. При $t = 0$ сигмоїда вироджується в постійну функцію із значенням 0,5. Область значень даної функції знаходиться в інтервалі (0,1). Важливою перевагою цієї функції є простота її похідної:

$$\frac{d\sigma(x)}{dx} = tf(x)(1 - f(x)), \qquad (2.4)$$

де $f(x)$ –ознака об'єкта , $\sigma(x)$ - функція втрат, t – час.

Ця структура призводить до чутливості логістичної функцію втрат до викидів в даних.

Те, що похідна цієї функції може бути виражена через її значення, полегшує використання цієї функції при навчанні мережі за алгоритмом зворотного поширення. Це дозволяє запобігти насиченню від великих сигналів

Ця функція не визначена, коли p (1 | x) = 1 або p (1 | x) = 0 (прямуючи до ∞ і -∞ відповідно), але прогнозує плавну криву, яка росте, коли p (1 | x) збільшується і дорівнює 0, коли p (1 | x) = 0,5

Етапи побудови системи прийняття рішення у задачі класифікації :

1) Розуміння задачі та даних
2) Первинна обробка даних та ознак
3) Побудова моделі
4) Приведення навчання до оптимізації
5) Розв'язок проблем оптимізації і перенавчання
6) Оцінка якості
7) Впровадження та експлуатація

Прийняття рішення за допомогою моделі штучної нейронної мережі

Розглянемо алгоритм навчання мережі для прийняття рішення щодо класифікації. Нейронна мережа - це суперпозиція нейронів з нелінійною функцією активації.

$$a(x, w) = \sigma(\langle w, x \rangle) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right), \qquad (2.5)$$

де $f_j(x)$ - ознака об'єкта x, $w_j \in \mathbb{R}$ - ваги ознак, $\sigma(x)$- функція активації.

Вибір параметрів моделі $w \equiv (w_{jh}, w_{hm}) \in \mathbb{R}^{H(n+M+1)+M}$:

Отримуємо задачу оптимізації:

$$Q(w) := \sum_{i=1}^{\ell} \mathscr{L}(w, x_i, y_i) \to \min_{w}, \qquad (2.6)$$

1. Ініціалізація мережі: вагові коефіцієнти і зсуви мережі приймають малі випадкові значення.

2. Визначення елемента навчальної множини: (вхід - вихід).

Входи $(x_1, x_2, \ldots x_n)$, повинні розрізнятися для всіх прикладів навчальної множини.

3. Обчислення вихідного сигналу: $y_{im} = f(S_{jm})$

$i_m = 1, \ldots, N_m, m = 1, \ldots, L$

де S - вихід суматора, w - вага зв'язку, y - вихід нейрона, b - зсув, i - номер нейрона, N - число нейронів у прошарку, m - номер прошарку, L - число прошарків, f- передатна функція.

4. Налаштування синаптичних ваг:

$$w_{ij}(t+1) = w_{ij}(t) + r\, g_j\, x_i', \qquad (2.7)$$

де w_{ij} - вага від нейрона i або від елемента вхідного сигналу i до нейрона j у момент часу t, x_i'- вихід нейрона i, r - швидкість навчання, g_j - значення похибки для нейрона j.

Якщо нейрон з номером j належить останньому прошарку, тоді

$$g_j = y_j(1 - y_j)\,(d_j - y_j)\,, \qquad (2.8)$$

де d_j - бажаний вихід нейрона j, y_j - поточний вихід нейрона j.

Якщо нейрон з номером j належить одному з прошарків з першого по передостанній, тоді k пробігає всі нейрони прошарку з номером на одиницю більше, ніж у того, котрому належить нейрон j.

Вибір числа шарів потребує системного підходу. Якщо в конкретному завданні гіпотеза про лінійну роздільність класів виглядає правдоподібно, то можна обмежитися одношаровим перцептроном. Двошарові мережі дозволяє представляти звивисті нелінійні границі, і в більшості випадків цього вистачає. Чим більше шарів, тим багатший клас функцій реалізує мережу, але тим гірше сходяться градієнтні методи, і тим важче її навчити.

Вибір числа нейронів в прихованому шарі Н виробляє різні способи, але жоден з них не є найкращим.

1. Візуальний спосіб. Якщо межа класів (або крива регресія) занадто згладжена, значить, мережа занадто спрощена, і необхідно збільшувати число нейронів в прихованому шарі. Якщо межа класів (або крива регресії) відчуває занадто різкі коливання, на тестових даних спостерігаються великі викиди, ваги мережі приймають великі по модулю значення, то мережа переускладнена, і прихований шар слід скоротити

2. Оптимізація H по зовнішньому критерію, наприклад, за критерієм сковзного контролю або середньої помилки на незалежній контрольній вибірці $Q(X^k)$. Залежність зовнішніх критеріїв від параметра складності, яким є Н, звичайно має характерний оптимум.

