

혼자 공부하며 함께 만드는

혼공 용어 노트

목차

가나다순

RMS 정규화 root mean square normalization	31	로지스틱 회귀 logistic regression	13
top-k 샘플링	32	릿지 회귀 ridge regression	12
top-p 샘플링	32	말뭉치 corpus	26
가중치(또는 계수) weight(또는 coefficient)	11	맷플롯립 matplotlib	07
강인공지능 vs 약인공지능	06	머신러닝과 딥러닝 machine learning과 deep learning	06
검증 세트 validation set	16	멀티 헤드 어텐션 multi-head-attention	29
결정 트리 Decision Tree	16	미니배치 경사 하강법 minibatch gradient descent	14
결정계수 coefficient of determination	10	배열 인덱싱 array indexing	09
과대적합 vs 과소적합 overfitting vs underfitting	10	배치 경사 하강법 batch gradient descent	15
교차 검증 cross validation	16	밸리드 패딩 valid padding	23
군집 clustering	19	변환기 transformer	12
그레이디언트부스팅 gradient boosting	18	부트스트랩 샘플 bootstrap sample	17
그리드 서치 Grid Search	17	불리언 인덱싱 boolean indexing	13
넘파이 numpy	09	브로드캐스팅 broadcasting	09
다중 분류 multi-class classification	13	비지도 학습 unsupervised learning	08
다중 회귀 multiple regression	11	샘플링 편향 sampling bias	09
다항 회귀 polynomial regression	11	선형 회귀 linear regression	11
단어 임베딩 word embedding	27	셀 cell	26
대규모 언어 모델 large language model, LLM	30	셀 상태 cell state	27
데이터 전처리 data preprocessing	09	셀프 어텐션 Self-attention	29
드롭아웃 dropout	22	소프트맥스 함수 softmax function	14
딥러닝 deep learning	20	손실 함수 loss function	15
라쏘 회귀 lasso regression	12	순차 데이터 sequential data	25
랜덤 서치 Random Search	17	순환 신경망 recurrent neural network, RNN	26
랜덤 포레스트 Random Forest	17	스트라이드 stride	24
렐루 함수 ReLU Function	21	시계열 데이터 time series data	25
로지스틱 손실 함수 logistic loss function	15	시그모이드 함수 sigmoid function	13

심층 신경망 deep neural network, DNN	21	크로스엔트로피 손실 함수 cross-entropy loss function	15
앙상블 학습 ensemble learning	17	클로즈드 소스 LLM	31
어텐션 메커니즘 Attention mechanism	29	텐서플로 TensorFlow	20
에포크 epoch	14	토큰 token	27
엑스트라 트리 extra trees	18	토큰화 tokenization	31
오픈소스 LLM	31	트랜스포머 Transformer	29
옵티마이저 optimizer	22	특성 feature	07
원-핫 인코딩 one-hot encoding	20	특성 맵 feature map	23
위치 인코딩 positional encoding	30	패딩과 세임 패딩 padding과 same padding	23
은닉 상태 hidden state	26	풀링 pooling	24
은닉층 hidden layer	21	피드포워드 네트워크 feedforward network	29
이너서 inertia	19	피드포워드 신경망 feedforward neural network, FFNN	25
이진 분류 binary classification	06	필터 filter	23
인공신경망 artificial neural network, ANN	20	하이퍼파라미터 hyperparameter	12
인공지능 artificial intelligence	06	합성곱 convolution	23
적응적 학습률 adaptive learning	22	허깅페이스 HuggingFace	30
전이 학습 transfer learning	30	확률적 경사 하강법 Stochastic Gradient Descent	14
정형 데이터 vs 비정형 데이터		활성화 함수 activation function	20
structured data vs unstructured data	17	회귀 regression	10
주성분 분석 principal component analysis, PCA	19	훈련 training	07
지도 학습 supervised learning	08	훈련 데이터 training data	08
차원 축소 dimensionality reduction	19	훈련 세트와 테스트 세트 train set와 test set	09
최대 풀링과 평균 풀링 max pooling과 average pooling	24	히스토그램 histogram	19
층 정규화 layer normalization	29	히스토그램 기반 그레이디언트 부스팅	
코랩과 노트북 Colab과 Notebook	06	Histogram-based Gradient Boosting	18
콜백 callback	22	힌지 손실 hinge loss	15
크로스 어텐션 cross attention	30		

ABC 순

activation function 활성화 함수	20	dropout 드롭아웃	22
adaptive learning 적응적 학습률	22	ensemble learning 앙상블 학습	17
ANN artificial neural network 인공신경망	20	epoch 에포크	14
array indexing 배열 인덱싱	09	EXAONE	31
artificial intelligence 인공지능	06	extra trees 엑스트라 트리	18
Attention mechanism 어텐션 메커니즘	29	feature map 특성 맵	23
BART	30	feature 특성	07
batch gradient descent 배치 경사 하강법	15	feedforward network 피드포워드 네트워크	29
binary classification 이진 분류	06	FFNN feedforward neural network 피드포워드 신경망	25
boolean indexing 불리언 인덱싱	13	filter 필터	23
bootstrap sample 부트스트랩 샘플	17	GPT Generative Pre-trained Transformer	32
broadcasting 브로드캐스팅	09	gradient boosting 그래디언트부스팅	18
callback 콜백	22	Grid Search 그리드 서치	17
cell state 셀 상태	27	GRU Gated Recurrent Unit	28
cell 셀	26	hidden layer 은닉층	21
clustering 군집	19	hidden state 은닉 상태	26
coefficient of determination 결정계수	10	hinge loss 힙지 손실	15
Colab과 Notebook 코랩과 노트북	06	Histogram-based Gradient Boosting	18
convolution 합성곱	23	histogram 히스토그램	19
corpus 말뭉치	26	HuggingFace 허깅페이스	30
cross-entropy loss function 크로스엔트로피 손실 함수	15	hyperparameter 하이퍼파라미터	12
cross attention 크로스 어텐션	30	inertia 이너셔	19
cross validation 교차 검증	16	k-최근접 이웃 분류 vs k-최근접 이웃 회귀	10
data preprocessing 데이터 전처리	09	k-최근접 이웃 알고리즘	07
Decision Tree 결정 트리	16	k-means algorithm k-평균 알고리즘	19
deep learning 딥러닝	20	large language model, LLM 대규모 언어 모델	30
dimensionality reduction 차원 축소	19	lasso regression 라쏘 회귀	12
DNN, deep neural network 심층 신경망	21	layer normalization 층 정규화	29

