벡터 매트릭스의 계산과 적용

이 호 석
뉴미디어학과 공과대학 호서대학교
hslee@office.hoseo.ac.kr

A Computation and Application of Vector Matrix

Ho Suk Lee
New Media Dept. College of Engineering Hoseo University

요 약

본 논문은 벡터를 원소로 가지는 새로운 매트릭스인 벡터 매트릭스의 정의, 기본 연산 규칙, 그리고 적용에 대하여 논의한다. 이러한 벡터 매트릭스는 노드와 노드 사이에 여러 개의 에지가 형성되는 다중 에지 네트워크를 적절하게 표현하고 관련된 응용에 쉽게 적용할 수 있을 것으로 생각한다. 본 논문은 이러한 벡터 매트릭스를 노드와 노드 사이에 2개의 에지가 있는 feedforward 네트워크에 적용하여 출력값을 계산하는 과정을 보여준다.

1. 서 론

참고 문헌[1]의 19장에 제시된 설명에 의하면 인간의 두뇌에 새로운 기억이 형성되는 것은 뉴런과 뉴런 사이에 새로운 시냅스가 추가로 형성되는 것이라고 한다. 즉, 기존의 시냅스이외에 새로운 시냅스가 추가로 뉴런과 뉴런 사이에 형성되는 것이다. 이것을 네트워크로 모델링하면 기존의 에지이외에 새로운 에지가 노드와 노드 사이에 형성되는 것이라고 할 수 있다. 그런데 일반적으로 네트워크를 구성하는 노드와 노드 사이에는 하나의 에지가 있는 것으로 생각한다. 본 논문에서는 노드와 노드 사이에 2개의 에지가 있는 것을 가정하고 이것을 표현하기 위하여 새로운 매트릭스인 벡터 매트릭스를 정의하였다. 그리고 벡터 매트릭스에 대한 기본 연산 규칙을 계산하여 보았다.

일반적으로 매트릭스는 다음과 같이 표시된다. 여기서 매트릭스의 원소 $a_{11}, a_{12}, a_{21}, a_{22}$ 값은 스칼라 값이다.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

그리고 매트릭스 B와 C를 다음과 같이 정의할 경우에

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, \quad C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

매트릭스 곱셈 $A \times B = C$ 에서 C 의 원소 c_{ij}는 $c_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj}$ 로 정의된다[2].

그런데 본 논문은 매트릭스의 원소가 스칼라가 아니고 벡터인 경우를 고려하기로 한다. 즉, 다음 매트릭스 S 의 원소인 s_{11} s_{12}, s_{21}, s_{22} 는 스칼라가 아니고 벡터이다.

$$S = \begin{pmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{pmatrix}$$

즉, s_{11}~s_{22}는 다음과 같이 벡터로 표시된다. 일단 벡터들의 크기는 4인 동일한 것으로 가정한다.

$$s_{11} = (s_1, s_2, s_3, s_4),\ s_{12} = (s'_1, s'_2, s'_3, s'_4)$$
$$s_{21} = (s''_1, s''_2, s''_3, s''_4),\ s_{22} = (s^+_1, s^+_2, s^+_3, s^+_4)$$

이러한 경우에 매트릭스 S 를 벡터 매트릭스라고 새롭게 정의하고, 이러한 벡터 매트릭스에 대하여 곱셈, 뺄셈, 덧셈, 나눗셈과 대수연산 등을 논의하고, 2개의 에지를 가지는 다중 에지 feedforward 네트워크에 적용하여 출력값을 계산하는 과정을 보여준다.

