# 6 장: 딥 하이퍼넷 (Deep Hypernetworks)

## 6.1 하이퍼넷 구조

하이퍼넷(hypernetworks, 줄여서 hypernet)은 데이터에 존재하는 고차 관계를 학습하는 확률그래프 모델이다[1]. 데이터 변수들간의 고차적 관계를 표현하기 위해서 하이퍼그래프(hypergraph) 구조를 이용한다. 관측된 데이터를 재생성하도록 무감독 학습을 수행하며, 학습 데이터의 결합확률분포를 추정하는 알고리듬으로 볼 수 있다. 환경과의 상호작용을 통해서 끊임없이 새로 발생하는 온라인 데이터를 끊임없이 학습하는 평생 점진적 학습을 수행한다. 변화하는 환경에 빠르고 점진적으로 적응하기 위해서 One-shot 학습이 가능한 구조 재조직 연산을 수행한다.

실제적인 온라인 학습 방법을 논의하기 전에 먼저 확률통계적인 이론을 살펴보자. 하이퍼넷은 데이터의 확률분포, 즉 우도(likelihood)를 최대화하도록 학습된다. 후에 점진적 알고리듬을 기술할 때 보겠지만 실제로는 우도 외에 사전 확률 분포(prior distribution)를 고려하여 이를 점진적으로 갱신하는 순차적 베이지안 학습을 수행한다. 여기서는 이론적으로 좀 더 명확한 우도 최대화 학습 방법으로 먼저 설명한다. 하이퍼넷을 구성하는 모델을 기술하는 파라미터를 $W$라 하면, 모델로부터 데이터가 생성될 확률은 다음과 같이 표현될 수 있다.

$$P(D \mid W) = \prod_{d=1}^{N} P(\mathbf{x}^{(d)} \mid W)$$

위에서 학습집합 $D$의 전체 확률을 각 성분확률의 곱으로 표현할 수 있는 것은 학습 데이터가 순서에 무관하게 즉 독립적으로 생성된다고 보기 때문이다. 학습 데이터에 대한 확률분포는 통계역학에서 사용하는 볼쯔만(Boltzmann) 분포를 사용한다. 즉, 물리계에서 상태 에너지 분포 $E(\mathbf{x})$인 시스템이 상태 $\mathbf{x}$에 있을 확률은 다음의 볼쯔만 분포를 가진다:

표 6-1. 하이퍼넷의 주요 특성

| 특성 | 설명 |
|---|---|
| **고차 연관관계 표현** | · Feature 의 조합을 명시적으로 표현<br>· 일반적인 그래프 구조만을 허용하는 베이지안망과 비교할 때 빠른 학습을 기대 |
| **구조 학습** | · 학습 과정에서 구조의 진화 과정을 적용<br>· 의미 있는 모듈 발견을 기대 |
| **군집 기반 부호화** | · 하이퍼엣지와 그 조합으로 구성된 모듈로 데이터의 정보가 표현됨. 점진적 학습 가능<br>· 베이지안망의 조건확률표 기반 정보 표현과 대비 |
| **조합성(compositionality)** | · 학습 과정에서 선택, 생성된 하이퍼엣지와 모듈의 조합으로 새로운 모듈 생성<br>· 추상화 및 추상화 수준을 높인 기호 기반 계산 |
| **자가-감독학습** | · 레이블이 없는 데이터로도 학습 가능<br>· 학습 과정에서 레이블 탐색 |
| **동적 조정 가능 체계** | · 동적 자기조직 방식의 학습과 추론<br>· 상시 추론 가능 |

$$P(\mathbf{x}|W) = \frac{1}{Z(W)} \exp\left[ -\frac{E(\mathbf{x}|W)}{k_B T} \right]$$

위에서 $k_B$ 는 볼쯔만 상수, $T$ 는 이 시스템의 절대 온도이며 정규화 상수 $Z$ 는 모든 가능한 상태들이 나타날 확률의 합이다.

$$Z(W) = \sum_{\mathbf{x}} \exp\left[ -\frac{E(\mathbf{x}|W)}{k_B T} \right]$$

앞의 두 식을 결합하면 feature vector **x** 에 대한 우도 확률은 다음과 같이 정리된다.

$$P(\mathbf{x}|W) = \frac{\exp\left[ -E(\mathbf{x}|W) \right]}{\sum_{\mathbf{x}'} \exp\left[ -E(\mathbf{x}'|W) \right]}$$

하이퍼그래프는 일반적인 그래프를 확장한 고차의 그래프이다. 일반적인 그래프에서 에지(edge)는 두 개의 정점을 연결한다. 즉 에지의 차수는 $k \le 2$ 이다. 하이퍼그래프에서는 정점을 2개 이상으로 일반화하여 다수의 정점을 연결한 하이퍼에지를 허용한다. 즉 $k \ge 1$. 하나의 하이퍼에지에 있는 정점의 갯수를 그 하이퍼에지의 차수라 한다. 하이퍼네트워크는 다시 하이퍼그래프를 일반화하여 각각의 하이퍼에지가 가중치를 가지도록 확장한 것이다. 이는 신경망이 연결 가중치를 가진 것처럼 하이퍼에지의 연결선들이 가중치를 가진 것으로 볼 수 있다. 형식화하면, 하이퍼네트워크는 다음의 트리플로 정의된다.

$$\begin{aligned}
&H = (X, E, W), \\
&X = \{x_1, x_2, \ldots, x_{|I|}\}, \\
&E = \{E_1, E_2, \ldots, E_{|E|}\}, \\
&W = \{w_1, w_2, \ldots, w_{|E|}\} \\
&E_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}
\end{aligned}$$

특수한 경우로 차수 k인 하이퍼에지로만 구성된 하이퍼네트워크를 k-하이퍼네트워크라 한다.

그림 6-1에 하이퍼네트워크의 한 예를 도식화하였다. 7개의 정점 $X = \{x_1, x_2, \ldots, x_7\}$ 과 5개의 하이퍼에지 $E = \{E_1, E_2, E_3, E_4, E_5\}$ 로 구성되었으며 각각의 하이퍼에지는 다른 차수 $k_i = |E_i|$ 와 가중치 $W = \{w_1, w_2, w_3, w_4, w_5\}$ 를 가진다. 하이퍼네트워크 $H$ 는 인시덴스 행렬 $[a_j^i]$ 로 표시할 수 있다. 매트릭스는 $|E|$ 개의 행을 가지며 각각 하이퍼에지를 나타낸다. $I+1$ 개의 열은 $H$ 의 정점을 나타낸다. 행렬의 요소는 다음을 나타낸다.

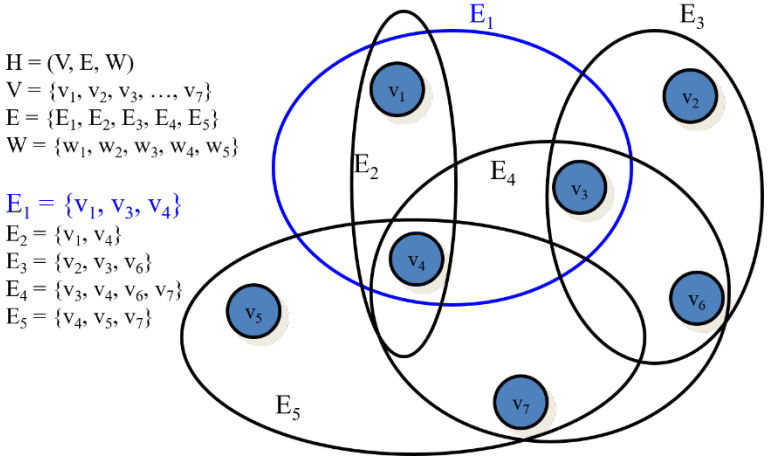



그림 6-1. 하이퍼네트워크의 예.

하이퍼그래프 구조와 하이퍼엣지 각각에 부여된 가중치로 모델이 정의된다.

$$a_j^i = \begin{cases} w_j & x_i \in E_j \quad j = 0 \\ 1 & x_i \in E_j \quad j = 1,..,|E| \\ 0 & x_i \notin E_j \quad j = 1,..,|E| \end{cases}$$

여기서 $i$=1,…,$I$는 열의 인덱스이고 $j$=1,…,|$E$|는 행의 인덱스이다.

하이퍼네트워크는 학습데이터셋 $D_N = \{\mathbf{x}^{(d)}\}_{d=1}^N$을 저장한 후에 내용 기반으로 인출할 수 있는 확률연상메모리로 사용될 수 있다.

$$\mathbf{x}^{(d)} = (x_1, x_2, ..., x_I),$$
$$E_i = (x_{i_1}, x_{i_2}, ..., x_{i_j}, ..., x_{i_k}),$$
$$i_j \in \{1, 2, ..., I\},\ j = 1, ..., k$$

하이퍼넷의 에너지 함수를 다음과 같이 정의한다.

$$\varepsilon(\mathbf{x}^{(d)}; W) = -\sum_{i=1}^{|E|} w_{i_1 i_2 ... i_{E_i}} x_{i_1}^{(d)} x_{i_2}^{(d)} ... x_{i_{E_i}}^{(d)},$$

여기서 $W$는 하이퍼넷 모델의 파라미터인 하이퍼에지의 가중치벡터를 나타낸다. $x_{i_1}^{(n)} x_{i_2}^{(n)} ... x_{i_{|E_i|}}^{(n)}$는 데이터 아이템 $\mathbf{x}^{(d)}$의 $k_i = |E_i|$개의 요소를 가진 변수들의 조합이다. 하이퍼넷으로부터 데이터가 생성될 확률은 깁스(Gibbs) 분포로 주어진다.

$$P(\mathbf{x}^{(d)} | W) = \frac{1}{Z(W)} \exp\left\{-\varepsilon(\mathbf{x}^{(d)}; W)\right\},$$

여기서 $\exp\{-\varepsilon(\mathbf{x}^{(d)}; W)\}$는 볼쯔만 인자이고 정규화 항 $Z(W)$은 다음과 같다.

