

가변적 output layer를 이용한 LVQ 기반 물체 분류

김헌기, 조성원, 김재민, 이진형, 김석호
 홍익대학교 전기 정보 제어 공학과

Object Classification Based on LVQ with Flexible output layer

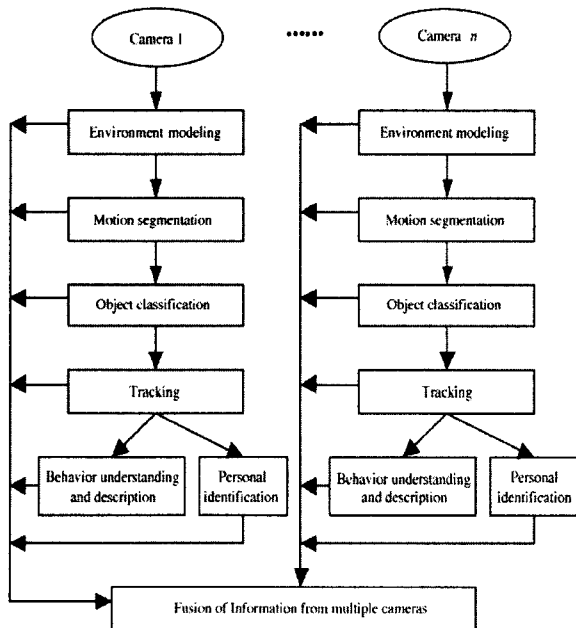
Hunki Kim, Seongwon Cho, Jaemin Kim, Jinhyung Lee, Seokho Kim
 Department of Electrical, Information and Control Engineering, Hongik University

Abstract - In this paper, we present a new method for classifying object using LVQ (Learning Vector Quantization) with flexible output layer. The proposed LVQ is a supervised learning method that dynamically generates output neurons and initializes automatically the weight vectors from training patterns. If the classes of the nearest output neuron is different from the class of the training pattern, a new output neuron is created and the given training pattern is used to initialize the weight vector of the created neuron. The proposed method is significantly different from the previous competitive learning algorithms in the point that the output neurons are dynamically generated during the learning process.

1. 서 론

최근 영상을 이용한 물체 추적은 컴퓨터 비전 분야에서 활발하게 연구되는 주제 중 하나이다. 물체 추적의 과정을 크게 구분하면 움직임 검출, 검출된 물체 분류, 물체 추적, 물체의 행동 분석, 다중 채널 카메라에서 들어오는 영상 처리 결과 결합 등으로 나눌 수 있다. 그림1은 이러한 영상 감시의 흐름을 나타내고 있다[1].

검출된 물체를 크게 분류하면 사람, 자동차, 사람그룹, 클러스터로 구분할 수 있다. 클러스터에는 동물, 가방 등 신원을 확인할 수 없는 물체를 포함한다. 물체를 분류하는 방법에는 크게 물체의 형태 기반의 분류 방법과 물체의 움직임 기반의 분류 방법으로 나눌 수 있다. 신경회로망의 한 방법인 LVQ [2],[3]는 형태 기반의 분류 방법으로 데이터의 학습이 빠르고 연산량이 적어 실시간으로 물체를 분류할 수 있는 장점이 있다. 하지만 데이터의 훈련 시 output layer의 개수를 정확히 예측할 수 없고 output layer의 개수에 따라 물체를 분류하는 정확도가 매우 달라질 수 있다. 그러므로 본 논문에서는 output layer의 개수를 데이터의 특성에 맞게 결정해주는 알고리즘을 제시한다.



〈그림 1〉 일반적인 영상 추적의 구성도

2. 본 론

2.1 가변적 output layer를 이용한 LVQ

2.1.1 Learning Vector Quantization

기존 LVQ의 훈련 방법은 샘플 데이터를 직접 가중치 벡터(Weight Vector)의 초기 값으로 이용하고 샘플과 각각 가중치 벡터와의 거리(Euclidean Distance)를 구하여 가장 거리가 짧은 벡터를 워너(Winner)로 설정하고 워너로 결정된 벡터의 부류(Class)와 샘플의 부류의 상이를 판단하여 가중치 벡터를 업데이트시키는 방법이다. 이러한 LVQ의 훈련 방법은 샘플 데이터 중에서 가중치 벡터의 초기 값을 정할 때 각각의 부류를 잘 대표할 수 있는 샘플의 선택 여부에 따라 인종률이 달라질 수 있고, 같은 부류의 샘플 중에서도 각기 다른 모양이 있을 수 있기 때문에 같은 부류의 샘플 중에서 몇 개의 output layer를 설정하느냐에 따라 인종률이 달라질 수 있다. 그러므로 각각의 데이터의 특성에 따라 output layer의 개수는 달라지고 가장 최적화된 output layer의 개수를 결정하는 것은 LVQ 훈련 방법의 가장 중요한 문제이다. 본 논문에서 제안하는 방법은 LVQ 훈련 시 output layer의 개수를 데이터의 특성에 맞게 결정해주는 방법이다.