2. 본 론

2.1 벡터 매트릭스의 기본 계산

먼자, 벡터 매트릭스의 곱셈에 대하여 논의하기로 한다. 벡터 매트릭스 T와 U를 다음과 같이 정의하자.

$$T = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}$$

예를 들어서 s_{11}~s_{22}와 t_{11}~t_{22}에 다음과 같이 값을 부여하고 연산을 해 보자.

s_{11} = (1, 2, 3, 4), s_{12} = (-1, -2, -3, -4)
s_{21} = (5, 6, 7, 8), s_{22} = (-5, -6, -7, -8)

t_{11} = (10, 20, 30, 40), t_{12} = (-10, -20, -30, -40)
t_{21} = (50, 60, 70, 80), t_{22} = (-50, -60, -70, -80)

그러면

$$U=\begin{pmatrix} s_{11}t_{11}+s_{12}t_{21}, s_{11}t_{12}+s_{12}t_{22} \\ s_{21}t_{11}+s_{22}t_{21}, s_{21}t_{12}+s_{22}t_{22} \end{pmatrix}$$ 이므로

u_{11} = s_{11} x t_{11} + s_{12} x t_{21} = (1, 2, 3, 4)x(10, 20, 30, 40) + (-1, -2, -3, -4)x(50, 60, 70, 80) = (10, 40, 90, 160) + (-50, -120, -210, -320) = (-40, -80, -120, -160)

u_{12} = s_{11} x t_{12} + s_{12} x t_{22} = (1, 2, 3, 4)x(-10, -20, -30, -40) + (-1, -2, -3, -4)x(-50, -60, -70, -80) = (-10, -40, -90, -160) + (50, 120, 210, 320) = (40, 80, 120, 160)

u_{21} = s_{21} x t_{11} + s_{22} x t_{21} = (5, 6, 7, 8)x(10, 20, 30, 40)+(-5, -6, -7, -8)x(50, 60, 70, 80) = (50, 120, 210, 320) + (-250, -360, -490, -640) = (-200, -240, -280, -320)

u_{22}= s_{21} x t_{12} + s_{22} x t_{22} = (5, 6, 7, 8)x(-10, -20, -30, -40)+(-5, -6, -7, -8)x(-50, -60, -70, -80) = (-50, -120, -210, -320) + (250, 360, 490, 640) = (200,240,280,320)

가 된다. 여기서 연산자 x 는 벡터에서 위치에 따른 (positionwise) 곱셈 연산자이고, 연산자 + 는 벡터에서 위치에 따른 덧셈 연산자라고 정의한다. 다음은 계산된 U 벡터 매트릭스이다.

$$U=\begin{pmatrix} (-40,-80,-120,-160),(40,80,120,160) \\ (-200,-240,-280,-320),(200,240,280,320) \end{pmatrix}$$

벡터 매트릭스에 대한 스칼라 곱셈은 다음과 같이 표현할 수 있다.

$$\alpha S=\begin{pmatrix} s_{11}\ s_{12} \\ s_{21}\ s_{22} \end{pmatrix} = \begin{pmatrix} \alpha s_{11}\ \alpha s_{12} \\ \alpha s_{21}\ \alpha s_{22} \end{pmatrix}$$ 이다.

여기서 αs_{11} = (αs_1, αs_2 , αs_3 , αs_4) 이 된다. s_{12}, s_{21}, s_{22}의 경우도 마찬가지이다. 벡터 매트릭스에 대한 스칼라 나눗셈은 다음과 같이 표현할 수 있다.

$$1/\alpha\ \ S=\begin{pmatrix} s_{11}\ s_{12} \\ s_{21}\ s_{22} \end{pmatrix} = \begin{pmatrix} s_{11}/\alpha, s_{12}/\alpha \\ s_{21}/\alpha, s_{22}/\alpha \end{pmatrix}$$

여기서, s_{11}/α = (s_1/α, s_2/α, s_3/α, s_4/α)이 된다. s_{12}, s_{21}, s_{22}의 경우도 동일하게 계산할 수 있다.