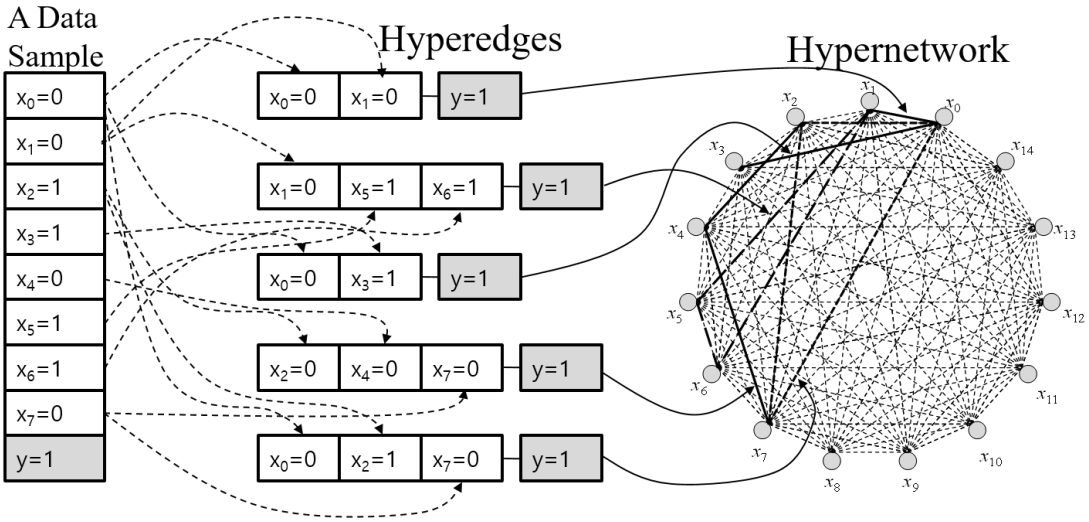



그림 6-2. 관측 데이터 벡터에서 feature의 고차 조합을 통한 하이퍼엣지 구성 및 하이퍼넷 업데이트 과정. 감독 학습을 위한 데이터의 경우 레이블(y)은 하나의 벡터에서 나온 모든 하이퍼엣지에 공통으로 추가된다.

$$Z(W) = \sum_{x^{(m)}} \exp\{-\varepsilon(\mathbf{x}^{(m)}; W)\}$$
$$= \sum_{x^{(m)}} \exp\left\{-\sum_{i=1}^{|E|} w_{i_1 i_2 \cdots i_{|E_i|}} x_{i_1}^{(m)} x_{i_2}^{(m)} \ldots x_{i_{|E_i|}}^{(m)}\right\}.$$

하이퍼넷은 가중치를 가진 하이퍼에지들의 군집으로 데이터의 결합확률분포를 표시하는 확률 모델이다.

## 6.2 하이퍼넷 학습

학습은 최대 우도값을 갖는 가중치 벡터를 찾는 문제와 같다.

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} P_{\mathbf{w}}(\mathbf{v}) = -\arg\min_{\mathbf{w}} \log P_{\mathbf{w}}(\mathbf{v})$$

가중치 연결 $w_{ij}$하나에 대해서 다시 표현하면 이는 확률변화에 대한 가중치의 기울기가 0 이 되는 것을 찾는 문제로 볼 수 있다. 즉

$$\frac{\partial \log P_{\mathbf{w}}(\mathbf{v})}{\partial w_{ij}} = 0$$

이 조건을 만족하는 경우 다음을 유도할 수 있다.

$$\frac{\partial \log P_{\mathbf{w}}(\mathbf{v})}{\partial w_{ij}} = < v_i h_j >^0 - < v_i h_j >^\infty$$

따라서 가중치 변경식은 다음과 같으며

$$\Delta w_{ij} = \eta(< v_i h_j >^0 - < v_i h_j >^\infty)$$

이를 대조분산학습(contrastive divergence learning) 방법이라 한다. 학습 절차를 구체적으로 기술하면 다음과 같다.

학습 데이터 $D$가 주어졌다고 하자.

$$D_N = \{\mathbf{x}^{(d)}\}_{d=1}^N$$

학습은 다음의 우도 함수를 최대화하는 하이퍼넷 모델을 찾는 것이다.

$$P(D|W) = \prod_{n=1}^{N} P\left(x^{(n)}|W\right)$$

우도를 최대화하는 대신 로그우도를 최대화할 수 있다.

$$\ln P(D|W) = \ln \prod_{d=1}^{N} P(\mathbf{x}^{(n)}|W)$$
$$= \sum_{d=1}^{N} \left\{ \left[ \sum_{k=1}^{K} \frac{1}{C(k)} \sum_{i_1 i_2 \cdots i_k} w_{i_1 i_2 \cdots i_k}^{(k)} x_{i_1}^{(n)} x_{i_2}^{(n)} \cdots x_{i_k}^{(n)} \right] - \ln Z(W) \right\},$$

학습은 로그우도의 값이 더 이상 변하지 않은 파라미터를 찾는 것이며, 수학적으로는 다음의 미분값을 구하는 것이다.

$$
\begin{aligned}
&\frac{\nabla}{\nabla w^{(k)}_{i_1,i_2,\cdots,i_k}} \ln \prod_{d=1}^{N} P(\mathbf{x}^{(d)}|W) \\
&= \frac{\nabla}{\nabla w^{(k)}_{i_1,i_2,\cdots,i_k}} \sum_{d=1}^{N} \left\{ \left[ \sum_{k=1}^{K} \frac{1}{C(k)} \sum_{i_1,i_2,\cdots,i_k} w^{(k)}_{i_1 i_2 \cdots i_k} x^{(d)}_{i_1} x^{(d)}_{i_2} \cdots x^{(d)}_{i_k} \right] - \ln Z(W) \right\} \\
&= \sum_{d=1}^{N} \left\{ \frac{\nabla}{\nabla w^{(k)}_{i_1,i_2,\cdots,i_k}} \left[ \sum_{k=1}^{K} \frac{1}{C(k)} \sum_{i_1,i_2,\cdots,i_k} w^{(k)}_{i_1 i_2 \cdots i_k} x^{(d)}_{i_1} x^{(d)}_{i_2} \cdots x^{(d)}_{i_k} \right] - \frac{\nabla}{\nabla w^{(k)}_{i_1,i_2,\cdots,i_k}} \ln Z(W) \right\} \\
&= \sum_{d=1}^{N} \left\{ x^{(d)}_{i_1} x^{(d)}_{i_2} \cdots x^{(d)}_{i_k} - \left\langle x_{i_1} x_{i_2} \cdots x_{i_k} \right\rangle_{P(\mathbf{x}|W)} \right\} \\
&= N \left\{ \left\langle x_{i_1} x_{i_2} \cdots x_{i_k} \right\rangle_{Data} - \left\langle x_{i_1} x_{i_2} \cdots x_{i_k} \right\rangle_{P(\mathbf{x}|W)} \right\},
\end{aligned}
$$

마지막의 식에서 $< x_{i_1} x_{i_2} \cdots x_{i_k} >_{Data}$는 데이터의 분포를 나타내며 $< x_{i_1} x_{i_2} \cdots x_{i_k} >_{P(\mathbf{x}|W)}$는 하이퍼넷 모델이 표현하는 데이터의 분포이다.

$$
\begin{aligned}
\left\langle x_{i_1} x_{i_2} \cdots x_{i_k} \right\rangle_{Data} &= \frac{1}{N} \sum_{d=1}^{N} \left[ x^{(d)}_{i_1} x^{(d)}_{i_2} \cdots x^{(d)}_{i_k} \right] \\
\left\langle x_{i_1} x_{i_2} \cdots x_{i_k} \right\rangle_{P(\mathbf{x}|W)} &= \sum_{\mathbf{x}} \left[ x_{i_1} x_{i_2} \cdots x_{i_k} P(\mathbf{x}|W) \right].
\end{aligned}
$$

학습은 이 두 분포가 수렴하는 지점에서 일어난다.

하이퍼넷은 입력층과 하이퍼에지 노드로 구성된 이층 신경망으로 표현될 수 있다. 학습 과정은 다음과 같이 상향 추론과 하향 추론의 반복을 통해서 수렴할 때까지 진행된다.

1. 데이터 벡터 $\mathbf{x}$를 입력층에 할당한다.
2. 입력노드에 연결된 은닉층 뉴런들을 활성화한다. 활성화된 은닉벡터를 $\mathbf{h}$라 하자. $P(\mathbf{h}|\mathbf{x},W)$
3. 활성화된 은닉뉴런들로부터 입력층의 재생성 이미지 $\mathbf{x}'$를 생성한다. $P(\mathbf{x}|\mathbf{h},W)$
4. $\mathbf{x}$와 $\mathbf{x}'$의 차이를 줄여주는 방향으로 하이퍼넷의 구조와 파라미터 $W = \{(e_i, w_i) | i = 1,\ldots,M\}$를 조정한다.

$$
\begin{aligned}
H &\leftarrow H \cup \{e_i\} \\
w_i &\leftarrow w_i + \Delta w_i
\end{aligned}
$$

하이퍼넷은 하이퍼에지들의 멀티셋 $H$으로 표현될 수 있으며 이를 이용하여 위의 4 번 단계는 다음 세 가지의 연산을 통해서 구현된다.