linear regression 선형 회귀	11	sampling bias 샘플링 편향	09
logistic loss function 로지스틱 손실 함수	15	Self-attention 셀프 어텐션	29
logistic regression 로지스틱 회귀	13	sequential data 순차 데이터	25
loss function 손실 함수	15	sigmoid function 시그모이드 함수	13
LSTM Long Short-Term Memory	27	softmax function 소프트맥스 함수	14
machine learning과 deep learning 머신러닝과 딥러닝	06	Stochastic Gradient Descent 확률적 경사 하강법	14
matplotlib 맷플롯립	07	stride 스트라이드	24
max pooling과 average pooling 최대 풀링과 평균 풀링	24	structured data vs unstructured data	17
minibatch gradient descent 미니배치 경사 하강법	14	supervised learning 지도 학습	08
multi-class classification 다중 분류	13	TensorFlow 텐서플로	20
multi-head-attention 멀티 헤드 어텐션	29	time series data 시계열 데이터	25
multiple regression 다중 회귀	11	token 토큰	27
numpy 넘파이	09	tokenization 토큰화	31
one-hot encoding 원-핫 인코딩	20	train set와 test set 훈련 세트와 테스트 세트	09
optimizer 옵티마이저	22	training data 훈련 데이터	08
overfitting vs underfitting 과대적합 vs 과소적합	10	training 훈련	07
padding과 same padding 패딩과 세임 패딩	23	transfer learning 전이 학습	30
PCA, principal component analysis 주성분 분석	19	transformer 변환기	12
polynomial regression 다항 회귀	11	Transformer 트랜스포머	29
pooling 풀링	24	transformers	31
positional encoding 위치 인코딩	30	unsupervised learning 비지도 학습	08
Random Forest 랜덤 포레스트	17	valid padding 밸리드 패딩	23
Random Search 랜덤 서치	17	validation set 검증 세트	16
regression 회귀	10	weight 가중치 (또는 계수)	11
ReLU Function 렐루 함수	21	word embedding 단어 임베딩	27
ridge regression 릿지 회귀	12		
RNN recurrent neural network 순환 신경망	26		
root mean square normalization RMS 정규화	31		

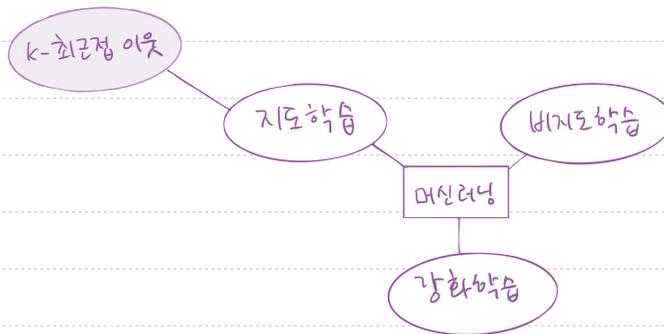
01 장

□ 인공지능	artificial intelligence [01장 029쪽]
	학습하고 추론할 수 있는 지능을 가진 컴퓨터 시스템을 만드는 기술
□ 강인공지능 vs 약인공지능	[01장 028쪽, 029쪽]
	강인공지능은 인공일반지능이라고도 하고 사람의 지능과 유사(영화 속 전지전능한 AI)함. 약인공지능은 특정 분야에서 사람을 돕는 보조 AI(음성 비서나 자율 주행도 여기 포함)
□ 머신러닝과 딥러닝	machine learning과 deep learning [01장 029쪽, 030쪽]
	머신러닝은 데이터에서 규칙을 학습하는 알고리즘을 연구하는 분야(대표 라이브러리는 사이킷런). 딥러닝은 인공신경망을 기반으로 한 머신러닝 분야를 일컫음(대표 라이브러리는 텐서플로와 파이토치)
□ 코랩과 노트북	Colab과 Notebook [01장 034쪽, 038쪽]
	코랩은 웹 브라우저에서 텍스트와 프로그램 코드를 자유롭게 작성 할 수 있는 온라인 에디터로 이를 코랩 노트북 또는 노트북이라 부름. 최소 실행 단위는 셀이며 코드 셀과 텍스트 셀이 있음
□ 이진 분류	binary classification [01장 048쪽]
	머신러닝에서 여러 개의 종류(혹은 클래스) 중 하나를 구별해 내는 문제를 분류(classification)라고 부르며 2개의 종류(클래스) 중 하나를 고르는 문제를 이진 분류라 함

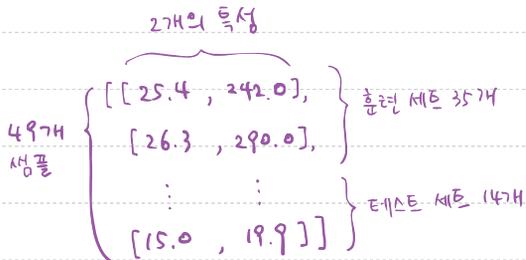
02장

□ 지도 학습 supervised learning [02장 071쪽]
 지도 학습은 입력(데이터)과 타겟(정답)으로 이뤄진 훈련 데이터가 필요하며 새로운 데이터를 예측하는 데 활용함. 1장에서 사용한 k-최근접 이웃이 지도 학습 알고리즘임

□ 비지도 학습 unsupervised learning [02장 071쪽]
 타겟 데이터 없이 입력 데이터만 있을 때 사용. 이런 종류의 알고리즘은 정답을 사용하지 않으므로 무언가를 맞힐 수가 없는 대신 데이터를 잘 파악하거나 변형하는 데 도움을 줌



