

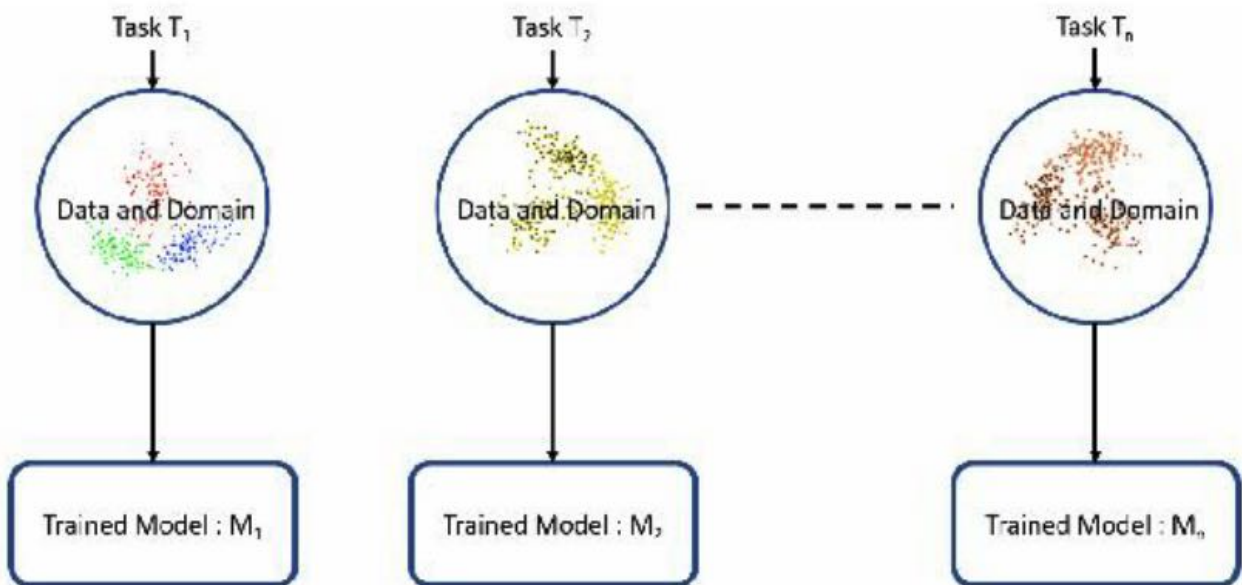
■ 전이학습(Transfer Learning)

○ 등장배경

사람은 task(과제)마다 지식을 전이할 수 있는 능력이 있다. 어떤 task로부터 지식을 얻고, 얻은 지식을 활용하여 그와 비슷한 task를 수행할 수 있다는 말이다. Task가 유사할수록 지식을 전이하고 활용하기는 더욱 쉬워진다. 기존의 기계학습과 딥러닝 알고리즘은 독립적으로 작동하도록 고안되어있다. 이 알고리즘들은 특정한 task만 수행하도록 학습한다. 만약 feature-space(피쳐 공간)의 분포가 바뀌면, 모델은 처음부터 새로 학습해야한다. Transfer Learning(전이학습)은 독립적으로 학습하는 패러다임을 극복하고 어떤 task로부터 얻은 지식을 비슷한 task에 활용하자는 아이디어이다. Transfer Learning의 개념과 딥러닝에서의 의의에 대해 알아보자.

○ 전이학습(Transfer Learning)

전통적으로, learning 알고리즘은 독립적으로 어떤 문제를 해결하도록 고안되어왔다. 사용처의 요구사항과 당면한 데이터에 따라서, 알고리즘은 모델이 주어진 특정한 task만 수행하도록 학습하는데 적용된다. 전통적인 기계학습은 특정한 domain, data, task에 기반해서 모델을 독립적으로 학습한다.



전통적인 기계학습

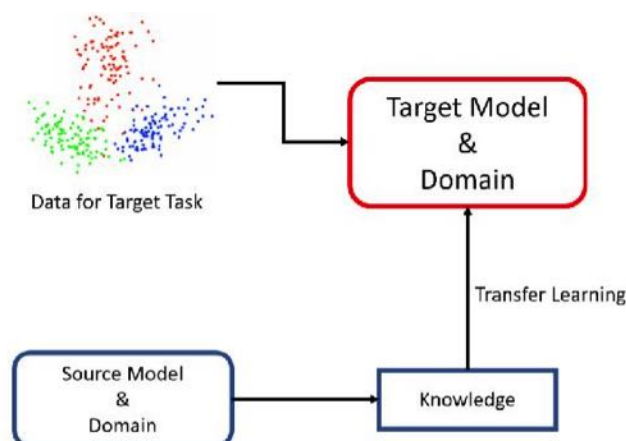
Transfer Learning은 학습 절차에서 한 스텝 더 나아가 사람이 지식을 활용하는 방법과 비슷하게 학습하는 방법이다. Transfer Learning은 비슷한 task를 수행하기 위해서 이전 모델 또는 지식을 재사용한다. Transfer Learning은 기존 기계학습 알고리즘의 확장판으로 여겨지기도 한다. 어떻게 task간에 지식을 전이할 수 있을지 이해하기 위해서 Transfer Learning에 대한 연구가 광범위하게 이루어졌지만, 1995년 Neural Information Processing System(NIPS)에서 발표된 *Learning to Learn: Knowledge Consolidation and*

*Transfer in Inductive Systems*가 이 분야 연구의 최초 태동으로 여겨진다. 그 때부터 Learning to Learn, Knowledge Consolidation, Inductive Transfer와 같은 용어들이 Transfer Learning과 함께 혼용되었다. 연구자와 학회들은 저마다의 배경에 맞추어 각자의 정의대로 용어를 사용하였다. Goodfellow의 저서, *Deep Learning*에서 Transfer Learning을 일반화하여 언급하였으며 다음과 같은 정의로 사용하였다.