1. 새로운 하이퍼에지의 추가:
   $\text{If } x_i = 1 \wedge x_i' = 0, \text{ then } x_i$들로만 구성된 하이퍼에지 $e_k$를 새로 만들어 $H$에 추가한다.
2. 하이퍼에지 가중치 값의 변경:
   $\text{If } x_i = 1 \wedge x_i' = 0, \text{ then } w_k \leftarrow w_k + \varepsilon \text{ for } e_k \text{containing } x_i$
   $\text{If } x_i = 0 \wedge x_i' = 1, \text{ then } w_k \leftarrow w_k - \varepsilon \text{ for } e_k \text{containing } x_i$
3. 하이퍼에지의 제거: $w_k \le w_{\min}$인 하이퍼에지를 제거한다.

## 6.3 하이퍼넷 응용 사례

하이퍼넷은 확률 기반 연상 메모리적 특성을 가진 머신러닝 방법론이며, 이러한 특성을 기반으로 패턴 분류(classification), 그림/텍스트/음악 생성(generation), 다중모달 정보의 고차 관계 탐색(retrieval, discovery), 인지 모델링(cognitive model) 등에 응용되었다.

하이퍼넷은 데이터를 표현하는 feature 의 고차 연관 관계를 고려한 패턴 분류 기법으로서 장점이 있으며, 필기체 숫자[2], 질병 예측[3], 문서 내 패턴 인식[4] 등에 적용된 바 있으며, 최근에는 다중 레이블 분류 문제를 위한 해법이 제시되었다[5].

하이퍼넷의 학습 및 추론 알고리즘은 은닉 노드를 포함하는 경우와 은닉 노드 없이 데이터를 표현한 feature vector 만으로 구성하는 경우가 있다. 후자의 경우 하이퍼엣지의 집합으로 표현되는 모델 내에서의 병렬적 계산 과정을 통한 패턴 인식 및 추론이 가능하며, 이러한 알고리즘의 특성으로 인해 DNA 컴퓨팅, 분자 컴퓨팅 환경에서 머신러닝 알고리즘 구현을 위한 프레임웍으로도 연구되고 있다[6][7].

하이퍼넷은 순서가 있는 데이터에서의 패턴 분류와 생성에도 적용되었다. 주가의 상승/하락 예측[8], 음악 생성[9], 대화 구문[10] 등의 사례가 있다.

하이퍼넷의 확률연상메모리적 특성을 가장 잘 보여주는 사례는, '사진-문장' 쌍과 같이 둘 이상의 표현이 다른 정보 간의 연관 관계를 학습하여 활용하는 '다중모달' 문제에 대한 응용이다. 한 예로, 사진과 문장 간의 상호 연상을 통한 검색과 생성 문제 적용 사례가 있다. 잡지 기사에 담긴 사진과 본문 문장 사이의 연관 검색[11], 설명에 기반한 사진 탐색[12], 드라마 정지영상과 대사 간의 연관 검색[13] 및 생성[14] 등이 주목할 만한 결과이며, 다음 절에서 소개할 딥 하이퍼넷으로 확장되는 연장선상에 있는 연구 결과이다. 생물정보학 및 의료 정보에서의 다중모달 연관관계 탐색에 적용된 사례와[15][3], 지속적 학습을 수행하는 인지 에이전트 모델링에 적용된 사례도 주목할 만하다[16][17][18].

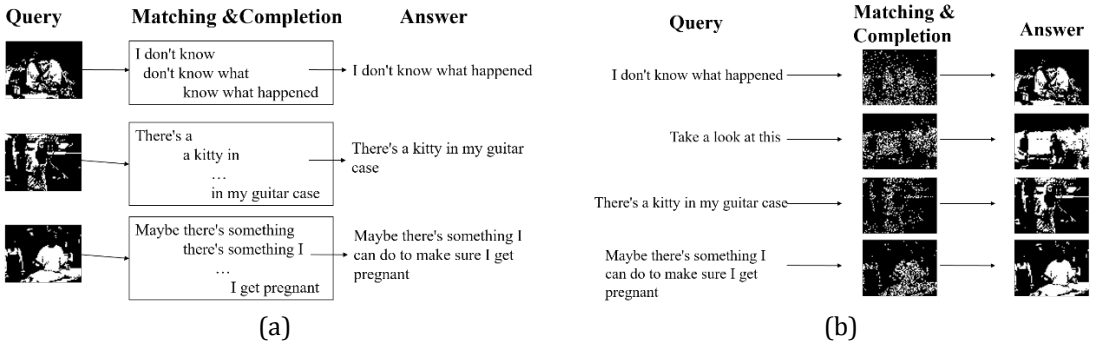



그림 6-3. 하이퍼넷 응용 사례. Videome 프로젝트. 드라마 이미지(image)와 대사(text) 간 연관성을 학습하고, (a) 이미지를 질문으로 한 대사 연상(I2T) (b) 대사를 질문으로 한 이미지 연상 작업을 수행(T2I) [13]

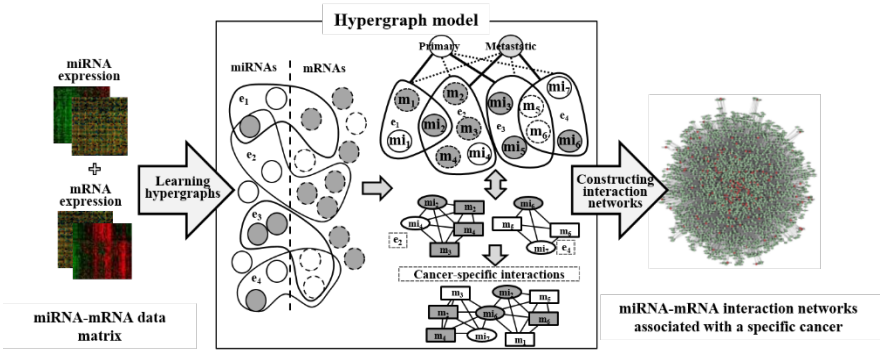



그림 6-4. 하이퍼넷을 이용한 다중 모달 연관관계 탐색 사례. 암 진단을 위해 두 형태의 유전자 발현 패턴을 하이퍼넷을 이용하여 종합 분석하고, 암의 유전적 특징 발견에 기여[3]

## 6.4 딥 하이퍼넷

딥 하이퍼넷은 하이퍼넷을 빌딩블록으로 하여 다층을 스택킹한 딥러닝 구조이다.
딥 하이퍼넷(deep hypernetwork, DHN)은 생성적 딥러닝 모델이다[19]. 먼저 한 개의 은닉층만을 가진 단순한 하이퍼넷 구조(아래에 설명)를 은닉변수 모델로 생각해보면, 식으로는 다음과 같다.

$$P(\mathbf{x}) = \sum_{\mathbf{h}_1} P(\mathbf{x}|\mathbf{h}_1)P(\mathbf{h}_1)$$

딥하이퍼넷의 핵심 아이디어는 (다른 딥러닝 생성 모델과 마찬가지로) 은닉층 $\mathbf{h}_1$을 모두 탐색하는 대신에 또 다른 하나의 은닉층을 상위층에 둠으로써 계층적으로 층을 쌓아나가는 것이다. 즉 $\mathbf{h}_1$층 위에 $\mathbf{h}_2$층을 하나 더 쌓음으로써 다음과 같이 식을 확장한다.

$$\begin{aligned} P(\mathbf{x}) &= \sum_{\mathbf{h}_1} P(\mathbf{x}|\mathbf{h}_1)P(\mathbf{h}_1) \\ &= \sum_{\mathbf{h}_1} P(\mathbf{x}|\mathbf{h}_1) \sum_{\mathbf{h}_2} P(\mathbf{h}_1|\mathbf{h}_2)P(\mathbf{h}_2) \\ &= \sum_{\mathbf{h}_2} \sum_{\mathbf{h}_1} P(\mathbf{x}|\mathbf{h}_1)P(\mathbf{h}_1|\mathbf{h}_2)P(\mathbf{h}_2) \end{aligned}$$

위 식에서 마지막 등호는 $\mathbf{h}_1$와 $\mathbf{h}_2$의 공간이 공통적인 전체 은닉벡터 공간이라는 가정에 기반한다. 이와 같은 방식으로 층을 쌓아서 결국 관측된 학습 데이터의 우도를 $n$개의 은닉층을 사용하여 확장된 계층적 은닉 표현 구조 즉 딥하이퍼넷 구조로 표시할 수 있다.

$$P(\mathbf{x}) = \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{x}|\mathbf{h}_1)P(\mathbf{h}_1|\mathbf{h}_2)\cdots P(\mathbf{h}_{n-1}|\mathbf{h}_n)P(\mathbf{h}_n)$$

다음 절에서 유도하는 바와 같이 이 식은 다음과 같이 다시 쓸 수 있다.

$$P(\mathbf{x}) = \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{h}_n|\mathbf{h}_{n-1})\cdots P(\mathbf{h}_2|\mathbf{h}_1)\cdots P(\mathbf{h}_1|\mathbf{x})P(\mathbf{x})$$

위와 같이 상향 추론과 하향추론이 같다는 것을 이용하여 다양한 추론과 학습이 가능하다.

