

Development of a Multi-criteria Pedestrian Pathfinding Algorithm by Perceptron Learning

Kyeonah Yu*, Chojung Lee**, Inyoung Cho***

Abstract

Pathfinding for pedestrians provided by various navigation programs is based on a shortest path search algorithm. There is no big difference in their guide results, which makes the path quality more important. Multiple criteria should be included in the search cost to calculate the path quality, which is called a multi-criteria pathfinding. In this paper we propose a user adaptive pathfinding algorithm in which the cost function for a multi-criteria pathfinding is defined as a weighted sum of multiple criteria and the weights are learned automatically by Perceptron learning. Weight learning is implemented in two ways: short-term weight learning that reflects weight changes in real time as the user moves and long-term weight learning that updates the weights by the average value of the entire path after completing the movement. We use the weight update method with momentum for long-term weight learning, so that learning speed is improved and the learned weight can be stabilized. The proposed method is implemented as an app and is applied to various movement situations. The results show that customized pathfinding based on user preference can be obtained.

▶ Keyword: Path searching algorithm, Multi-criteria pathfinding, Perceptron learning, Customized pathfinding

I. Introduction

수많은 내비게이션 장치들과 포털 사이트가 제공하는 길찾기 기능은 각기 이용하는 데이터에 따라 품질의 차이가 있긴 하지만 기본적으로는 최단 경로 탐색 알고리즘에 기반하고 있다. 여기서 ‘최단’은 물리적으로 가장 짧은 것을 의미하지는 않고 시간이나 그 밖에 여러 종류의 비용이 기준이 될 수 있어 이를 다기준 길찾기(multi-criteria pathfinding)라고 하며 여러 기준들을 탐색 알고리즘의 비용함수로 정의하고 최소화하는 방향으로 경로를 탐색한다. 예를 들어 차량용 경로안내의 경우 물리적인 거리와 시간이외에 도로의 종류, 통행료, 회전 여부 등이 길찾기에 고려되며 도보용 길안내의 경우에는 계단 존재 여부, 도로의 종류 여부 등이 길찾기에 고려된다. 이와 같은 요소들은 우선순위 방식이나 가중 점수를 부여하는 방식으로

프로그램되어 길찾기 결과에 반영되지만 사용하고 있는 사람의 특성에 따라 요소의 반영을 자동으로 조정하는 길찾기 프로그램은 아직 상용화되지 않고 있다. 본 논문에서는 GPS 시스템에서 사용자의 선호도를 자동으로 학습하여 같은 제품의 길찾기 프로그램이라도 사용하는 사람에 따라 다른 길을 안내하게 되는 사용자 맞춤형 길찾기 알고리즘을 소개한다.

본 논문에서는 다기준 길찾기를 위한 비용함수를 여러 가지 기준들(criteria)의 가중치합으로 정의한다. 이와 같이 하면 가중치의 크기로 요소들의 중요도를 표현해 낼 수 있으며 단층 신경망회로에서 도출된 퍼셉트론 학습이라는 잘 정립된 이론으로 가중치를 학습할 수 있다는 장점이 있다. 학습이 탐색에 통합되어 동작하므로 기준을 평가하기 위해 많은 조건문을 거쳐

• First Author: Kyeonah Yu, Corresponding Author: Kyeonah Yu
*Kyeonah Yu (kyeonah@duksung.ac.kr) Dept. of Computer Science, Duksung Women's University
**Chojung Lee (chwjdfjqm@naver.com) Dept. of Computer Science, Duksung Women's University
***Inyoung Cho (jiyinyoung@naver.com) Dept. of Computer Science, Duksung Women's University
• Received: 2017. 09. 22, Revised: 2017. 10. 19, Accepted: 2017. 11. 14.
• This work was supported by 2016 Duksung Women's Univ. Research Grant.

야하는 별도의 과정이 필요하지 않을 뿐 아니라 새로운 기준을 용이하게 추가할 수 있는 확장성을 제공한다.

