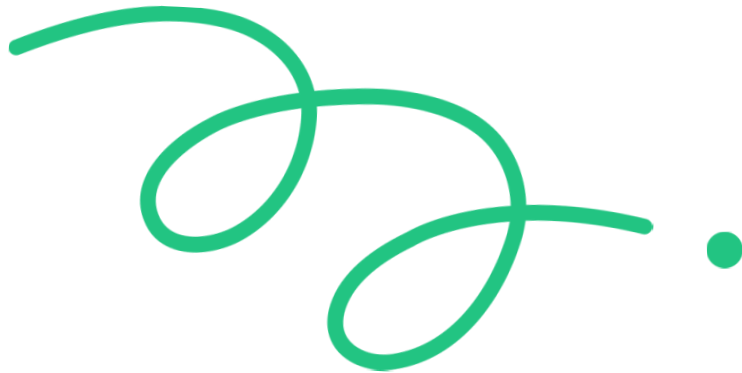


AI 프롬프트 작성 가이드북

AI 학습 메이트와 똑똑하게 소통하는 기술



목 차

01

프롬프트 엔지니어링, 뭘까?

- 프롬프트, 어떻게 생겼을까? (기본 구성 요소)
- 프롬프트 잘 쓰는 5가지 원칙
- 레벨 업! 고급 프롬프트 패턴
- 이걸 덤~ 프롬프트 샘플 세트

02

이론 공부할 때 쓰는 똑똑한 질문법

- 이해를 쑥쑥 올리는 질문법
- 생각의 힘을 키우는 비판적 질문

03

코드 실습할 때 써먹는 질문 스킬

- 코드 오류, 시랑 같이 잡아내기
- 코드 생성과 리뷰까지 한 번에
- 코드 프롬프트 예시 모음

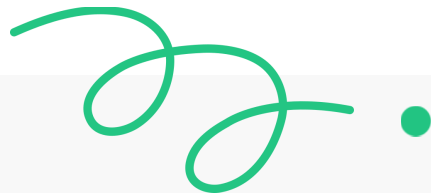
04

내 프롬프트, 점검해볼까?

- 자가진단 체크리스트
- 이걸 피하자! 잘못된 프롬프트 습관
- 자주 하는 질문 (FAQ)

1 프롬프트 엔지니어링, 뭘까?

- 프롬프트, 어떻게 생겼을까? (기본 구성 요소)
- 프롬프트 잘 쓰는 4가지 핵심 원칙
- 레벨 업! 고급 프롬프트 패턴



프롬프트 엔지니어링

AI 학습 파트너를 위한 고급 소통 기술

이제 AI는 단순히 검색만 해주는 도구가 아니에요.
우리 공부를 옆에서 든든하게 도와주는 '학습 파트너'로 성장하고 있죠.

앞으로는 AI와 함께 복잡한 문제도 풀어내고, 마치 팀 프로젝트 메이트처럼 협력할 수 있어요.
이번 가이드에서는 프롬프트 작성의 기본부터 고급 스킬까지 차근차근 배우면서,
AI를 진짜 똑똑한 파트너로 활용하는 방법을 알려드릴 거예요!

| 프롬프트, 어떻게 생겼을까? (기본 구성 요소)

1. 지시사항

Instruction

LLM 모델이 수행해야 할 구체적인 작업에 대한 설명
명확하고 구체적인 행동 지침 포함

2. 맥락

Context

모델이 더 나은 답변을 할 수 있도록 제공되는 배경 정보
관련된 예시나 참조 정보 포함

3. 입력 데이터

Input Data

해결하고자 하는 문제나 질문에 대한 구체적인 데이터
모델이 처리해야 할 실제 내용

4. 출력 지시자

Output Indicator

원하는 결과물의 형식이나 구조를 명시
응답의 길이, 형식, 스타일 등을 지정

| 프롬프트 잘 쓰는 4가지 핵심 원칙



**명확하고
구체적으로 묻기**

1 무엇을, 어느 범위까지, 어떤 조건으로 필요한지 숫자/예시/제약을 붙여서 물어보세요.

- 나쁜 예시: "머신러닝이 뭐예요?"
- 좋은 예시: "비전공자가 이해할 수 있게 머신러닝의 개념과 중요한 특징을 설명해줘."



**필요한
맥락 제공**

2 목표(왜), 상태(레벨, 도구, 버전), 이미 시도한 것과 결과를 함께 알려주면 품질이 확 올라가요.

- 나쁜 예시: "셔츠를 어디에서 살 수 있나요?"
- 좋은 예시: "서울 강남에서 정장용 남성 셔츠를 구매할 곳을 추천해 줄래?"



**원하는 출력
형식 지시**

3 목록/표/코드블록/단계별 등 형식, 길이, 톤, 언어를 미리 지정하면 정리된 답을 받기 쉬워요.

- 나쁜 예시: "데이터 분석 결과를 알려줘."
- 좋은 예시: "데이터 분석 결과를 표로 정리하고, 주요 통계 값(평균, 중앙값, 표준편차)을 포함해줘."



**반복과
개선**

4 첫 답변은 초안! "여기서 A는 유지하고 B만 더 자세히" 처럼 피드백을 주며 다듬어 가세요.

- 나쁜 예시: "다시 해줘."
- 좋은 예시: "결과가 마음에 들지 않아. 문제의 배경과 중요성을 추가하고, 문단 구성을 더 논리적으로 작성해줘."

| 레벨 업! 고급 프롬프트 패턴



역할 부여 (Role, Persona)

5 역할/대상/평가기준을 명확히 주면, 맥락에 맞는 전문적인 답변을 끌어낼 수 있어요.

