Sztuczne sieci neuronowe

Krzysztof A. Cyran
POLITECHNIKA ŚLĄSKA
Instytut Informatyki, p. 335

Wykład 10

Mapa cech Kohonena i jej modyfikacje

- uczenie sieci samoorganizujących się
- kwantowanie wektorowe
- algorytm WTA
- uczenie mapy cech Kohonena algorytmem WTM
- typy sąsiedztwa
- algorytm stochastycznej relaksacji
- algorytm SCS
- algorytm gazu neuronowego

Zastosowania mapy cech Kohonena

Mapa cech Kohonena

- Sieć konkurencyjna uczona bez nadzoru
- Uczeniu podlega zwycięski neuron oraz jego „otoczenie”
- Mapa wykorzystuje sąsiedztwo neuronów 1, 2 lub rzadziej więcej wymiarowe
- Zastosowanie:
  - Do znajdowania regionów aktywizowanych podobnymi wartościami cech
  - w kompresji stratnej obrazów

Uczenie sieci samoorganizujących się

- Celem uczenia samoorganizującego jest taki dobór wag neuronów, który zminimalizuje wartość oczekiwaną zniekształcenia, mierzonego jako błąd aproksymacji wektora wejściowego **x** wartościami wag neuronu zwycięskiego:

$$E = \frac{1}{p}\sum_{i=1}^{p}\left\|\mathbf{x}_i - \mathbf{w}_{w(i)}\right\|$$

gdzie $w(i)$ to numer zwycięskiego neuronu przy prezentacji wektora $\mathbf{x}_i$, a $\mathbf{w}_{w(i)}$ to wektor wag tego neuronu

Kwantowanie wektorowe

- Sieć po nauczeniu się, realizuje tzw. kwantowanie wektorowe VQ

  (ang. Vector Quantization), czyli aproksymację dowolnego wektora, poprzez wektor wzorca, najbliższy rozważanemu wektorowi

  (jest to równoważne kwantyzacji wektorowej przestrzeni wejściowej)
- Kwantowanie to zwane jest w tym przypadku LVQ

  (ang. Learning Vector Quantization), ze względu na poprzedzajacy proces uczenia, konieczny do osiągnięcia poprawnego kwantowania.

Algorytm WTA

- Przy wykorzystaniu sieci neuronowych, algorytm WTA (Winner Takes All) jest odpowiednikiem algorytmu K-uśrednień (K-means) znanego z analizy skupień
- Algorytm WTA polega na obliczaniu aktywacji każdego neuronu, a następnie wyborze zwycięzcy o największym sygnale wyjściowym

Uczenie mapy cech Kohonena

- Zwycięski neuron, a więc ten dla którego odległość między wektorem wag $\mathbf{w}_w$ a wektorem wejściowym $\mathbf{x}$ jest najmniejsza podlega uczeniu, polegającym na adaptacji swoich wag w kierunku wektora $\mathbf{x}$:

$$\mathbf{w}_w(k+1) = \mathbf{w}_w(k) + \eta(\mathbf{x} - \mathbf{w}_w(k))$$

- Uwaga: jeżeli wektory wejściowe są normalizowane, wówczas minimalna odległość między wektorem wag $\mathbf{w}_w$ a wektorem wejściowym $\mathbf{x}$ odpowiada równocześnie maksymalnej wartości iloczynu skalarnego: $\mathbf{w}_w \cdot \mathbf{x}$

Uczenie mapy cech Kohonena

- W ścisłym algorytmie WTA pozostałe neurony nie podlegają uczeniu, co prowadzi do słabej zbieżności
- W wersji uogólnionej WTM (Winner Takes Most), wykorzystywanej w mapie cech Kohonena, wprowadza się sąsiedztwo neuronu zwycięskiego, również podlegające modyfikacji
- Dodatkowo można wprowadzić modyfikację uwzględniającą zmęczenie zwycięskich neuronów: jest to modyfikacja zaczerpnięta z biologii, mająca na celu faworyzowanie neuronów o najmniejszej aktywności dla wyrównania szans

Uczenie mapy cech Kohonena metodą WTM

- Działanie algorytmu WTM przedstawia się następująco:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta_i G(i,\mathbf{x})(\mathbf{x} - \mathbf{w}_i(k))$$

dla wszsytkich neuronów i należących do sąsiedztwa S_w zwycięzcy, przy czym funkcja G określa wpływ odległości od zwycięzcy na wielkość modyfikacji

WTA jako szczególny przypadek WTM

- Definiując $G(i, \mathbf{x})$ w postaci:

$$G(i,\mathbf{x}) = \begin{cases} 1 & \text{dla} \quad i = w \\ 0 & \text{dla} \quad i \neq w \end{cases}$$

gdzie w oznacza numer zwycięzcy, dostajemy klasyczny algorytm WTA, jako przypadek szczególny algorytmu WTM

Uczenie mapy cech Kohonena metodą WTM

- W samoorganizujących się mapach Kohonena określa się najpierw zwycięzcę używając euklidesowej odległości, po czym ustala się wartość współczynnika adaptacji dla neuronów należących do sąsiedztwa.