2.1.2 학습알고리즘

DLVQ(Dynamic Learning Vector Quantization) 알고리즘은 가중치 벡터의 초기화, 가중치 벡터의 업데이트, 새로운 가중치 벡터의 생성 등으로 구성되어 있다. DLVQ가 기존 LVQ와 다른 점은 데이터 훈련 시 가중치 벡터의 개수를 미리 결정하는 것이 아니라 데이터 훈련과 함께 가중치 벡터를 생성한다는 것이다. 이러한 훈련 알고리즘을 살펴보면,

Step 0. 각각 변수의 초기화

$$t = 0, n = 0$$

$$\alpha(0) \in (0, 1)$$

$$n_j = 0 \text{ for } j = 1, 2, \dots, J$$

Step 1. 정지 조건이 만족되지 않을 경우 Steps 2-8을 계속 반복

Step 2. 샘플 데이터를 각각 입력 벡터 x_t 로 하여 Steps 3-7을 수행

Step 3. $r=1, \dots, k$ 에 대하여

a. Cr을 찾음

$$\|x_j - w_{c_r}\| = \min_{c_1, \dots, c_r} \|x_j - w_{c_r}\|$$

b. 만약 $u_{c_r} = j$ 라면, 가중치 벡터를 업데이트 하고 Step 5를 실행
 $w_{c_r}(new) = w_{c_r}(old) + \alpha(t) \cdot [x_j - w_{c_r}(old)]$

c. r을 다음 숫자로 넘김

Step 4. 만약 $u_{c_r} \neq j$ 이라면 새로운 output layer를 만들고 가중치 벡터를 초기화한다.

$$n = n + 1, n_j = n_j + 1$$

$$w_n(new) = x_j, u_n = j$$

Step 5. 각각 변수의 초기화

Step 6. 학습률 $\alpha(t)$ 를 업데이트

Step 7. t를 다음 숫자로 넘김

Step 8. 정지 조건을 판별 : 미리 정해진 회수로 결정

t : 현재 학습 시간

n : 생성된 output layer의 전체 개수

J : 분류될 총 개수

j : 부류의 번호 $1 \leq j \leq J$

I : output layer의 번호 $1 \leq i \leq n$

x_j : 부류 j에 속해있는 입력 벡터

k : 최종 output layer의 번호

n_j : 부류 j에 속한 output neuron의 개수

w_i : output layer i에 대한 가중치 벡터

u_i : output layer i의 워너 부류

u : 모든 output layer의 부류로 구성된 벡터

$$u = [u_1, u_2, \dots, u_n]^T$$

c_i : 가장 거리가 짧은 output layer
 $r(t)$: t에서의 학습률

DLVQ 알고리즘은 SCL(Simple Competitive Learning)[4], FSCL(Frequency Sensitive Competitive Learning)[5],[6]과 같은 기존 LVQ 학습 방법과 달리 훈련 도중 입력 벡터에 대한 위너 벡터의 부류가 입력 벡터의 부류와 같으면 업데이트 하고 입력 벡터의 부류와 같지 않으면 새로운 가중치 벡터로 생성하는 방법이다. 이러한 방법을 사용하면 같은 부류에 속한 입력 벡터들 중에서 가중치 벡터로 결정되는 벡터의 수를 데이터에 따라 다르게 설정할 수 있고, 전체 가중치 벡터의 수도 데이터의 특성에 맞게 결정할 수 있다.

2.2 실험 결과

본 실험은 기존의 LVQ 방법과 제안하는 알고리즘을 사용한 방법을 비교하기 위하여 실행하였다. 실험을 위한 샘플 데이터는 비디오 영상의 각 프레임에서 추출한 물체의 이미지를 사용하였다. 각 부류당 400개의 영상을 사용하였고, 1000개의 결과 확인 영상을 사용하여 총 1200개의 샘플 영상과 3000개의 결과 확인 영상을 사용하였다.



