왜 이 글을 쓰는가

딥러닝은 최근 괄목할만한 성과를 내고 있지만, 모델들의 우수한 성능이 정확히 어떤 원리에 근거한 것인지는 아직도 잘 알려진 바가 없다. 대부분의 경우 심지어는 모델을 설계하고 개발한 연구진들마저도 세세하게 이해하지 못한다. 그마저도 두루뭉술하게 직관에 의존한 설명을 내놓는 것이 전부다.

한 가지 확실한 것은, 딥러닝이 요 몇 년 새 눈부시게 발전해왔다는 것이다. 수많은 일자리들을 창출해냈고, 언어학, 생물학처럼 예상치 못한 분야들과의 시너지도 보여줬다. 수요가 높아진만큼 공급도 증가했다. 많은 인재와 자원들이 딥러닝으로 흘러들어갔고, 그에 따라 새로운 연구 결과들도 매번 경쟁적으로 쏟아지고 있다. 이렇게 짧은 간격의 빠른 발전이 가능한 것은, 어쩌면 여타 전통적 학문과는 달리 이론적 베이스가 잘 잡혀있지 않아서일 수도 있겠다. 모델 설계 방식과 성능 사이의 명확한 인과관계를 아직 학계에서 밝혀내지 못했기 때문에, 이런 '주먹구구'식 접근이 결국 유일한 방법인 것이고, 결과적으로 참신한 모델들이 우후죽순으로 생겨날 수 있는 것이다.

이런 방식이 꼭 '잘못되었다'고 할 수는 없다. 현재의 딥러닝 패러다임을 반박하기에는, 연구 결과들의 성능이 너무 좋다. 또한 아마도 딥러닝 종사자의 대다수에 해당할 공학적 마인드를 가진 이들은 이러한 bottom-up 접근방식을 내심 더 선호할 수도 있겠다. 장황한 이론적 토대를 세운 후 그를 기반으로 구체적 모델들을 구현하는 것이 아니라, 실제로 좋은 성과를 내는 모델이 있으면 그걸 가지고 향후에 나름의 작동 원리를 유추해보는 방식 말이다.

그렇지만 약간 개운치 못한 감이 있는 것은 사실이다. 특히나 딥러닝 모델들이 어떻게 이렇게 좋은 성과를 내는지 규명하는 것은 단순한 학문적 구색 맞추기 또는 지적 유희에 불과하지 않다는 점에서 그렇다. 모델이 어떻게, 또 왜 잘 작동하는 것인지 이해할 수 있으면, 반대로 모델이 어떤 상황에서 제대로 작동하지 못할지도 알 수 있다. 즉 모델의 성공이 애매한 직관과 랜덤한 운에 달려있는 게 아니라, 우리가 선제적으로 그 성공과 실패를 조절할 수 있게 되는 것이다.

시장 논리에 따라 많은 인재와 자원이 투입되는 딥러닝 커뮤니티는 고성능 모델을 개발해내는데 능하다. 지금과 같은 추세라면 앞으로도 계속 기존의 것들을 뛰어넘는 모델들이, 기술들이 발표될 것이다. 하지만 동시에 이런 상황을 조금 멀리서 조망하면서, 그래서 대체 우리가 지금 무엇을 하고 있는 건지 제대로 이해하려는 노력도 필요하다. 그런 이해가 오히려 반대로 딥러닝 연구에 새로운 방향성을 제시해줄 수 있기에 더더욱 그렇다.

그래서 하고 싶은 말은?

서론이 길었다. 이 글의 본 주제는 '머신러닝에서의 정보이론'이다. 원래 정보이론에서나 등장할법한 개념들이 머신러닝에 심심찮게 등장하는데, 그 유래와 함의를 고찰해보려는 글이다. 그렇다면 대체 왜 앞에서 그렇게 '이론적 토대'니 '정확한 이해'니 하는 것들에 대해 말을 늘어놓은 것인가?

저자는 딥러닝에 대한 정보이론적 접근이, 현재의 딥러닝 방법론이 왜 실제로 잘 작동하는 것인지를 설명해줄 수 있는 열쇠라고 생각한다.