- 나쁜 예시: “알고리즘에 대해 설명해줘.”
- 좋은 예시: “당신은 알고리즘 강사입니다. 초보자를 대상으로 정렬 알고리즘의 개념과 예제를 쉽게 설명해주세요.”



예시 제공 (Shot Prompting)

6 원하는 스타일의 입력-출력-예시(또는 선호 표현)를 보여주면, AI가 그 패턴을 정확히 따라가요.

- 나쁜 예시: “데이터를 분석해줘.”
- 좋은 예시: “다음과 같은 형식으로 데이터를 분석해줘. 1. 데이터의 요약통계, 2. 이상치 탐지 결과, 3. 시각화 포함”



단계적 접근 (Chain-of-Thought)

7 큰 문제를 작은 단계로 쪼개 순서대로 진행하며, 각 단계를 확인하고 넘어가세요.

- 나쁜 예시: “이 데이터로 최적의 머신러닝 모델을 만들어줘.”
- 좋은 예시: “1. 데이터를 탐색하고, 전처리 과정을 제안해주세요.
2. 적합한 머신러닝 후보를 제시해주세요.
3. 각 알고리즘의 장단점을 비교해 최적의 모델을 추천해주세요.”

INSIGHT

‘맥락 제공’과 ‘역할 부여’는 사실 AI의 넓고 방대한 지식 창고를, 내가 필요한 영역으로 딱 맞게 좁혀주는 핵심 기술이에요.

이걸 잘 활용하면 AI가 훨씬 예측 가능한 답을 주고, 답변의 정확도와 품질이 눈에 띄게 높아진답니다! ☺

| 이건 덤~ 프롬프트 샘플 세트

1. [대상 수준]에게 [주제]를 [길이/형식]으로, [비유/예시] 포함해서 설명해줘.

- 고등학생 수준으로, 은유 1개와 150자 내 요약으로 딥러닝 기술의 핵심을 설명해줘.

2. 나는 [배경/수준], 목표는 [학습 목표]. [제약]을 지켜 설명해줘.

- 나는 [비전공 대학생]이고, 오늘 수업한 주제는 [경사하강법]이야. 나의 학습 목표는 [직관적인 원리 이해]야.
수식은 최소화하고, [일상 비유 1개 + 3단계 요약]으로 설명해줘.

3. [개념A] vs [개념B]를 [항목1/항목2/항목3] 표로 [분량] 이내 요약해줘.

- 지도학습과 비지도학습을 [정의/대표 알고리즘/데이터 요구/장단점]을 4열 표로 정리해줘. 각 칸은 25자 이내.
- 정확도/정밀도/재현율/F1을 [입력 - 계산 - 해석] 3단계로 불릿 정리해줘. 숫자 예시는 간단히.

4. 이 프롬프트를 [평가기준]으로 평가하고, 수정안을 전/후로 제시해줘.

- 내 프롬프트와 AI 응답을 보고, SCORE 기준으로 개선 3가지를 제안해줘. 각 개선은 수정 전과 후의 예시를 포함.

| 이건 덤~ 프롬프트 샘플 세트

5. 너는 [역할]. [주제]를 가르치고 2문항
퀴즈+정답을 제공해줘.

- 너는 'AI 개론 조교'야. '과적합'을 시험공부
비유로 설명하고, 마지막에 2문항 퀴즈를
출제해줘 (정답 포함).

6. [개념]의 대표 예시 1, 작동하지 않는 반례 1과
이유를 설명해줘.

- 경사하강법이 실패하거나 지역해에 머무는 상황
2가지를 제시하고, 각각 왜 그런지 직관적으로
설명해줘.

7. [개념]을 [단계 : 정의-비유-절차-오개념검증-
100자 요약] 순서로 정리해줘.

- 강화학습을 스포츠 훈련에 비유해서 3문장으로
설명해줘. 연습-피드백-보상 연결이 드러나게.

I 이건 덤~ 프롬프트 샘플 세트

❖ 종합해서 한번에 쓴다면?

나는 비전공 대학생이고,
경사하강법을 직관적으로 이해하고 싶어.

수식은 최소화하고, 언덕 내려가기 비유로 3단계
절차를 불릿으로, 마지막에 실패/수렴 문제 2가지
와 해결 힌트(러닝메이트, 초기값)를 덧붙여줘.

형식 : [3단계 절차 표]

+ [실패 상황 2개 불릿] + [100자 요약]

- 경사하강법은 **가장 낮은 지점(최솟값)**을 찾아가는 과정을 말해요. 마치 안개가 자욱한 산꼭대기에서 가장 낮은 곳으로 내려가는 것과 같아요. 어디가 정상인지 모를 때, 발밑의 경사만 보고 가장 가파른 방향으로 한 걸음씩 내려가는 거죠.

경사하강법 3단계 절차

단계	설명	비유
1. 현재 위치 파악	현재 위치에서의 **기울기(경사)**를 계산해요. 이 기울기는 현재 위치에서 어느 방향으로 가야 더 낮아질지 알려줘요.	발밑의 경사가 어느 방향으로 가장 가파른지 느껴봐요.
2. 다음 위치 이동	계산된 기울기에 학습률 을 곱한 만큼 이동해요. 학습률은 한 걸음에 얼마나 멀리 갈지 정하는 보폭이에요.	발밑 경사가 가파를수록, 그리고 보폭을 크게 할수록 더 멀리 내려가요.
3. 반복	목표 지점(골짜기)에 도착하거나, 더 이상 내려갈 곳이 없을 때까지 1, 2단계를 계속 반복해요.	평지에 도착하거나 더 이상 발밑 경사가 느껴지지 않을 때까지 계속 걸어 내려가는 거예요.