벡터 매트릭스끼리의 덧셈은 다음과 같다.

$$S+T = \begin{pmatrix} s_{11}\ s_{12} \\ s_{21}\ s_{22} \end{pmatrix} + \begin{pmatrix} t_{11}\ t_{12} \\ t_{21}\ t_{22} \end{pmatrix} = \begin{pmatrix} s_{11}+t_{11}, s_{12}+t_{12} \\ s_{21}+t_{21}, s_{22}+t_{22} \end{pmatrix}$$

여기서 $s_{11}+t_{11}$ = (s_1,s_2,s_3,s_4)+(t_1,t_2,t_3,t_4) =($s_1+t_1,s_2+t_2,s_3+t_3,s_4+t_4$) 이며 다른 항의 경우도 동일하게 계산할 수 있다. 벡터 매트릭스끼리의 뺄셈도 비슷하게 계산될 수 있다.

그리고 벡터 매트릭스 정의로부터 덧셈에 있어서 다음과 같은 교환(commutative) 법칙[3]과 결합(associative) 법칙[3]이 성립하는 것을 알 수 있다.

1. A + B = B + A
2. (A + B) + C = A + (B + C)
3. a(A + B) = aA + aB
4. (a + b)A = aA + bA
5. (ab)A = a(bA)

이제 벡터 매트릭스의 곱셈에 대하여 다음과 같은 결합 법칙과 분배 법칙이 성립하는지 증명해 보자.

1. (AB)C = A(BC)
2. (A + B)C = AC + BC
3. A(B+C) = AB + AC

$$A=\begin{pmatrix} a_{11}\ a_{12} \\ a_{21}\ a_{22} \end{pmatrix},\quad B=\begin{pmatrix} b_{11}\ b_{12} \\ b_{21}\ b_{22} \end{pmatrix},\quad C=\begin{pmatrix} c_{11}\ c_{12} \\ c_{21}\ c_{22} \end{pmatrix}$$

라고 하면

$$(AB)C = \begin{pmatrix} a_{11}b_{11}+a_{12}b_{21}, a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21}, a_{21}b_{12}+a_{22}b_{22} \end{pmatrix}\begin{pmatrix} c_{11}, c_{12} \\ c_{21}, c_{22} \end{pmatrix}$$

$$= \begin{pmatrix} (a_{11}b_{11}+a_{12}b_{21})c_{11}+(a_{11}b_{12}+a_{12}b_{22})c_{21}, \\ (a_{11}b_{11}+a_{12}b_{21})c_{12}+(a_{11}b_{12}+a_{12}b_{22})c_{22} \\ (a_{21}b_{11}+a_{22}b_{21})c_{11}+(a_{21}b_{12}+a_{22}b_{22})c_{21}, \\ (a_{21}b_{11}+a_{22}b_{21})c_{12}+(a_{21}b_{12}+a_{22}b_{22})c_{22} \end{pmatrix}$$

이고

$$A(BC) = \begin{pmatrix} a_{11}, a_{12} \\ a_{21}, a_{22} \end{pmatrix}\begin{pmatrix} b_{11}c_{11}+b_{12}c_{21}, b_{11}c_{12}+b_{12}c_{22} \\ b_{21}c_{11}+b_{22}c_{21}, b_{21}c_{12}+b_{22}c_{22} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}(b_{11}c_{11}+b_{12}c_{21})+a_{12}(b_{21}c_{11}+b_{22}c_{21}), \\ a_{11}(b_{11}c_{12}+b_{12}c_{22})+a_{12}(b_{21}c_{12}+b_{22}c_{22}) \\ a_{21}(b_{11}c_{11}+b_{12}c_{21})+a_{22}(b_{21}c_{11}+b_{22}c_{21}), \\ a_{21}(b_{11}c_{12}+b_{12}c_{22})+a_{22}(b_{21}c_{12}+b_{22}c_{22}) \end{pmatrix}$$