□ 훈련 데이터 training data [02장 071쪽]
 지도 학습의 경우 필요한 입력(데이터)과 타겟(정답)을 합쳐 놓은 것



□ 훈련 세트와 테스트 세트	train set와 test set [02장 072쪽]
	모델을 훈련할 때는 훈련 세트를 사용하고 평가는 테스트 세트로 함. 테스트 세트는 전체 데이터에서 20~30%
□ 샘플링 편향	sampling bias [02장 076쪽]
	훈련 세트와 테스트 세트에 샘플이 고르게 섞여 있지 않을 때 샘플링 편향이 나타나며 제대로 된 지도 학습 모델을 만들 수 없음
	<p>The diagram shows two purple ovals representing data sets. The left oval is labeled '훈련 세트' (Training Set) and contains approximately 15 fish icons. The right oval is labeled '테스트 세트' (Test Set) and contains approximately 8 fish icons. Above the training set, there is a handwritten note in purple ink that says '온바를 훈련 데이터' (Only use training data), indicating that the test set is not being used for training, which is a common mistake leading to sampling bias.</p>
□ 넘파이	numpy [02장 077쪽]
	파이썬의 대표적인 배열array 라이브러리로 고차원의 배열을 손쉽게 만들고 조작할 수 있는 간편한 도구를 많이 제공함.
□ 배열 인덱싱	array indexing [02장 081쪽]
	넘파이 기능으로 여러 개의 인덱스로 한 번에 여러 개의 원소를 선택할 수 있음
□ 데이터 전처리	data preprocessing [02장 103쪽]
	머신러닝 모델에 훈련 데이터를 주입하기 전 가공하는 단계로 특성값을 일정한 기준으로 맞추어 주는 작업. 데이터를 표현하는 기준이 다르면 알고리즘을 올바르게 예측할 수 없음
□ 브로드캐스팅	broadcasting [02장 104쪽]
	조건을 만족하면 모양이 다른 배열 간의 연산을 가능하게 해 주는 기능

03장

□ 회귀 regression [03장 121쪽]

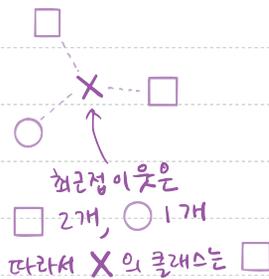
클래스 중 하나로 분류하는 것이 아니라 임의의 어떤 숫자를 예측하는 문제

□ k-최근접 이웃 **참고 용어** k-최근접 이웃 알고리즘 [03장 121쪽, 122쪽]

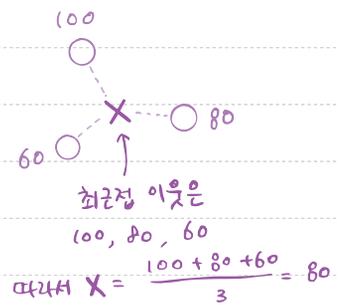
분류 vs k-최근 k-최근접 이웃 알고리즘을 사용해 각각 분류 문제와 회귀 문제를 해결하는 방법

접 이웃 회귀

k-최근접 이웃 분류



k-최근접 이웃 회귀



□ 결정계수 (R^2) coefficient of determination [03장 126쪽]

회귀 모델에서 예측의 적합도를 0과 1 사이의 값으로 계산한 것으로 1에 가까울수록 완벽함

$$R^2 = 1 - \frac{(\text{타겟} - \text{예측})^2}{(\text{타겟} - \text{평균})^2}$$

□ 과대적합 vs **overfitting vs underfitting** [03장 128쪽]

과소적합

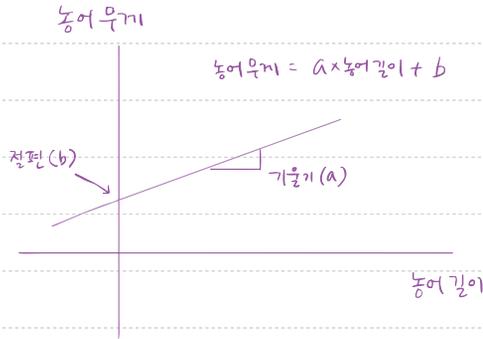
과대적합은 모델의 훈련 세트 점수가 테스트 세트 점수보다 훨씬 높을 경우를 의미함. 과소적합은 이와 반대로 모델의 훈련 세트와 테스트 세트 점수가 모두 동일하게 낮거나 테스트 세트 성능이 오히려 더 높을 경우를 의미함

□ 선형 회귀

linear regression

[03장 141쪽]

널리 사용되는 대표적인 회귀 알고리즘으로 특성이 하나인 경우 어떤 직선을 학습하는 알고리즘(예를 들면, 농어 무게 학습 그래프)



□ 가중치

weight (또는 coefficient)

[03장 143쪽]

(또는 계수)

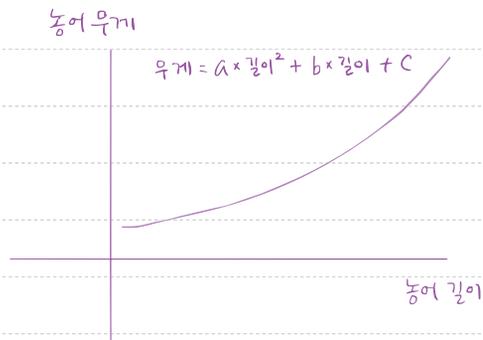
선형 회귀가 학습한 직선의 기울기를 종종 가중치 또는 계수라 함
 (위 그림에서 기울기(a))

□ 다항 회귀

polynomial regression

[03장 145쪽]

다항식을 사용하여 특성과 타깃 사이의 관계를 나타낸 선형 회귀

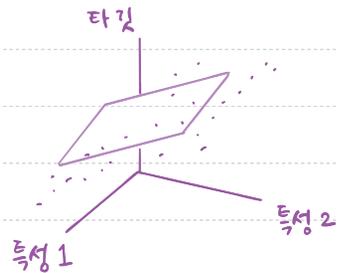


□ 다중 회귀

multiple regression

[03장 157쪽]

여러 개의 특성을 사용한 선형 회귀



□ 변환기 **transformer** [03장 160쪽]
특성을 만들거나 전처리하는 사이킷런의 클래스로 타깃 데이터 없이 입력 데이터를 변환함

□ 릿지 회귀 **ridge regression** [03장 166쪽]
규제가 있는 선형 회귀 모델 중 하나로 모델 객체를 만들 때 alpha 매개변수로 규제의 강도를 조절함. alpha 값이 크면 규제 강도가 세지므로 계수 값을 더 줄이고 조금 더 과소적합되도록 유도하여 과대적합을 완화시킴

□ 하이퍼파라미터 **hyperparameter** [03장 167쪽]
머신러닝 모델이 학습할 수 없고 사람이 지정하는 파라미터

□ 라쏘 회귀 **lasso regression** [03장 170쪽]
또 다른 규제가 있는 선형 회귀 모델로 alpha 매개변수로 규제의 강도를 조절함. 릿지와 달리 계수 값을 아예 0으로 만들 수도 있음



04장 [✓]

□ 다중 분류 **multi-class classification** [04장 191쪽]