형식화하면, 딥하이퍼넷은 관측된 데이터 변수들 $\mathbf{x} = (x_1, x_2, ..., x_V)$의 결합확률분포를 표현하는 다층구조의 확률그래프 모델이다. 관측 변수들의 값 벡터를 $\mathbf{x}$, 은닉 변수들의 값 $h_i$들의 벡터를 $\mathbf{h}$라 하자.

$$\mathbf{x} = (x_1, x_2, ..., x_V)$$
$$\mathbf{h} = (h_1, h_2, ..., h_H)$$

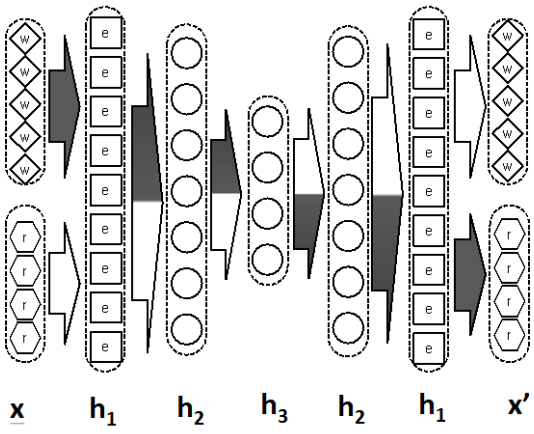



그림 6-5. **딥하이퍼넷의 구조와 확률전파 과정**

은닉층의 벡터값들을 층을 표시하는 인자를 사용하여 $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n$ 로 표시하면, 데이터의 확률분포는 앞에서 살펴본 바와 같이 딥구조 형태로 다시 쓸 수 있다.

$$P(\mathbf{x}) = \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{x} \mid \mathbf{h}_1) P(\mathbf{h}_1 \mid \mathbf{h}_2) \cdots P(\mathbf{h}_{n-1} \mid \mathbf{h}_n) P(\mathbf{h}_n)$$

$P(\mathbf{h}_s)$는 $s$ 번째의 하이퍼넷 층의 은닉뉴런들 $h_i^{(s)}$의 결합확률분포를 나타낸다. 하이퍼넷을 통계물리 시스템으로 보자면 그 상태 에너지 함수 $E(\mathbf{h}_s)$로부터(아래에 정의됨) 확률분포는 소프트맥스(softmax)를 사용하여 다음과 같이 기술할 수 있다.

$$P(\mathbf{h}_s) = \frac{\exp(-E(\mathbf{h}_s))}{\sum_j \exp(-E(\mathbf{h}_j))}$$

확률의 기본법칙인 $P(\mathbf{x} \mid \mathbf{y}) = P(\mathbf{x}, \mathbf{y}) / P(\mathbf{y})$을 사용하면, $s$ 번째 뉴런층에 대한 $s$-1 번째 뉴런층의 조건부 확률분포 $P(\mathbf{h}_s \mid \mathbf{h}_{s-1})$는 다음과 같이 계산될 수 있다.

$$P(\mathbf{h}_s \mid \mathbf{h}_{s-1}) = \frac{\exp(-E(\mathbf{h}_s, \mathbf{h}_{s-1}))}{\exp(-E(\mathbf{h}_{s-1}))}$$

하이퍼넷은 에너지 함수는 $E(\mathbf{h}_j) = h(s(\mathbf{h}_j))$로 표시할 수 있으며, 여기서 $h(\cdot)$는 뉴런의 활성화 함수이며(아래에 설명) $s(\mathbf{h}_j) = \sum_{i1} w_{i1}^{(j)} h_{i1}^{(j)} + \sum_{i1,i2} w_{i1i2}^{(j)} h_{i1}^{(j)} h_{i2}^{(j)} + \cdots$이다.

$$P(\mathbf{h}_s) = \frac{\exp(-E(\mathbf{h}_s))}{\sum_j \exp(-E(\mathbf{h}_j))}$$

$$E(\mathbf{h}_j) = h(s(\mathbf{h}_j))$$

$$s(\mathbf{h}_j) = \sum_{i1} w_{i1}^{(j)} h_{i1}^{(j)} + \sum_{i1,i2} w_{i1i2}^{(j)} h_{i1}^{(j)} h_{i2}^{(j)} + \cdots + \sum_{i1,i2,\cdots,ik} w_{i1\_ik}^{(j)} h_{i1}^{(j)} \cdots h_{ik}^{(j)}$$

위의 식에서 에너지함수를 구성하는 $s(\mathbf{h}_j)$에 있는 합의 항들은 각각 2 차부터 $k$차까지의 하이퍼에지를 갖는 하이퍼넷 구조를 기술한 것이다. 하이퍼에지를 모두 다 포함하는 것은 항수가 폭발적으로 증가하기 때문에 딥하이퍼넷 학습 알고리즘은 중요한 항을 선별하여 간소한 모델 구조를 찾아 낸다. 아래에서 살펴보겠지만 여기에 진화 알고리즘적인 탐색기법에 의한 최적화 방법을 사용한다.

딥하이퍼넷은 다른 딥러닝 구조와는 달리 각각의 뉴런층이 fully connected 된 구조가 아닌 고차 희소 그래프 구조의 하이퍼넷 형태를 취한다[1]. 즉, CNN 이나 DBN 이 뉴런층의 아래층 뉴런들을 선형으로 결합하는 구조를 갖는데 반해서

$$s(\mathbf{x}) = \sum_i w_i x_i$$

DHN 은 아래층 뉴런들의 하이퍼그래프 구조로 새로운 층을 형성한다. 하이퍼그래프 구조를 형성하는 하이퍼에지가 뉴런들의 곱을 형성하며, 예를 들어서 최대 세 개까지의 뉴런을 포함하는 차수 3 의 하이퍼에지를 갖는 하이퍼그래프는 다음식으로 표현된다.

$$s(\mathbf{x}) = \sum_i w_i x_i + \sum_{i,j} w_{ij} x_i x_j + \sum_{i,j,k} w_{ijk} x_i x_j x_k$$

즉, 딥하이퍼넷은 하이퍼넷을 다층으로 적층한 딥러닝 모델로서 각각의 하이퍼넷 층은 아래층의 하이퍼넷 유닛들을 결합하여 새로운 하이퍼에지를 만들고 이를 새로운 유닛으로 사용한다. 새로운 유닛으로 출력을 생성하는 방법은 기존의 다른 딥러닝 방식과 마찬가지고 다양한 활성화 함수를 사용할 수 있다. 즉 시그모이드(sigmoid) 출력함수를 사용할 경우, 입력의 총합 $s = s(\mathbf{x})$으로부터 다음과 같이 출력값을 계산한다.

$$h(s) = \frac{1}{1+\exp(-s)}$$

ReLU(rectified linear unit)를 사용할 경우는 다음 식을 계산한다.

$$h(s) = \max(0, s)$$

이 두 함수 외에 딥하이퍼넷 모델은 응용에 따라서 다음과 같은 이진 함수나 선형함수 및 가우시안 함수 등을 사용할 수 있다.

$$h(s) = \begin{cases} 1, & \text{if } s > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$h(s) = \exp\left(\frac{-|s|^2}{\sigma^2}\right)$$

$$h(s) = s$$

하이퍼에지를 생성하는 방식에 따라서 다양한 구조의 하이퍼넷이 구성된다. 앞에서도 논의한 바와 같이 하이퍼에지를 생성하는 방식은 도메인 지식을 사용하는데, DBN 과는 달리 완전연결된 구조를 갖지 않고 간략한 구조를 갖는다. CNN 과는 달리 정규 구조의 커널을 사용하지 않은 점도 하이퍼넷이 다른 점이다. 어떤 커널을 사용하는지 그 자체를 학습하는 것을 하이퍼넷에서는 중요한 학습문제로 본다.

### 6.4.1. 딥하이퍼넷 학습 알고리즘

딥하이퍼넷 모델 $W$를 학습하는 문제는 결국 모델 $W$로부터 데이터 $\mathbf{x}$가 생성될 확률 즉 모델의 우도

$$P(\mathbf{x} \mid W) = \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{x} \mid \mathbf{h}_1) P(\mathbf{h}_1 \mid \mathbf{h}_2) \cdots P(\mathbf{h}_{n-1} \mid \mathbf{h}_n) P(\mathbf{h}_n)$$

를 최대화하도록 딥하이퍼넷의 하이퍼에지 구조와 연결 가중치의 값들 $W$을 포함하는 다음 식

$$s(\mathbf{h}_j) = \sum_{i1} w_{i1}^{(j)} h_{i1}^{(j)} + \sum_{i1,i2} w_{i1i2}^{(j)} h_{i1}^{(j)} h_{i2}^{(j)} + \cdots$$

을 조정하는 과정으로 볼 수 있다. 이를 위해서 각 층에 있는 하이퍼에지의 종류와 갯수 및 가중치를 진화적 탐색을 통해서 변형하면서 더욱 좋은 해가 나오면 받아들이고 그렇지 않으면 버리는 방식을 취한다. 이 과정은 하이퍼그래프의 탐색 공간에서 몬테칼로 시뮬레이션하는 통계적인 방법과 유사하며 Graph Monte Carlo (GMC) 로 불린다[20]. 이를 구현하는 방법을 요약하면 다음과 같다. 보다 구체적인 것은 다음 절에서 만화영화 비디오 학습 예를 통해서 다시 설명될 것이다.