딥러닝의 지위가 요 몇 년 동안 급부상하면서 배우려는 사람도, 배우는 사람도 크게 늘어났다. 이 글을 읽는 독자들도 그 중 일부일 것이라 생각한다. 최신 기술을 공부할 때, 특히 해당 기술이 상당한 수준의 수학적 배경을 전제로 하고 있다면, 학습자로서 쉽게 빠지기 쉬운 함정은 '모든 것을 의심 없이 있는 그대로 받아들이는' 것이다. 그 기술의 전문성에 압도당한 나머지, 연구자가 왜 하필이면 이러한 접근방식을 택했을지 고민하지도 않는다. 내용을 의심하고 곱씹는 과정이 없기 때문에, 그것을 본인의 사고로 체화할 수 있는 기회를 빼앗기는 것이다.

예를 들어 loss function으로 흔히 사용되는 아래의 cross entropy 식을 보자.

$$H(p,q) = \sum_{\forall x} p(x) \log \frac{1}{(q(x))}$$

실제 확률 분포 (true distribution) $p(x)$와 추정 확률 분포 (estimated distribution) $q(x)$ 가 있을 때 이 둘이 얼마나 유사한지를 측정할 수 있는 함수이다.

여기서 $q(x)$에 로그를 취하는 이유는 무엇인지, 두 확률 분포 간의 '유사도'라는 게 어떤 의미인지, 왜 위 식을 하필 '엔트로피'라 이름 붙인 건지를 이해한다면, 이를 바탕으로 본인 이해의 층위를 높일 수 있다. 우선 어떤 상황에, 어떤 모델에 cross entropy를 loss function으로 사용해야할지에 대한 감을 잡을 수 있을 것이다. 더 나아가서는 딥러닝 방법론에서 loss function 이 어떤 작용을 하는 것인지에 대한 직관을 얻을 수 있을 것이다.

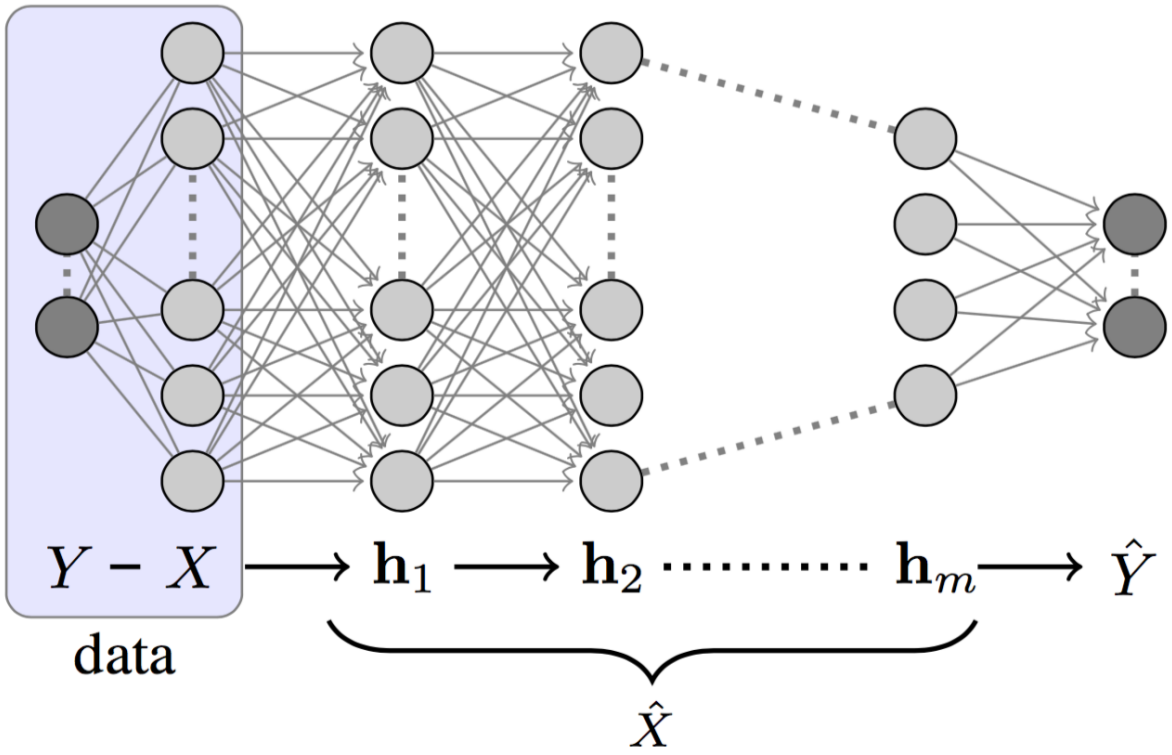
이런 소화 과정 없이, 단순히 'cross entropy 는 뛰어난 연구자들이 만들어낸, 딥러닝에서 광범위하게 쓰이는 마법같은 함수' 라고 액면 그대로 받아들이고 넘어가면 안 된다. 당장은 편리할 수 있어도, 향후 새로운 문제 상황에서 해결책을 찾을 수 있는 직관을 얻는데는 별 도움이 되지 않기 때문이다.