탐색 알고리즘의 결과 경로를 관찰된 출력으로, 사용자가 실제 택한 경로가 목표 출력으로 설정하고, 이 두 출력이 다르면 가중치 학습이 일어나도록 한다. 가중치 학습은 단기(short-term)와 장기(long-term)의 두 가지 방식으로 구현한다. 단기 가중치 학습은 사용자가 실제 이동하는 중에 일어나는 모든 탐색 오차에 대해 가중치 갱신을 하고 실시간으로 경로 탐색에 반영하는 것이며 장기 가중치 학습은 이동이 종료된 후에 전체 경로에 대한 평균값으로 가중치를 갱신하는 것이다. 실제 사용자 프로파일에는 장기 가중치 학습 결과만 반영하는데 사용자의 특정 선택에 대해 민감하게 가중치를 갱신하는 것은 바람직하지 않기 때문에 본 논문에서는 모멘텀을 이용한 가중치 갱신 방법 [1]을 이용할 것을 제안한다. 제안한 방법은 [2]에서 제시한 가이드라인에 따라 사용자의 안전과 도로의 특성을 포함한 보행자용 길찾기 앱으로 구현되어 서울시에서 제공한 동작구 CCTV 위치 정보[3]와 티맵[4]에서 제공한 도로 정보 및 좌표 정보를 가공하여 서로 다른 선호도를 갖는 사용자들에 대해 테스트하며 효율적으로 사용자에게 적용됨을 보인다.

본 논문의 구성은 2장에서는 내비게이션 길찾기와 다기준 길찾기에 대한 연구 동향을 살펴보고 3장에서는 다기준 길찾기 문제를 정의하고 가중치 학습을 위한 수정된 퍼셉트론 학습에 의한 가중치 갱신 방법을 소개한다. 4장에서는 구현된 앱을 소개하고 에뮬레이터를 통해 제안된 알고리즘을 적용한 결과를 제시하며 5장에서 향후 과제에 대한 제안과 함께 맺는다.

II. Related works

GPS 시스템은 맵과 길찾기 알고리즘의 두 가지 구성요소로 이루어진다. GPS기반 길찾기에서 맵을 다루는 방법은 노드와 링크로 이루어진 방향성 있는 그래프이다. 이 경우, 기존의 로봇 경로계획이나 게임에서의 길찾기 등과의 가장 큰 차이점은 광범위한 스케일과 조밀한 세그먼트로 인해 엄청난 수의 노드를 다루기 때문에 많은 문헌들이 이를 효과적으로 해결하는데 초점을 맞추고 있다. [5]와 [6]에서는 최단 경로 알고리즘을 가장 빠르게 돌리기 위한 데이터 구조 및 실행 절차를 비교 제시하였다. [7]에서는 내비게이션 길찾기의 특성이 맞게 광역 수준에서 좁은길까지 계층적 길찾기를 통해 문제의 복잡도를 줄이고 동적인 환경에도 효율적으로 대응할 수 있는 방법을 제안하였으며 많은 노드수를 다루는 네트워크의 라우팅 알고리즘을 응용하여 적용하는 방식을 제안하기도 하였다 [8]. [9]에서는 전처리 과정을 통해 경로를 미리 계산하고 저장하여 실시간 탐색없이 빠르게 길찾기 안내를 하는 방법을 제안하였다.

GPS 길찾기의 또 하나의 주된 연구 분야는 비용함수를 다양하는 다기준 길찾기(multi-criteria pathfinding) 방법으로 추천 경로의