1) 새로운 학습예 $\mathbf{x}$를 가져온다. 입력층에 관측 변수값들을 할당한다.

2) 이로부터 상위층으로 확률값 $P(\mathbf{h}_1|\mathbf{x})$을 계산하고 변수값을 순차적으로 할당하며 $P(\mathbf{h}_2|\mathbf{h}_1)$를 계산하면 이를 반복하여 최상위 은닉변수 값을 $P(\mathbf{h}_n|\mathbf{h}_{n-1})$로부터 할당한다.
3) 최상위의 확률분포 $P(\mathbf{h}_n)$로부터 은닉변수 값을 항당하고 이를 거꾸로 하위층으로 전파하면서 확률값 $P(\mathbf{h}_{n-1}|\mathbf{h}_n)$을 계산하고 은닉변수값들을 순차적으로 할당하며 최하위층의 확률 분포 $P(\mathbf{x})$를 추정하고 이로부터 변수값 즉 관측 변수값 $\mathbf{x}'$을 할당한다.
4) 관측변수 벡터와 생성한 관측변수 벡터를 비교하여 그 차이를 줄여주는 방향으로 중간 은닉층들에 있는 하이퍼에지의 구성과 그 가중치 $W$를 변경한다.

$$\frac{\nabla}{\nabla W^{(k)}_{i_1 i_2 \cdots i_k}} \ln \prod_{n=1}^{N} P\left(\mathbf{x}^{(n)}|W\right) = N\left\{\left\langle x_{i_1} x_{i_2} \cdots x_{i_k}\right\rangle_{Data} - \left\langle x_{i_1} x_{i_2} \cdots x_{i_k}\right\rangle_{P(x|W)}\right\}$$

5) 위의 4 의 과정을 $N$ 번 반복한다.

이 알고리즘을 구현하는 데는 다양한 변형이 있을 수 있다. 예를 들어, DBN 에서 사용하는 방법과 같이 층별로 먼저 아래층을 학습 후 상위층으로 전파하는 Layer-wise Pre-training 방법을 사용할 수 있다. 이 경우 학습 예가 하나 대신에 여러 개를 한꺼번에 사용하는 Minibatch 방법을 사용할 수 있다. 단, DHN 은 기본적으로 학습예가 스트림으로 들어오는 문제에서 점진적 학습을 하도록 고안된 점을 고려하여 아주 작은 사이즈의 미니배치를 사용한다.

위의 학습 절차에서 확률값을 계산하는 방법과 은닉변수들을 결정하는 것을 여러 번의 값을 할당하고 순차적으로 계산하며 이 과정을 여러번 반복하는 방법을 사용한다는 데에 주목하자. 이는 정확한 베이지안 확률 계산이 실제로는 불가능하기 때문에 근사하는 방법이며, 딥하이퍼넷은 하이퍼에지의 랜덤 생성과 선택적 결합 등을 이용한 진화탐색에 기반한 몬테칼로 시뮬레이션으로 이를 구현한다. 이러한 그래프 몬테칼로 방법이 실제로 얼마나 유용한지는 다음 절에서의 응용예를 통해 실험적으로 보여줄 것이다.

이론적으로 위의 알고리즘 스텝 2 와 3 에서 상향, 하향 추론에 의해서 구현될 수 있음을 보이기 위해서 다음 식을 변형해 보자.

$$P(\mathbf{x}) = \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{x}|\mathbf{h}_1)P(\mathbf{h}_1|\mathbf{h}_2)\cdots P(\mathbf{h}_{n-1}|\mathbf{h}_n)P(\mathbf{h}_n)$$

베이스 규칙으로 조건부 확률의 순서를 바꿀 수 있다는 것에 착안하여 위의 식의 각 항에 다음과 같이 베이스 규칙을 적용하자.

$$P(\mathbf{x}|\mathbf{h}_1) = \frac{P(\mathbf{h}_1|\mathbf{x})P(\mathbf{x})}{P(\mathbf{h}_1)}$$

$$P(\mathbf{h}_1|\mathbf{h}_2) = \frac{P(\mathbf{h}_2|\mathbf{h}_1)P(\mathbf{h}_1)}{P(\mathbf{h}_2)}$$

$$\cdots$$

$$P(\mathbf{h}_{n-1}|\mathbf{h}_n) = \frac{P(\mathbf{h}_n|\mathbf{h}_{n-1})P(\mathbf{h}_{n-1})}{P(\mathbf{h}_n)}$$

원래의 딥하이퍼넷의 우도 식은 다음과 같이 다시 쓸 수 있다.

$$
\begin{aligned}
&P(\mathbf{x}) \\
&= \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{x}\,|\,\mathbf{h}_1) P(\mathbf{h}_1\,|\,\mathbf{h}_2) \cdots P(\mathbf{h}_{n-1}\,|\,\mathbf{h}_n) P(\mathbf{h}_n) \\
&= \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} \frac{P(\mathbf{h}_1\,|\,\mathbf{x})P(\mathbf{x})}{P(\mathbf{h}_1)} \frac{P(\mathbf{h}_2\,|\,\mathbf{h}_1)P(\mathbf{h}_1)}{P(\mathbf{h}_2)} \cdots \frac{P(\mathbf{h}_n\,|\,\mathbf{h}_{n-1})P(\mathbf{h}_{n-1})}{P(\mathbf{h}_n)} P(\mathbf{h}_n) \\
&= \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{h}_1\,|\,\mathbf{x}) P(\mathbf{x}) P(\mathbf{h}_2\,|\,\mathbf{h}_1) \cdots P(\mathbf{h}_n\,|\,\mathbf{h}_{n-1}) \\
&= \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{h}_n\,|\,\mathbf{h}_{n-1}) \cdots P(\mathbf{h}_2\,|\,\mathbf{h}_1) \cdots P(\mathbf{h}_1\,|\,\mathbf{x}) P(\mathbf{x})
\end{aligned}
$$

이와 같이 우도 계산이 상향 추론과 하향 추론의 어떤 형태로 계산해도 동등함을 알 수 있다.

마지막으로, 딥하이퍼넷을 감독학습 모델로 사용하는 경우 학습 방법과 추론 방법을 살펴보다. 감독학습은 입력벡터 $\mathbf{x}$ 로부터 출력벡터 $\mathbf{y}$ 을 예측하는 것이므로, 은닉층을 사용하여 풀어 쓰면 다음과 같이 표현된다.

$$
\begin{aligned}
P(\mathbf{y}\,|\,\mathbf{x}) &= \sum_{\mathbf{h}} P(\mathbf{y}, \mathbf{h}\,|\,\mathbf{x}) \\
&= \sum_{\mathbf{h}} P(\mathbf{y}\,|\,\mathbf{h}) P(\mathbf{h}\,|\,\mathbf{x}) \\
&= \sum_{\mathbf{h}_n} \cdots \sum_{\mathbf{h}_1} P(\mathbf{y}\,|\,\mathbf{h}_n) P(\mathbf{h}_n\,|\,\mathbf{h}_{n-1}) \cdots P(\mathbf{h}_2\,|\,\mathbf{h}_1) P(\mathbf{h}_1\,|\,\mathbf{x})
\end{aligned}
$$

즉 학습된 딥하이퍼넷의 입력단에 관측 벡터 $\mathbf{x}$를 넣고 이로부터 은닉변수들의 값을 순차적으로 구한 후 최상위층의 은닉변수층에 연결된 출력벡터값 $\mathbf{y}$의 확률을 계산하면 된다. 생성모델의 특성을 살려서 출력으로부터 입력을 생성해 낼수도 있다. 즉 먼저 최상위층에 출력벡터 $\mathbf{y}$를 할당한 후 하위층으로 확률을 전파하여 입력벡터 $\mathbf{x}$의 확률을 계산할 수도 있다.

## 6.5 딥 하이퍼넷 응용 사례

딥모델 중 CNN 은 영상 패턴인식 능력에서 좋은 성능을 보인다. 그러나 인간수준의 인공지능 실현을 위해서는 패턴인식뿐만 아니라 패턴회상 또는 패턴생성 능력이 더욱 중요하다. 과거의 기억을 되살려 새로운 정보를 생성해 내는 능력은 인간 지능의 기반이다. 생성적 학습모델은 이러한 것을 가능하게 한다. Hinton 은 DBN 을 이용하여 영상 생성이 가능함을 숫자 이미지 MNIST 데이터에 데모하였다[21]. 그러나 숫자 영상은 정적인 패턴이다. 모바일폰 데이터와 같이 개인의 일상 생활을 오랜동안 또는 평생학습하여 이를 재현하고 예측하는 것이 가능할까[22]? 기계가 TV 드라마를 보고 그 줄거리를 학습하여 새로운 드라마 줄거리를 만들어 내는 상상력 기계를 만들수 있을까[23]? DHN 은 이러한 인간수준의 인공지능 기계를 만들기 위한 인지메모리와 학습 모델로서 개발되었다. 앞에서 강조한 것처럼 DHN 은 스트림 형태로


그림 6-6. **딥하이퍼넷을 이용한 개념망 학습의 대표적인 사례로 뽀로로 애니메이션 학습 결과를 소개한다.**

들어오는 데이터로부터 새로운 개념을 형성하여 지식베이스를 자동으로 구축하고 이를 이용하여 스토리 생성과 같은 고급 정보를 예측하고 생성하는 딥러닝 모델이다. 이 절에서는 이러한 응용의 예로서 만화영화 비디오로부터 개념들의 지식망을 학습하고 이를 기반으로 상상력을 발휘하는 응용에 대한 실험 결과를 소개한다.