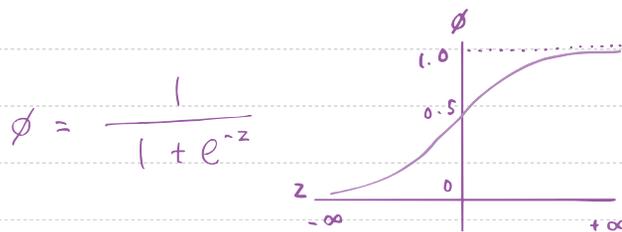
타깃 데이터에 2개 이상의 클래스가 포함된 문제

□ 로지스틱 회귀 **logistic regression** [04장 193쪽]

선형 방정식을 사용한 분류 알고리즘으로 선형 회귀와 달리 시그모이드 함수나 소프트맥스 함수를 사용하여 클래스 확률을 출력

□ 시그모이드 함수 **sigmoid function** *logistic regression* [04장 193쪽]

시그모이드 함수 또는 로지스틱 함수라고 부르며 선형 방정식의 출력을 0과 1 사이의 값으로 압축하며 이진 분류를 위해 사용. 이진 분류일 경우 시그모이드 함수의 출력이 0.5보다 크면 양성 클래스, 0.5보다 작으면 음성 클래스로 판단



□ 불리언 인덱싱 **boolean indexing** [04장 195쪽]

넘파이 배열은 True, False 값을 전달하여 행을 선택할 수 있으며 이를 불리언 인덱싱이라고 함

□ 소프트맥스 함수

softmax function

[04장 200쪽]

여러 개의 선형 방정식의 출력값을 0~1 사이로 압축하고 전체 합이 1이 되도록 만들며 이를 위해 지수 함수를 사용하기 때문에 정규화된 지수 함수라고도 함

$$S_1 = \frac{e^{z_1}}{e_sum}, S_2 = \frac{e^{z_2}}{e_sum}, \dots, S_7 = \frac{e^{z_7}}{e_sum}$$

□ 확률적 경사

Stochastic Gradient Descent

[04장 211쪽]

하강법

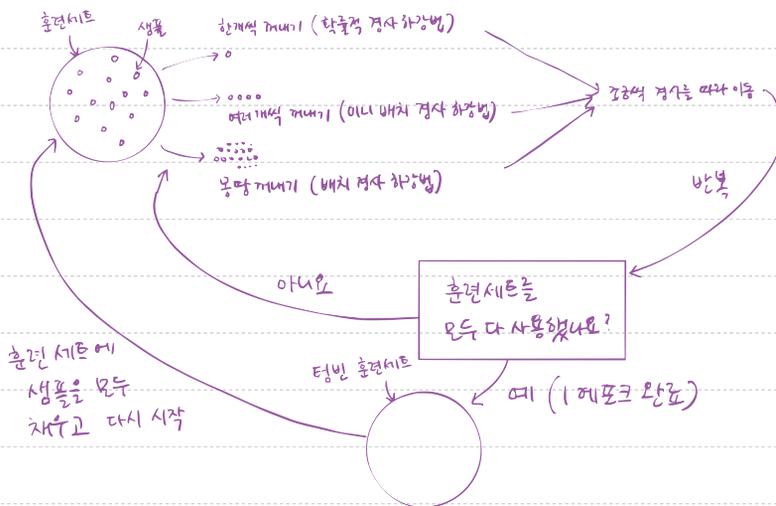
훈련 세트에서 랜덤하게 하나의 샘플을 선택하여 손실 함수의 경사를 따라 최적의 모델을 찾는 알고리즘

□ 에포크

epoch

[04장 213쪽]

확률적 경사 하강법에서 훈련 세트를 한 번 모두 사용하는 과정



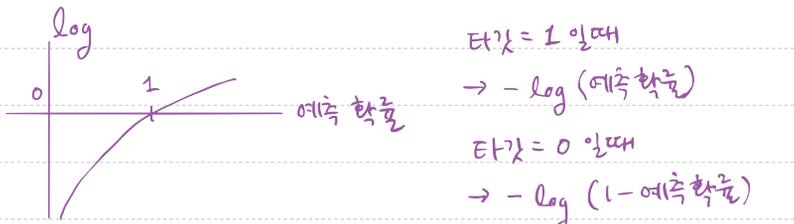
□ 미니배치 경사

minibatch gradient descent

[04장 213쪽]

하강법

1개가 아닌 여러 개의 샘플을 사용해 경사 하강법을 수행하는 방법으로 실전에서 많이 사용

□ 배치 경사 하강법	batch gradient descent	[04장 213쪽]
	한 번에 전체 샘플을 사용하는 방법으로 전체 데이터를 사용하므로 가장 안정적인 방법이지만 그만큼 컴퓨터 자원을 많이 사용함. 또한 어떤 경우는 데이터가 너무 많아 한 번에 전체 데이터를 모두 처리하지 못할 수 있음.	
□ 손실 함수	loss function	[04장 213쪽]
	어떤 문제에서 머신러닝 알고리즘이 얼마나 엉터리인지를 측정하는 기준.	
□ 로지스틱 손실 함수	logistic loss function <i>이진 크로스엔트로피 손실 함수라고도 함</i>	[04장 215쪽]
	양성 클래스(타겟 = 1)일 때 손실은 $-\log(\text{예측 확률})$ 로 계산하며, 1 확률이 1에서 멀어질수록 손실은 아주 큰 양수가 됨. 음성 클래스(타겟 = 0)일 때 손실은 $-\log(1 - \text{예측 확률})$ 로 계산함. 이 예측 확률이 0에서 멀어질수록 손실은 아주 큰 양수가 됨	
		
□ 크로스엔트로피 손실 함수	cross-entropy loss function	[04장 217쪽]
	다중 분류에서 사용하는 손실 함수	
□ 힙지 손실	hinge loss	[04장 223쪽]
	서포트 벡터 머신(support vector machine)이라 불리는 또 다른 머신러닝 알고리즘을 위한 손실 함수. SGDClassifier가 여러 종류의 손실 함수를 loss 매개변수에 지정하여 다양한 머신러닝 알고리즘을 지원함	



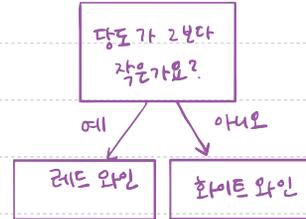
05장

□ 결정 트리

Decision Tree

[05장 234쪽]

스무고개와 같이 질문을 하나씩 던져 정답을 맞춰가며 학습하는 알고리즘으로 비교적 예측 과정을 이해하기 쉬움



□ 검증 세트

validation set **참고 용어** 하이퍼파라미터

[05장 257쪽]