품질을 높이는데 관한 연구이다. [10]에서는 A* 알고리즘이나 다익스트라 알고리즘을 이용하여 공사장 두 지점 사이의 경로를 탐색하는데 기존의 최단 경로이외에 안전(safety) 최적화, 가시성(visibility) 최적화 등의 기준을 충족할 수 있는 방법을 수학적으로 제시하였으나 사용에 따라 적응적으로 변하지 않는다. [11]에서는 본 연구에서 제안하려는 방법과 가장 근접한 접근 방식을 보여 준다. 사용자는 툴박스를 이용하여 선호도를 입력하게 되어 있으며 이를 기반으로 생성된 가상의 경로에 대해 사용자의 피드백을 받고 뉴로 퍼지 모델 방식으로 분류 기준을 정하게 된다. 본 논문에서는 사용자의 명백한 피드백이 없이도 실행중에 학습이 일어날 수 있으며 학습에 앞서 미리 경로들을 분류할 필요성도 없다는 장점이 있다. [12]와 [13]은 보행자 길찾기 문제를 다루었다. [12]에서는 보행자의 특성을 커버하기 위해 도로가 아닌 오픈 공간에 대한 분석을 통해 그래프를 확대하는 방법을 제시하였으며 [13]에서는 동적으로 경로를 선택하는 방법을 제시하였으나 실시간 학습 방법을 제시하지는 않았다.

III. Design of Multi-Criteria Path-Finding Algorithm

기존의 길찾기 문제에 경로의 품질을 결정하는 기준들을 추가하고 각 기준에 대한 가중치를 할당하여 다기준 길찾기 문제를 정의하고 사용자 맞춤형 길찾기를 위해 가중치 튜닝 알고리즘을 소개한다.

3.1 Definition of Multi-Criteria Pathfinding Problem

그림 1과 같이 방향 그래프로 표현된 맵에서 다기준 길찾기 문제는 다음과 같이 정의된다.

$$(G, C, W, f, n_s, n_g) \rightarrow P(n_1 = n_s, n_2, \dots, n_l = n_g), l \geq 0$$

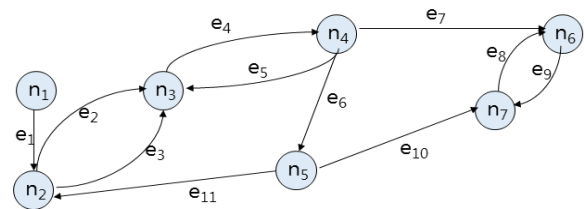


Fig. 1. Directed Graph
 $G = (N, E), N = \{n_i | 1 \leq i \leq 7\}, E = \{e_i | 1 \leq i \leq 11\}$

$G = (N, E)$ 는 방향 그래프, N 은 노드 집합, $E \subseteq N \times N$ 는 링크 집합이다.

C 는 기준(criteria)의 집합이다. $k = |C|$ 라고 할 때, $R_+^k \ni C_i = (c_{i1}, \dots, c_{ik})$ 는 i 번째 링크의 비용을 정의하는 벡터이다. 길찾기의 가장 기본인 거리 기준은 적은 값일수록 좋은 값이므로 다른 모든 기준들도 이에 맞추어 스케일과 함께 정규화시킨다.

W 는 C 에 해당하는 가중치 벡터이므로 $|W|=k$ 이다.

f 는 경로 탐색에 사용되는 비용함수로 $f(n)=g(n)+h(n)$ 으로 정의되며 여기서 $g(n)$ 은 시작 노드에서 노드 n 까지 실제 소요된 비용을 말하며 $h(n)$ 은 노드 n 부터 목표 노드까지의 추정치이다. $h(n)$ 은 A* 알고리즘의 허용성(admissibility)을 유지하기 위해 노드 n 부터 목표 노드까지의 직선거리를 사용하고 $g(n)$ 은 여러 기준들을 포함하여 다음과 같이 정의한다.

링크의 비용함수 $\phi: E \rightarrow R_+^k$ 를 정의하면 $\phi(e_i)=\phi(n_i, n_{i+1})=(w_1 \cdot c_{i1}, \dots, w_k \cdot c_{ik}), \sum_j w_j = 1$ 이

고, $g(n_i)=\sum_{i=1}^{i-1} \phi(e_i)$ 이 된다.

n_s 와 n_g 는 각각 시작노드와 목표노드를 나타낸다.

$P(n_1=n_s, n_2, \dots, n_l=n_g), l \geq 0$ 는 결과 경로를 나타낸다.