정보이론과 딥러닝

심층 신경망 (DNN, Deep Neural Network)은 초기에 인간의 뇌구조에서 영감을 받아 고안되었다. 뉴런 (Neuron)과 같은 용어 사용에서부터 뇌과학의 흔적을 찾을

수 있다. 하지만 초기 구상 단계에서는 뇌에서 아이디어를 따왔다 하더라도, 그 이후부터는 분명 독자적인 발전 과정을 거쳤다. 현재의 심층 신경망 구조는 생물학적 뇌와는 매우 다른 무언가가 되었기 때문이다. 당장 gradient back-propagation 만 하더라도 인간의 뇌에서 구현 가능한 작동 방식은 아니라는 느낌이 온다. 현재 변수 설정 값을 가지고 output을 미리 계산한 다음 오차를 바탕으로 변수를 조절하는, 이런 앞뒤로 반복적으로 왔다 갔다하는 메커니즘은 생물학적으로 썩 타당해보이진 않는다.

DNN이 인간의 뇌와 크게 유사하다고 보기도 어렵다면, 대체 이걸 무어라고 정의할 수 있을까?



위 다이어그램을 참조해가며 DNN의 구조를 매우 high-level 하게 되짚어보자. 우선 input data X, target label Y와 최종 예측 라벨 $\hat{Y}$, 또 hidden layers $h_1, h_2, \cdots, h_m$ 이 있다. X는 비교적 high-dimensional, Y는 low-dimensional 일 것이다. 즉 이 DNN은 X를 Y로 변환할 수 있는 방법을 학습하는 것이다. X와 Y 사이에 존재하는 알려지지 않은 복잡한 관계를, 여러 함수의 합성과 매개변수 등으로 quantitative 하게 표현하는 방법을 배우는 것이다.

이는, 즉 X를 '압축'하는 것이라 볼 수 있다. X에서 최대한 '불필요한' 정보를 없애는 것 말이다. 딥러닝을 정보 이론적 관점에서 오래 연구해왔고, 이 글을 쓰는데 많은 영감이 된 Naftali Tishby는 실제로 "The most important part of learning is

actually forgetting.” 이라는 말을 남긴 적 있다. 이런 관점에서 보자면 딥러닝에서 우리가 ‘학습’이라고 부르는 것은 ‘불필요한 정보를 잊어버리는 것’이라고 볼 수 있겠다.

주의깊은 독자라면 위 문단을 읽으며 불편했을 수 있다. ‘불필요한’ 정보란 무엇인지 정확한 정의를 내리지 않은 채 사용했기 때문이다. 지금까지의 맥락에서 ‘불필요한’ 정보란, X에서 Y와 관련 없는 정보를 뜻한다. 하지만 문제가 또 하나 생긴다. ‘관련 없는’을 제대로 정의하려면, 두 데이터 간의 정확한 관련성을 양적으로 측정할 수 있어야 한다. 어떻게?

여기가 바로 정보 이론이 등장하는 지점이다. 정보이론은 정보를 정량화하고, 저장하고, 통신하는 방법을 다루는 학문이다. 예컨대 정보 이론에서 주요하게 다뤄지는 rate-distortion theory에서는 정해진 한계 내에서 주어진 데이터를 가장 효율적으로 압축하는 방법 (특정 distortion 수치 D를 넘지 않으면서 하나의 기호당 드는 비트의 수 R을 최소화)를 연구한다. 또한 정보이론에서의 Mutual Information 개념을 이용하면 두 랜덤 변수 사이의 관련성을 정량화할 수도 있다. 딥러닝의 본질을 ‘데이터 압축’으로 해석했을 때, 결국 정보이론적 관점이 필요해지는 이유이다.

딥러닝과 정보이론의 연관성을 충분히 강조했으니, 이번 글에서는 딥러닝에서 찾아볼 수 있는 가장 기본적인 정보 이론 개념 몇 가지를 소개하고 마무리하도록 하자.

