

## Машинно обучение

### Лабораторно упражнение 7

#### Класификация с множество класове. Представяне на невронна мрежа

Упражнението демонстрира класификация в множество класове чрез методи логаритмична регресия и невронни мрежи. Целта е да се покаже как се извършва класификация на входни данни към един от множество възможни класове, да се демонстрира как се представя структурата на невронна мрежа и да се сравни точността на обучението, постигнато чрез логаритмична регресия и невронна мрежа с предварително обучени параметри.

Като пример се разглежда задачата за разпознаване на ръкописни цифри от 0 до 9.

#### 1. Класификация в множество класове

Автоматично разпознаване на ръкописни цифри от 0 до 9 се използва например при разпознаване на пощенските кодове, написани на пратки. Тази задача е пример за класификация, при която входните данни (изображения на ръкописни цифри) се отнасят към един от множество възможни класове (цифрите от 0 до 9).

##### 1.1. Обучителни данни

Файлът `data.mat` съдържа 5000 форматирани изображения на ръкописни цифри от 0 до 9. Данните са взети от еталонната база данни MNIST (<http://yann.lecun.com/exdb/mnist/>), съдържаща множество примери за ръкописни цифри. Файлът е в `.mat` формат, т.е. данните са записани в MATLAB матрична форма вместо в текстов ASCII формат. Изображението на всяка цифра се представя с матрица, която се зарежда директно в програмата.

Във файла са дадени 5000 обучителни примери, всеки от които е изображение, представено с матрица от пиксели  $20 \times 20$ . Всеки пиксел е представен с числова стойност за неговия интензитет от сивата скала. Матрицата  $20 \times 20$  се преобразува във вектор  $400 \times 1$ . Като резултат всеки обучителен пример става един ред в матрицата  $X$ . Тя има размер  $5000 \times 400$  и всеки ред в нея и обучителен пример с изображение на ръкописна цифра.

Обучителните примери съдържат също вектор  $y$  ( $5000 \times 1$ ) с етикети `labels`) за всяка цифра.

##### 1.2. Визуализиране на обучителните данни

Избират се 20 реда от матрицата с обучителните примери  $X$ . Скриптът изобразява цифрите (фиг. 1).



Фиг. 1. Обучителни примери

##### 1.3. Логаритмична регресия за класификация с множество класове

Ще използваме логаритмична регресия, за да построим множествен класификатор. Тъй като в случая имаме 10 възможни класа, ще обучим 10 отделни логаритмични класификатори.

### 1.3.1. Регуляризирана функция на цената

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

За да изчислим функцията  $J(\theta)$ , трябва да изчислим стойността на хипотезата  $h_{\theta}(x^{(i)})$  за всеки пример  $i$ , където:

$$h_{\theta}(x^{(i)}) = g(\theta^T x^{(i)})$$

$$g(\theta^T x^{(i)}) = \frac{1}{1 + e^{-\theta^T x^{(i)}}} - \text{сигмоидна функция}$$

Регуляризация на параметър  $\theta_0$  не се прави.

**Задача 1. Довършете функцията sigmoid(z), която връща стойността на сигмоидната функция.**

**Задача 2. Довършете функцията lrcostFunctionReg(theta, reg, X, y), която изчислява регуляризираната цена. Реализирайте векторна форма на цената (т.е. без да използвате никакъв цикъл за всички обучителни примери).**

### 1.3.2. Регуляризиран градиент

Регуляризираният градиент се изчислява по следния начин:

$$\frac{\delta J(\theta)}{\delta \theta_j} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j$$

Векторна форма на регуляризиания градиент:

$$\frac{\delta J(\theta)}{\delta \theta_j} = \frac{1}{m} X^T (g(X\theta) - y) + \frac{\lambda}{m} \theta_j$$

Регуляризация на параметър  $\theta_0$  не се прави.

**Задача 3. Довършете функцията lrgredientReg(theta, reg, X,y), която изчислява регуляризиания градиент. Реализирайте векторна форма на градиента (т.е. без да използвате никакъв цикъл за всички обучителни примери).**

## 1.4. Класификация с множество класове

### 1.4. 1. Построяване на класификатори за всички класове

За всеки клас обучаваме отделен класификатор чрез логаритмична регресия. Функцията oneVsAll връща матрица  $\theta$  с размерност  $K \times (N + 1)$ , където  $K$  - брой класове,  $N$  - брой променливи (features). Матрицата  $\theta$  съдържа намерените стойности на параметрите за всеки класификатор. Всеки ред от матрицата  $\theta$  съдържа намерените стойности на параметрите за класификатора на съответния клас.

**Задача 4. Попълнете параметрите при извикване на функцията oneVsAll за построяване на класификатори за всеки клас чрез логаритмична регресия. Използвайте стойност 0.1 за параметър на регуляризацията.**

### 1.4.2. Предсказване чрез построените класификатори

След като сме обучили класификаторите за отделните класове, ще ги използваме за предсказване (т.е. за разпознаване) на цифрата в дадено изображение. За всеки обучителен пример изчисляваме каква е вероятността изображението да принадлежи на даден клас, като използваме обучените класификатор за съответния клас. Функцията `predictOneVsAll` връща номера на класа, към който цифрата принадлежи с най-голяма вероятност.