3.2 Weight Tuning by Perceptron Learning

단기 가중치 학습은 사용자가 실제 이동하는 중에 실시간으로 학습되어 경로 탐색에 반영된 후에 리셋되며, 장기 가중치 학습은 이동이 종료된 후에 사용자 프로파일에 반영할 가중치 학습으로, 전체 경로에 대한 평균값으로 가중치를 갱신하는 것이다.

3.2.1 Short-term Weight Learning

사용자가 실제 이동을 시작할 때에는 현재의 가중치로 계산된 이동 루트 $P(n_1=n_s, n_2, \dots, n_l=n_g), l \geq 0$ 가 제공되고 제안된 루트를 벗어나면 가중치 학습이 일어나며 알고리즘은 그림 2와 같다.

```

input:  $P(n_1=n_s, n_2, \dots, n_l=n_g)$ 
 $i=1$ ;
while (t) {
  if ( $n_i \equiv n_g$ ) then halt;
   $e_i=(n_i, n_{i+1})$ ;
   $e_i^* \leftarrow$  edge taken by the user;
  if ( $e_i \neq e_i^*$ ) then // planned path  $\neq$  user path
  {
     $W \leftarrow$  updateWeight( $W, e_i, e_i^*$ );
     $x \leftarrow$  successor of  $e_i^*$ ;
     $P \leftarrow A^*(G, C, W, f, x, n_g)$ ;
     $i=1$ ;
    continue;
  }
  else {
     $i=i+1$ ;
    continue;
  }
}

```

Fig. 2. Weight Learning Algorithm

updateWeight(W, e_i, e_i^*)는 계산된 출력값과 목표값의 차이를 최소화시키는 단층 퍼셉트론의 학습 규칙으로 계산한다.

$$W = W + \alpha [C(e_i) - C(e_i^*)] \quad (1)$$

바람직한 노드를 의미하는 e_i^* 가 현재 계산된 e_i 와 같지 않

을 때만 가중치 수정이 이루어지기 때문에 식 (1)은 바람직한 노드의 기준 함수 값은 낮추고 관찰된 노드들의 기준 함수 값은 높이는 방향으로 가중치가 수정됨을 의미한다. 선행 노드까지는 $g(n)$ 값이 같았다는 의미이므로 i 번째 링크의 기준 벡터만으로 가중치를 갱신하면 된다. α 는 학습 속도를 나타내는 인자이다. α 가 크면 가중치의 변화가 크게 일어나 학습 중단 조건을 만족하지 못하는 경우가 발생하고 너무 작으면 가중치의 변화가 너무 더딘 단점이 있는데 학습을 위한 적절한 α 는 실험적으로 정하도록 한다.

3.2.2 Long-term Weight Learning

사용자의 이동 경로를 $P^*(n_s, n_2^*, \dots, n_g)$ 라 하고 현재의 가중치로 탐색한 경로를 $P(n_s, n_2, \dots, n_g)$ 라고 할 때, 장기 가중치 학습에서는 두 경로를 구성하는 모든 링크들의 기준(criteria)의 합계를 델타 규칙에 적용한다 (식 (2)). α 는 앞 절에서 설명한 바와 같이 학습 속도를 나타내는 인자이며, $(|P|-1)$ 은 현재 가중치로 탐색한 경로의 링크 수, $(|P^*|-1)$ 은 사용자 이동 경로의 링크 수이다.

$$\Delta W = \alpha \left[\sum_{i=1}^{(|P|-1)} C(e_i) - \sum_{i=1}^{(|P^*|-1)} C(e_i^*) \right] \quad (2)$$