Information

‘정보 이론’에서 단언컨대 키워드는 ‘정보’가 될 것이다. 따라서 이 분야에서 얘기하는 ‘정보’가 무엇인지 의미를 짚고 넘어갈 필요가 있다. 한 가지 문제는, 명확한 정의를 내리기가 쉽지 않다는 것이다. 당장 ‘Philosophy of Information’ 이라는 명칭의, 정보의 의미를 탐구하기 위한 분과 학문이 따로 존재할 정도로, ‘정보’가 무엇인지 정확히 규정하는 것은 꽤 까다로운 일이다. 그렇지만 최소한 정보 이론 내에서 우리가 ‘정보’라고 지칭하는 것은 수식으로 애매함없이 표현할 수 있다. 게다가 상당히 직관적이기까지 하다.

‘정보’를 ‘놀라움’의 척도로 바라보면 된다. 어떤 사건이 ‘놀랍다’면, 그 사건을 통해서 우리는 많은 ‘정보’를 얻은 것으로 간주할 수 있다. 기존 사고의 흐름에 큰 도전이 되었기 때문이다. 또 어떤 사건이 얼마나 ‘놀라운’지는, 그 확률로 표현할 수 있다. 높은 확률을 갖고 있는 사건은 ‘덜 놀라운’ 것이고, 비교적 낮은 확률의 사건은 ‘더 놀라운’ 것으로 볼 수 있기 때문이다. 예를 들어 어제 일기예보에서 오늘 비가 올 확률이 100%라고 예측을 했다면, 오늘 비가 내린 것은 ‘놀라움’이 0이고, ‘정보’도 0이다. 반면 어제 1%의 확률로 오늘 눈이 온다고 했는데, 오늘 눈이 아닌 비가 내렸다면 이는 매우 놀라운 일이며, 이 일로 인해 우리가 얻은 정보량도 그만큼 많다고 볼 수 있다.

이런 맥락에서, 우리는 '정보'라는 것을 일종의 양함수 (positive function) I라고 바라볼 수 있다:

$$I : [0,1] \to [0,\infty)$$

여기서 I는 사건의 발생 확률 $p \in [0,1]$을 '놀라움', 또는 '정보(량)'으로 변환해주는 함수라고 생각하면 되겠다. 이 때 I가 자연스레 만족해야하는 성질 다섯 가지가 있다:

1. 연속적 (정의역 $[0,1]$에서)
2. 음이 아님 (non-negative)
3. 단조 감소 $\iff p_1 < p_2 \Rightarrow I(p_1) > I(p_2)$
4. $I(1) = 0$
5. Additive (i.e. 어느 두 독립사건의 확률이 각각 p_1, p_2 일 때, $I(p_1 p_2) = I(p_1) + I(p_2)$ 성립)

그리고 위 다섯 가지 조건을 모두 만족하는 함수는,

$$I(p) := \log \frac{1}{p}$$

뿐임이 알려져 있다.

따라서 이제부턴 어떤 사건의 '정보(량)'을 알고 싶으면, $I(p)$를 구하면 된다. 이 때 $I(p)$가 사건 자신의 정보를 알려준다는 의미에서 self-information 이라고 부르기도 한다.