하이퍼파라미터 튜닝을 위해 모델을 평가할 때, 테스트 세트를 사용하지 않기 위해 훈련 세트에서 다시 떼어 낸 데이터 세트



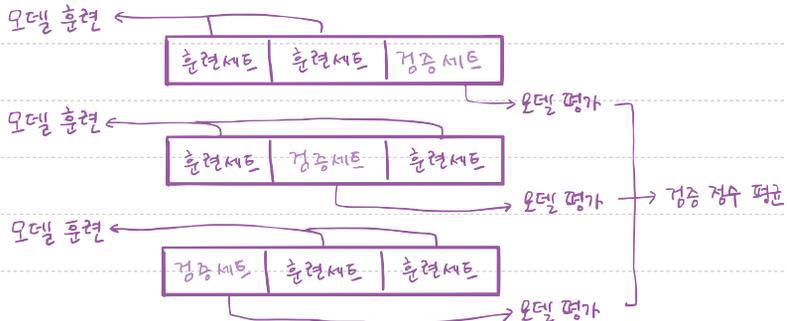
□ 교차 검증

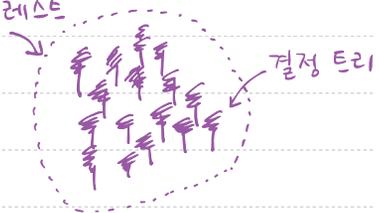
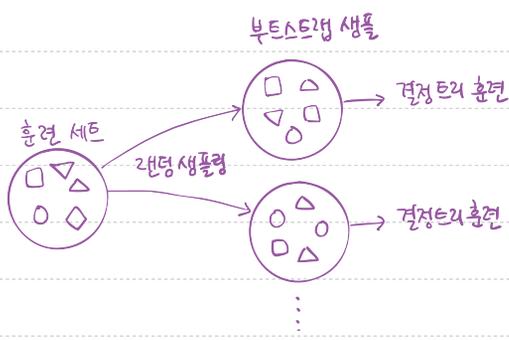
cross validation

[05장 259쪽]

훈련 세트를 여러 폴드로 나눈 다음 한 폴드가 검증 세트의 역할을 하고 나머지 폴드에서는 모델을 훈련함.

이렇게 모든 폴드에 대해 검증 점수를 얻어 평균하는 방법으로 교차 검증을 이용하면 검증 점수가 안정적이며, 훈련에 더 많은 데이터를 사용할 수 있음



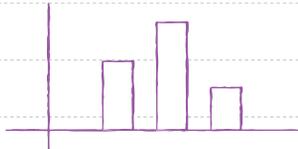
□ 그리드 서치	Grid Search	[05장 262쪽]
하이퍼파라미터 탐색을 자동화해 주는 도구		
□ 랜덤 서치	Random Search	[05장 266쪽]
랜덤 서치는 연속적인 매개변수 값을 탐색할 때 유용		
□ 정형 데이터 vs 비정형 데이터	structured data vs unstructured data	[05장 278쪽]
특정 구조로 이루어진 데이터를 정형 데이터라 하고, 반면 정형화되기 어려운 텍스트나 이미지 등을 비정형 데이터라 함 <i>CSV나 데이터베이스 등</i>		
□ 앙상블 학습	ensemble learning	[05장 278쪽]
여러 알고리즘(예, 결정 트리)을 합쳐서 성능을 높이는 머신러닝 기법		
□ 랜덤 포레스트	Random Forest	[05장 279쪽]
대표적인 결정 트리 기반의 앙상블 학습 방법. 안정적인 성능 덕분에 널리 사용됨.		
부트스트랩 샘플을 사용하고 <i>랜덤 포레스트</i>		
랜덤하게 일부 특성을 선택하		
여 트리를 만드는 것이 특징 <i>결정 트리</i>		
		
□ 부트스트랩 샘플	bootstrap sample	[05장 279쪽]
데이터 세트에서 중복을 허용하여 데이터를 샘플링하는 방식		
		

□ 엑스트라 트리	extra trees 참고 용어 랜덤 포레스트, 앙상블 학습	[05장 283쪽]
	랜덤 포레스트와 비슷하게 동작하며 결정 트리를 사용하여 앙상블 모델을 만들지만 부트스트랩 샘플을 사용하지 않는 대신 랜덤하게 노드를 분할하여 과대적합을 감소시킴	
□ 그레이디언트 부스팅	gradient boosting	[05장 285쪽]
	깊이가 얇은 결정 트리를 사용하여 이전 트리의 오차를 보완하는 방식으로 앙상블하는 방법. 깊이가 얇은 결정 트리를 사용하기 때문에 과대적합에 강하고 일반적으로 높은 일반화 성능을 기대할 수 있음	
□ 히스토그램 기반 그레이디언트 부스팅	Histogram-based Gradient Boosting	[05장 286쪽]
	그레이디언트 부스팅의 속도를 개선한 것으로 과대적합을 잘 억제하며 그레이디언트 부스팅보다 조금 더 높은 성능을 제공. 안정적인 결과와 높은 성능으로 매우 인기가 높음	



06장 ✓

□ 히스토그램	histogram [06장 310쪽] 값이 발생한 빈도를 그래프로 표시한 것으로 보통 x축이 값의 구간(계급)이고, y축은 발생 빈도(도수)임
□ 군집	clustering [06장 314쪽] 비슷한 샘플끼리 그룹으로 모으는 작업으로 대표적인 비지도 학습 작업 중 하나
□ k-평균 알고리즘	k-means algorithm [06장 321쪽] 처음에 랜덤하게 클러스터 중심을 정하여 클러스터를 만들고 그다음 클러스터의 중심을 이동하여 다시 클러스터를 결정하는 식으로 반복해서 최적의 클러스터를 구성하는 알고리즘
□ 이너셔	inertia [06장 328쪽] k-평균 알고리즘은 클러스터 중심과 클러스터에 속한 샘플 사이의 거리를 쥌 수 있는데 이 거리의 제곱 합을 이너셔라고 함. 즉 클러스터의 샘플이 얼마나 가깝게 있는지를 나타내는 값임
□ 차원 축소	dimensionality reduction [06장 336쪽] 데이터를 가장 잘 나타내는 일부 특성을 선택하여 데이터 크기를 줄이고 지도 학습 모델의 성능을 향상시킬 수 있는 방법
□ 주성분 분석	principal component analysis, PCA [06장 336쪽] 차원 축소 알고리즘의 하나로 데이터에서 가장 분산이 큰 방향을 찾는 방법이며 이런 방향을 주성분이라 함. 원본 데이터를 주성분에 투영하여 새로운 특성을 만들 수 있음