$$M_t = \Delta W + \beta M_{t-1}, 0 \leq \alpha, \beta \leq 1 \quad (3)$$

$$W_{t+1} = W_t + M_t \quad (4)$$

식 (3)과 (4)는 모멘텀(M)을 델타 규칙에 반영하는 식이다. 모멘텀의 원래 의미는 물체가 한 방향으로 지속적으로 변동하려는 경향인데 델타 규칙에서는 과거의 가중치 갱신값들을 추가로 더해 가중치 벡터값이 바로 바뀌지 않고 어느 정도 일정 방향을 유지하면서 변화도록 하는 것을 의미한다 [14]. 모멘텀이 가중치 학습에 도움이 되는 이유는 사용자의 한 번의 선택에 따라 즉각 반응하게 되면 가중치의 진동(oscillation)이 심해져 수렴하는 속도가 늦어지게 되는데 모멘텀의 추가로 학습의 안정성이 높아지기 때문이다. β 는 모멘텀 반영 인자인데 β 값을 초기에는 작은 값으로 설정, 즉 모멘텀의 비중을 작게 하여 학습이 빠르게 일어나도록 하고 학습이 어느 정도 진행된 후에는 크게 하여 사용자가 여러 번 다른 선택을 해야 가중치가 바뀌도록 안정성을 제공하고자 한다.

IV. Multi-Criteria Path-Finding App and Simulation Results

경로의 거리뿐만 아니라 도로의 종류 및 안전성을 고려하는 다기준 보행자 앱을 구현하고 다양한 선호도의 사용자들에 의해 직접 테스트된 후, 앱의 결과가 어떻게 달라지는가를 안드로

이드 에뮬레이터로 실험한다.

4.1 Multi-criteria Pedestrian Pathfinding App

이동 거리와 안전성, 도로의 종류를 고려한 보행자 길찾기 앱을 안드로이드 기반으로 제작하였다. 경로 탐색 알고리즘과 가중치 학습 알고리즘은 자바 프로그래밍 언어로 작성하여 다른 리소스들과 함께 패키지로 통합하였다. 사용한 리소스들을 정리하면 다음과 같다.

- T-map: 지도, 도로와 노드 정보
- 서울시 오픈 데이터: CCTV 위치 정보
- 다음 오픈 API: 목적지 주소 검색

노드 정보와 CCTV 위치 정보는 위도와 경도 쌍으로 제공되었으며 도로 정보는 도로의 종류를 간소화시켜 0~3까지의 값을 갖는 연결 그래프로 제공되었다. 앱의 기능은 크게 두 가지이다.

1. 사용자가 목적지를 입력하면 현재 위치로부터 목적지까지의 현재의 가중치로 탐색한 경로를 화면에 붉은 선으로 표시한다. 그 밖의 기준 관련 정보들도 각기 다양한 방법으로 화면상에 제공한다. 예를 들어 CCTV는 각 해당하는 위치에 카메라 아이콘으로 표시되고 도로에 대한 정보는 지도에서 직접 확인할 수 있다. (그림 3)
2. 사용자는 경로와 기타 정보를 참고하여 이동하는데 선택한 경로가 제안한 경로와 다른 경우에는 가중치 학습을 수행하고 새로운 가중치로 경로를 재탐색하여 결과를 화면에 표시한다.

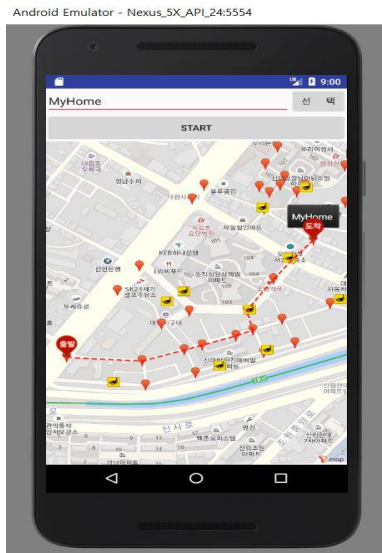


Fig. 3. Pathfinding App for Pedestrians Providing the Route and CCTV Locations

경로 탐색 알고리즘은 A* 알고리즘으로 구현하였으며 세 가지 기준을 이용하는 링크의 비용함수는 식 (5)와 같다.