〈그림 2〉 실험 영상 (a) 차 (b) 사람 (c) 사람그룹

DLVQ를 이용하여 훈련한 결과 약 330개의 output layer가 나왔다. 공정한 비교를 위하여 기존 LVQ 훈련 시 output layer의 개수를 330개로 설정하였고 입력 차원(Input Dimension)은 256(16x16 사이즈로 정규화)으로 설정하였다. 두 방법 모두 같은 학습률을 적용하였다.

$$\alpha(t) = 0.3 \cdot \left(1 - \frac{t}{\text{전체 회수}}\right)$$

표1과 표2는 DLVQ와 기존 LVQ를 이용하여 훈련된 결과를 나타내고 있다. DLVQ의 경우 layer의 개수는 전체 샘플 데이터에 대한 회수에 따라 다소 차이가 있지만 약 330개의 layer가 생성되었고, 훈련(%)은 샘플 데이터를 직접 인증 데이터로 사용하여 인증하였을 경우 인증에 성공한 백분율이고, 차(%), 사람(%), 사람그룹(%)은 샘플 데이터가 아닌 결과 인증 데이터를 사용하여 인증하였을 경우 인증에 성공한 백분율이다. 표에서 확인한 것과 같이 훈련(%), 차(%), 사람(%)의 경우 각각 2.06%, 2.39%, 2.86% 정도로 기존 LVQ 방법에 비해 높은 인증률을 보였다. 하지만 사람 그룹의 경우 다른 객체의 인증률보다 낮고 오히려 2.60% 정도의 낮은 인증률을 보였다. 이는 사람 그룹의 경우 두 사람의 밀착 정도나 사람의 수 등 변수가 많기 때문에 두 사람이 너무 밀착될 경우 한 사람으로 오인하거나 또는 너무 많은 사람이 영상에 잡혀서 차로 오인한 결과이다.

3. 결 론

본 논문에서는 새로운 방식의 훈련 방법을 사용하는 DLVQ를 이용하여 물체를 분류하는 방법에 대하여 제안하였다. DLVQ는 데이터의 특성에 맞게 output layer의 개수를 결정해주는 방법으로 기존의 LVQ 방법에 비해 인증률이 매우 높고 데이터의 훈련 속도가 빠를 뿐만 아니라 인증 과정에서도 실시간으로 처리할 수 있는 장점이 있다. 그러나 샘플 데이터와 실제 인증 시 사용되는 데이터의 차이가 심할 경우 인증률이 떨어질 수 있다. 그리고 클러스터의 모양이 사람이나 차의 모양과 매우 흡사하거나 영상의 노이즈가 검출될 경우 인증률이 떨어질 수 있다. 그러므로 여러 상황에 대해서도 인증률을 높게 만들기 위하여 대표 샘플을 되도록 많이 확보해야 하고 물체의 형태 분류뿐만 아니라 움직임도 이용한 분류도 같이 병행하여 인증률을 높일 필요가 있다.

〈표 1〉 DLVQ의 인증률

회수	layer의 개수	훈련(%)	차(%)	사람(%)	사람그룹(%)
50	322	100	98.23	97.30	84.00
60	333	100	98.42	96.68	83.20
70	327	100	98.55	96.49	84.27
80	339	100	98.55	96.68	84.27
90	332	100	98.48	97.37	83.13
100	326	100	98.36	97.23	83.27
평균	329.83	100	98.43	96.96	83.69

〈표 2〉 기존 LVQ의 인증률

회수	layer의 개수	훈련(%)	차(%)	사람(%)	사람그룹(%)
50	330	97.83	96.27	93.85	85.81
60	330	98.25	96.15	94.87	85.01
70	330	97.91	95.77	94.39	88.04
80	330	98.32	96.59	95.15	87.97
90	330	97.37	95.89	93.91	83.74
100	330	97.93	95.58	92.41	87.16
평균	330	97.94	96.04	94.10	86.29

〈참 고 문 헌〉

- [1] Weiming Hu, Tieniu Tan, Liang Wang, Maybank, S, "A survey on visual surveillance of object motion and behaviors", in Systems, Man and Cybernetics, Part C, IEEE Transactions on , pp. 334 - 352, 2004
- [2] Kohonen, T.: Learning Vector Quantization. Neural Networks 1(1) 303, 1988
- [3] Fausett, L.: Fundamentals of Neural Networks. Prentice Hall, 1994
- [4] Rummelhart, D.E., McClelland, J.L.: Parallel Distributed Processing. In: Rummelhart, D.E., Zipser, D. (eds.): Feature Discovery by Competitive Learning. MA: MIT Press, Cambridge, 1988
- [5] Ahalt, S.C., Krishnamurthy, A.K., Chen, P., and Melton, D.E.: Competitive Learning Algorithms for Vector Quantization. Neural Networks 3, 277-290, 1990
- [6] DeSieno, D.: Adding a Conscience to Competitive Learning. Proc. IEEE International Conference on Neural Networks, SanDiego,California, I 117-II24, 1990