07 장

- **인공신경망** **artificial neural network, ANN** [07장 365쪽]

생물학적 뉴런에서 영감을 받아 만든 머신러닝 알고리즘. 신경망은 기존의 머신러닝 알고리즘으로 다루기 어려웠던 이미지, 음성, 텍스트 분야에서 뛰어난 성능을 발휘하면서 크게 주목을 받고 있으며 종종 딥러닝이라고도 부름

- **딥러닝** **deep learning** [07장 368쪽]

딥러닝은 인공신경망과 거의 동의어로 사용되는 경우가 많으며 혹은 심층 신경망(deep neural network, DNN)을 딥러닝이라고 부름. 심층 신경망은 여러 개의 층을 가진 인공신경망임

- **케라스** **Keras** [07장 368쪽]

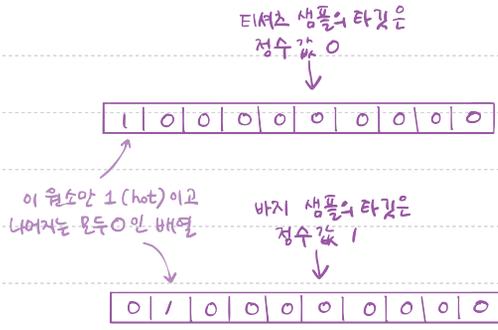
딥러닝을 위한 고수준 API를 제공하는 파이썬 라이브러리. 실제 연산은 텐서플로와 같은 백엔드가 담당하며 3.0버전부터는 텐서플로, 파이토치, 잭스를 백엔드로 사용할 수 있음

- **활성화 함수** **activation function** 참고 용어 소프트맥스 함수 [07장 375쪽]

소프트맥스와 같이 뉴런의 선형 방정식 계산 결과에 적용되는 함수

- **원-핫 인코딩** **one-hot encoding** [07장 377쪽]

타깃값을 해당 클래스만 1이고 나머지는 모두 0인 배열로 만드는 것. 다중 분류에서 크로스 엔트로피 손실 함수를 사용하려면 0, 1, 2와 같이 정수로 된 타깃값을 원-핫 인코딩으로 변환해야 함

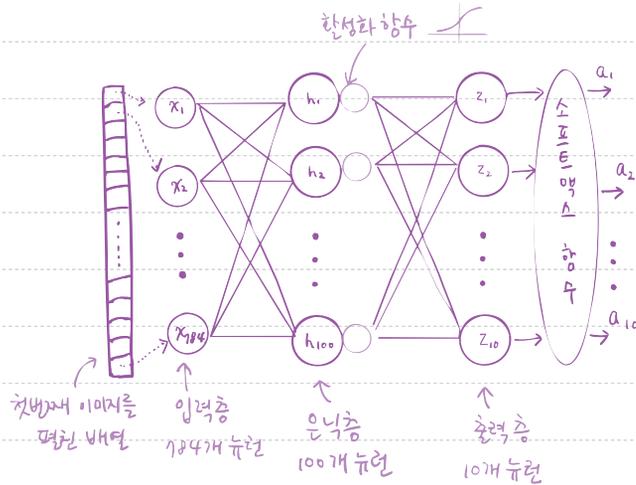


□ 은닉층

hidden layer

[07장 389쪽]

입력층과 출력층 사이에 있는 모든 층을 은닉층이라고 부름



□ 심층 신경망

deep neural network, DNN

[07장 391쪽]

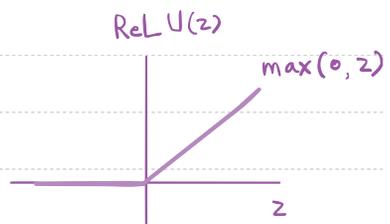
2개 이상의 층을 포함한 신경망으로 종종 다층 인공신경망, 심층 신경망, 딥러닝을 같은 의미로 사용함

□ 렐루 함수

ReLU Function

[07장 397쪽]

입력이 양수일 경우 마치 활성화 함수가 없는 것처럼 그냥 입력을 통과시키고 음수일 경우에는 0으로 만드는 함수



□ 옵티마이저	optimizer	[07장 400쪽]
	신경망의 가중치와 절편을 학습하기 위한 알고리즘 또는 방법. 케라스에는 다양한 경사 하강법 알고리즘이 구현되어 있으며 대표적으로 SGD, 네스테로프 모멘텀, RMSprop, Adam 등이 있음	
□ 적응적 학습률	adaptive learning rate	[07장 402쪽]
	모델이 최적점에 가까이 갈수록 안정적으로 수렴하도록 학습률을 낮추도록 조정하는 방법. 이런 방식들은 학습률 매개변수를 튜닝하는 수고를 덜 수 있는 것이 장점	
□ 드롭아웃	dropout	[07장 431쪽]
	훈련 과정에서 층에 있는 일부 뉴런을 랜덤하게 꺼서(즉 뉴런의 출력을 0으로 만들어) 과대적합을 막음	
□ 콜백	callback	[07장 437쪽]
	케라스에서 훈련 과정 중간에 어떤 작업을 수행할 수 있게 하는 객체로 keras.callbacks 패키지 아래에 있는 클래스로 fit() 메서드의 callbacks 매개변수에 리스트로 전달하여 사용	



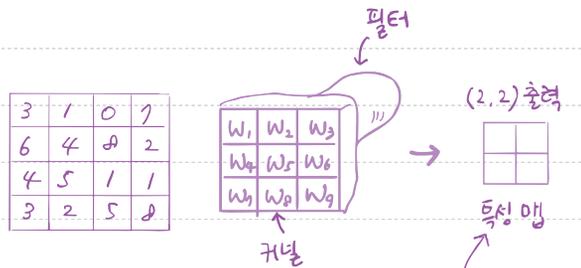
08장 ✓

□ 합성곱 **convolution** [08장 461쪽]

합성곱은 밀집층과 비슷하게 입력과 가중치를 곱하고 절편을 더하는 선형 계산이지만 밀집층과 달리 합성곱은 입력 데이터 전체에 가중치를 적용하는 것이 아니라 일부에 가중치를 곱함

□ 필터 **filter** [08장 463쪽]

밀집층의 뉴런에 해당. 뉴런 = 필터 = 커널 모두 같은 말이라 생각해도 좋음



□ 특성 맵 **feature map** [08장 465쪽]