$$\phi(e_i) = \phi(n_i, n_{i+1}) = (w_1 \cdot c_{i1}, w_2 \cdot c_{i2}, w_3 \cdot c_{i3}) \quad (5)$$

c_{i1}, c_{i2}, c_{i3} 는 각각 거리, 도로 등급, 위험도를 나타내며 $c_{i1} = |n_{i+1} - n_i|$ 으로 계산되고 c_{i2} 는 도로가 넓을수록 작은 값을 갖도록, c_{i3} 은 CCTV가 도로상에 어떻게 배치되었는가에 따라 위험할수록 높은 값을 갖도록 설정하였으며 노드 간의 거리를 나타내는 c_{i1} 의 평균값의 스케일에 맞추어 다른 기준의 범위를 정하고 정규화하였다. w_1, w_2, w_3 는 각각 거리, 도로 등급, 위험도에 대한 가중치이며 초기 가중치는 (1.0, 0, 0)으로 하여 기준의 최단 거리 길찾기 결과와 동일하도록 하였다. 가중치 학습이 일어난 후에는 $\|W\|$ (norm)로 나누어 w_i 가 -1~+1 사이의 값을 갖도록 정규화하였다.

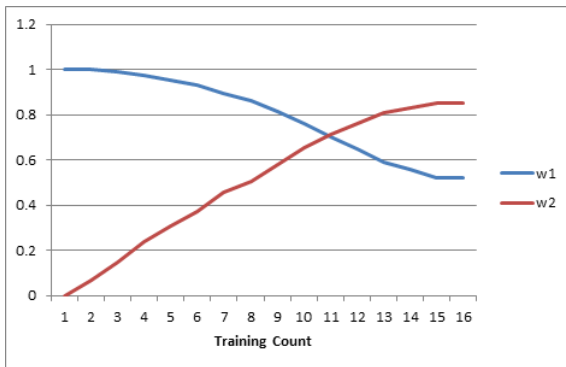
4.2 Simulation Results

그림 4는 사용자가 앱에서 제안한 경로와 다른 길을 택하여 이동할 때 가중치 학습을 통해 사용자의 선호도를 파악하여 경로를 재탐색한 결과를 보여준다. 이때의 가중치 학습은 식 (1)의 단기 가중치 학습이다. 앱에서 제안한 경로가 목표까지의 최단 경로인 반면 사용자는 넓은 길을 선택한 것을 알 수 있다. 그림 4에서 점선은 길찾기 앱에서 제안한 경로이고 실선으로 표시된 경로는 출발이후, 사용자의 선택이 제안한 경로와 다를 경우 감지하고 가중치 학습 후에 재탐색한 결과이다. 그림 4(a)의 α 값이 그림 4(b)의 α 보다 큰 경우인데 사용자가 선택한 길의 특성이 가중치에 얼마나 빠르게 반영되는가의 차이를 보여준다. 그림 4(a)가 (b)에 비해서 사용자의 선호도가 재탐색 결과에 빠르게 반영됨을 알 수 있다. 그러므로 사용자의 선호도를 실시간으로 신속히 반영하기 위해 학습 속도를 나타내는 단기 가중치 학습 인자는 장기 가중치 학습에 비해 크게 설정하였는데 본 논문에서 제안한 가중치 학습 방법의 핵심은 실제 이동 중에 일어나는 단기 가중치 학습 결과는 리셋하고 이동이 완료된 후에 모멘텀을 이용하여 가중치를 갱신하는 것이므로 큰 α 값 사용함으로써 초래될 수 있는 문제점은 발생하지 않았다.

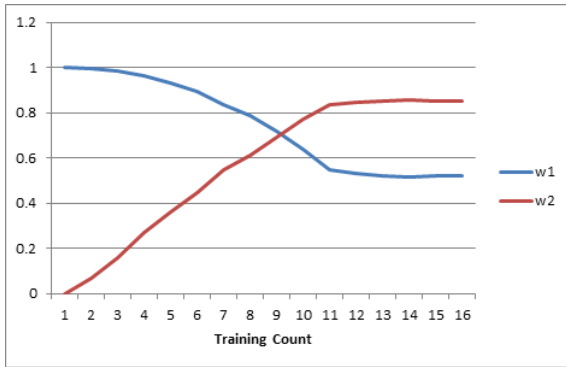


Fig. 4. Re-searching results with different α values : (α of (a)) > (α of (b))