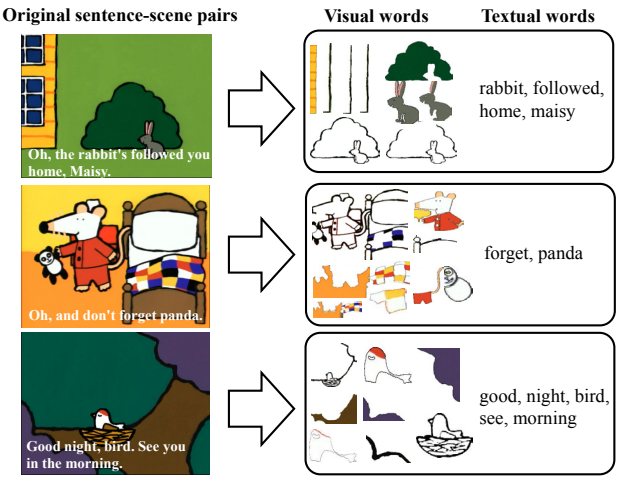


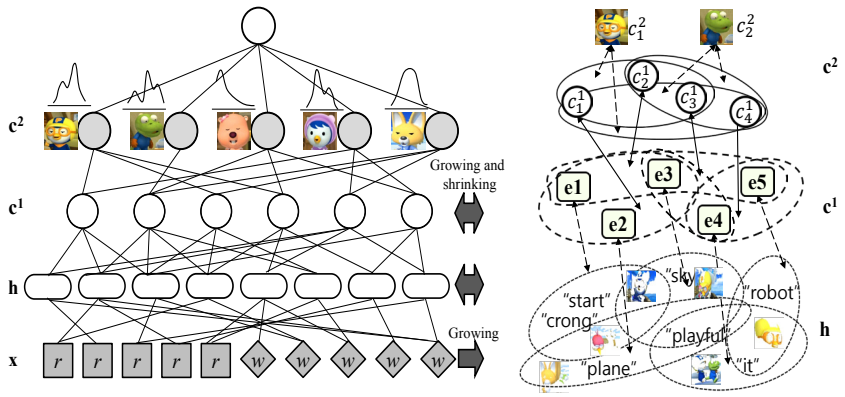

그림 6-7. **만화영화 비디오로부터 물체와 단어 추출**

아래 실험에서는 Pororo 만화 비디로를 학습 소재로 사용하였다(그림 6-6). 183 개의 에피소드를 DHN 에 의해 무감독 학습하였다. 실험의 목적은 DHN 구조가 이와 같이 시공간적인 멀티모달 데이터를 (배치 방식이 아닌) 온라인 학습에 의해서 끊임없이 학습 할 수 있도록 하고 학습된 지식구조를 이용하여 여러가지 응용을 데모하는데 있다. 응용의 예는 세가지이다. 하나는 학습된 비전-언어의 멀티모달 지식 구조로부터 언어(대사)를 입력으로 주면 비전(장면)을 생성하는 심상(mental imagery) 또는 상상력 기계(imagination machine)의 가능성을 데모하는 것이다. 다른 하나는 반대로 비전(장면)을 입력으로 주면 학습된 지식 구조를 이용하여 언어(대사)를 생성하는 것이다. 이는 시각 장면을 글로 기술하는 능력에 해당한다. 만약 로봇이 이러한 능력을 갖춘다면 이동하면서 장면을 말로 중계를 할 수 있을 것이다. 궁극적인 목표는 학습된 183 개의 에피소드로부터 184 번째의 에피소드를 기계가 생성하는 스토리텔링 기계를 개발하는 것이다.

비디오를 학습하는데 있어서, 모든 프레임을 다 학습하는 것은 현재 컴퓨팅 파워로 가능하지 않을 뿐만 아니라 사람도 그렇게 하지 않는다. 사람은 주의 집중에 의해서 장면과 프레임을 선별적으로 일부만을 학습한다. 본 연구에서는 대사가 나온 장면만을 학습에 사용하는 서브샘플링 방법을 사용하였다. 그림 6-7 은 비디오로부터 장면-대사의 쌍으로 구성된 비디오 스토리 학습 데이터를 만드는 과정이다. 대사가 나온 장면을 취하여 여기에 영상처리를 통해서 물체 인식을 수행한다. 이러한 처리를 통해서 DHN 의 학습 데이터 $\mathbf{x}$는 하나의 장면-대사 쌍에 나타난 물체들 $r_1, r_2, ..., r_n$과 단어들 $w_1, w_2, ..., w_m$의 조합로 기술된다($n$ 과 $m$ 은 비전과 언어의 각 어휘수).

$$\mathbf{x} = (\mathbf{r}, \mathbf{w})$$
$$\mathbf{r} = (r_1, r_2, ---, r_n)$$
$$\mathbf{w} = (w_1, w_2, ---, w_m)$$

그림 6-8 은 추출한 물체와 단어들로부터 개념망을 온라인 학습하는 딥하이퍼넷 구조를 보여준다. 제일 아래층은 그림 조각과 대사의 단어들로 구성된 두 가지 모달리티의 관측 변수들 $\mathbf{x}$ 이다. 그 다음층은 이들을 상호 결합하여 구성한 다양한 하이퍼에지들에 해당하는 은닉변수들 $\mathbf{h}$ 이다. 그 위에 이어지는 층들은 하이퍼에지들을 다시 결합함으로써 상위의 개념을 표현하는 은닉층들이다. 이 실험에서는 문제의 특성을 고려하여 추가의 은닉층 두 개를 을 도입하였으며 이를 각각 $\mathbf{c}^1, \mathbf{c}^2$ 로 표시하기로 한다. 이는 앞절에서의 $\mathbf{h} = \mathbf{h}_1$, $\mathbf{c}^1 = \mathbf{h}_2$, $\mathbf{c}^2 = \mathbf{h}_3$ 에 해당한다.




(a) Example of deep concept hierarchy learned from Pororo videos (b) Hypergraph representation of (a)

그림 6-8. **비디오 학습을 위한 딥하이퍼넷 구조**

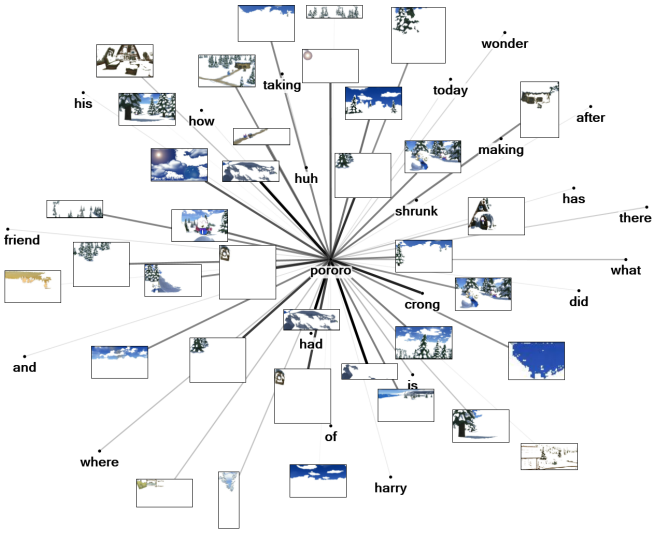



**그림 6-9. 딥하이퍼넷으로 학습한 개념망 구조**

마지막의 최상위층 $\mathbf{c}^2$은 비디오에 등장하는 인물들을 나타낸다. 뽀로로 만화영화의 경우 마지막 층은 뽀로로, 통통 등의 등장인물들이다. 아래의 스토리 학습 실험에서는 등장 인물을 라벨링해서 학습에 사용하는 감독학습 방법을 사용하였다. 다른 모든 실험에서는 입력층에만 관측 데이터가 할당되고 은닉층에 있는 하이퍼에지들은 관측데이터를 재생성하도록 하는 무감독 학습 원리를 따른다. 이는 표상 학습(representation learning)을 하고자 하는 일반적인 딥러닝 철학과 일치하며, 층을 순차적으로 추가함에 따라서 표상의 점차 복잡해지게 된다. 마지막 층에 이름을 부여하면 이는 언어적 카테고리화 과정으로 볼 수 있다.

학습의 결과는 그림 6-9 와 같은 개념망 구조이다. 이 그림은 학습된 딥하이퍼넷으로부터 가중치가 높은 연결선을 가진 하이퍼에지들에 나타난 개념들만을 연결한 개념신경망 구조이다. 이 구조는 어떤 개념(언어)들이 어떤 물체(영상)과 상호 연상작용을 강하게 일으키는지를 간접적으로 알수 있는 일종의 멀티모달 시맨틱 네트워크이다. 중요한 것은 이러한 개념망이 비디오 데이터로부터 자동으로 구성되었으며 비디오를 관찰함에 따라서 점진적으로 재구성된다는 것이다. 마치 어린아이들이 만화영화를 보면 머리속에 새로운 개념을 학습하며 인지적인 발달 능력을 키워가는 것과 같다[17].