경사하강법의 실패 상황

- 학습률이 너무 클 때:** 보폭이 너무 커서 가장 낮은 골짜기를 지나쳐 버리는 거예요. 마치 큰 걸음으로 강충깡충 뛰다가 골짜기를 못 찾고 계속 왔다 갔다 하는 상황이지요. **학습률**을 적절히 조절해야 해요. 🏹 🌀
- 지역 최솟값 문제:** 여러 개의 골짜기(최솟값)가 있는 지형에서, 시작한 위치에서 가장 가까운 작은 골짜기에 갇혀버리는 거예요. 더 깊은 골짜기가 옆에 있는데도 말이지요. **초기값**을 여러 군데에서 시작해보면 더 깊은 골짜기를 찾을 확률이 높아져요. 🏠 🔍

100자 요약

경사하강법은 함수의 최솟값을 찾기 위해 현재 위치의 **경사**를 계산하고, 정해진 ****보폭(학습률)****만큼 **반복적으로 이동**하며 가장 낮은 지점에 도달하는 최적화 방법입니다. 너무 큰 보폭은 골짜기를 놓치고, 초기 위치에 따라 얕은 골짜기에 갇힐 수 있어 **학습률과 초기값 설정**이 중요합니다.

👉 🔄 🔄 🔄 🔄 🔄 :

△ 작성한 프롬프트에 대한 'Gemini 2.5-Flash' 응답 결과

I 이건 덤~ 프롬프트 샘플 세트

❖ 종합해서 한번에 쓴다면?

나는 SW전공 대학생이고, Java를 사용하는 웹 개발자의 입장에서 경사하강법을 이해하고 싶어.

수식은 최소화하고, 웹 기술에 빗대어 직관적으로 설명해줘. 마지막에는 실패/수렴 문제 2가지와 해결 힌트(학습률 튜닝, 초기값 설정)를 덧붙여줘.

형식 : [3단계 절차 표] + [실패 상황 2개 불릿] + [100자 요약]



웹 개발자를 위한 경사하강법(Gradient Descent) 설명

경사하강법은 **모델이 예측한 값과 실제 값의 차이(오차)**를 줄여나가는 최적화 알고리즘입니다. 이 과정을 웹 개발에 비유하자면, 웹사이트의 성능을 개선하는 작업과 매우 유사합니다.

예를 들어, 웹사이트 로딩 속도를 최적화한다고 생각해 봅시다.

- **모델**은 웹사이트입니다.
- **예측**은 현재 웹사이트의 로딩 속도입니다.
- **실제 값**은 우리가 목표로 하는 로딩 속도(예: 1초 이내)입니다.
- **오차**는 현재 속도와 목표 속도의 차이입니다.
- **경사하강법**은 이 오차를 최소화하기 위해 백엔드 코드 최적화, 이미지 용량 줄이기, CDN 도입 등 다양한 개선 작업을 시도하는 과정입니다.

경사하강법 3단계 절차

단계	웹 개발 비유	머신러닝 과정
1단계: 현재 위치 파악	웹사이트 로딩 속도를 측정합니다. (예: 5초)	**손실 함수(Loss Function)**를 사용해 현재 모델의 오차를 계산합니다. (예: $y_{\text{예측}} - y_{\text{실제}}$)

(중략)

- **해결 힌트:** 초기값을 다양하게 설정하여 여러 지점에서 경사하강법을 시작해 볼 수 있습니다.

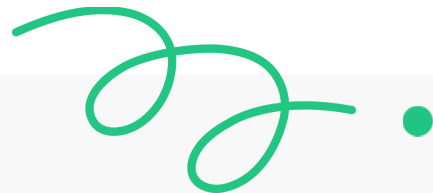
100자 요약

경사하강법은 웹 개발자가 웹사이트 성능을 최적화하는 과정과 유사합니다. 현재 성능을 측정(오차 계산)하고, 개선 방향을 파악(기울기 계산)한 뒤, 개선 작업을 반복(가중치 업데이트)하여 오차를 최소화합니다. 학습률이 너무 크면 최적화에 실패하고, 초기값에 따라 국소 최소값에 갇힐 수 있어 주의해야 합니다.

△ 작성한 프롬프트에 대한 'Gemini 2.5-Flash' 응답 결과

2 이론 공부할 때 쓰는 똑똑한 질문법

- 이해를 쑥쑥 올리는 질문법
- 생각의 힘을 키우는 비판적 질문

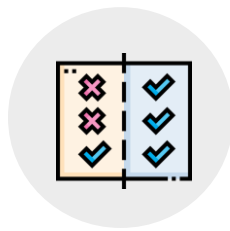
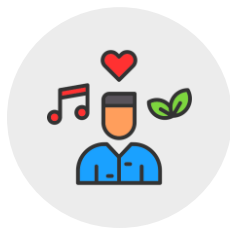


깊은 이해를 위한 AI 질문 전략

AI 이론은 추상적이고 어렵게 느껴질 때가 많아요.
그래서 단순히 정보를 찾는 데 그치지 않고,
프롬프트를 똑똑하게 써서 AI를 내 전담 튜터처럼 활용하는게 중요해요.