합성곱 계산을 통해 얻은 출력을 특별히 특성 맵이라 부름

□ 패딩과 **padding과 same padding** [08장 467쪽]

세임 패딩 입력 배열의 주위를 가상의 원소(보통 0)로 채우는 것을 패딩이라고 하고 합성곱 신경망에서는 세임 패딩을 많이 사용함

□ 밸리드 패딩 **valid padding** [08장 469쪽]

패딩 없이 순수한 입력 배열에서만 합성곱을 하여 특성 맵을 만드는 경우이며 특성 맵의 크기가 줄어들 수밖에 없음

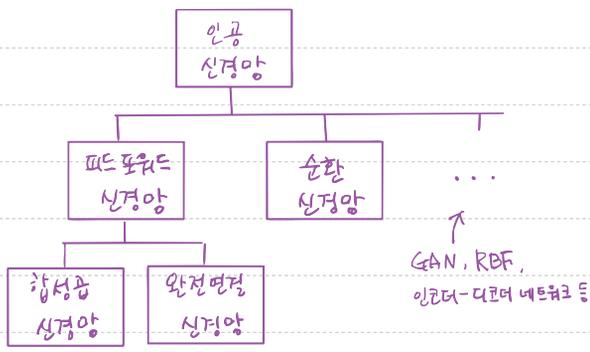
09장 ^④

□ 순차 데이터 sequential data [09장 545쪽]
 텍스트나 시계열 데이터와 같이 순서에 의미가 있는 데이터를 말함. 예를 들어 "I am a boy"는 쉽게 이해할 수 있지만 "boy am a I"는 말이 되지 않음

순차 데이터 → 텍스트. 예) "I am a boy"
 시계열. 예) 1일 15°C, 2일 17°C, 3일 16°C, ...

□ 시계열 데이터 time series data [09장 545쪽]
 일정한 시간 간격으로 기록된 데이터
 주식, 일자별 날씨 등등

□ 피드포워드 신경망 feedforward neural network, FFNN [09장 546쪽]
 입력 데이터의 흐름이 앞으로만 전달되는 신경망. 완전 연결 신경망과 합성곱 신경망이 모두 피드포워드 신경망에 속함

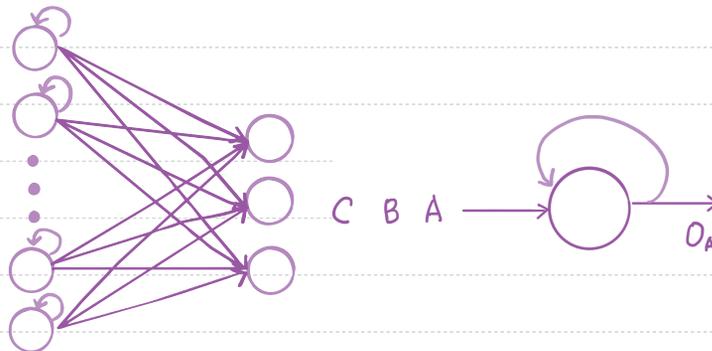


□ 순환 신경망

recurrent neural network, RNN

[09장 547쪽]

완전 연결 신경망과 거의 비슷함. 순차 데이터에 잘 맞는 인공신경망의 한 종류로 순차 데이터를 처리하기 위해 고안된 순환 층을 1개 이상 사용한 신경망



□ 셀

cell

[09장 548쪽]

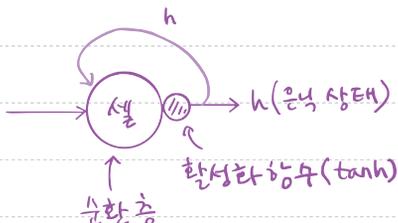
순환 신경망에서는 특별히 층을 셀이라 부르며 한 셀에는 여러 개의 뉴런이 있지만 완전 연결 신경망과 달리 뉴런을 모두 표시하지 않고 하나의 셀로 층을 표현함

□ 은닉 상태

hidden state

[09장 548쪽]

순환 신경망에서는 셀의 출력을 은닉 상태라 부름. 은닉 상태는 다음 층으로 전달 될 뿐만 아니라 셀이 다음 타임스텝의 데이터를 처리할 때 재사용됨



□ 말뭉치

corpus

[09장 559쪽]

자연어 처리 분야에서는 훈련 데이터를 종종 말뭉치라고 부름. 예를 들어 IMDB 리뷰 데이터셋이 하나의 말뭉치임

□ 토큰

token

[09장 559쪽]

일반적으로 영어 문장은 모두 소문자로 바꾸고 구두점을 삭제한 다음 공백을 기준으로 분리하는데 이렇게 텍스트에서 공백으로 구분되는 문자열 또는 단어를 토큰이라고 부름

I am a boy는 4개의 토큰. 한글은 조사가 발달해 형태소 분석을 해야 함

□ 단어 임베딩

word embedding

[09장 573쪽]

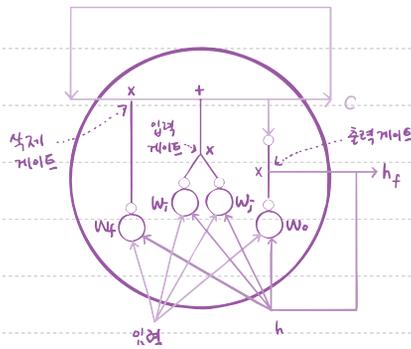
순환 신경망에서 텍스트를 처리할 때 즐겨 사용하는 방법으로 입력으로 정수 데이터를 받아 메모리를 훨씬 효율적으로 사용할 수 있음

□ LSTM

Long Short-Term Memory

[09장 596쪽]

단기 기억을 오래 기억하기 위해 고안된 순환층. 입력 게이트, 삭제 게이트, 출력 게이트 역할을 하는 작은 셀이 포함



□ 셀 상태

cell state

[09장 596쪽]

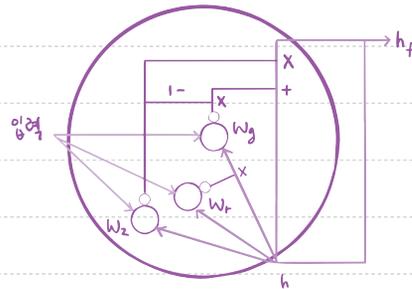
LSTM 셀은 은닉 상태 외에 셀 상태를 출력. 셀 상태는 다음 층으로 전달되지 않으며 현재 셀에만 순환됨

□ GRU

Gated Recurrent Unit

[09장 610쪽]