그림 5와 표 1은 모멘텀을 사용한 장기 가중치 학습의 두 가지 장점을 각각 보여준다. 직관적인 비교를 위해 기준 중에 2가지 특성만을 고려하여 사용자가 자신의 선호도와 일치하는 선택을 하는 실험의 결과이다. 그림 5(a)는 모멘텀을 사용하지 않고 가중치를 학습한 결과이고 같은 시나리오를 모멘텀을 사용하여 반복한 결과가 그림 5(b)이다. 모멘텀을 이용한 경우(그림 5(b))에 그림 5(a)의 경우에 비해 가중치 값의 역전도 빨리 일어나고 빨리 안정화되어 빠른 학습이 가능함을 알 수 있다.



(a) Learning Results without Momentum



(b) Learning Results with Momentum

Fig. 5. Graphs of Weight Changes

사용자가 경로를 선택하는데 있어서 경우에 따라서는 선호도와 다른 경로로 이동하는 경우도 있다. 경로 선택이 선호도와 무관하게 랜덤하게 이루어지는 극단적인 케이스에는 가중치가 수렴하지 못하고 그림 6과 같이 진동한다. 이런 경우에는 전 단계까지의 학습 결과를 반영하는 모멘텀이 큰 진동을 피할 수 있게 하여 학습된 가중치가 안정되도록 해준다. 표 1은 장기 가중치 학습에서 모멘텀의 반영 비율인 β 의 값에 따른 가중치의 진동폭의 변화를 보여준다. 모멘텀을 사용하지 않은 경우에는 β 값과 무관하므로 가중치 변화량($\Delta W1$) 일정하지만 모멘텀을 사용하는 경우에는 β 값이 커짐에 따라 변화량($\Delta W2$)이 적어짐을 확인할 수 있다. β 값의 효과를 극대화하기 위해서는 학습 초기에는 작은 β 값을 이용하고 학습이 안정됨에 따라 커지도록 설정하는 것이 효과적임을 보여준다.

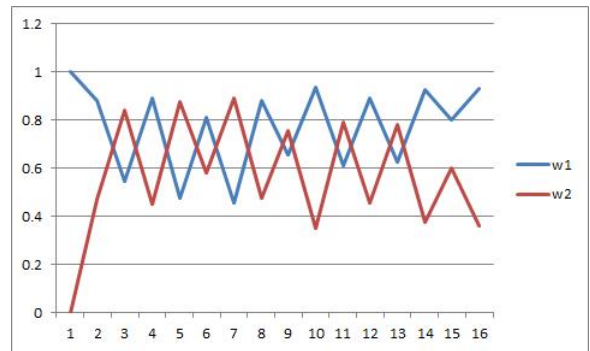


Fig. 6. Weight Changes due to Random Path Selection

Table 1. Average Weight Changes with Different β

β	Average ΔW		Ratio ($\Delta W2/\Delta W1$)
	w/o momentum $\Delta W1$	with momentum $\Delta W2$	
0.05	0.33	0.317	0.961
0.1	0.33	0.305	0.924
0.15	0.33	0.294	0.891
0.2	0.33	0.28	0.848
0.25	0.33	0.275	0.833
0.3	0.33	0.267	0.809
0.35	0.33	0.259	0.785

V. Conclusions

본 논문에서는 다기준 길찾기에서 사용자의 선호도를 자동으로 학습하여 같은 제품의 길찾기 프로그램이라도 사용하는 사람에 따라 다른 길을 안내하는 사용자 맞춤형 길찾기 앱을 개발하였다. 대부분의 내비게이션 길찾기 안내 결과는 큰 차이가 없어진 현 상황에서 차별화를 위해서는 경로의 질이 문제가 될 것으로 보여 본 논문이 추구하는 바가 의미가 있다고 할 것이다. 본 논문의 실험에서는 3가지의 판단 기준을 사용하였지만 일반성을 잃지 않고 여러 가지 기준의 추가가 가능하다. 또한 보행자용 길찾기로 소개했으나 같은 원리를 전체 내비게이션 길찾기에 확장 적용할 수 있을 것이다.