이 전략을 쓰면, 단순 암기가 아니라 스스로 생각하고 질문하며 개념을 깊이 이해할 수 있어요.
결국 AI는 검색 엔진이 아니라 나만의 학습 파트너가 되는 거죠!

| 이해를 쑥쑥 올리는 질문법



질문 패턴	개념 정의 및 난이도 조절	비유와 일상 예시 활용	개념 비교 및 대조	요약 및 핵심 정리	오개념 검증 및 예외 케이스 질문
설명	새로운 개념을 수준에 맞춰 설명 요청 (ELI5: Explain Like I'm 5)	추상적 개념을 구체적인 비유로 연결	유사하지만 다른 개념들의 차이점 파악	핵심을 빠르게 파악하기 위한 요약 요청	이해한 내용 검증 및 적용 범위 확인 요청
예시	“5살 어린이에게 말하듯이 뉴럴 네트워크가 뭔지 설명해줘.”	“강화학습을 스포츠 훈련에 비유해서 설명해줘.”	“지도학습과 비지도학습을 정의, 데이터 특성, 활용 사례 측면에서 표로 비교해줘.”	“‘Attention Is All You Need’ 논문의 핵심 기여 3가지를 bullet point로 요약해줘.”	“경사하강법이 최적해를 찾지 못하고 실패하는 대표적인 상황은 뭐야? 이 방법이 작동하지 않는 반대 예시를 들어줘.”
효과	<ul style="list-style-type: none"> 인지적 과부하 감소 핵심 개념 이해 촉진 	<ul style="list-style-type: none"> 정보의 인코딩 및 장기 기억 전환 촉진 	<ul style="list-style-type: none"> 개념 간 관계 이해 강화 <ul style="list-style-type: none"> 복잡한 개념의 직관적 이해 도움 	<ul style="list-style-type: none"> 정보의 밀집도를 높여 시간적 효율 극대화 중요한 내용을 빠르게 학습하여 집중도 향상 	<ul style="list-style-type: none"> 대안적 문제 해결 전략 탐색 유도 탐구 기반 학습과 비판적 사고 촉진

| 이해를 쏙쏙 올리는 질문법

다음의 예시와 같이 질문해 보세요! (멀티턴 방식)

Level 1 (개요)

“Transformer의 핵심 개념 3가지를 각각 한 문장으로 요약해줘. 비전공자가 이해할 수 있게 설명해줘.”

Level 2 (상세)

“방금 설명한 ‘Self-Attention’, ‘Positional Encoding’, ‘Feed-Forward Network’가 Transformer 아키텍처에서 각각 어떤 역할을 하는지 구체적으로 설명해줘. 이해하기 쉽게 일상의 개념에 비유해서 알려줘.”

Level 3 (응용)

“Transformer와 RNN을 정의, 구조적 차이, 활용 사례 측면에서 비교하고, Transformer가 긴 문장 처리에서 더 효율적인 이유를 설명해줘. 그리고 이것 표로 정리해줘.”

INSIGHT

설명을 들은 후, “왜 그런 결과가 나오나요?” 또는 “이 개념이 왜 중요한가요?” 같은 질문을 던져보세요. 이렇게 하면 근본적인 원리와 중요성을 더 잘 이해할 수 있을 거예요. ☺

I 이건 덤~ 프롬프트 샘플 세트

❖ 3단계 멀티턴으로 쓰다면?

- ① 나는 비전공 대학생이고, CNN(합성곱 신경망)이 뭔지 쉽게 알고 싶어. 대학생 초급 수준으로 정의와 기본 아이디어를 한 문단으로 설명해줘.
- ② 좋아, 이제 일상 비유를 활용해서 CNN을 설명해줘. 예를 들어 사람 얼굴을 알아 본다던가, 음식 사진을 분류하는 비유를 써도 좋아. 그리고 CNN의 주요 구성 요소(합성곱, 풀링, 완전연결층)를 표로 정리해줘.
- ③ 마지막으로 사람들이 CNN에 대해 흔히 아는 오해(오개념) 2가지를 알려주고, 각각을 어떻게 바로잡아야 하는지 설명해줘. 그리고 전체 내용을 100자 핵심 요약으로 정리해줘.

◆ CNN(합성곱 신경망)은 컴퓨터가 이미지나 동영상 같은 시각 데이터를 이해하고 인식하도록 돕는 인공 신경망의 한 종류입니다. 일반적인 인공 신경망과 달리, CNN은 이미지의 공간적 구조(픽셀의 위치 관계)를 효율적으로 처리하도록 설계되었어요. 핵심 아이디어는 ****필터(filter)****라는 작은 겹겹의 패턴 탐지기를 사용하여 이미지 전체를 훑으며 특징을 추출하는 겁니다. 마치 사람이 특정 사물의 특징(예: 코너, 선, 색상)을 인식하듯이, CNN은 이런 필터를 통해 이미지의 다양한 특징들을 계층적으로 학습하고 결합하여 최종적으로 이미지가 무엇인지 분류하거나 식별하는 거죠.

◆ 좋아요, CNN의 작동 원리를 일상적인 비유로 쉽게 설명해 드릴게요.

사람 얼굴을 알아보는 CNN

CNN을 사람의 뇌가 얼굴을 인식하는 과정에 비유할 수 있습니다. 우리는 누군가의 얼굴을 볼 때, 전체를 한 번에 파악하기보다 눈, 코, 입, 그리고 얼굴형 같은 ****특징들(features)****을 먼저 인식합니다.

1. 눈 코 입 특징 찾기 (합성곱 층, Convolutional Layer):

CNN의 첫 번째 단계는 마치 우리가 얼굴의 세부 특징을 찾는 것과 같습니다. CNN은 ****필터(filter)****라고 불리는 작은 돋보기를 가지고 이미지의 모든 부분을 꼼꼼히 훑습니다. 예를 들어, 한 필터는 '수평선'을, 다른 필터는 '곡선'을 찾아내도록 학습됩니다. 이 필터들은 이미지 위를 한 칸씩 이동하며, 이미지에서 필터와 비슷한 패턴(예: 눈, 코, 입의 윤곽)을 발견하면 그 위치에 신호를 보냅니다.