LSTM 셀의 간소화 버전으로 생각할 수 있지만 LSTM처럼 셀 상태를 계산하지 않고 은닉 상태 하나만 포함. LSTM보다 가중치가 적기 때문에 계산량이 적지만 LSTM 못지않은 좋은 성능을 내는 것으로 알려져 있음



10장

□ 어텐션 메커니즘	Attention mechanism	[10장 636쪽]
	인코더-디코더 RNN 구조에서 디코더가 인코더가 만든 모든 타임스텝의 은닉 상태를 참조하기 위해 고안된 구조	
□ 트랜스포머	Transformer	[10장 637쪽]
	RNN 대신 어텐션만을 사용하여 인코더-디코더 구조를 구현한 인공 신경망으로, 셀프 어텐션, 층 정규화, 잔차 연결, 드롭아웃, 피드포워드 네트워크 등으로 구성됨	
□ 셀프 어텐션	Self-attention	[10장 639쪽]
	트랜스포머의 인코더와 디코더에서 모두 사용하는 어텐션 메커니즘으로, 입력 토큰 사이의 어텐션을 계산	
□ 멀티 헤드 어텐션	multi-head-attention	[10장 641쪽]
	여러 개의 셀프 어텐션을 병렬로 처리하는 구성 요소	
□ 층 정규화	layer normalization	[10장 642쪽]
	배치 정규화와 비슷하게 입력 데이터를 정규화하는 방법이지만, 배치 정규화와 달리 각 토큰 별로 정규화를 수행함	
□ 피드포워드 네트워크	feedforward network	[10장 644쪽]
	트랜스포머 인코더와 디코더의 어텐션 층 다음에 오며, 일반적으로 두 개의 밀집 층으로 구성됨. 첫 번째 밀집 층에만 활성화 함수를 사용하며, 초기에는 ReLU를 사용했지만 근래에는 다양한 활성화 함수를 사용한 변종이 많음	

□ 위치 인코딩	positional encoding [10장 646쪽]
	토큰 임베딩 벡터에 토큰의 위치를 보상하기 위해 더해주는 실수 벡터. 트랜스포머는 입력 텍스트를 순차적으로 처리하지 않기 때문에 토큰 위치에 대한 정보를 추가하기 위해 사용함
□ 크로스 어텐션	cross attention [10장 648쪽]
	트랜스포머 모델의 디코더 블록에 있는 두 번째 어텐션 메커니즘. 크로스 어텐션에서는 인코더의 최종 출력을 키와 값으로 사용하고, 디코더의 은닉 벡터를 쿼리로 사용함
□ 대규모 언어 모델	large language model, LLM [10장 650쪽]
	기술적인 정의는 모호하지만, 일반적으로 많은 수의 모델 파라미터를 가진 트랜스포머 기반 언어 모델을 말함. 이런 모델들은 보통 수십억 개에서 수백억 개의 파라미터를 가지고 있으며, 1조 개가 넘는 파라미터를 가진 경우도 있음
□ 전이 학습	transfer learning [10장 657쪽]
	대규모 데이터셋에서 훈련된 신경망을 비슷하거나 다른 작업에 활용하는 방법으로 사전 훈련된 신경망을 그대로 사용하거나 부분적으로 미세 튜닝할 수 있음
□ BART	[10장 658쪽]
	메타에서 공개한 인코더-디코더 기반의 트랜스포머 모델. 번역, 요약과 같은 시퀀스-투-시퀀스 작업에 활용할 수 있음
□ 허깅페이스	HuggingFace [10장 663쪽]
	트랜스포머 기반의 모델을 개발하고 공유하기 위한 플랫폼으로, 많은 모델이 등록되어 있고 자연어 처리뿐만 아니라 비전과 오디오 분야의 모델도 제공함

□ transformers	[10장 663쪽]
	허깅페이스에서 만든 인기 많은 파이썬 패키지로, 사전 훈련된 트랜스포머 기반 모델을 전이 학습에 활용하고 미세 튜닝하기 위한 다양한 기능을 제공
□ 토큰화	tokenization [10장 670쪽]
	입력 텍스트를 신경망 모델에 전달하기 위해 작은 단위로 쪼개는 방법. 대표적인 방법으로 BPE, 워드피스, 셴텐스피스 등이 있음
□ 오픈소스 LLM	[10장 663쪽]
	대규모 언어 모델의 구조와 가중치가 모두 공개된 모델로, 가중치가 공개되었지만 이를 상업적인 목적으로 사용하는 기준은 모델마다 다를 수 있음. 대표적인 오픈소스 LLM: Llama, Gemma, Phi, Qwen 등
□ 클로즈드 소스	[10장 656쪽]
LLM	모델의 구조와 가중치가 공개되지 않은 독점적인 모델로, 이런 모델을 사용하려면 제공하는 회사의 인터페이스나 API를 사용해야 함. 대표적인 클로즈드 소스 LLM: GPT, Claude, Gemini 등
□ EXAONE	[10장 685쪽]
	LG AI 연구원에서 만든 디코더 기반의 트랜스포머 모델이며, 오픈소스 파운데이션 모델. 비교적 적은 파라미터 개수를 가진 모델로도 영어와 한국어 텍스트를 생성하는데 높은 성능을 냄
□ RMS 정규화	root mean square normalization [10장 687쪽]
	층 정규화의 변종으로, 분산을 계산할 때 평균을 고려하지 않는 정규화 기법. 계산량도 줄어들고 성능에도 영향이 없기 때문에 최신 LLM에서 널리 사용됨

□ top-k 샘플링

[10장 696쪽]

가장 높은 로짓(또는 확률)을 가진 상위 몇 개의 토큰을 선택하는 샘플링 전략. 이를 1로 지정하면 가장 높은 값을 가진 토큰 하나만 선택하기 때문에 항상 결정적인 결과를 얻게 됨

□ top-p 샘플링

[10장 697쪽]

확률 크기 순으로 토큰을 정렬한 후 지정한 임계 확률에 도달할 때까지 토큰을 선택하므로 샘플링할 때마다 선택 가능한 토큰의 개수가 달라짐. 이를 1로 지정하면 모든 토큰이 후보 토큰이 됨

□ GPT

Generative Pre-trained Transformer

[10장 700쪽]

오픈AI에서 만든 대규모 언어 모델로 GPT-3부터는 클로즈드 소스로 제공. 최신 GPT 모델을 사용하려면 ChatGPT 웹 인터페이스나 오픈AI에서 제공하는 API를 사용해야 함