“Situation where what has been learned in one setting is exploited to improve generalization in another setting.”

예시를 들어서 위의 정의를 이해해보자. 레스토랑이라는 domain에서, 사진 안에 있는 물체를 인식하는 task가 있다고 하자. 주어진 조건에서의 이 task를 T_1 이라고 하자. 이 task에 대한 dataset이 주어지고, 모델을 잘 학습시킨다. 잘 학습되었다는 것은 모델이 일반화가 되어서, 같은 domain(레스토랑)이지만 본 적 없는 data에 대해서도 성능이 좋은 상태를 말한다. 전통적인 지도-기계학습 알고리즘은 주어진 domain에서 요구되는 task를 학습하는데 데이터가 충분하지 않으면 학습이 되지 않는다. 이제부터 공원 또는 카페에서 찍은 사진으로부터 물체를 탐지하는 새로운 task, T_2 가 생겼다고 하자. T_1 에 대해서 학습한 model을 이용해서 해결된다면 이상적이겠지만, 현실에서는 성능저하와 일반화가 되지 않은 모델을 마주하게 될 것이다. 여기는 다양한 이유가 있는데, 흔히 모델의 bias(편향)가 학습 data와 domain에 치중되었기 때문이라고 표현한다. Transfer Learning은 이전 task로부터 학습한 지식을 비슷한 새로운 task에 활용할 수 있게 한다. 만약 T_1 에 대한 데이터가 상당히 많다면, 우리는 이 지식을 이용하여 일반화시켜 T_2 에 적합하게 만들 수 있다. T_2 에 대한 데이터가 더 적어도 가능하다. 이미지 분류를 예로 들자면, low level feature들 (edges, shapes, and lighting)은 task마다 공유될 수 있으므로 task별로 지식을 전이할 수 있다.

다음 그림을 통해 Transfer Learning이 어떻게 비슷하고 새로운 task에 기존 지식을 활용하는지 살펴보자.



위 그림에서 보았듯이, 기존 task로부터 얻은 지식은 target task를 학습할 때 추가 입력 정보로 활용된다.

1. Transfer Learning의 장점

Target task를 학습할 때 성능 향상을 위해서 소스 모델로부터 얻은 지식을 이용함으로써, target task 학습에 다음과 같은 효과를 얻을 수 있다.

▷ Improved baseline performance(기본 성능 향상)

- 고립된 모델의 지식을 증대시킬 때 소스 모델로부터 얻은 지식을 이용하면, 지식 전이 효과로 기본 성능이 향상된다.

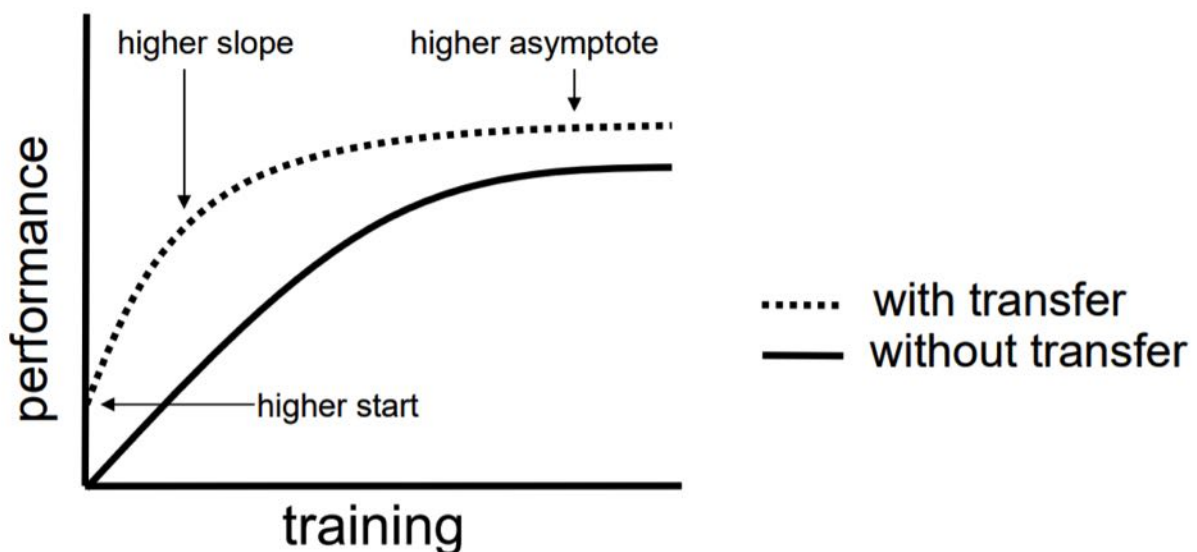
▷ Model-development time(모델 개발 시간)

- 소스 모델로부터 얻은 지식을 이용하는 것은 target task를 처음부터 학습하는 것에 대비해서 전체적으로 학습에 도움을 준다. 고로, 모델을 학습하는데 소요하는 시간도 전반적으로 단축되는 결과를 가져온다.

▷ Improved final performance(최종 성능 향상)

- Transfer Learning의 영향으로 최종적으로 더 높은 성능을 얻을 수 있다.

다음 그림을 통해서 기본 성능 향상(더 높은 시작점), 효율성(가파른 학습 곡선), 더 나은 최종 성능(더 높은 수렴지점)을 확인할 수 있다.



2. Transfer Learning 이해하기

Transfer Learning은 Neural networks(신경망)과 같은 Inductive learning(귀납 학습)과 Bayesian networks에도 적용할 수 있다. 즉, Transfer Learning 개념은 딥러닝에만 한정되지 않는다. 이번 장에서 Transfer Learning의 형식적 정의부터 시나리오, 주의할 점, 전통적 기계학습에의 적용 전략을 알아보자.