◆

CNN에 대한 흔한 오해와 진실

1. CNN은 이미지 전체를 한 번에 학습한다? ❌

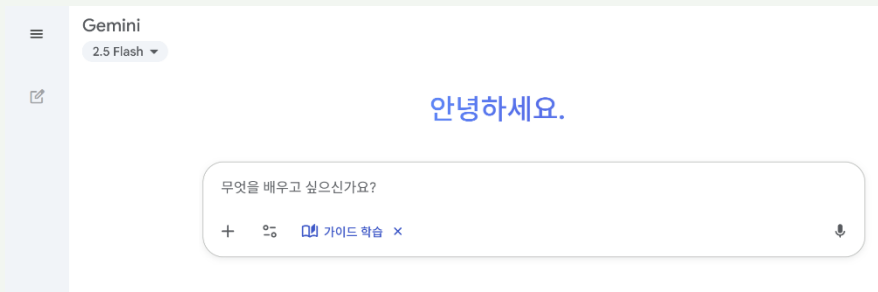
- **오해:** CNN이 이미지를 통째로 보고 한 번에 사물을 인식하는 것으로 생각하기 쉽습니다.
- **진실:** CNN은 이미지 전체를 한 번에 보는 것이 아니라, **합성곱 층**을 통해 작은 필터로 이미지를 **부분 부분 훑어가며** 특징을 추출합니다. 이 작은 특징들을 여러 층을 거치며 점차 결합하여 최종적으로 전체 이미지를 이해하게 됩니다. 이는 마치 거대한 퍼즐을 맞추는 것처럼, 조각들을 먼저 파악하고 나서 전체 그림을 완성하는 과정과 같습니다.

△ 작성한 프롬프트에 대한 'Gemini 2.5-Flash' 응답 결과

| 생각의 힘을 키우는 비판적 질문

소크라테스식 질문 기법 (Socratic Prompting)

- 고대 소크라테스의 협력적 논쟁 대화 방식에서 영감을 받은 질문 기법
- 단순히 사실과 정답을 전달하는 대신, 단계적/명료한 질문으로
 - ① 가정을 드러내고, ② 함의를 탐구하며, ③ 스스로 논리적인 결론에 도달하도록 지원
- 서비스 적용 예시 : OpenAI 'ChatGPT - 공부하기', Google 'Gemini - 가이드 학습'



| 생각의 힘을 키우는 비판적 질문

다음의 예시와 같이 질문해 보세요! (싱글턴 방식)

[역할]	너는 비전공 대학생을 돕는 AI 개론 조교이자 소크라테스식 안내자다.
[대상/제약]	나는 비전공 대학생이다. 수식·전문용어를 최소화하고, 비유와 일상 예시를 활용해 설명해줘.
[목표]	LangChain의 핵심 개념과 구성요소(LLM·프롬프트·체인·도구·메모리·리트리버/벡터저장소·에이전트)를 스스로 도출하고, RAG와 에이전트 활용 맥락, 한계/예외 상황까지 이해한다.
[진행 방식]	한 번에 질문 1개만 해. 내가 답하면 그것을 바탕으로 다음 질문을 이어가. 내가 1분 내 답하지 않는다고 가정되면, 합리적 가정을 두고 Q→A를 한 번에 진행해도 좋다.
[단계]	1) 선개념 확인: “LangChain을 들어본 적이 있나요? 알면 한 문장 정의를, 모르면 예상 정의를 말해보세요.” 2) 정의 정교화: 내 답을 바탕으로 일상 비유(레고 블록/레시피)로 2문장 설명 + 핵심 용어 3개(한 줄 정의). 3) 구성요소 찾기: 입력→처리→출력 흐름에서 필요한 블록을 스스로 말하게 유도하고, 누락 블록을 질문으로 채워줘. 4) 사용 맥락 구분: “언제 LangChain이 유리하고, 언제 API 직호출이 간단한가?” 반례·예외 2가지 제시 요청. 5) 미니 설계: “메모 몇 줄짜리 RAG 앱”을 3단계로 설계하게 유도(문서 준비→임베딩/검색→응답 생성). 6) 오개념 점검: 흔한 오해 2개를 물어보고, 맞/틀린 이유를 내게 검토시켜줘. 7) 요약/실천: 100자 요약 + 내가 바로 할 수 있는 다음 행동 1가지를 제안.
[출력 형식]	• 각 단계: 질문 1개 + 내 답 입력칸 표기(예: □내 답) • 표 1개: “구성요소/역할/한줄예” 3열 • 마지막 줄: 100자 요약
[톤]	따뜻하고 간결하게. 핵심만. 지금 1단계 질문부터 시작해줘.

I 이건 덤~ 프롬프트 샘플 세트

❖ Gemini '가이드 학습'으로 질문하기

AI가 스스로 학습한다는 게 어떤 의미인지 잘 모르겠어요.

INSIGHT

Gemini '가이드 학습' 기능은 소크라테스식 질문 기법을 이미 이해하고 있기 때문에, 별도의 복잡한 설정 없이도 학습에 최적화된 AI 응답 환경을 바로 경험할 수 있어요!

AI가 제시하는 꼬리 질문에 직접 답하면서, 이해가 부족했던 개념을 정리하고 더 깊이 있는 학습으로 확장할 수 있습니다.

- ◆ AI가 스스로 학습한다는 개념, 정말 좋은 질문이에요. 함께 그 의미를 찬찬히 파헤쳐 봐요. 제가 질문을 드리며 가이드를 해 드릴게요.

AI가 '스스로 학습한다'는 건, 우리가 사람에게 "고양이 사진"이라고 알려주는 것처럼 AI에게 수많은 예시를 (중략)

자, 이 개념을 더 깊이 이해하기 위해 다음 중 어떤 것부터 알아보고 싶으신가요?