Sąsiedztwo prostokątne

- W klasycznym algorytmie Kohonena funkcja $G(i, \mathbf{x})$ jest zdefiniowana jako:

$$G(i,\mathbf{x}) = \begin{cases} 1 & \text{dla} \quad d(i,w) \le \lambda \\ 0 & \text{dla pozostalych} \end{cases}$$

gdzie $d(i, w)$ oznacza odległość euklidesową między wektorami wag neuronu zwycięzcy w oraz neuronu i-tego.

- Współczynnik λ jest promieniem sąsiedztwa o wartości malejącej wraz z postępem uczenia
- Sąsiedztwo takie nosi nazwę prostokątnego

Sąsiedztwo gaussowskie

- Innym typem sąsiedztwa stosowanym w mapach odwzorowań topologicznych Kohonena, jest sąsiedztwo gaussowskie, w którym funkcja $G(i, \mathbf{x})$ jest zdefiniowana jako:

$$G(i,\mathbf{x}) = \exp\left(-\frac{d^2(i,w)}{2\lambda^2}\right)$$

- Sąsiedztwo gaussowskie prowadzi do lepszej organizacji sieci, niż sąsiedztwo prostokątne

Algorytm stochastycznej relaksacji

- W algorytmie stochastycznej relaksacji neuron *i*-ty podlega modyfikacji z prawdopodobieństwem danym przez rozkład Gibbsa:
- *T* jest parametrem zwanym temperaturą, jak w symulowanym wyżarzaniu

$$P(i) = \frac{\exp\left(-\frac{\|\mathbf{x}-\mathbf{w}_i\|^2}{T}\right)}{\sum_{j=1}^{n}\exp\left(-\frac{\|\mathbf{x}-\mathbf{w}_j\|^2}{T}\right)}$$

Zachowanie algorytmu stochastycznej relaksacji

- Przy wysokiej temperaturze wszystkie neurony zmieniają swe wektory wagowe z prawdopodobieństwem:

$$\lim_{T \to \infty} P(i) = \frac{1}{N}$$

- W skrajnym przypadku, gdy $T \to 0$,

  $P(i) \to 1$ dla zwycięzcy oraz

  $P(i) \to 0$ dla neuronów różnych od zwycięzcy
- W trakcie działania algorytmu temperatura stopniowo obniża się od wartości maksymalnej do zera, dzięki czemu w granicy algorytm przechodzi w algorytm WTA

Sąsiedztwo w algorytmie stochastycznej relaksacji

- Sąsiedztwo w tym algorytmie jest również stochastyczne i dane jest wzorem:

$$G(i,\mathbf{x}) = \begin{cases} 1 & \text{dla} \quad P(i) \le P \\ 0 & \text{dla pozostalych} \end{cases}$$

gdzie P jest liczbą losową o rozkładzie równomiernym z przedziału [0,1]

Algorytm SCS

- Algorytm SCS (Soft Competition Scheme) jest deterministyczną wersją algorytmu stochastycznej relaksacji, zmierzająca do poprawy efektywności wolnego algorytmu stochastycznego.
- W modyfikacji tej wartość binarna funkcji *G* przyjmowana z prawdopodobieństwem *P* jest zastąpiona wartością identycznie obliczanego prawdopodobieństwa, tzn.:

$$G(i, \mathbf{x}) = P(i)$$

Algorytm gazu neuronowego

- Wszystkie neurony sortuje się narastająco, według ich odległości od wektora **x**.
- Wartość funkcji sąsiedztwa wyznacza się ze wzoru:

$$G(i,\mathbf{x}) = \exp\left(-\frac{m(i)}{\lambda}\right)$$

gdzie $m(i)$ oznacza kolejność neuronu uzyskaną w procesie sortowania, a λ jest malejącym w czasie parametrem analogicznym do promienia sąsiedztwa w algorytmie Kohonena

Zachowanie się algorytmu gazu neuronowego

- Przy $\lambda=0$, tylko zwycięzca podlega adaptacji, i algorytm staje się algorytmem WTA
- Przy $\lambda\neq0$ algorytm przypomina podejście zbiorów rozmytych, przy czym każdemu neuronowi odpowiada wartość funkcji przynależności do sąsiedztwa, określona poprzednim wzorem