▷ Formal definition(형식적 정의)

- Transfer Learning의 형식적 정의를 살펴봄으로써 차후 알아볼 전략들을 더 잘 이해할 수 있다. Domain, task, marginal probabilities(주변확률)을 이용해서 Transfer Learning의 구조를 표현할 수 있다.
- Domain (D)는 feature space(χ)와 주변확률 $P(X)$ - X 는 sample data point임- 로 이루어진 원소가 2개인 tuple(튜플)이다. 임의의 벡터 x_i 에 대하여 $X = \{x_1, x_2, \dots, x_n\}$ 이고, $X \in \chi$ 이므로

$$D = \{\chi, P(X)\}$$

- 또한, Task (T)는 label space(γ)와 목적 함수 f 로 이루어진 원소가 2개인 tuple이다. 목적 함수는 feature vector/label((x_i, y_i) , $x_i \in \chi$, $y_i \in \gamma$) 쌍으로부터 학습되고, feature vector에 대응하는 label을 예측($f(x_i) = y_i$)한다. 목적함수는 확률론적 관점에서 $P(\gamma|X)$ 로 나타낼 수 있으므로

$$T = \{\gamma, P(Y|X)\} = \{\gamma, f\} \quad Y = \{y_1, \dots, y_n\}, y_i \in \gamma$$

- 이러한 구조를 사용함으로써, Transfer Learning을 D_S domain의 T_S source task로부터 얻은 지식을 이용하여 D_T domain의 target 목적함수(또는 target task T_T)를 향상시키려는 절차로 정의할 수 있다.

▷ Scenarios(시나리오)

- 위의 수식 구조로부터 다음과 같은 4가지 시나리오를 도출할 수 있다. 이해를 돕기 위해 문서 분류를 예를 들어서 살펴보자.
- $\chi_S \neq \chi_T$ Source domain의 feature space와 target domain의 feature space가 다른 경우이다. 예를 들어 문서들이 서로 다른 언어로 쓰여진 상황이다.
- $P(X_S) \neq P(X_T)$ Source domain과 target domain의 주변확률분포가 다른 경우이다. 예를 들어 문서들이 서로 다른 주제를 이야기하고 있는 상황이다. 이 시나리오는 일반적으로 Domain Adaption으로 알려져있다.
- $\gamma_S \neq \gamma_T$ Source task와 target task 간에 label space가 다른 경우이다. 예를 들어 문서들이 서로 다른 label이 붙은 상황이다. 실제로 이 시나리오는 다음 4번 시나리오와 거의 함께 일어난다. 두 task의 label이 다른데, 조건부 확률분포는 정확히 일치할 가능성은 희

박하기 때문이다.

- $P(Y_S|X_S) \neq P(Y_T|X_T)$ Source task와 target task 간에 조건부 확률분포가 다른 경우이다. 예를 들어 source 문서와 target 문서의 클래스가 서로 다른 경우이다. 실제로 이 시나리오가 자주 일어난다.