1. **AI가 스스로 학습하는 '과정'**은 어떻게 되는지? (예: 지도 학습, 비지도 학습)
2. **AI의 학습이 실제로 쓰이는 '사례'**는 어떤 것들이 있는지? (예: 추천 시스템, 자율 주행)
3. AI의 학습에 숨겨진 **'한계점'**이나 **'문제점'**은 무엇인지? (예: 데이터 편향, 설명 불가능한 판단)

1번부터 알아볼래요.

- ◆ 좋아요, AI가 실제로 어떻게 학습하는지 그 **과정**부터 차근차근 알아볼까요?

(중략)

이전에 우리가 이야기했던 고양이 사진 학습이 바로 지도 학습의 대표적인 예시예요.

그렇다면, 이 고양이 사진 학습에서 **'선생님'**의 역할은 무엇이었을까요?

어떤 모습이 고양이 귀의 특징인지 알려주는 것이라고 생각해요.

- ◆ 맞아요, 아주 좋은 생각이에요. 🐾

더 정확히 말하면, AI는 **이 사진은 고양이다**라는 **정답 라벨**과 함께 사진을 보게 됩니다. 그리고 그 라벨 덕분에 AI는 스스로 사진 안에서 '고양이 귀', '수염' 같은 특징들을 찾아내고 학습하는 거죠.

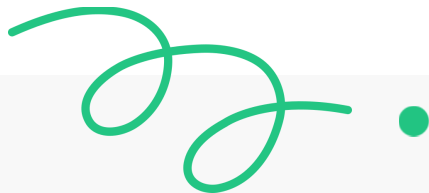
바로 이런 식으로 **정답이 있는 데이터**(라벨이 있는 데이터)를 가지고 학습하는 것이 ****지도 학습(Supervised Learning)****이에요.

△ 작성한 프롬프트에 대한 'Gemini 2.5-Flash - 가이드 학습' 응답 결과

3

코드 실습할 때 써먹는 질문 스킬

- 코드 오류, AI와 같이 잡아내기
- 코드 생성과 리뷰까지 한 번에
- 코드 프롬프트 예시 모음



| 코드 오류, 시랑 같이 잡아내기



에러 메시지 분석 요청

- 에러 메시지를 그대로 프롬프트에 포함하여 질문
- 오류 메시지의 의미와 해결책 요청
- 예시 : "TypeError: 'int' object is not iterable 오류 발생. 이 오류가 무슨 뜻이고, 어떻게 해결할 수 있을까?"



증상 기반 디버깅

- 명확한 오류 메시지 없이 예상과 다르게 동작할 때 활용
- 구체적인 증상을 설명하고 원인 진단 요청
- 예시 : "Training loss는 감소하는데 validation loss는 증가해. 가능한 원인과 해결책은?"

| 코드 생성과 리뷰까지 한 번에

AI를 '페어 프로그래머'로 활용하기

AI는 단순한 코드 생성기가 아니에요.
조금만 똑똑하게 물어보면, 개발 과정 전체를 함께하는
든든한 페어 프로그래머가 될 수 있죠!

효과적인 프롬프트 전략을 쓰면,
코드 작성, 코드 이해, 디버깅, 최적화까지
모든 과정에서 AI가 옆에서 같이 도와줍니다.

이렇게 활용하면 문제 해결은 더 빨라지고,
결과물의 품질은 훨씬 좋아져요!



코드 생성 요청

- 기능, 언어, 입출력 형식, 구현 방식 등 명확한 제약 조건 제시
- 예시: "파이썬으로 리스트에서 짝수만 추출하는 함수를 작성해줘."



코드 이해 및 분석

- 코드의 기능과 구조를 체계적으로 분석 요청
- 전체 개요, 라인별 분석, 잠재적 버그 등 형식 지정



코드 최적화 및 리팩토링

- 기존 코드 제시 후 명확한 개선 목표 명시
- 예시: "이 퀵소트 알고리즘을 더 빠르게/메모리 효율적으로 사용하는 방법이 있을까? 개선된 코드를 제안해 줘."



기타 유용한 패턴

- 언어 간 코드 변환
- 문서 내 주석 자동 추가

| 코드 프롬프트 예시 모음

다음의 예시와 같이 질문해 보세요! (싱글턴 방식)

[역할/수준] 당신은 시니어 ML 멘토입니다. 저는 비전공 대학생입니다.

[목표] 아래 경사하강법 코드의 흐름을

 1) 단계 요약

 2) 수식 대응(MSE, 그래디언트)

 3) 하이퍼파라미터(lr, n_iter, tol) 영향으로 설명해 주세요.

[형식] 불릿 포인트 + 100자 요약. 마지막에 개선점 3가지(학습률 스케줄, 정규화, 수렴 체크 강화)를 제안해 주세요.

[코드] (이 파일 전체 또는 핵심 함수 붙여넣기)

[역할/수준] 당신은 디버깅 코치입니다.

[상황] 아래 에러 로그를 읽고, 오류 원인(한줄) → 상세 설명(쉬운말) → 수정 방안(코드 조각) → 재발 방지 체크리스트(3개) 순서로 알려주세요.

[에러 로그] (콘솔 에러 내용 붙여넣기)

[추가] 입력/라벨 차원 불일치, 학습률 과대, dtype 문제 같은 흔한 원인도 점검해 주세요.

INSIGHT

그래도 만약 프롬프트를 쓰는 게 어렵다면? 일단 요구사항을 꼭 나열해서 작성하고, AI에게 템플릿에 맞춰서 수정해달라고 요청해 보세요!
‘메타 프롬프트’ 를 통해 더 나은 프롬프트를 생성하거나 개선할 수 있습니다 ☺

I 이건 덤~ 프롬프트 샘플 세트

❖ Gemini 에게 코드 해석 물어보기

[역할/수준] 당신은 시니어 ML 멘토입니다. 저는 비전공 대학생입니다.

