TUTORIAL OF WEEK 11

INT301 Bio-computation, Week 11, 2025

Example of Simple Competitive Learning

- **We are given 6 data samples:**

```
[0 1 1]       [1 1 0.5]       [0.2 0.2 0.2]
[0.5 0.5 0.5] [0.4 0.6 0.5]   [0 0 0]
```

- **3 ouput units each associated with a prototype vector.**
- **Learning rate: 0.5**
- **You are required to update the prototype vectors using Simple Competitive Learning (SCL) according to such upadating rule:**

$$\Delta w_{ji} = \eta y_j \left(x_i - w_{ji} \right)$$

$$\begin{cases} y_{j^*} = 1 \\ y_j = 0 \qquad if \qquad j \neq j^* \end{cases}$$

Example of Simple Competitive Learning

- **Question 1: The 3 prototype vectors are randomly initialized as:**

[0.14 0.75 0.71] [0.99 0.51 0.37] [0.73 0.81 0.87]

and the winner units are determined by minimizing the Euclidean distance. Please upadate the vectors one epoch and list the clustering result.

$$\Delta w_{ji} = \eta y_j \left(x_i - w_{ji} \right)$$

$$\begin{cases} y_{j^*} = 1 \\ y_j = 0 \quad if \quad j \neq j^* \end{cases}$$

Training on Input Vectors

Input vector # 1: [0.00 1.00 1.00]

Computing the distance beween Input vector to each prototype vector:

[0.14 0.75 0.71] [0.99 0.51 0.37] [0.73 0.81 0.87]

↓ 0.1662 ↓ 1.6171 ↓ 0.5859

Winning weight vector # 1: [0.14 0.75 0.71]

Updated winning weight vector:

[0.14 0.75 0.71]+0.5*([0.00 1.00 1.00]-[0.14 0.75 0.71])

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 2: [1.00 1.00 0.5]

Computing the distance beween Input vector to each prototype vector:

[0.07 0.87 0.85] [0.99 0.51 0.37] [0.73 0.81 0.87]

↓ ↓ ↓

1.0043 0.2571 0.2459

Winning weight vector # 3: [0.73 0.81 0.87]

Updated winning weight vector:

[0.73 0.81 0.87]+0.5*([1.00 1.00 0.5]-[0.73 0.81 0.87])

[0.07 0.87 0.85][0.99 0.51 0.37] **[0.87 0.90 0.69]**

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 3: [0.2 0.2 0.2]

Computing the distance beween Input vector to each prototype vector:

[0.07 0.87 0.85] [0.99 0.51 0.37] [0.87 0.90 0.69]

↓ ↓ ↓

0.8883 0.7491 1.1790

Winning weight vector # 2: [0.99 0.51 0.37]

Updated winning weight vector:

[0.99 0.51 0.37]+0.5*([0.2 0.2 0.2]-[0.99 0.51 0.37])

[0.07 0.87 0.85]**[0.59 0.36 0.29]** [0.87 0.90 0.69]

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 4: [0.5 0.5 0.5]

Computing the distance beween Input vector to each prototype vector:

[0.07 0.87 0.85] [0.59 0.36 0.29] [0.87 0.90 0.69]

↓ ↓ ↓

0.4443 0.0718 0.3330

Winning weight vector # 2: [0.59 0.36 0.29]

Updated winning weight vector:

[0.59 0.36 0.29]+0.5*([0.5 0.5 0.5]-[0.59 0.36 0.29])

[0.07 0.87 0.85]**[0.55 0.43 0.39]** [0.87 0.90 0.69]

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 5: [0.4 0.6 0.5]

Computing the distance beween Input vector to each prototype vector:

[0.07 0.87 0.85] [0.55 0.43 0.39] [0.87 0.90 0.69]

↓ ↓ ↓

0.3043 0.0635 0.3470

Winning weight vector # 2: [0.59 0.36 0.29]

Updated winning weight vector:

[0.55 0.43 0.39]+0.5*([0.4 0.6 0.5]-[0.55 0.43 0.39])

[0.07 0.87 0.85]**[0.47 0.51 0.45]** [0.87 0.90 0.69]

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 6: [0.0 0.0 0.0]

Computing the distance beween Input vector to each prototype vector:

[0.07 0.87 0.85] [0.47 0.51 0.45] [0.87 0.90 0.69]