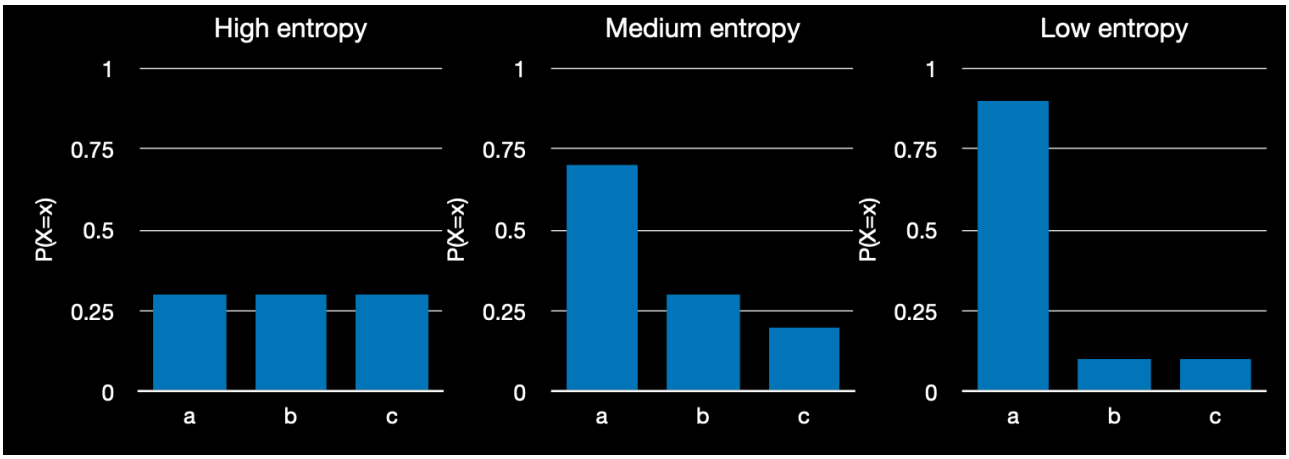
Entropy

이제 위에서 내린 '정보'의 의미를 바탕으로 엔트로피를 정의할 수 있다.
이산 확률 변수 X에 대해서, X의 엔트로피 또는 $H(X)$는 X의 self-information 들의 평균으로 정의한다. 즉,

$$H(X) := E[I(P(X))] = \sum_{x \in \chi} P(X = x) \log \frac{1}{P(X = x)}$$

가 된다. 이 때 χ는 X의 정의역이고, P는 X의 확률 질량함수이다.

그러니 쉽게 생각했을 때 엔트로피는 확률 변수의 평균적인 '놀라움'의 척도가 된다. 엔트로피가 높을 수록 해당 확률 변수는 더 '놀랍'고 예측 불가능하다. 확률이 어느 쪽에 쏠려있지 않고, 균일하게 분포해있다는 뜻이기도 하다.



덧붙여 엔트로피를 조금 다른 관점에서 바라볼 수도 있는데, '최대 압축 가능 정도'의 지표로 해석할 수도 있다. 두 명의 사람이, 어떤 확률 분포 X에서 추출한 표본을 서로 통신하는 상황이라고 해보자. 이 때 X의 엔트로피는 표본을 전달할 때 필요한 최소 (평균) 기호 수가 된다.

예컨대 철수는 대전에, 영희는 서울에 사는 두 명의 친구이다. 철수와 영희는 각각 주사위를 던져 나온 눈의 수를 서로에게 전보로 주고받는 상황이다. 이 때 주사위의 눈(1~6)은 이진수로 변환해 전보로 친다고 해보자.

표본 x를 보낼 때 사용할 수 있는 가장 최적화된 비트 수는 $\log \frac{1}{P(X=x)}$임이 알려져 있다. 약간의 직관을 위해서, 간단하게는 아래처럼 생각해볼 수 있다.

> 만약 모든 확률 공간의 모든 사건들이 확률 p로 발생한다면, 전체 사건의 개수는 $\frac{1}{p}$ 일 것이다. 그렇기 때문에 사용 가능한 기호의 수가 n개일 때 (만약 이진수를 사용한다면 $n = 2$), 사건의 내용을 인코딩하기 위해서 사용해야하는 최소 비트 개수는 $\log_n \frac{1}{p}$ 가 된다.

따라서 이들의 평균인 $H(X) = \sum_{x \in \chi} P(X=x) \log \frac{1}{P(X=x)}$가 최소 평균 비트 수가 되는 것이다.

즉 1부터 6까지의 수를 어떤 방식으로 인코딩하든, 결국 메시지의 평균 길이 (비트 수)의 하한은 엔트로피 $H(X)$ 이다. 데이터를 어느 정도까지 압축 (변환)할 수 있는지 그 한계를 뜻하는 셈이다.

Cross-Entropy

글의 앞부분에서 잠깐 cross-entropy를 간략히 소개했었는데, 수식은 다음과 같다:

$$H(p,q) = -\sum_{\forall x} p(x)\log\frac{1}{q(x))}$$

앞서 entropy를 봤다면 이 식이 매우 친숙하게 느껴질 것이다. 사실상 유일하게 달라진 부분은, 새로운 확률 분포 q가 등장했다는 것이다.

Cross-entropy는 딥러닝에서 loss function으로 자주 사용되는데, 그 이유는 두 확률분포 p, q가 얼마나 유사한지/차이나는지 보여주는 정량화된 지표이기 때문이다.

확률변수 X에 대해, p가 ground truth이고 q가 우리가 예측한 모델이라고 해보자. 그러면, 앞서 살펴봤듯 추출한 각 표본 x를 통신할 때 필요한 최소 비트 수는 $\log\frac{1}{p(x)}$이다. 하지만 위 cross-entropy 식에서는 대신 $\log\frac{1}{q(x)}$를 사용했다. 최적에서 멀어진 것이다. 그러므로 당연히 $H(p,q) \geq H(p,p) = H(X)$가 될 수밖에 없다.