[목표] 아래 경사 하강법 코드의 흐름을 (1) 단 개요 요약(2) 수식 대응(MSE, 그래디언트). (3) 하이퍼 파라미터 영향으로 설명해 주세요.

[형식] 불릿 포인트 + 100자 요약. 마지막에 개선점 3가지(학습률 스케줄, 정규화, 수렴 체크 강화)를 제안해 주세요.

[코드] << 생략 >>

◆ 안녕하세요. 머신러닝을 공부하는 비전공자이시군요. 아주 좋은 출발입니다! 경사 하강법 코드를 분석하고 이해하기 쉽게 설명해 드릴게요.

100자 요약

이 코드는 경사 하강법을 이용해 주어진 데이터를 가장 잘 설명하는 직선($y = wx + b$)을 찾는 과정이에요. 예측값과 실제값의 차이(오차)를 줄이는 방향으로 기울기(w)와 절편(b)을 조정합니다.

$$y = wx + b$$

(오차)를 줄이는 방향으로 기울기(w)와 절편(b)을 조정합니다.

w

(절편)

(중략)

개선점 3가지 제안

1. 학습률 스케줄링 적용:

현재는 고정된 학습률(0.5)을 사용하지만, 학습 초반에는 빠르게, 후반에는 세밀하게 학습률을 줄여 나가는 **학습률 스케줄링(Learning Rate Scheduling)**을 적용하면 더 안정적으로 수렴할 수 있습니다.

2. 정규화 도입:

데이터가 여러 개의 피처(Feature)를 가질 경우, 각 피처의 스케일이 다르면 학습이 불안정해질 수 있습니다. **정규화(Normalization)**를 통해 모든 피처의 스케일을 맞춰주면 더 효율적인 학습이 가능합니다.

3. 수렴 체크 강화:

현재는 오차가 특정 값 이하로 떨어지면 학습을 멈추지만, 학습을 멈출지 결정하는 다른 조건들을 추가할 수 있습니다. 예를 들어, 일정 횟수 동안 오차가 더 이상 감소하지 않을 때 학습을 멈추는 **조기 종료(Early Stopping)**를 고려해 볼 수 있습니다.

혹시 더 궁금한 점이 있으시면 언제든지 물어보세요!

△ 작성한 프롬프트에 대한 'Gemini 2.5-Flash' 응답 결과

I 이건 덤~ 프롬프트 샘플 세트

❖ Gemini 에게 코드 에러 물어보기

[역할/수준]

당신은 디버깅 코치입니다.

[상황]

아래 에러 로그를 읽고, 오류 원인(한줄) → 상세 설명(쉬운말) → 수정 방안(코드 조각) → 재발 방지 체크리스트(3개) 순서로 알려주세요.

[에러 로그]

RuntimeError: CUDA out of memory

❖ 오류가 발생했군요. 걱정 마세요, 함께 해결해 봐요!

오류 원인

GPU 메모리가 부족해서 발생하는 문제입니다.

상세 설명

RuntimeError: CUDA out of memory 는 **GPU(그래픽 카드)**의 VRAM(비디오 램) 용량이 부족할 때 발생하는 에러예요. 딥러닝 모델은 학습 과정에서 GPU에 연산에 필요한 데이터(이미지, 텍스트 등)와 모델

(중략)

수정 방안

배치 크기(Batch Size)를 줄여주세요.

딥러닝 모델은 한 번에 여러 개의 데이터를 묶어서 학습하는데, 이 묶음의 크기를 **배치 크기**라고 합니다. 이 크기를 줄이면 한 번에 GPU에 올라가는 데이터의 양이 줄어들어 메모리 부족 문제를 해결할 수 있습니다.

(중략)

```
# 수정 후
batch_size = 32 # 또는 16, 8 등 더 작은 값으로 변경
# 학습 코드가 여기에 있다고 가정
```

재발 방지 체크리스트

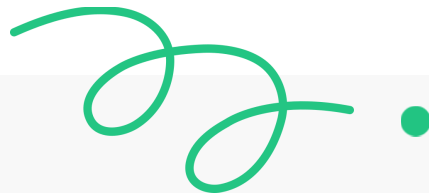
- **배치 크기 확인:** 모델을 실행하기 전에 **배치 크기**가 GPU 메모리 용량을 초과하지 않는지 확인하세요. 배치 크기를 점진적으로 늘려가면서 적절한 값을 찾는 것이 좋습니다.
- **모델 크기 확인:** 사용하려는 **모델의 크기**가 너무 크지는 않은지 확인하세요. 모델의 레이어(층)가 많거나 파라미터 수가 많으면 메모리 사용량도 늘어납니다. 필요하다면 모델의 구조를 경량화하는 방법을 고려해 볼 수 있습니다.
- **리소스 모니터링:** `nvidia-smi` 같은 도구를 사용해 학습 중 **GPU 사용량**을 주기적으로 모니터링하세요. GPU 메모리가 꽉 차기 전에 미리 문제를 감지할 수 있습니다.

△ 작성한 프롬프트에 대한 'Gemini 2.5-Flash' 응답 결과

4

내 프롬프트, 점검해볼까?

- 자가진단 체크리스트
- 이걸 피하자! 잘못된 프롬프트 습관
- 자주 하는 질문 (FAQ)



| 자가진단 체크리스트

내가 쓴 프롬프트, 과연 잘 만든 걸까? 체크리스트로 확인하고, 더 나은 프롬프트 습관을 만들어 보세요!