이다. 여기서 예를 들어 위 (AB)C의 첫 번째 항, $(a_{11}b_{11}+a_{12}b_{21})c_{11}+(a_{11}b_{12}+a_{12}b_{22})c_{21}$ 을 전개하면 $a_{11}b_{11}c_{11}+a_{12}b_{21}c_{11}+a_{11}b_{12}c_{21}+a_{12}b_{22}c_{21}$ 이 된다. 그리고 A(BC)의 첫 번째 항 $a_{11}(b_{11}c_{11} + b_{12}c_{21}) + a_{12}(b_{21}c_{11}+b_{22}c_{21})$ 을 전개하면 $a_{11}b_{11}c_{11} + a_{11}b_{12}c_{21} + a_{12}b_{21}c_{11}+a_{12}b_{22}c_{21}$ 가 되어 결과가 동일함을 알 수 있다. 다른 항도 전개하면 동일한 것을 알 수 있다. 따라서 결합법칙과 분배법칙이 성립함을 알 수 있다. 2번과 3번의 법칙도 성립함을 알 수 있다.

벡터 매트릭스의 치환은 스칼라 매트릭스와 동일하게 정의할 수 있다.

$$S^T = \begin{pmatrix} s_{11}\ s_{21} \\ s_{12}\ s_{22} \end{pmatrix}$$

벡터 매트릭스의 치환에 대하여 다음과 같은 연산 법칙이 성립하는지 증명해 보자.

1. $(A + B)^T = A^T + B^T$
2. $(aA)^T = aA^T$
3. $(AB)^T = B^TA^T$
4. $(A^{-1})^T = (A^T)^{-1}$

1번과 2번은 자명하다. 3번을 계산해 보자. 앞에서 ST=U 이므로 $(ST)^T = U^T$ 이다. 따라서

$$U^T = \begin{pmatrix} s_{11}t_{11}+s_{12}t_{21}, s_{21}t_{11}+s_{22}t_{21} \\ s_{11}t_{12}+s_{12}t_{22}, s_{21}t_{12}+s_{22}t_{22} \end{pmatrix}$$

이다. 그러면, T^TS^T를 계산해 보자.

$$T^T = \begin{pmatrix} t_{11}\ t_{21} \\ t_{12}\ t_{22} \end{pmatrix} \text{ 이고 } \quad S^T = \begin{pmatrix} s_{11}\ s_{21} \\ s_{12}\ s_{22} \end{pmatrix} \text{ 이므로}$$

$$T^TS^T = \begin{pmatrix} t_{11}s_{11}+t_{21}s_{12}, t_{11}s_{21}+t_{21}s_{22} \\ t_{12}s_{11}+t_{22}s_{12}, t_{12}s_{21}+t_{22}s_{22} \end{pmatrix}$$

이 된다. 따라서 $(ST)^T = T^TS^T$가 성립한다고 할 수 있다.

이제 벡터 매트릭스의 결정요소(determinant)[4]를 계산해 보자. S 벡터 매트릭스를 예로 들자.

$$\det(S) = s_{11} \times s_{22} - s_{12} \times s_{21}$$

$$S^{-1} = \frac{1}{\det(S)}\begin{pmatrix} s_{22}, & -s_{12} \\ -s_{21}, & s_{11} \end{pmatrix}$$

그러면

$$(S^{-1})^T = \frac{1}{\det(S)}\begin{pmatrix} s_{22}, & -s_{21} \\ -s_{12}, & s_{11} \end{pmatrix}$$

$$(S^T)^{-1} = \begin{pmatrix} s_{11}\ s_{21} \\ s_{12}\ s_{22} \end{pmatrix}^{-1} = \frac{1}{\det(S^T)}\begin{pmatrix} s_{22} & -s_{21} \\ -s_{12} & s_{11} \end{pmatrix}$$

$$\det(S^T) = s_{11} \times s_{22} - s_{21} \times s_{12}$$

따라서 성립한다. 다만, 여기서 결정요소는 스칼라 값이 아니고 벡터 결정요소이다. 따라서 4번의 법칙도 성립한다.

이로서 벡터 매트릭스에 대한 연산 기본 법칙이 성립하는 것을 모두 제시하였다.