> 예를 들어 알파벳 (A,B,C,D)의 실제 확률 분포 $p = (\frac{1}{2},\frac{1}{2},0,0)$ 이고, 추정 확률 분포 $q = (\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4})$ 라고 해보자. 그러면
> $H(p,p) = \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 1 + 0 + 0 = 1$이다. 반면
> $H(p,q) = \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 = 2$ 이다.

확률분포 p와 q의 차이에서 entropy과 cross-entropy의 차이가 기인하기 때문에, 거꾸로 cross-entropy를 줄이면 q를 p에 근접시킬 수 있다. 이런 이유로 cross-entropy 가 loss function으로 많이 사용되는 것이다.

참고로 딥러닝에서 자주 등장하는 K-L divergence라는 개념도 있는데, $D_{KL}(P\|Q) = H(p,q) - H(p,p)$ 이다. 즉 cross-entropy와 실제 entropy의 차이에 불과한 것이다. 따라서 cross-entropy minimization과 K-L divergence minimization은 결국 같은 것인데, 일반적으로는 cross-entropy를 loss function으로 더 많이 사용한다.

Mutual Information

마지막으로 살펴볼 개념은 mutual information이다. Mutual information은 한 확률변수가 다른 확률변수에 대해 갖고 있는 정보량을 알려준다. 두 확률변수 X, Y가 있을 때 둘 간의 mutual information $I(X,Y)$는 다음과 같이 정의된다:

$$I(X,Y) = \sum_{x\in\chi}\sum_{y\in\Upsilon} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} = D(p(x,y)\|p(x)p(y))$$

즉, x, y의 joint distribution과 product distribution 사이의 K-L divergence라 볼 수 있다. 식을 잘 유도하면

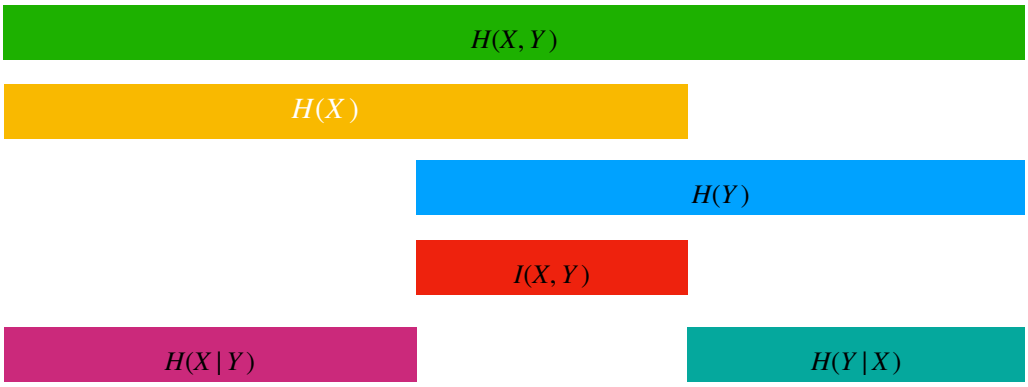
$$I(X,Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X\,|\,Y) = H(Y) - H(Y\,|\,X) = H(X,Y) - H(X\,|\,Y) - H(Y\,|\,X)$$

임도 확인할 수 있다.

이 때 $H(X\,|\,Y)$는 conditional entropy를 뜻하는 기호로, 확률변수 Y를 관측했다는 가정 하의 X의 엔트로피이다.

$$H(X\,|\,Y) := \sum_y p(y)H(X\,|\,Y=y) = \sum_y p(y)\sum_x p(y\,|\,x)\frac{1}{\log p(y\,|\,x)} = \sum_x\sum_y p(y,x)\log\frac{1}{p(y\,|\,x)}$$

아래 막대 다이어그램을 보면 좀 더 이해가 쉬울 것이다.



즉 mutual information $I(X,Y)$는, Y를 관측하고 나서의 X에서의 불확실성 감소 (= 엔트로피 감소 = 정보 이득)의 정도라고 해석할 수도 있다.

참고자료

https://analyticsindiamag.com/how-tishbys-information-bottleneck-can-break-open-the-black-box-of-deep-learning/

Opening the Black Box of Deep Neural Networks via Information (https://arxiv.org/pdf/2202.06749.pdf)

The information bottleneck method (https://arxiv.org/pdf/physics/0004057.pdf)

https://mbernste.github.io/posts/self_info/

https://stats.stackexchange.com/questions/87182/what-is-the-role-of-the-logarithm-in-shannons-entropy/87194#87194