**Задача 5.** Довършете функцията `predictOneVsAll`, като изчислите вероятностите за принадлежност на всеки обучителен пример към всеки клас. За целта използвайте реализираната по-горе сигмоидна функция.

**Задача 6.** Попълнете параметрите при извикване на функцията `predictOneVsAll`.

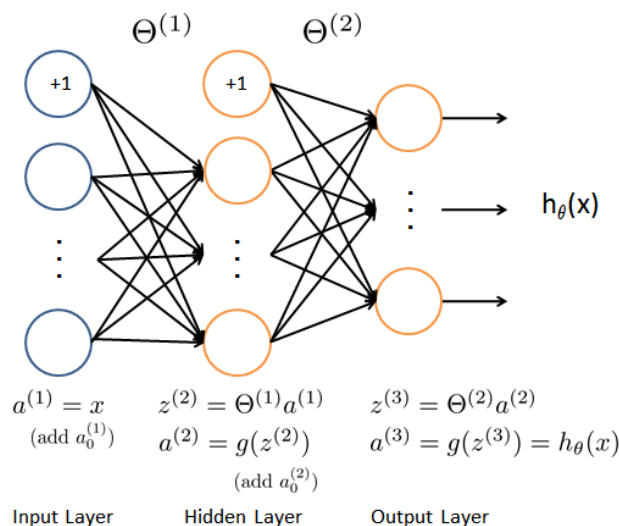
Би трябвало да получите резултат за точност на обучението с логаритмична регресия 93.24 %.

## 2. Невронни мрежи

Логаритмичната регресия не е ефективен метод за построяване на по-сложни нелинейни класификатори. В такива случаи приложение имат невронните мрежи.

### 2.1. Модел на невронната мрежа

Моделът на невронната мрежа за разпознаване на ръкописни цифри е представен на фиг. 2. Мрежата се състои от 3 слоя - входен слой (input layer), скрит слой (hidden layer) и изходен слой (output layer). Входните данни са стойностите на интензитетите на пикселите от изображението на цифра. Тъй като изображенията са с размерност 20x20, входният слой има 400 върха (+1 за допълнителен връх за т.нар. отклонение (bias unit)).



Фиг. 2. Модел на невронна мрежа за разпознаване на ръкописни цифри

$\theta^{(1)}$  - матрица със стойности на параметри в първия слой (input layer)

$\theta^{(2)}$  - матрица със стойности на параметри в скрития слой (hidden layer)

$\theta^{(1)}$  се зарежда в `Theta1` с размерност 25x401, а  $\theta^{(2)}$  се зарежда в `Theta2` с размерност 10x26. Тези размерности са определени от това, че във втория (скрития) слой има 25 върха, а в изходния слой има 10 върха (по един за всеки възможен клас).

Файлът weights.mat съдържа стойности на предварително обучени параметри.

## 2.2. Feedforward propagation

Feedforward propagation е алгоритъм за изчисляване на стойностите във върховете от всеки слой на невронната мрежа. В дадената невронна мрежа стойностите във върховете се изчисляват по следния начин:

За първи слой (input layer):

Добавяме връх  $a_0^{(1)} = 1$ .

$$a^{(1)} = X$$

, където

$a^{(1)}$  - стойности на върховете на мрежата в първия слой

$X$  - матрица със стойности на входните променливи

За втори слой (hidden layer)

Добавяме връх  $a_0^{(2)} = 1$ .

$$z^{(2)} = \theta^{(1)} a^{(1)}$$

$$a^{(2)} = g(z^{(2)})$$

$\theta^{(1)}$  - стойности на параметрите в първи слой на мрежата

$a^{(1)}$  - стойности на върховете в първи слой на мрежата

$g(z^{(2)})$  - сигмоидна функция с параметър  $z^{(2)}$

За трети слой (output layer)

$$z^{(3)} = \theta^{(2)} a^{(2)}$$

$$a^{(3)} = g(z^{(3)}) = h_{\theta}(x)$$

$\theta^{(2)}$  - стойности на параметрите във втори слой на мрежата

$a^{(2)}$  - стойности на върховете във втори слой на мрежата

$g(z^{(3)})$  - сигмоидна функция с параметър  $z^{(3)}$

Стойността на хипотезата  $h_{\theta}(x)$  се получава в изходния слой на мрежата като стойност в интервала 0-1 за всеки възможен клас. Тази стойност определя вероятността за принадлежност на входните данни към дадения клас. Крайният резултат от невронната мрежа е класът с най-голяма стойност на хипотезата, т.е. класът с най-голяма вероятност.

**Задача 7.** Довършете функцията `predict`, която изчислява стойностите на върховете в невронната мрежа и връща най-високата вероятност за принадлежност на входен пример към даден клас.

**Задача 8.** Попълнете параметрите при извикване на функцията `predict`.

Би трябвало да получите, че точността на обучението с невронната мрежа 97.52 %.