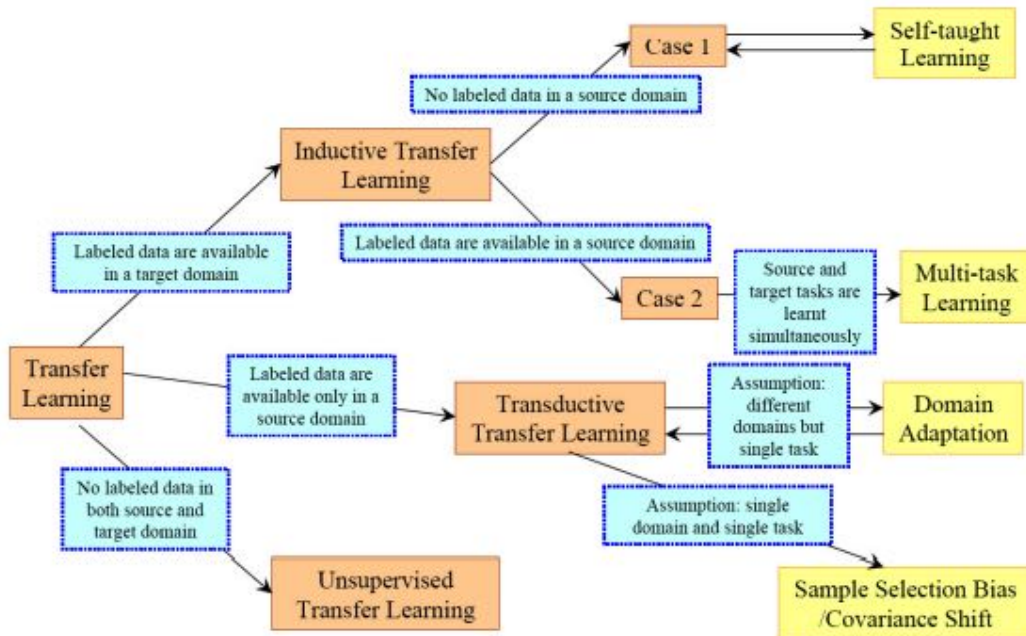
▷ 주의할 점

- Transfer Learning의 절차 동안, 다음 3가지 중요한 질문에 대하여 반드시 답을 구할 수 있어야 한다.
- **What to transfer:** 전체 프로세스 중 가장 첫 번째 질문이자 가장 중요한 단계이다. 우리는 Target task의 성능을 높이기 위해서 source 모델로부터 어떤 지식을 가져와야 하는지 찾고 있다. 이 질문에 답변하기 위해서는, 어떤 지식이 source에 특화된 지식이고 어떤 지식이 source와 target에 공통되는 지식인지 명확히 식별할 수 있어야 한다.
- **When to transfer:** 우리의 의도와는 달리 transfer Learning을 적용한 후 성능이 더 나빠질 수도 있고 엉뚱한 곳에서 영향을 받았을 수도 있다. (이를 Negative transfer라고 한다) 우리의 목적은 target task의 성능을 높이는 것이지 망치는 것이 아니다. 고로 transfer를 적용해야 할 때와 적용해서는 안 될 때를 신중히 판단할 수 있어야 한다.
- **How to transfer:** 상기 두 개의 질문에 답변이 되었을 때, 우리는 실제 지식 전이를 하는 방법을 고민하는 단계로 나아갈 수 있다. 이는 기존 알고리즘과 다른 테크닉에 대한 변화를 포함하고 있으며, 뒷 장에서 이에 대하여 다시 다룰 것이다. 또한 전이하는 방법에 대한 이해를 돕기 위해서 다음 장에서는 구체적인 용례를 설명할 것이다.

▷ Strategies(전략)

- Transfer Learning의 전략과 테크닉에 대하여 알아보자. Domain, task, 데이터의 가용성에 따라서 갈래가 나뉜다.
- 전이학습이 적용되는 용례를 전통적인 기계학습 알고리즘을 포함해서 다음 3가지로 분류할 수 있다.
- **Inductive Transfer Learning:** Source domain과 target domain이 같으나, source task와 target task가 서로 다른 경우이다. 알고리즘은 source domain의 귀납적 편향을 이용하여 target task의 성능을 향상시킨다. Source domain이 label이 있는 data를 쓰는지, 또는 label이 없는 데이터를 쓰는지에 따라 Multitask learning(멀티태스크 학습)과 Self-taught learning(자기교사 학습)으로 다시 세분화할 수 있다.
- **Unsupervised Transfer Learning:** 이 세팅은 target domain에서 비지도학습에 초점을 둔 inductive transfer learning과 유사하다. Source domain과 target domain은 유사하지만, task가 서로 다른 경우이다. 이 경우, label이 붙은 data는 어느쪽에서도 사용할 수 없다.
- **Transductive Transfer Learning:** Source task와 target task는 비슷하지만, domain이 서로 다른 경우이다. 이 세팅은 feature space와 주변확률에 따라서 세부 항목으로 나뉘어질 수 있다.

- 다음 2개의 그림은 상기 내용을 이해하기 쉽게 도표로 표시한 것이다.



Learning Strategy	Related Areas	Source & Target Domains	Source Domain Labels	Target Domain Labels	Source & Target Tasks	Tasks
Inductive Transfer Learning	Multi-task Learning	The Same	Available	Available	Different but Related	Regression Classification
	Self-taught Learning	The Same	Unavailable	Available	Different but Related	Regression Classification
						Clustering Dimensionality Reduction
Unsupervised Transfer Learning		Different but Related	Unavailable	Unavailable	Different but Related	
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias & Co-variate Shift	Different but Related	Available	Unavailable	The Same	Regression Classification

▷ Transferable Components(전이 요소)

- 이번 파트에서는 “What to transfer”에 대한 답이 될 수 있는 목록에 대해서 알아보자.
- **Instance transfer:** Source domain으로부터 얻은 지식을 target task에 재사용하는 것은 이상적인 시나리오이다. 대부분의 경우, source domain data는 즉시 재사용이 불가능하다. 그보다는, source domain으로부터 특정 인스턴스가 재사용될 수 있는 경우가 있다. Inductive learning에서는 AdaBoost 등의 방법으로 source domain의 training 인스턴스를 target task에 활용할 수 있다.
- **Feature-representation transfer:** 이 전략은 Source domain으로부터 target domain에 활용 가능한 feature representations(특징 표현)을 식별함으로써, domain 간의 차이를 최소화하고 오차율을 낮추는데 그 목적이 있다. Label이 붙은 데이터의 가용 여부에 따라서 지도학습과 비지도 학습에 적용된다.
- **Parameter transfer:** 이 접근법은 관련성이 있는 모델들이 어떤 파라미터 혹은 하이퍼 파라미터의 사전분포를 공유한다는 전제하에서 효과가 있다. Source data와 domain 데이터를 동시에 학습하는 Multitask learning과 달리, transfer learning하기 위해서 target domain의 손실에 추가적인 가중치를 더해서 학습함으로써 전반적인 성능을 개선한다.
- **Relational-knowledge transfer:** 상기 3개의 접근법과 달리, Relational-knowledge

transfer는 non-IID를 이용한다. 독립적이지 않고 동일하게 분포되지 않은 데이터라는 뜻으로 각각의 data point는 서로 연관이 있다는 말이다. 예를 들어 사회적 연결망(social network) data는 relational-knowledge-transfer 전략을 사용한다.

- 아래는 Strategies와 Components의 관계를 나타낸 표이다.

