Podstawy uczenia maszynowego

DWUWARSTWOWA SIEĆ NEURONOWA

KLASYFIKACJA IRYSÓW

Jakub Gulcz - nr. 75999

1 Dane

1.1 Pobieranie danych

Pierwotne dane:

ID	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
...					
81	5.5	2.4	3.8	1.1	Iris-versicolor
82	5.5	2.4	3.7	1.0	Iris-versicolor
83	5.8	2.7	3.9	1.2	Iris-versicolor
84	6.0	2.7	5.1	1.6	Iris-versicolor
...					
145	6.7	3.3	5.7	2.5	Iris-virginica
146	6.7	3.0	5.2	2.3	Iris-virginica
147	6.3	2.5	5.0	1.9	Iris-virginica
148	6.5	3.0	5.2	2.0	Iris-virginica
149	6.2	3.4	5.4	2.3	Iris-virginica
150	5.9	3.0	5.1	1.8	Iris-virginica

Dane są pobierane i wstępnie obrabiane przez funkcję GetData(FileName). Funkcja ta wyciąga z pliku informacje, dzieli je oraz pozbywa się kolumny ID (zbędna informacja).

FileName ← Nazwa pliku

1.2 Obróbka danych

Za dalszą obróbkę danych odpowiada funkcja ConvertData(RawData,LEP). Funkcja wyciąga występujące etykiety, tworzy listę przykładów uczących i ich rozwiązań oraz liste przykładów sprawdzających i ich rozwiązań.

RawData ← Wstępnie przygotowane dane
LEP ← Procent danych, które mają zostać wykorzystane jako przykłady uczące

Przetworzone dane:

SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
...			
5.5	2.4	3.8	1.1
5.5	2.4	3.7	1.0
5.8	2.7	3.9	1.2
6.0	2.7	5.1	1.6
...			
6.7	3.3	5.7	2.5
6.7	3.0	5.2	2.3
6.3	2.5	5.0	1.9
6.5	3.0	5.2	2.0
6.2	3.4	5.4	2.3
5.9	3.0	5.1	1.8

Answer	Species
1,0,0	Iris-setosa
1,0,0	Iris-setosa
1,0,0	Iris-setosa
1,0,0	Iris-setosa
1,0,0	Iris-setosa
1,0,0	Iris-setosa
1,0,0	Iris-setosa
...	
0,1,0	Iris-versicolor
0,1,0	Iris-versicolor
0,1,0	Iris-versicolor
0,1,0	Iris-versicolor
...	
0,0,1	Iris-virginica
0,0,1	Iris-virginica
0,0,1	Iris-virginica
0,0,1	Iris-virginica
0,0,1	Iris-virginica
0,0,1	Iris-virginica

2 Sieć neuronowa

2.1 Inicjalizacja

Za inicjalizację sieci odpowiada funkcja Init(S,K1,K2). Tworzy ona dwie macierze wag dla połączeń neuronów oraz wypełnia je początkowo losowymi wartościami z zakresu -0.1 do 0.1.

Wejście $\rightarrow$ Warstwa ukryta: $W^{(1)}_{S+1\times K_1}$

Warstwa ukryta $\rightarrow$ Wyjście: $W^{(2)}_{K_1+1\times K_2}$

S $\leftarrow$ Liczba wejść do sieci
K1 $\leftarrow$ Liczba neuronów w warstwie ukrytej
K2 $\leftarrow$ Liczba wyjść sieci

2.2 Symulacja sieci

SimulateNN(W1,W2,Example) symuluje działanie sieci dwuwarstwowej.

$$\overline{X}_{S\times 1} \xrightarrow{[^{-1}_{\bullet}]} \overline{X}^{(1)}_{S+1\times 1} \xrightarrow{W^{(1)}_{S+1\times K_1}} \overline{U}^{(1)}_{K_1\times 1} \xrightarrow{f(\bullet)} \overline{Y}^{(1)}_{K_1\times 1} \xrightarrow{[^{-1}_{\bullet}]}$$

$$\xrightarrow{[^{-1}_{\bullet}]} \overline{X}^{(2)}_{K_1+1\times 1} \xrightarrow{W^{(2)}_{K_1+1\times K_2}} \overline{U}^{(2)}_{K_2\times 1} \xrightarrow{f(\bullet)} \overline{Y}^{(2)}_{K_2\times 1}$$

Zgodnie ze schematem, do wektora danych wejściowych dodajemy bias o wartości -1.

Obliczamy pobudzenie neuronów warstwy ukrytej zgodnie ze wzorem:

$$U = \sum_{s=1}^{S} x_s \cdot w_s$$

Sprawdzamy funkcję aktywacji zgodnie ze wzorem:

$$Y = \frac{1.}{1+e^{(-\beta * U)}}$$

Dodajemy do wktora danych wyjściowych wartwy ukrytej bias (-1).

Obliczamy pobudzenie neuronów warstwy wyjściowej zgodnie ze wzorem:

$$U = \sum_{s=1}^{S} x_s \cdot w_s$$

Sprawdzamy funkcję aktywacji zgodnie ze wzorem:

$$Y = \frac{1.}{1+e^{(-\beta * U)}}$$

2.3 Uczenie

Za naukę sieci odpowiada funkcja Learn(W1przed,W2przed,P,T,n). Na podstawie podanych przykładów oraz liczby powtorzeń sieć jest wstanie dopasować swoje wagi między neuronami by jak najlepiej rozwiązywać problemy danego typu.

W1przed ← Macierz wag stworzona w funkcji Init()

W2przed ← Macierz wag stworzona w funkcji Init()

P ← Wybrane przyklady na, których sieć ma się uczyć

T ← Odpowiedzi do wcześniej wybranych przykładów

n ← liczba powtórzeń uczenia się sieci

2.3.1 schemat uczenia

krok.1 Losujemy przyklad z dostępniej puli

krok.2 Podajemy wektor danych wejściowych do funkcji SimulateNN() i obliczamy wyjścia

krok.3 Liczymy błąd na wyjściu sieci zgodnie ze wzorem:

$$D2 = PoprawnaOdpowiedz - WyjcieSieci$$

krok.4 Obliczamy błąd na wyjściu warstwy ukrytej:

$$D1 = MacierzWagNaWyjciuWarstwyUkrytej[1:] * D2$$

Kolejność obliczania błędów jest istotna.

krok.5 Obliczamy poprawki do wag

$$E1 = \beta \cdot D1 \times Y1 \times (1 - Y1)$$
$$E2 = \beta \cdot D2 \times Y2 \times (1 - Y2)$$

oraz

$$dW1 = WspolczynnikUczenia \times X1 \times E1'$$
$$dW2 = WspolczynnikUczenia \times X2 \times E2'$$

Krok.6 Dodajemy poprawki do wag

$$W1 = W1 + dW1$$
$$W2 = W2 + dW2$$

krok.7 Powtórzyć Kroki od 1 do 6 n razy