향후에는 A* 탐색의 성능 향상을 위해 허용성(admissibility)을 잃지 않는 범위에서 정보량이 많은 휴리스틱을 고안하는 연구와 많은 기준 정보들을 효율적으로 저장하고 검색(retrieve)하는 방법에 대한 연구를 수행이 필요하다.

REFERENCES

[1] S. Roy and J. Shynk, "Analysis of the momentum LMS algorithm", IEEE transactions on acoustics, speech, and signal processing, 38(12), pp. 2088-2098, Dec. 1990.
 [2] R. B. Pereira, "Pedestrian Navigation", <http://www>.

navipedia.net/index.php/Pedestrian_Navigation, 2011 updated in 2013.

- [3] Seoul City Open Data, <http://data.seoul.go.kr>.
- [4] SK Planet Developers, <https://developers.skplanetx.com/apidoc/kor/tmap/>.
- [5] F. B. Zhan, "Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures", *Journal of Geographic Information and Decision Analysis*, vol.1(1), pp. 69-82, Jan. 1998.
- [6] D. Ostrowski, I. Pozniak-Koszalka, L. Koszalka, and A. Kasprzak, "Comparative Analysis of the Algorithms for Pathfinding in GPS Systems", *ICN 2015 : The Fourteenth International Conference on Networks*, pp. 102-108, 2015.
- [7] H. Antikainen, "Using the Hierarchical Pathfinding A* Algorithm in GIS to Find Paths through Rasters with Nonuniform Traversal Cost", *International Journal of Geo-Information*, pp. 996-1014, Dec. 2013.
- [8] T. Miksa, L.Koszalka, and A. Kasprzak, "Comparison of heuristic methods applied to optimization of computer networks", *Proc. of 11th International Conference on Networks,IARIA*, pp. 34-38, 2012.
- [9] A. Botea, "Ultra-Fast Optimal Pathfinding without Runtime Search", *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 122-127, 2011.
- [10] A. R. Soltani, H. Tawfik, and T. Fernando, "A multi-criteria based path finding application for construction site layouts," *Proceedings of Sixth International Conference on Information Visualization*, pp. 779-784, 2002.
- [11] P. Pahlavani, M. R. Delavar, "Multi-criteria route planning based on a driver's preferences in multi-criteria route selection", *Transportation Research Part C*, Vol. 40, pp. 14-35, 2014.
- [12] A. Simeon, et al. "Towards Realistic Pedestrian Route Planning", *OASlcs-OpenAccess Series in Informatics*. Vol. 48, Schloss Dagstuhl Leibniz Zentrum fuer Informatik, pp. 1-15, 2015.
- [13] A. U. Kemloh Wagoum, A. Seyfried, and S. Holl, "Modelling dynamic route choice of pedestrians to assess the criticality of building evacuation", *Advances in Complex Systems* Vol 15(7), Jul. 2012.
- [14] S. Ruder, "An overview of gradient descent optimization algorithms", *arXiv preprint arXiv: 1609. 04747*, <http://ruder.io/optimizing-gradient-descent/>, 2016.

Authors



Chojung Lee is an undergraduate student of the Department of Computer Science at Duksung Women's University. Her research interests include Algorithms and Artificial Intelligence.



Inyoung Cho is an undergraduate student of the Department of Computer Science at Duksung Women's University. Her research interests include Algorithms and Artificial Intelligence.



Kyeonah Yu received the B.S. and M.S. degrees in Control and Instrumentation Engineering from Seoul National University, Korea, in 1986 and 1988, respectively and the Ph.D degree in Computer Science from University of

Southern California, USA, in 1995. She joined the faculty of the Department of Computer Science at Duksung Women's University, Seoul, Korea, in 1996. She is currently a Professor in the Department of Computer Science, Duksung Women's University. Her research interests include artificial intelligence, path finding algorithms, and applying knowledge-based AI to neural networks.