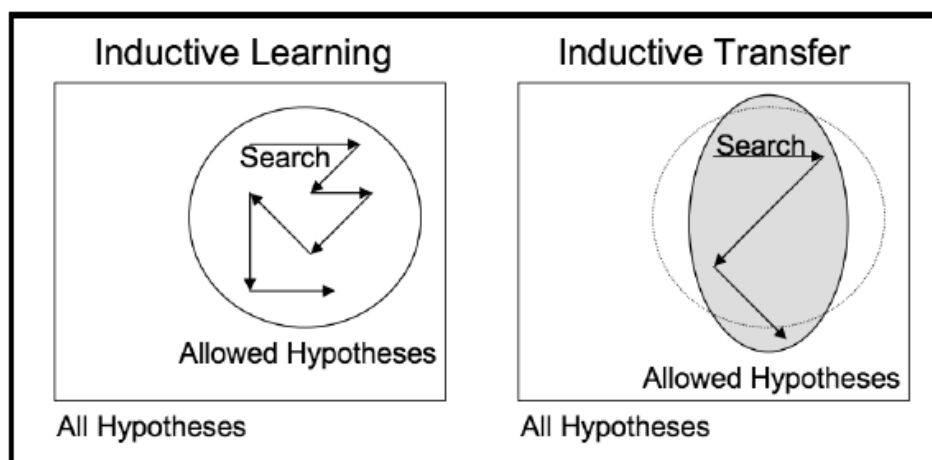
	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
Instance transfer	✓	✓	
Feature-representation transfer	✓	✓	✓
Parameter-transfer	✓		
Relational-knowledge transfer	✓		

3. Transfer Learning과 Deep Learning

딥러닝 모델은 inductive learning의 일종이다. inductive learning 알고리즘의 목적은 학습 데이터셋으로부터 패턴을 찾는 것이다. 분류 문제를 예로 들어보면, 모델은 입력되는 feature와 정답 label 사이의 패턴을 찾는다. 알고리즘은 학습 데이터의 분포에 관련된 가정을 바탕으로 학습 모델이 한번도 보지 않은 데이터에 대해서도 잘 맞추도록 학습모델을 일반화한다. 이러한 가정들을 inductive bias(귀납 편향)이라고 한다.

Inductive bias 또는 가정은 다양한 요소들에 의해서 특징지어지는데, 요소에는 가설 공간 내에서 검색 절차와 가설 공간을 제한하는 것 등이 있다. 따라서, 이러한 편향들은 모델이 주어진 task와 domain에서 무엇을 배우고 어떻게 배울 것인지에 큰 영향을 미친다.

Inductive transfer는 source task의 inductive bias를 target task에 활용하는 기법이다. 이는 다양한 방법으로 구현될 수 있는데, 모델의 가설공간을 좁게 제한함으로써 target task의 inductive bias를 조정하거나 source task로부터 얻은 지식을 활용해서 모델 내에서의 검색 절차를 조정하는 방식이 있다. 이 방식은 아래 그림처럼 시각화할 수 있다.



Inductive learning 알고리즘은 Inductive transfer 기법 뿐만 아니라 Bayesian transfer,

hierachical transfer 기법을 통해서도 target task 학습과 성능을 높일 수 있다.

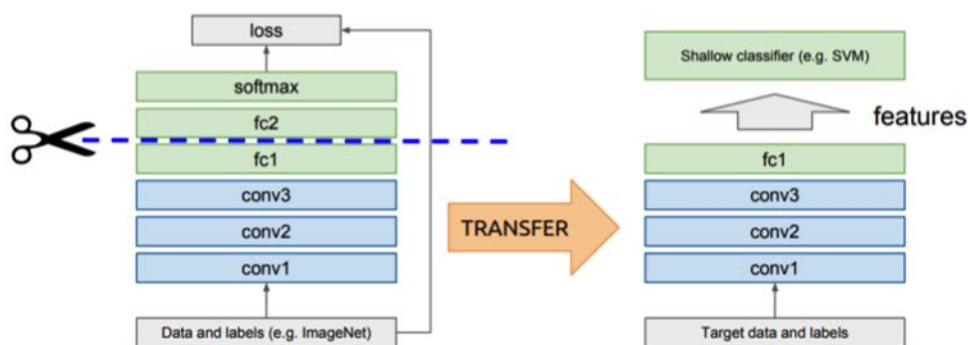
4. Transfer Learning 방법론

최근 몇 년간 딥러닝 분야가 상당한 발전과 놀라운 성과를 이룩하였다. 그러나 학습 시간과 학습에 필요한 양은 전통적인 기계학습에 비할 수 없이 어마어마하게 소요된다.

때로는 사람보다 뛰어난 성능의 딥러닝 모델이 나오기도 하는데, Computer vision, NLP 등 다양한 domain을 막론하고 최신 딥러닝 모델이 쏟아져나오고 있다. 대부분의 경우, 연구팀과 관련 종사자들은 다른 사람들이 모델을 사용할 수 있도록 세부 명세를 공개한다. 이를 pre-trained model이라고 하며 딥러닝 분야에서 transfer learning을 이해하고 활용하는데 큰 축을 차지하고 있다.

▷ Feature-extraction

- 딥러닝은 레이어가 쌓여있는 구조로 되어 있으며 각각의 레이어는 서로 다른 feature를 학습한다. 이러한 레이어들은 마지막 레이어까지 서로 연결되어서 최종 결과를 산출한다. 딥러닝이 이렇게 레이어가 겹겹이 쌓인 구조로 되어있는 덕분에, 최종 레이어를 제외한 pre-trained model(VGG 또는 Inception V3 모델 등)을 다른 task에서 feature를 추출하기 위해 사용할 수 있다. 다음 그림을 보면 transfer learning이 feature extraction으로 사용되는 것을 직관적으로 이해할 수 있다.