딥하이퍼넷을 이용하여 비디오 장면이 하나씩 늘어나면서 개념신경망이 발달해 나가는 과정에 대한 학습 알고리즘을 논문에 상세히 기술되어 있다[20]. 여기서는 그 절차를 요약해서 간략히 설명한다. 전체 학습 과정은 베이지안 추론 과정이다. 즉 현재까지의 개념 지식 즉 사전 확률분포에 기반하여 관측된 데이터 즉 비디오 장면을 기반으로 우도를 측정하고 이 둘을 결합하여 개념들에 대한 확률분포 즉 사후확률분포를 갱신해 가는 과정이다. 등장 인물들의 변화를 통한 스토리텔링을 학습하는 과정을 예로 들어 기술한다. 이 경우 관측변수는 딥하이퍼넷의 최하단 입력벡터 $\mathbf{x}=(\mathbf{r},\mathbf{w})$과 최상단의 출력벡터 $\mathbf{y}=\mathbf{c}^2$이다. 즉

$$\mathbf{x}=(\mathbf{r},\mathbf{w})$$
$$\mathbf{y}=\mathbf{c}^2$$

학습은 다음과 같이 베이스 추론 과정으로 볼 수 있다.

$$P_t(\mathbf{h},\mathbf{c}^1\,|\,\mathbf{r},\mathbf{w},\mathbf{c}^2)=\frac{P(\mathbf{r},\mathbf{w}\,|\,\mathbf{h},\mathbf{c}^1,\mathbf{c}^2)P(\mathbf{c}^2\,|\,\mathbf{c}^1,\mathbf{h})P_{t-1}(\mathbf{h},\mathbf{c}^1)}{P(\mathbf{r},\mathbf{w},\mathbf{c}^2)}$$

$$P(\mathbf{r},\mathbf{w},\mathbf{c}^2)=\iint_{\mathbf{h},\mathbf{c}^1}P(\mathbf{r},\mathbf{w}\,|\,\mathbf{h},\mathbf{c}^1,\mathbf{c}^2)P(\mathbf{c}^2\,|\,\mathbf{c}^1,\mathbf{h})P_{t-1}(\mathbf{h},\mathbf{c}^1)d\mathbf{h}\,d\mathbf{c}^1$$

여기서 $P_{t-1}(\mathbf{h},\mathbf{c}^1)$은 데이터 $\mathbf{r},\mathbf{w},\mathbf{c}^2$를 관측하기 전의 사전 확률에 해당하고 $P_t(\mathbf{h},\mathbf{c}^1|\mathbf{r},\mathbf{w},\mathbf{c}^2)$은 데이터를 관측한 후의 사후 확률에 해당한다. 그러나 이 식에서 분모의 $P(\mathbf{r},\mathbf{w},\mathbf{c}^2)$를 계산하는 것은 실제로 불가능하다. 모든 가능한 가설공간 $d\mathbf{h}\,d\mathbf{c}^1$에 대해서 적분을 수행해야 하기 때문이다. 모든 베이지안 학습 방법이 이 문제에 봉착한다. 딥하이퍼넷의 핵심 아이디어는 이 가설공간을 Sparse Population Coding 기법과 진화탐색 방법을 결합한 몬테칼로 시뮬레이션을 통해서 효율적으로 근사하는 것이다[20]. 이를 위해서 경험적인 분포를 사용한다.

$$P_t(\mathbf{h},\mathbf{c}^1|\mathbf{r},\mathbf{w},\mathbf{c}^2) \propto \prod_{d=1}^{D_t}\left\{P(\mathbf{r}^{(d)},\mathbf{w}^{(d)}|\mathbf{h},\mathbf{c}^1,\mathbf{c}^2)P(\mathbf{c}^2|\mathbf{c}^1)P(\mathbf{c}^1|\mathbf{h})P_{t-1}(\mathbf{h})\right\}$$

여기서 $D$ 는 학습예의 갯수이다. 온라인 학습의 경우 $D=1$마다 식이 갱신된다. 첫번째 항은 모델로부터 관측 데이터가 생성될 확률 즉 우도를 나타내며 이의 로그를 취한 로그우도는 다음과 같다.

$$\log P(\mathbf{r}^{(d)},\mathbf{w}^{(d)}|\mathbf{c}^2,\mathbf{c}^1,\mathbf{h}) = \sum_{n=1}^{N}\log P(r_n^{(d)}|\mathbf{c}^2,\mathbf{c}^1,\mathbf{h}) + \sum_{m=1}^{M}\log P(w_m^{(d)}|\mathbf{c}^2,\mathbf{c}^1,\mathbf{h})$$

두 번째부터 네 번째 항은 다음과 같다.

$$P(w_m^{(d)}=1|\mathbf{c}^2,\mathbf{c}^1,\mathbf{h}) = \exp\left(s_m^{\mathbf{w}} - \sum_{i=1}^{|\mathbf{e}^{\mathbf{c}}|}\alpha_i\right)$$

$$P(r_n^{(d)}=1|\mathbf{c}^2,\mathbf{c}^1,\mathbf{h}) = \exp\left(s_n^{\mathbf{r}} - \sum_{i=1}^{|\mathbf{h}^{\mathbf{c}}|}\alpha_i\right)$$

$$\mathbf{s}^{\mathbf{w}} = \sum_{i=1}^{|\mathbf{e}^{\mathbf{c}}|}\alpha_i e_i^{\mathbf{w}} \text{ and } \mathbf{s}^{\mathbf{r}} = \sum_{i=1}^{|\mathbf{e}^{\mathbf{c}}|}\alpha_i e_i^{\mathbf{r}}$$

두 번째 항은 클러스터로부터 캐릭터를 예측하는 확률을 나타내고, 세 번째 항은 클러스터간의 유사도를 측정하며, 네번째 항은 모델 복잡도를 반영한다. 세 번째 항은 복잡도가 높은 모델 구조에 페널티를 줌으로써 확률을 낮게 하는 정규화 항의 역할을 한다. 이는 딥하이퍼넷과 같이 구조 학습을 하는 머신러닝 방법에서 모델 복잡도를 낮춤으로써 성능과 효율을 최대화하는 Occam's Razor 의 원리를 구현하는 아주 중요한 방법이다. 위 식에서 $e_i$ 는 하이퍼에지를 표시하고 $\alpha_i$ 는 그 가중치이다. 딥하이퍼넷에서 학습은 두 단계가 있다. 하나는 다양한 하이퍼에지를 구성적으로 탐색하는 구조학습이다. 이를 위해서 현재의 개체(하이퍼에지) 집합에서 관측된 데이터를 고려하여 새로운 개체(하이퍼에지)를 생성하는 베이지안 진화연산 기법을 사용한다[24]. 다른 하나는 하이퍼에지의 가중치를 결정하는 파라메터 학습이며 다음의 식에 의해서 점진적으로 수정된다.

$$\alpha_i = \sum_{d=1}^{D}\left\{g(e_i)f(\mathbf{r}^{(d)},\mathbf{w}^{(d)};e_i)\right\},\ \alpha_i^t = \lambda\alpha_i + (1-\lambda)\alpha_i^{t-1}$$

$$f(\mathbf{r}^{(d)},\mathbf{w}^{(d)};e_i) = \begin{cases} 1, \text{ if } \left(\mathbf{r}^{(d)}\cdot e_i^{\mathbf{r}} + \mathbf{w}^{(d)}\cdot e_i^{\mathbf{w}}\right)\Big/ e_i e_i^{\mathrm{T}} > \kappa \\ 0, \text{ otherwise} \end{cases}$$

위 식은 모델에 들어 있는 하이퍼에지를 통해서 생성한 데이터가 얼마나 관측한 데이터를 반영하는지를 측정하고 있다. 즉 딥하이퍼넷은 생성모델로서 관측한 데이터를 생성하는 능력이 좋을수록 하이퍼에지 가중치를 높여준다.

### 6.5.1 기계상상 실험 결과

| Query sentences | Episodes 1~52 (1 season) | Episodes 1~104 (2 seasons) | Episodes 1~183 (all seasons) |
|---|---|---|---|
| | | | |
| | |  | |

그림 6-10. **학습된 딥하이퍼넷에 의한 대사의 생성**

딥하이퍼넷에 의해 학습된 개념신경망은 뽀로로 만화영화 183 편에 들어 있는 일반적인 개념과 지식을 저장하고 있다. 이를 활용하는 세가지 실험을 하였다. 첫 번째 실험은 뽀로로 만화영화 중에서 나온 대사를 주고 그 장면을 떠오르게 하는 상상력 실험이다. 이는 기계가 사람과 같이 연상 작용에 의해서 심상을 떠 올릴 수 있는지에 대한 실험으로서, 딥하이퍼넷 신경망이 사람의 인지적인 연상기억 능력을 보이는지를 알아보기 위한 것이다. 그림 6-10 은 그 실험 결과의 예를 보여준다. 입력으로 “Tongtong, please change this book using magic. Kurikuri, Kurikuri Tongtong!” 이라고 하면 딥하이퍼넷은 구축된 개념신경망을 이용하여 그림에 보이는 것과 같은 영상들을 조합해 낸다. 이러한 상상력 능력이 학습이 진행됨에 따라 향상되는지를 알아보기 위해서 각각 52 편, 104 편, 183 편의 에피소드를 보았을 때 상상되는 심상의 그림을 비교하였다. 그림에 보이듯이 떠 올리는 심상의 영상 복잡도와 정확도가 학습이 진행됨에 따라서 향상되는 것을 알 수 있다. 이 결과는 아직 완전하지는 않지만 다양한 응용에 활용될 수 있다. 예를 들어, 상상된 그림을 질의어로 만화영화 전편을 검색할 경우 사람과 같은 인지적 순간 검색 능력을 가지는 교차 모달리티 비디오 검색 엔진을 만들 수 있을 것이다.