번호	프롬프트 원칙	자가 진단 문항	체크
1	명확하고 구체적으로 묻기	목적/대상/범위를 한 문장에 담고, 글자/항목 같은 수치 제약도 함께 적었나요?	<input type="checkbox"/>
2	필요한 맥락 제공	내 수준/현재 목표/시간/수학 등 제약을 먼저 알려주었나요?	<input type="checkbox"/>
3	원하는 출력 형식 지시	표/불릿/코드 등 형식과 열/항목 수 규격까지 지정했나요?	<input type="checkbox"/>
4	반복과 개선	1차 응답 뒤, 개선 기준(정확성, 간결성 등)과 구체적인 수정 요청을 덧붙였나요?	<input type="checkbox"/>
5	역할 부여	답변자의 역할(조교/코치 등)과 톤/난이도를 정해두었나요?	<input type="checkbox"/>
6	예시 제공	핵심 개념을 돕는 일상 예시나 반례를 꼭 요구했나요?	<input type="checkbox"/>
7	단계적 접근	정의-비유-절차-검증-요약 순서를 따라 달라고 요청했나요?	<input type="checkbox"/>

| 이건 피하자! 잘못된 프롬프트 습관

너무 모호함

- 나쁜 예시: “RNN 코드 짜줘.”
- 좋은 예시: “감정 분석용 양방향 LSTM 코드를 PyTorch로 작성해줘. 입력은 토큰화된 문장 시퀀스이고, 출력은 긍정/부정 이진 분류야.”

맥락 없음

- 나쁜 예시: “이 에러 왜 나와? [에러 메시지 붙여넣기]”
- 좋은 예시: “다음 코드를 실행하면 [에러 메시지]가 나와. [코드 붙여넣기] 원인과 해결책을 알려줘.”

한번에 너무 많은 요구

- 나쁜 예시: “Transformer 완전히 설명하고, 코드도 짜주고, 최적화 방법도 알려줘.”
- 좋은 예시: “먼저 Transformer의 Encoder 부분만 설명해줘. 이해되면 그 다음에 Decoder로 넘어갈게.”

수동적 태도

- 나쁜 예시: “코드 좀 고쳐줘”
- 좋은 예시: “이 코드에서 메모리 누수가 발생하는 것 같아. 어느 부분을 확인해봐야 할까? 진단방법도 알려줘.”

I 자주 하는 질문 (FAQ)

Q1. AI가 틀린 답변을 주면 어떻게 하나요?

A. 다음과 같이 대응해 보세요 :

“방금 답변에서 [구체적 부분]이 이상해 보여.

[이유/근거]를 고려하면 다른 결과가 나와야 하는 것 아닐까?

혹시 내가 잘못 이해한 건지, 아니면 답변에 오류가 있는지 확인해줘.

공식 문서나 논문을 참고해서 정확한 답변을 다시 해줄 수 있을까?”

| 자주 하는 질문 (FAQ)

Q2. 너무 긴 답변을 받았는데, 요약하고 싶어요.

A. 다음과 같이 대응해 보세요 :

“방금 설명이 매우 상세해서 좋았어. 하지만 핵심만 빠르게 파악하고 싶어.

다음 형식으로 요약해줘.

- 3줄 요약 : [가장 중요한 포인트 3개]
- 액션 아이템 : [내가 바로 할 수 있는 것]
- 더 공부할 키워드 : [3개] ”

I 자주 하는 질문 (FAQ)

Q3. 코드가 작동하지 않아요.

A. 다음과 같이 대응해 보세요 :

“제공받은 코드를 실행했는데 다음 에러가 발생했어 : [에러 메시지]

디버깅을 도와줘. 먼저 문제를 진단하고, 수정된 코드와 함께 왜 이런 문제가 발생했는지 설명해줘.”

내 환경 :

- Python : [버전]
- PyTorch : [버전]
- 실행 코드 : [전체 코드]

| 마치면서

좋은 질문은 AI 시대의 가장 강력한 무기입니다.

AI와 함께 지식을 탐구하고 문제를 해결하는 힘, 바로 여러분의 프롬프트에서 시작돼요!

- ✓ 프롬프트 핵심 원칙은 항상 기억하세요.
- ✓ 목적에 맞는 고급 기법을 전략적으로 써보세요.
- ✓ 연습하고 실험할수록 프롬프트 실력은 점점 성장합니다.
- ✓ 나만의 프롬프트 라이브러리를 쌓아가며, AI를 진짜 든든한 파트너로 만들어보세요!

Happy Prompting!

| 참고자료

- 삼성 SDS 인사이트 리포트 : <https://www.samsungsds.com/kr/insights/prompt-engineering.html>
- 인사혁신처 '인공지능(AI) 활용 가이드' :
https://www.mpm.go.kr/mpm/comm/pblcateList/?boardId=bbs_0000000000000036&mode=view&cntId=920&category=&pageIdx
- Learn Prompting : https://learnprompting.org/?srsId=AfmBOoq5ohihjQuK14a8bEM2m248bU8aBuHdzfhled4v-vBAv_QlRV7v
- OpenAI Academy : <https://academy.openai.com/>
- Google Prompt Engineering for AI Guide : <https://cloud.google.com/discover/what-is-prompt-engineering?hl=ko>
- Google Prompting guide 101 : <https://services.google.com/fh/files/misc/gemini-for-google-workspace-prompting-guide-101.pdf>
- Kaggle Whitepaper Prompt Engineering : <https://www.kaggle.com/whitepaper-prompt-engineering>
- Youtube [티타임즈TV] 강수진 박사의 프롬프트 엔지니어링의 매직 : <https://youtube.com/playlist?list=PL7d4-rFjtYdK4-RGBJTXbgLI5a-H7z7g&si=MGgUFdeETahe2iN7>