2.2 벡터 매트릭스의 적용

2.2.1 Feedforward 네트워크

이제 벡터 매트릭스를 feedforward 네트워크에 적용한 경우를 살펴보자. 다음은 참고문헌 [4]에 제시되어 있는 feedforward 네트워크 방정식이다.

$$c\frac{d}{dt}v = -v + F(W \times u)$$

여기서 시간에 의한 변화가 없다고 가정하고, 함수 F도 고려하지 않으면 다음과 같이 간단한 식을 얻을 수 있다.

$$v = W \times u$$

여기서 v 는 출력 벡터, u 는 입력 벡터, 그리고 W는 v와 u

feedforward 네트워크의 학습 값이다. 그리고 임의의 v와 u 사이에는 하나 이상의 에지가 있다고 가정하고 현재 2개가 있다고 - 일반적인 feedforward 네트워크는 노드와 노드 사이에 1개의 에지가 있는 것으로 생각한다[5] - 가정한다. 참고문헌 [1]에 의하면, 기억이 형성된다는 것은 뉴런과 뉴런 사이에 새로운 시냅스가 만들어지는 것이라고 한다. 에지의 학습 값은 0이상의 양수로 한다. 0값을 가진 경우에는 값이 없는 것으로 한다. 그러면 앞 절에서 설명한 벡터 매트릭스의 기본 계산 방법에 의하여 W x u 을 계산할 수가 있으므로 출력 v를 구할 수가 있다. 즉, 입력 노드를 u = (u_1,u_2), 출력 노드를 v=(v_1,v_2), 그리고 학습 값을 W=(w_1,w_2)라고 하자. 여기서 w_1=($\omega_1,\omega_2,\omega_5,\omega_6$)이고 w_2=($\omega_3,\omega_4,\omega_7,\omega_8$) 이다. 그리고 u와 v사이에 각각 2개의 에지에 의하여 feedforward 네트워크가 형성되어 있다고 가정하면, 다음과 같이 표현할 수 있다.

$$v_1 = w_1 \times u = \begin{pmatrix} w_1, w_5 \\ w_2, w_6 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} u_1 w_1 + w_5 u_2 \\ w_2 u_1 + w_6 u_2 \end{pmatrix}$$

$$v_2 = w_2 \times u = \begin{pmatrix} w_3, w_7 \\ w_4, w_8 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} w_3 u_1 + w_7 u_2 \\ w_4 u_1 + w_8 u_2 \end{pmatrix}$$

따라서, 벡터 매트릭스를 이용하여 노드와 노드 사이에 2개의 에지가 있는 feedforward 네트워크를 효과적으로 표현하고 계산할 수 있었다. 최종적인 값은 v_1 = F (($u_1 w_1 + w_5 u_2$)+($w_2 u_1 + w_6 u_2$))로 계산할 수 있다. v_2의 경우도 동일하게 계산할 수 있다.

3. 결론

본 논문에서는 새로운 매트릭스인 벡터 매트릭스를 정의하였다. 그리고 벡터 매트릭스에 대하여 기본 연산 규칙이 성립함을 제시하였으며, 노드와 노드 사이에 2개의 에지를 가지는 feedforward 네트워크에 적용하여 출력값을 계산하는 과정을 제시하였다.

참고 문헌

[1] Eric R. Kandel, In Search of Memory, W. W. Norton & Company Inc., 2006.
[2] Gene H. Golub, Charles F. Van Loan, Matrix Computation (3rd ed.), The Johns Hopkins University Press, 1996.
[3] F. R. Gantmacher, The Theory of Matrices Vol. 1 (2nd ed.), Chelsea Publishing Company, 1990.
[4] Erwin Kreyszig, Advanced Engineering Mathematics(7th ed.), John Wiley & Sons, Inc., 1993.
[5] Peter Dayan, L. F. Abbott, Theoretical Neuroscience, The MIT Press, 2001.