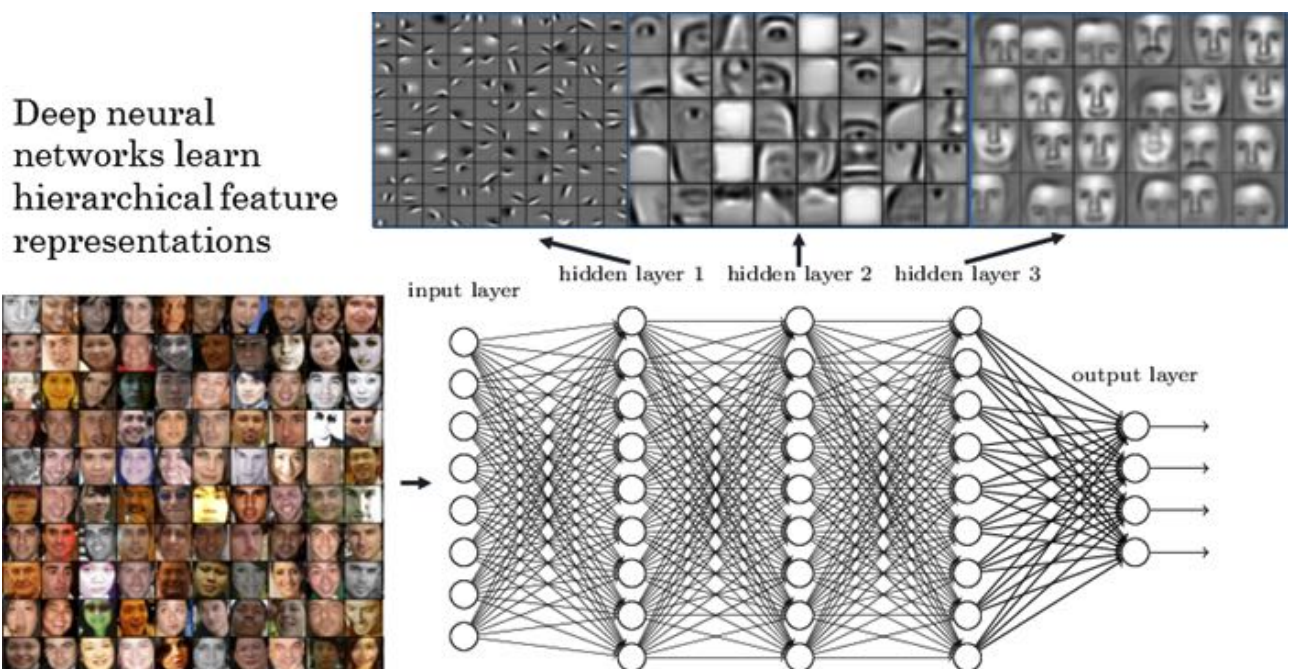


- Pre-trained model을 feature extraction으로 사용할 때는 weight에 대한 세팅에 주의해야 한다. 새로운 data로 새로운 task를 학습할 때에는, pre-trained model의 layer의 weight가 갱신되지 않도록 고정(freeze)시켜줘야 한다. Weight가 갱신되면 추출하는 feature의 의미가 달라질 수 있기 때문이다.
- 예를 들어, AlexNet의 pre-trained model을 feature extraction으로 사용한다고 가정해보자. AlexNet에서 분류를 담당하는 최종 레이어를 빼면, hidden layer들이 새로운 domain과 task로부터 받은 이미지를 4,096 차원의 벡터로 만들어주는 feature extractor가 된다.

Source domain task로부터 얻은 지식(pre-trained)으로부터 target domain task에 대한 feature를 추출할 수 있게 되는 것이다. 이후 새로운 task에 대응하는 분류 레이어만 학습하면 효율적으로 모델을 만들 수 있다. 이 기법은 딥러닝을 활용한 Transfer Learning으로 보편적으로 사용되고 있다.

▷ Fine-tuning

- Fine tuning은 분류 또는 회귀를 담당하는 최종 레이어만 바꾸는 것이 아니라, pre-trained model에서 가져온 레이어의 일부도 선택적으로 재학습한다. 딥러닝은 구조상 다양한 하이퍼파라미터 세팅이 가능하다. 다음 그림에서 보는 것처럼, 앞쪽의 layer는 일반적인 feature를 추출하고, 뒤쪽 layer일수록 주어진 task에 특화된 feature를 추출한다. 이 점에서 영감을 얻어서 목적에 따라 특정 layer는 weight를 고정시키고 나머지는 재학습되는 동안 fine tuning(미세 조정)되도록 하는 기법이 고안되었다.



- 이 기법은 모델의 전반적인 구조에 대한 관점에서 선행 지식을 활용하는 방법이며, 재학습 시점에 pre-trained model을 학습의 출발점으로 삼는 방법이다. 결론적으로, 학습 시간은 단축하면서 성능은 더 향상했다.

▷ Pre-trained models

- Transfer Learning의 기본 전제조건은 source task를 잘 학습한 모델이 존재해야 한다는 것이다. 다행히도, 딥러닝 분야에서는 공유가 활발히 이루어지고 있다. Computer vision과 NLP 등의 많은 도메인 분야에서, 많은 최신 딥러닝 모델 구조가 각 연구팀에 의해서 공개적으로 공유되고 있다. 특히 잘 알려진 모델들은 pre-trained model도 함께 공개하고 있다. Pre-trained model은 안정적으로 학습된 대개 수백만개의 parameter(weight)의 형태

로 공유된다.

- 유명 딥러닝 라이브러리 TensorFlow, Keras, Pytorch에서는 Xception, VGG16, Inception V3 등 유명 모델의 pre-trained model을 쉽게 다운로드할 수 있게 인터페이스를 제공하고 있다. 또한 많은 오픈소스 웹사이트에서도 pre-trained model을 구할 수 있다.
- Computer vision 분야에서는 VGG16, VGG19, Inception V3, Xception, ResNet50의 pre-trained model이 활발히 다뤄지고 있으며, NLP 분야에서는 Word2Vec, GloVe, FastText, Universal Sentence Encoder by Google, BERT의 pre-trained model이 많이 활용되고 있다.