↓ ↓ ↓

1.4843 0.6835 2.0430

Winning weight vector # 2: [0.47 0.51 0.45]

Updated winning weight vector:

[0.47 0.51 0.45]+0.5*([0.0 0.0 0.0]-[0.47 0.51 0.45])

[0.07 0.87 0.85]**[0.24 0.26 0.22]** [0.87 0.90 0.69]

Finish of Epoch 1

Example of Simple Competitive Learning

```
Clusters after epoch 1:
 Weight vector # 1: [ 0.07  0.87  0.85 ]
    Input vector # 1:   [ 0.00  1.00  1.00 ]
 Weight vector # 2: [ 0.24  0.26  0.22 ]
    Input vector # 3:   [ 0.20  0.20  0.20 ]
    Input vector # 4:   [ 0.50  0.50  0.50 ]
    Input vector # 5:   [ 0.40  0.60  0.50 ]
    Input vector # 6:   [ 0.00  0.00  0.00 ]
 Weight vector # 3: [ 0.87  0.90  0.69 ]
    Input vector # 2:   [ 1.00  1.00  0.50 ]
```

Example of Simple Competitive Learning

- **Question 2: Based on the 3 prototype vectors initialized in question 1 and the winner units are determined by maximizing inner product. Please upadate the vectors one epoch and list the clustering result again.**

Training on Input Vectors

Input vector # 1: [0.00 1.00 1.00]

Computing the distance beween Input vector to each prototype vector:

[0.14 0.75 0.71] [0.99 0.51 0.37] [0.73 0.81 0.87]

↓ ↓ ↓

1.4600 0.8800 1.6800

Winning weight vector # 3: [0.73 0.81 0.87]

Updated winning weight vector:

[0.73 0.81 0.87]+0.5*([0.00 1.00 1.00]-[0.73 0.81 0.87])

[0.14 0.75 0.71][0.99 0.51 0.37] **[0.37 0.91 0.94]**

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 2: [1.00 1.00 0.5]

Computing the distance beween Input vector to each prototype vector:

[0.14 0.75 0.71] [0.99 0.51 0.37] [0.37 0.91 0.94]

↓ ↓ ↓

1.2450 1.6850 1.7500

Winning weight vector # 3: [0.37 0.91 0.94]

Updated winning weight vector:

[0.37 0.91 0.94]+0.5*([1.00 1.00 0.5]-[0.37 0.91 0.94])

[0.14 0.75 0.71][0.99 0.51 0.37] **[0.69 0.96 0.72]**

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 3: [0.2 0.2 0.2]

Computing the distance beween Input vector to each prototype vector:

[0.14 0.75 0.71] [0.99 0.51 0.37] [0.69 0.96 0.72]

↓ ↓ ↓

0.3200 0.3740 0.4740

Winning weight vector # 3: [0.69 0.96 0.72]

Updated winning weight vector:

[0.69 0.96 0.72]+0.5*([0.2 0.2 0.2]-[0.69 0.96 0.72])

[0.14 0.75 0.71][0.99 0.51 0.37] **[0.45 0.58 0.46]**

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 4: [0.5 0.5 0.5]

Computing the distance beween Input vector to each prototype vector:

[0.14 0.75 0.71] [0.99 0.51 0.37] [0.45 0.58 0.46]

↓ ↓ ↓

0.8000 0.9350 0.7450

Winning weight vector # 2: [0.99 0.51 0.37]

Updated winning weight vector:

[0.99 0.51 0.37]+0.5*([0.5 0.5 0.5]-[0.99 0.51 0.37])

[0.14 0.75 0.71]**[0.71 0.51 0.44]** [0.45 0.58 0.46]

Example of Simple Competitive Learning

Training on Input Vectors

Input vector # 5: [0.4 0.6 0.5]

Computing the distance beween Input vector to each prototype vector:

[0.14 0.75 0.71] [0.71 0.51 0.44] [0.45 0.58 0.46]

↓ ↓ ↓

0.8610 0.8100 0.7580

Winning weight vector # 1: [0.14 0.75 0.71]

Updated winning weight vector:

[0.14 0.75 0.71]+0.5*([0.4 0.6 0.5]-[0.14 0.75 0.71])

[0.27 0.68 0.61][0.71 0.51 0.44] [0.45 0.58 0.46]