Метод оптимізації для знаходження параметрів моделі

Розглянемо метод стохастичного градієнтного спуску.

Вхід: вибірка X^ℓ темп навчання η, параметр λ

Вихід: $w \equiv \left(w_{jh}, w_{hm}\right) \in \mathbb{R}^{H(n+M+1)+M}$

Недоліки – для кожного об'єкту рахуємо функцію втрат

Кількість вагових коефіцієнтів $N = (n+1)H + (H+1)M$

Складність алгоритму: $O(N^2)$

Градієнтний спуск працює в просторах з будь-яким числом вимірів, навіть у нескінченновимірних. В останньому випадку простір пошуку зазвичай є простором функцій, і для визначення напрямку спуску здійснюється обчислення похідної Гато функціоналу, який мінімізують.

Розглянемо швидкий проксимальний градієнтний метод. А саме, якщо функція F є опуклою, а ∇F є ліпшицевою, і немає припущення, що F є сильно опуклою, то похибку цільового значення, породжувану методом градієнтного спуску на кожному кроці k, буде обмежено O(1 / k). Із застосуванням методики прискорення Нестерова похибка знижується до O(1 / k^2).

Стохастичний градієнтний спуск з імпульсом запам'ятовує оновлення Δ w на кожній ітерації і визначає наступне оновлення у вигляді випуклої(лінійної) комбінації градієнта і попереднього оновлення.

$$w := w - \eta \nabla Q_i(w) + \alpha \Delta w \quad , \qquad (2.9)$$

де $Q_i(w)$ - функція втрат, що мінімізується, параметр w, за яким мінімізується - η - довжина кроку (іноді званої темпом навчання в машинному навчанні).

У стохастичному (або «он-лайн») градієнтном спуску, істинний градієнт $Q_i(w)$апроксимується градієнтом по одному об'єкті.

Алгоритм стохастичного градієнтного спуску може бути представлений таким чином:

- Виберіть вихідний вектор параметрів w та темп навчання (learning_rate) η
- Повторити до тих пір, поки не буде отримано приблизне мінімальне.
- Випадково перетасувати об'єкти в навчальному наборі

Компроміс між обчислення істинного градієнта і градієнта для одного об'єкту - це обчислення градієнта для більш, ніж одного навчального об'єкту (так званої міні-порції - mini-batch) на кожному кроці. Збіжність стохастичного градієнтного спуску була проаналізована за допомогою теорії опуклої мінімізації та стохастичної апроксимації.

Розглянемо метод зворотного поширення.

Вхід: вибірка X^ℓ темп навчання η,

Параметр λ, H - число нейронів прихованого шару

Вихід: $w \equiv \left(w_{jh}, w_{hm}\right) \in \mathbb{R}^{H(n+M+1)+M}$

Складність алгоритму - $O(3N)$

У разі двошарової мережі прямий хід, зворотний хід і обчислення градієнта вимагають порядку O (Hn + HM) операцій. Тому даний метод легко реалізується на обчислювальних пристроях з паралельною архітектурою.

Методи вибору параметрів

З метою вибору оптимальних параметрів системи використовують ряд методів системного аналізу, статистики та теорії керування. Найбільш відомі з них: попередня зупинка, регуляризація системи, усереднення прийняття рішення, відсікання та нарощування.

Розглянемо метод попередньої (ранньої) зупинки. Ретельна підгонка параметрів моделі на фіксованій навчальній вибірці може призвести до надто точного налаштування на особливості конкретних даних, що призводить до неминуче збільшення реальної похибки. Очевидним виходом із цієї ситуації є зупинка процесу навчання до того моменту, доки реальна похибка не почне зростати з причини надлишкового регулювання. Зупинку виконують за наявності тенденції зростання реальної похибки. Для того, щоб уникнути попадання в локальний мінімум, оцінку реальної похибки виконують на підмножині параметрів.