3.2.4.2. Applications(적용 방안)

딥러닝은 실상 Transfer Learning의 수혜를 크게 입은 알고리즘 중 하나이다. 딥러닝에서 Transfer Learning이 성공적이었던 예시들을 소개한다.

▷ Transfer Learning for NLP

- 텍스트 데이터는 기계학습과 딥러닝을 적용하는데 많은 장애물이 있다. 텍스트 데이터는 보통 다양한 기법을 활용해서 변형시키거나 벡터화한다. Word2Vec이나 FastText 같은 임베딩 기법은 다른 학습 데이터셋을 이용해서 만들어졌다. 이 임베딩 모델의 선행지식을 전이해서 감성 분석, 문서 분류 등의 다른 task도 수행할 수 있다.

▷ Transfer Learning for Audio/Speech

- Computer vision과 NLP 분야와 마찬가지로, 오디오 데이터에 기반한 task에도 딥러닝 전이학습이 우수한 성능을 내고 있다. 예를 들어, 영어로 학습한 Automatic Speech Recognition(ASR, 자동 음성 인식)의 pre-trained model은 독일어 등의 타 언어 모델의 성능을 향상시키는데 기여했다. 또한 Automated speaker identification(자동 화자 인식) 분야에서도 딥러닝 전이학습이 큰 효과가 있었다.

▷ Transfer Learning for Computer Vision

- Computer vision 분야에서 딥러닝 전이학습은 CNN 기반의 많은 task (object recognition, object identification 등)에서 큰 효과가 있었다. Yosinski와 그의 동료들이 집필한 *How transferable are features in deep neural networks*이라는 논문은 어떻게 low level 레이어가 모서리 검출 등의 전통적인 computer vision 기법에서의 feature extractor와 같은 역할을 할 수 있는지, 왜 high level 레이어는 task에 특화된 feature를 추출하는지를 밝혔다. 이러한 발견은 VGG, AlexNet, Inceptions와 같은 최신 모델을 style transfer, face detection 등의 target task에 성공적으로 전이하는데 도움이 되었다.

5. Deep Transfer Learning(전이학습 심화 분류)

Transfer Learning에 대한 문헌이 많아지고 있고, 앞에서 말했듯이 Transfer Learning에 관련된 용어가 많고 모두 제각기 사용되고 있다. 따라서, 종종 Transfer Learning, domain adaption, multitask learning 등 그 차이가 혼란스러울 때가 있다. 그러나 안심해도 좋다. 이들 모두 비슷하고, 비슷한 문제를 해결하려는 시도일 뿐이다. 그러나 본 글에서는 일관성을 유지하기 위해서, source domain task의 지식을 바탕으로 target task를 해결하는 보편적인 의미로서의 Transfer Learning을 사용하였다.

▷ Domain adaption

- Domain adaption은 source domain의 주변확률이 target domain의 주변확률과 다른 경우를 일컬으며 $P(X_S) \neq P(X_T)$ 인 상황이다. 학습 내용을 전이하기 위해서는 source domain과 target domain의 data 분포의 차이가 있으므로 학습 내용을 전이하기 위해서는 조금 수정될 필요가 있다.
- 예를 들어, 긍정과 부정으로 라벨링이 붙은 영화 리뷰 말뭉치(corpus)는 제품 리뷰 감성 말뭉치와 다를 것이다. 영화 리뷰 감성을 학습한 분류기로 제품 리뷰 감성을 분류하면 데이터 분포 차이 때문에 결과가 다를 것이다. domain adaption 기법은 $P(X_S) \neq P(X_T)$ 시나리오에서 Transfer Learning을 하는 상황에 활용된다.

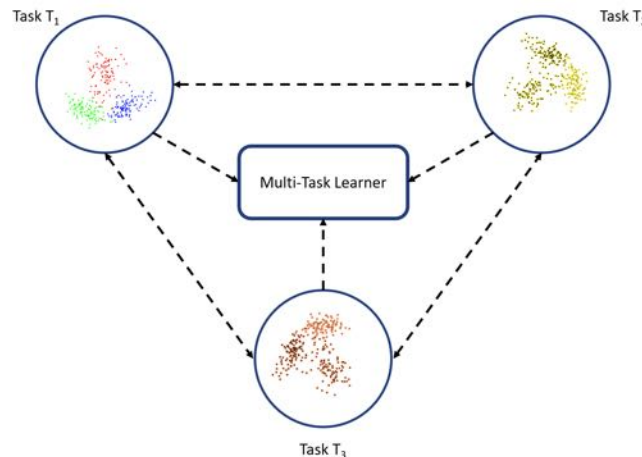
▷ Domain confusion

- 우리는 여러가지 Transfer Learning 전략에 대해서 알아보았고, 지식을 전이할 때 고민해야 할 세 가지 질문(what, when, how)에 대해 토의했다. 특히, 어떻게 feature representation transfer가 유용할 수 있는지에 대해서 알아보았는데, 딥러닝에서 각 레이어는 서로 다른 feature를 추출하기 때문이었다. 우리는 이 사실을 이용해서 domain이 변하지 않을 때 feature를 학습하고 domain 간의 지식 전이 가능성을 높일 수 있다. 모델이 어떤 representation을 학습하게 하는 대신에, 두 domain의 representation을 가능한 비슷하도록 유도한다.
- 이는 representation에 직접적으로 특정한 전처리 절차를 적용함으로써 달성할 수 있다. Baochen Sun, Jiashi Feng, Kate Saenko가 집필한 *Return of Frustratingly Easy Domain Adaption*에서 이러한 것들에 대해서 논의하였다. 또한 Ganin et. al. *Domain-Adversarial Training of Neural Networks*에서도 representation의 유사성 유도를 제시하였다. 이 기법에 숨겨진 간단한 아이디어는, source model에 다른 목적을 추가해서 모델이 domain을 혼동하게 함으로써 유사성을 이끌어내는 것이다. 그래서 Domain confusion이라는 이름이 붙었다.