두 번째 실험에서는 반대로 정지 비디오 그림을 딥하이퍼넷의 입력으로 주었다. 그리고 딥하이퍼넷의 확률적 생성능력을 이용한 연상기억에 기반하여 언어를 생성하였다. 즉 주어진 그림을 설명하는 글을 생성할 수 있는지를 시험하였다. 그림 6-11 은 그 결과를 보여준다. 해당 장면에 대한 뽀로로 원본의 대사와 비교해 볼 때 유사하면서도 변형이 된 문장들이 생성되는 것을 알수 있다. 예를 들어서, 주어진 그림에 대한 원본 대사는 “Clock, I have made another potion come and try it” 인데 딥하이퍼넷이 생성한 문장은 “as i don't have the right magic potion come and try it was nice”와 “ah, finished i finally made another potion come and try it we'll all alone” 였다. 역시 완전한 문장은 아니지만 의미가 상당히 통하며 창의적인 문장이 생성되는

as if that's the ship is being pulled
- the ship is being pulled

그림 6-11. **학습된 딥하이퍼넷에 의한 장면의 생성**

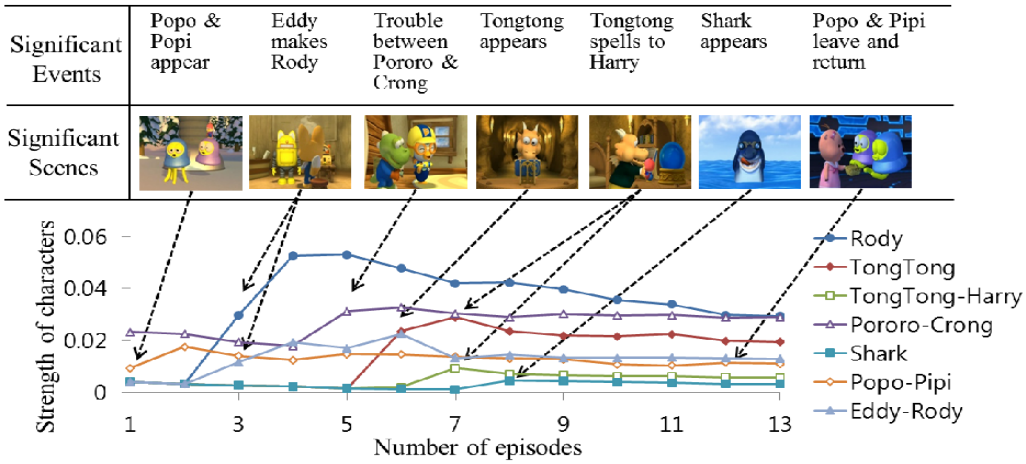



그림 6-12. **딥하이퍼넷 학습과정에서 만화 주인공들의 등장 패턴 변화**

것을 알수 있다. 이는 학습된 딥하이퍼넷이 주어진 그림에 들어 있는 그림 조각에 기반하여 확률적인 추론을 반복하여 단어를 생성하기 때문이다.

세 번째 실험은 뽀로로 장면 하나하나를 넘어서 시간의 흐름에 따른 에피소드 스토리를 학습할 수 있는지를 알아보고자 하였다. DHN 은 스트림으로 관측되는 데이터에 대해서 온라인 점진적 학습을 하기 때문에 시간적인 변화를 추적할 수 있다. 그림 6-12 는 에피소드가 진행됨에 따라

그림 6-13. **아이들과 만화비디오 보며 영어를 가르치는 Pororobot**

여러가지 사건이 발생되고 그에 따라 주인공들의 등장 빈도수가 달라지면 스토리 라인이 바뀌는 것을 딥하이퍼넷이 학습하고 있다는 것을 간접적으로 알수 있다. 이 실험의 결과에 힘입어 현재 만화영화를 보면서 아이들과 놀아주며 영어공부를 도와주는 로봇 Pororobot 을 개발하는 연구가 진행중이다(그림 6-13).

## 참고문헌


[1] B.-T. Zhang, “Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory,” *IEEE Comput. Intell. Mag.*, no. August, pp. 49–63, 2008.

[2] J.-K. Kim and B.-T. Zhang, “Evolving Hypernetworks for Pattern Classification,” in *IEEE Congress on Evolutionary Computation (CEC 2007)*, 2007, pp. 1856–1862.

[3] S.-J. Kim, J.-W. Ha, and B.-T. Zhang, “Bayesian evolutionary hypergraph learning for predicting cancer clinical outcomes,” *J. Biomed. Inform.*, vol. 49, pp. 101–111, 2014.

[4] J. Bootkrajang, S. Kim, and B.-T. Zhang, “Evolutionary hypernetwork classifiers for protein-proteininteraction sentence filtering,” in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO ’09*, 2009, p. 185.

[5] K. W. Sun, C. H. Lee, and J. Wang, “Multilabel Classification via Co-Evolutionary Multilabel Hypernetwork,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2438–2451, 2016.

[6] H. W. Lim, S. H. Lee, K. A. Yang, J. Y. Lee, S. I. Yoo, T. H. Park, and B.-T. Zhang, “In vitro molecular pattern classification via DNA-based weighted-sum operation,” *BioSystems*, vol. 100, no. 1, pp. 1–7, 2010.

[7] J.-H. Lee, C. Baek, H.-S. Chun, J.-H. Ryu, R. Deaton, and B.-T. Zhang, “Use of symmetric internal loops for molecular pattern classification,” in *International Conference on DNA Computing and Molecular Programming (DNA 20)*, 2014, p. 90.

[8] B. Elena, S. Kim, B. Andrei, H. Luchian, and B.-T. Zhang, “Evolving Hypernetwork Models of Binary Time Series for Forecasting Price Movements on Stock Markets,” in *IEEE Congress on Evolutionary Computation (CEC 2009)*, 2009, pp. 166–173.

[9] H.-W. Kim, B.-H. Kim, and B.-T. Zhang, “Evolutionary Hypernetworks for Learning to Generate Music from Examples,” in *IEEE International Confernece on Fuzzy Systems*, 2009, pp. 47–52.

[10] J.-H. Oh, H.-S. Chun, and B.-T. Zhang, “Generating Cafeteria Conversations with a Hypernetwork Dialogue Model,” in *Proceedings of the 14th International Symposium on Advanced Intelligent Systems (ISIS 2013)*, 2013, pp. 1424–1435.

[11] J.-W. Ha, B.-H. Kim, H.-W. Kim, W. Yoon, J.-H. Eom, and B.-T. Zhang, “Text-to-image cross-modal retrieval of magazine articles based on higher-order pattern recall by hypernetworks,” in *The 10th International Symposium on Advanced Intelligent Systems (ISIS 2009)*, 2009, no. Isis, pp. 274–277.

[12] J.-W. Ha, B.-J. Lee, and B.-T. Zhang, “Text-to-image retrieval based on incremental association via multimodal hypernetworks,” in *IEEE Conference on Systems, Man, and Cybernetics (IEEE SMC 2012)*, 2012, pp. 3245–3250.

[13] 장병탁, “SNU Videome Project : 인간수준의 비디오 학습 기술,” 정보과학회지, vol. 29, no. 2, pp. 17–31, 2011.

[14] J.-W. Ha and B.-T. Zhang, “Text-to-Image Generation based on Crossmodal Association with Hierarchical Hypergraphs,” in *2011 NIPS Workshop on Integrating Vision and Language*, 2011.

[15] S.-J. Kim, J.-W. Ha, and B.-T. Zhang, “Constructing higher-order miRNA-mRNA interaction networks in prostate cancer via hypergraph-based learning.,” *BMC Syst. Biol.*, vol. 7, no. 1, p. 47, 2013.

[16] U. Fareed and B.-T. Zhang, “MMG: A learning game platform for understanding and predicting human recall memory,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6232 LNAI, pp. 300–309, 2010.

[17] B.-T. Zhang, J.-W. Ha, and M. Kang, “Sparse Population Code Models of Word Learning in Concept Drift,” *Annu. Meet. Cogn. Sci. Soc.*, pp. 1221–1226, 2012.

[18] H. Kim and J. H. Park, “Hypergraph-based recognition memory model for lifelong experience,” *Comput. Intell. Neurosci.*, vol. 2014, 2014.

[19] B.-T. Zhang, “Deep hypernetwork models,” *Commun. Korean Inst. Inf. Sci. Eng.*, vol. 33, no. 8, pp. 11–24, 2015.

[20] J.-W. Ha, K.-M. Kim, and B.-T. Zhang, “Automated construction of visual-linguistic knowledge via concept learning from cartoon videos,” in *AAAI Conference on Artificial Intelligence (AAAI 2015)*, 2015.

[21] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science (80-. ).*, vol. 313, no. 5786, pp. 504–507, 2006.

[22] B.-T. Zhang, “Information-Theoretic Objective Functions for Lifelong Learning,” in *Lifelong Machine Learning: Papers from the 2013 AAAI Spring Symposium*, 2013, pp. 62–69.

[23] B.-T. Zhang, “Ontogenesis of Agency in Machines : A Multidisciplinary Review,” in *AAAI 2014 Fall Symposium on The Nature of Humans and Machines: A Multidisciplinary Discourse*, 2014.

[24] B.-T. Zhang, P. Ohm, and H. Muhlenbein, “Evolutionary induction of sparse neural trees,” *Evol. Comput.*, vol. 5, no. 2, pp. 213–36, 1997.