Porównanie algorytmów samoorganizacji

- Jeżeli za kryterium przyjmiemy błąd kwantowania, wówczas otrzymamy następującą kolejność (od najlepszego do najgorszego) algorytmów samoorganizacji:

- gaz neuronowy
- SCS
- K-means
- Mapa Kohonena

Mapa cech Kohonena: zastosowanie w kompresji obrazów

- Obraz o rozmiarze $N_x \times N_y$ pikseli (np. 512 × 512) podzielony jest na ramki $n_x \times n_y$ pikseli każda (np. 4 × 4 piksele)
- Każdy piksel reprezentowany jest przez 8-bitowy odcień szarości
- Każda ramka stanowi pojedynczy ($n_x \times n_y$ wymiarowy) wektor wejściowy **x**
- Sieć samoorganizująca zawiera n (np. 512) neuronów, każdy połączony z wszystkimi składowymi wektora **x**

- Uczenie sieci dowolnym z prezentowanych algorytmów ma na celu minimalizację błędu kwantowania, tak by każdemu **x** odpowiadały wektor wag neuronu zwycięzcy

- Przy kolejnej prezentacji ramek na wejście nauczonej sieci, plik skompresowany zawiera numery zwyciężających neuronów, oraz słownik w którym każdemu neuronowi przyporządkowany jest jego wektor wag

Mapa cech Kohonena: zastosowanie w kompresji obrazów

- Ponieważ zazwyczaj liczba ramek $N_r = (N_x \times N_y)/(n_x \times n_y)$ jest dużo większa od liczby neuronów n

  (u nas N_r= 16K ramek vs. n=0.5k neuronów, tj. stosunek N_r/n=32), więc uzyskuje się kompresję obrazu o współczynniku kompresji K_r danym przez:

$$K_r = \frac{N_r n_x n_y T}{N_r \log_2 n + n n_x n_y t}$$

gdzie T jest liczbą bitów użytych na reprezentację piksela, a t liczbą bitów użytych na reprezentację wartości pojedynczej wagi (u nas T=8bitów, oraz t=8bitów)

Stąd mamy:

$$K_r = \frac{16K \times 4 \times 4 \times 8}{16K \times \log_2 512 + 0.5K \times 4 \times 4 \times 8} =$$

$$= \frac{(16 \times 128)K}{(16 \times 9 + 16 \times 4)K} = \frac{16K \times 128}{16K \times 13} = \frac{128}{13} \approx 10$$

Mapa cech Kohonena: zastosowanie w kompresji obrazów

Jeżeli zamiast obrazu 512 × 512 rozważymy obraz 1024 ×1024, wówczas kompresja wzrośnie do:

$$K_r = \frac{4\times 16K\times 4\times 4\times 8}{4\times 16K\times \log_2 512 + 0.5K\times 4\times 4\times 8} =$$

$$= \frac{(16\times 512)K}{(16\times 36+16\times 4)K} = \frac{16K\times 512}{16K\times 40} = \frac{512}{40} \approx 13$$

Mapa cech Kohonena: zastosowanie w kompresji obrazów

Natomiast jeżeli zamiast obrazu 512 × 512 rozważymy obraz 256 ×256, wówczas kompresja zmaleje do:

$$K_r = \frac{0.25 \times 16K \times 4 \times 4 \times 8}{0.25 \times 16K \times \log_2 512 + 0.5K \times 4 \times 4 \times 8} =$$

$$= \frac{(16 \times 32)K}{(16 \times 2.25 + 16 \times 4)K} = \frac{16K \times 32}{16K \times 6.25} = \frac{32}{6.25} \approx 5$$

Mapa cech Kohonena: zastosowanie w kompresji obrazów

Można zauważyć ponadto, iż dla bardzo dużych obrazów (lub serii mniejszych), współczynnik kompresji asymptotycznie dążyć będzie do wartości:

$$\lim_{N_r \to \infty} K_r = \lim_{N_r \to \infty} \frac{N_r n_x n_y T}{N_r \log_2 n + n n_x n_y t} = \frac{n_x n_y T}{\log_2 n}$$

Jeżeli nie zmienimy rozmiaru ramki, ani ilości neuronów wyjściowych, da to graniczną wartość równą:

$$K_r = \frac{4 \times 4 \times 8}{\log_2 512} = \frac{128}{9} = 14.22$$

Większą kompresję uzyskamy tylko poprzez zwiększenie rozmiaru ramki i/lub zmniejszenie liczby neuronów, co jednak pogorszy jakość odtwarzanych obrazów poniżej obecnej wartości PSNR≈26db