▷ Multitask learning

- Multitask learning은 Transfer Learning들과는 약간 결이 다른 주제이다. Multitask learning의 경우는 source 와 target을 구별하지 않고 동시에 학습한다. 이때, 모델은 여러

개의 task에 대한 정보를 동시에 입력받는데, Transfer Learning이 처음에는 target task에 대해서 전혀 몰랐던 것과 차이가 있다. 다음 그림처럼 multitask learning을 그림으로 표현할 수 있다.



▷ One-shot learning

- 딥러닝은 태생적으로 데이터가 늘 고픈 시스템이다. 가중치를 학습하기 위해서 많은 학습 예제가 필요하기 때문이다. 이는 사람의 학습법과 다른, 딥러닝의 한가지 제한적인 측면이다. 예를 들어, 어린 아이는 사과같이 생긴 것을 한번 보는 것만으로 다양한 생김새의 사과를 알아볼 수 있다. 사람은 학습 예제 중 하나 또는 몇 가지 예제만으로 학습이 가능한데 기계학습과 딥러닝 알고리즘은 불가능하다. One-shot learning은 Transfer Learning의 변종으로서, 한 개 또는 몇 개의 학습 데이터에 기반해서 필요한 output을 추론하고자 한다. 세상 모든 class에 라벨을 붙이는 것이 불가능하고 새로운 클래스가 추가되기도 하기 때문에 이 학습법은 현실세계에 매우 필요하다. Task를 분류 문제로 예시를 든 것이며 다른 task도 마찬가지로 한계가 존재한다.
- One-shot learning이라는 단어를 처음 만들고 이 방면의 하위 연구분야를 개척한, 저명한 논문 Fei-Fei, et al. *One Shot Learning of Object Categories* 은 object categorization을 위해서 representation 학습에 대한 Bayesian 프레임워크의 변주를 제안하였다. 이 접근 방법은 딥러닝 분야에서 계속해서 발전하고 있다.

▷ Zero-shot learning

- Zero-shot learning은 Transfer Learning의 또다른 극단적인 변종이다. 이 기법은 task를 학습하는데 있어서 label이 없는 학습 예제에 의존한다. 예제를 이용하는 학습은 대부분 지도학습 알고리즘이므로 이상하게 들릴 것이다. 그러나 Zero-data learning 또는 zero-shot learning은 한 번도 본 적 없는 data를 이해하는 데 도움이 되는 추가적인 정보를 개발하기 위해서 학습 단계에서 스스로 똑똑하게 조정한다. Goodfellow의 저서 *Deep Learning*에서는 zero-shot learning을 세 가지 변수를 학습하는 시나리오라고 소개한다. 세 가지 변수는 전통적인 입력변수 x , 전통적인 출력변수 y 그리고 추가로 task T 를 설명하는 무작위 변수 T 이다. 모델은 조건부확률 분포 $P(y|x, T)$ 를 학습하도록 훈련된다. Zero-shot learning은 target 언어에 label이 없는 기계번역같은 시나리오에서 활용되어

질 수 있다.

6. Transfer Learning, 앞으로의 과제

Transfer Learning은 방대한 가능성을 지니고 있으며, 기존의 학습 알고리즘을 강화하는 데도 필요하다. 그러나 Transfer Learning에 대한 몇 가지 이슈가 있어서, 이에 대한 조사와 연구가 더 필요하다. What, when, how 질문에 대한 답변이 어려움은 일단 차치하고, 중요한 과제인 Negative transfer와 transfer bound에 대해 알아보자.

▷ Negative transfer

- 지금까지는 source task로부터 지식을 전이한 target task의 성능 향상에 대해서만 이야기했다. 그러나 오히려 Transfer learning이 성능을 떨어뜨리는 경우도 있다. Negative transfer는 source task로부터 target task로의 지식 전이가 성능 향상을 가져다주지 않고, 오히려 전반적인 target task의 성능을 떨어뜨리는 상황을 일컫는다. Negative Transfer에는 다양한 이유가 있을 수 있는데, source task가 target task와 별로 관련이 없거나 혹은 전이시키는 방법이 전이 과정을 적절히 조절하지 못했기 때문일 수 있다. Negative Transfer를 피하는 것은 매우 중요하고 이에 대한 신중한 연구가 필요하다. Rosenstien은 동료들과의 연구를 통해서 source와 target이 서로 다를 때 강압적으로 전이하는 것이 target task의 성능을 얼마나 줄먹는지에 대하여 경험적인 결과를 제시했다. Negative transfer를 막기 위해서 Bakker의 Bayesian 접근법과, 이 외에도 관계를 파악하기 위한 클러스터링 기반의 솔루션들이 연구되고 있다.

▷ Transfer bounds

- Transfer Learning의 양적화는 전이의 질과 전이 가능성에 영향을 미치는 중요한 요소이다. 전이되는 양을 계량하기 위해서 Hassan Mahmud와 그의 동료들은 Kolmogorov 복잡도를 이용하여 특정한 이론적 한계를 입증하고, task 사이의 관련성을 측정했다. Eaton과 그의 동료들은 전이되는 양을 측정하기 위해서 그래프 기반의 새로운 접근법을 제안했다.

○ 참고

[1]

<https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>

[2] Dipanjan Sarkar, Raghav Bali, Tamoghna Ghosh. Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras, Packt Publishing, 2018

[3] Lisa Torrey, Jude Shavlik. Transfer Learning, 2009

[4] Sinno Jialin Pan, Qiang Yang. IEEE, A Survey on Transfer Learning, 10.1109/TKDE.2009.191, 2009