Example of Simple Competitive Learning

Training on Input Vectors
Input vector # 6: [0.0 0.0 0.0]
Computing the distance beween Input vector to each prototype vector:
[0.27 0.68 0.61] [0.71 0.51 0.44] [0.45 0.58 0.46]

0.0000 0.0000 0.0000

Winning weight vector # 1: [0.27 0.61 0.61]
Updated winning weight vector:
[0.27 0.68 0.61]+0.5*([0.0 0.0 0.0]-[0.27 0.68 0.61])
[0.14 0.34 0.31][0.71 0.51 0.44] [0.45 0.58 0.46]

Finish of Epoch 1

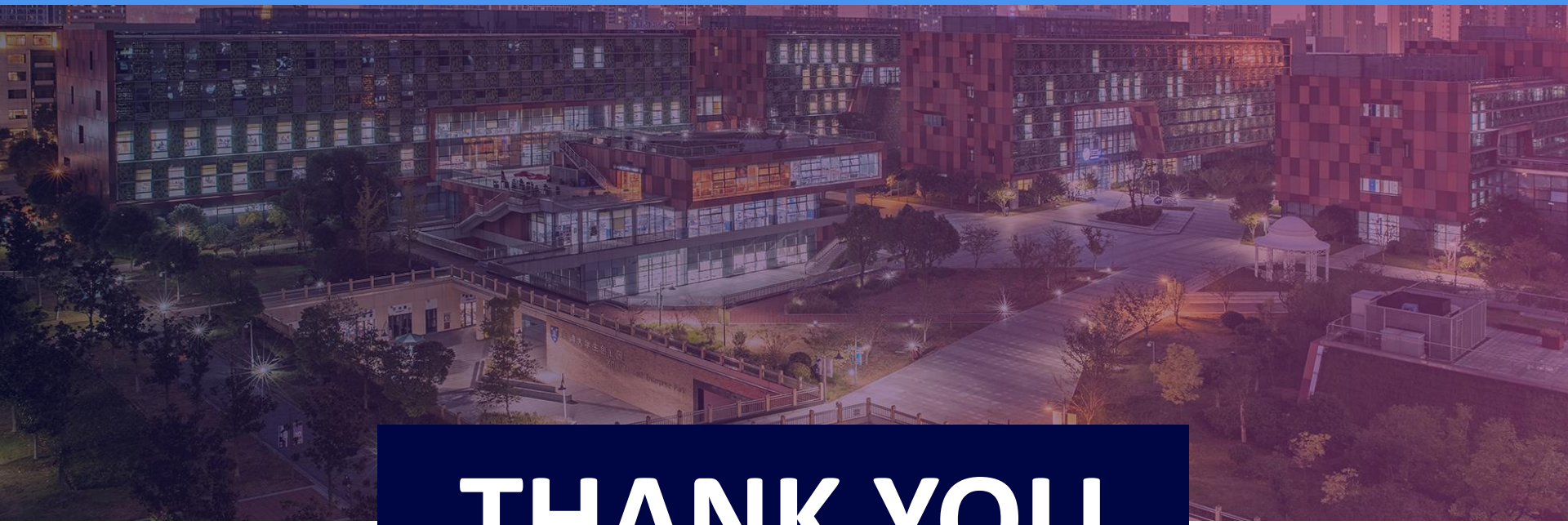
Example of Simple Competitive Learning

```
Clusters after epoch 1:
 Weight vector # 1: [ 0.14  0.34  0.31 ]
    Input vector # 5:   [ 0.40  0.60  0.50 ]
    Input vector # 6:   [ 0.00  0.00  0.00 ]
 Weight vector # 2: [ 0.71  0.51  0.44]
    Input vector # 4:   [ 0.50  0.50  0.50 ]
 Weight vector # 3: [ 0.45  0.58  0.46 ]
    Input vector # 1:   [ 0.00  1.00  1.00 ]
    Input vector # 2:   [ 1.00  1.00  0.50 ]
    Input vector # 3:   [ 0.20  0.20  0.20 ]
```

Summary

- SCL is sensitive to the choice of winning unit criterion
- Different similarity measures lead to different final prototypes

Will altering the presentation order of the training samples produce different update trajectories and hence different converged vectors? 会的，通常收敛到局部最优解



THANK YOU

VISIT US

WWW.XJTLU.EDU.CN

FOLLOW US

@XJTLU

Xi'an Jiaotong-Liverpool University
西交利物浦大學