Розглянемо регуляризацію. Реальну похибку представляють у вигляді суми середньоквадратичної похибки та деякої функції *R(W)*, що задає попередній стан моделі:

$$\boldsymbol{E} = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{y}^{(n)} - \tilde{\boldsymbol{y}}^{(n)}\right)^2 + k\boldsymbol{R}(\boldsymbol{W}), \qquad (2.10)$$

де *R(W)* – функція, що задає попередній стан мережі, *k* — коефіцієнт регуляризації, який задає ступінь впливу *R(W)* на реальну похибку, *N* - кількість об'єктів навчальної вибірки, $y^{(n)}$ – результат на об'єкті *n*,

Мінімізація *E* прямо відповідає вирішенню дилеми відхилення або дисперсії, оскільки середньоквадратична похибка задає статистичне відхилення, а функція *R(W)* – величину дисперсії. Коефіцієнт регуляризації k відіграє роль параметра, що формує співвідношення між статистичним відхиленням та дисперсією при вирішенні дилеми відхилення або дисперсії і змінюється в діапазоні $0 < k < \frac{2\sigma^2}{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}}$.

Розв'язок даної задачі одержують, виходячи з варіаційного принципу, при реалізації якого використовують вхідний набір даних та попередню інформацію про гладкість функції. Отже, регуляризація може

бути проведена шляхом формування вихідних даних у відповідності до функції.

Оцінка методу прийняття рішення

Реальною похибкою називають похибку, з якою функціонує в умовах реальних даних, на відміну від навчальної похибки, що визначається при роботі з навчальною вибіркою.

Перехресна перевірка (англ. cross-validation) — метод оцінювання достовірності математичної моделі з метою перевірки, наскільки результати статистичного аналізу узагальнюються на незалежному наборі даних.

Нехай дана вибірка . Розіб'ємо її N різними способами на дві неретинаючі підвибірки - навчальну X_n^l довжини l і контрольну X_n^k довжини $k = L - l$. Для кожного розбиття $n = 1,\ldots, N$ побудуємо алгоритм $a_n = \mu(X_n^\ell)$ і обчислимо значення $Q_n = Q(a_n, X_n^k)$. Середнє арифметичне значень Q_n за всіма розбиття називається оцінкою ковзного контролю (cross-validation, CV):

$$\mathrm{CV}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N} Q(\mu(X_n^\ell), X_n^k), \qquad (2.11)$$

де $\mu(X_n^l)$ – алгоритм моделі, X_n^l навчальна вибірка, X_n^k - контрольна вибіка

Одноразова перехресна перевірка передбачає розбиття вибірки на підвибірки з метою проведення аналізу на одній частині (навчальному наборі, англ. training set) і перевірки аналізу на іншій частині (контрольньому наборі, англ. validation set). Для зниження дисперсії здійснюється багаторазова перехресна перевірка із застосуванням різного розбиття, і результати цих перевірок усереднюються.

Висновки

У роботі розглянуто базові математичні поняття загальної постановки задачі машинного навчання. Представлено формальний опис нейронних мереж та їх структури. Описано методологію навчання багатошарових ШНМ прямого поширення, пошуку оптимальної кількості структурних одиниць, зокрема прихованих шарів та їх розмір.

Також приведено алгоритм пошуку вагових коефіцієнтів за допомогою методу стохастичного градієнтного спуску та детальний опис вибору параметрів для даної задачі оптимізації. Розглянуто способи оцінки побудованої моделі нейронної мережі.

Перевагами цих методів є їх паралелізм типових процесів для ефективного і ресурсно незатратного розв'язку єдиною глобальної задачі, здатність навчатися, що веде до універсальності, можливість вибору параметрів. Дані методи широко використовуються спеціалістами у галузі аналізу даних протягом останнього десятиріччя.

Література

1. Акулов П.В. Рішення задач прогнозування за допомогою нейронних мереж [Електронний ресурс] / Акулов Павло Володимирович - Режим доступу: www.dgtu.donetsk.ua
2. Rumelhart, D. E. and J. L. McClelland, 1986, Parallel Distributed Processing:
   Explorations in the Microstructure of Cognition, MIT Press, Cambridge, MA
3. McCulloch, W. S. and W. Pitts, 1943, "A logical calculus of ideas immanent in nervous activity," Bulletin of Mathematical Biophysics, vol. 5 pp. 115-133 Neural Networks, pp. 2476-2481
4. Rosenblatt, F., 1958, "The perceptron: a probabilistic model for information
   storage and organization in the brain," Psychological Review, vol. 65, pp. 386
5. Ширяев А.Н. Основы стохастической финансовой математики / Ширяев А.Н- М.: ФАЗИС, 1998.–415с.
6. Resampling Statistics [Електронний ресурс] / Edward Connor // Lecture notes Biology 710 - Advanced Biometry San Francisco State University, San Francisco, California, 2008. – Режим доступу: http://userwww.sfsu.edu/efc/classes/biol710/boots/rs-boots.htm