

# Final Project Description

Fall, 2025

M2177.004300 - 001

딥러닝의 기초

## 1 Introduction

본 프로젝트의 목적은 수업에서 배운 딥러닝 개념과 기법에 대한 종합적인 이해도를 평가하고, 최근 연구 동향을 직접 조사 및 실습을 통해 경험하는 데 있습니다. 주제는 자유롭게 선택할 수 있으며, 제시된 가이드라인 중 하나 또는 복수의 접근을 조합하여 수행할 수 있습니다. 다만, 가이드라인에 포함되지 않은 주제를 진행하고자 하는 경우에는 사전에 조교와 상의하기를 권장합니다.

수업 초반에 강조했듯이, 본인의 Computing 자원에 맞는 프로젝트를 수행하는 것이 중요합니다. **모델이 크거나 많은 자원을 사용했다고 해서 추가 점수를 부여하지 않습니다.**

## 2 Guidelines

가이드라인은 세 가지 접근 방식과 그 설명으로 구성되어 있습니다. 첫째, 기존 딥러닝 예제를 구현하고 발전시키는 방법, 둘째, 딥러닝을 새로운 task에 적용하여 성능을 향상시키는 방법, 셋째, 수학적 접근을 통해 더 나은 알고리즘을 제안하는 방법입니다.

아래에 제시된 각 가이드라인은 프로젝트 진행의 방향성을 잡는 데 도움을 주므로 참고하기를 권장합니다.

## 2.1 Improving existing task through deep learning method

수업 후반부에 다룰 예정인 CNN, RNN 등을 바탕으로 주요 딥러닝 과제(Image Classification, Natural Language Processing, Speech Recognition, Generative Adversarial Network 등)를 조사하고 구현하여 기존 결과를 재현하는 것이 출발점입니다. 이후, 재현한 모델을 바탕으로 기존 결과보다 성능을 향상시킬 수 있는 방법을 탐구하는 것이 목표입니다.

인터넷에 공개된 자료와 오픈 소스 코드가 풍부하기 때문에 선행 조사와 구현은 비교적 쉽습니다. 그러나, 그만큼 새로운 방법을 제시하는 것은 더 어려울 수 있습니다. 실험에 사용할 dataset은 자유롭게 선택 가능합니다.

수행 과정은 다음과 같이 진행됩니다. 먼저, 관련 논문을 조사하여 수행하고자 하는 task와 알고리즘을 이해합니다. 이어서, 코드를 실행하여 결과를 재현한 뒤, 네트워크 구조나 트레이닝 기법을 다양하게 변형·실험하면서 최적의 성능을 찾아갑니다.

이 접근은 기존 task에 대해 얼마나 다양한 실험을 성실하게 수행했는지를 중점적으로 평가합니다. 특히, 독창적인 아이디어를 제시하거나 여러 딥러닝 모델 혹은 서로 연관된 다양한 task를 종합적으로 탐구하고 그 결과 유의미한 결론을 도출한다면 **높은 평가를 받을 수 있습니다.**

## 2.2 Applying deep learning to the new tasks

수업에서 다룬 내용을 확장하여, 지금까지 딥러닝이 적용되지 않았던 새로운 task에 대해 딥러닝을 어떻게 활용할 수 있을지 탐구하고, 실제 구현을 통해 기존 방법 대비 유의미한 성능 향상이나 기타 장점을 확인하는 것을 목표로 합니다.

의미 있는 성능만 보장된다면 간단한 네트워크 구조를 사용해도 무방하기 때문에 구현 난이도는 높지 않습니다. 그러나 기존에 딥러닝이 활용되지 않았던 task의 경우,

- 학습 가능한 충분한 규모의 dataset 확보가 어렵고,
- novelty에 대한 요구 수준이 높으며,
- 기존 방법 대비 딥러닝의 성능이 반드시 우수하지 않을 수 있습니다.<sup>1</sup>

---

<sup>1</sup>보고서에는 해당 task가 새로운 문제임을 입증할 근거를 반드시 제시해야 합니다. Novelty가 낮다고 판단될 경우, 접근 1과 동일한 기준으로 평가됩니다.

평가에서는 다음과 같은 요소가 중요합니다:

- 제시한 task가 딥러닝 적용에 합리적으로 적합한지,
- novelty 주장에 대해 설득력 있는 근거를 제시했는지.

특히, 기존 방법에 비해 유의미한 성능 향상을 보였다면 **높은 평가를 받을 수 있습니다.**

## 2.3 Improving deep learning networks or mathematical algorithm

수업시간에 다룬 다양한 수학적 알고리즘 및 기법(예: ADAM, DROPOUT)을 바탕으로, 현업에서 활용되는 딥러닝 알고리즘 발전의 흐름을 이해하고 이를 수학적으로 확장·개선할 수 있는 방향을 탐구하는 것이 목표입니다. 이후, 제안한 알고리즘을 실제 코드로 구현하고 성능을 검증합니다.

이 접근은 수학적 이해와 프로그래밍 역량(PyTorch 등 딥러닝 라이브러리와 Python 활용 능력)이 모두 요구되므로 가장 어려운 방법일 수 있습니다. 그러나 실험 비용이 비교적 적게 들며, 성공적으로 수행할 경우 독창성과 학술적 기여도 측면에서 매우 높은 평가를 받을 수 있습니다. 데이터셋은 MNIST와 같은 간단한 것을 권장하나, 필요에 따라 다른 데이터셋을 활용해도 무방합니다.

수행 과정은 기존 수학적 알고리즘에서 출발하여, 이를 적절히 변형·확장해 새로운 수식 및 알고리즘을 제안하는 것입니다. 제시한 내용을 코딩으로 구현하고, 성능이 실제로 향상되는지를 검증해야 합니다. 또한, 제안한 알고리즘이 왜 성능을 개선할 수 있는지에 대해 수학적 또는 직관적인 설명을 제시하거나, 간단한 수준의 ablation study를 수행한다면 **높은 평가를 받을 수 있습니다.**

## 2.4 Collecting and labeling the new dataset

앞서 소개한 세 가지 주요 분류와는 달리, 데이터셋 구축은 독립적인 범주로 분류하기는 어렵지만 매우 중요한 과정이므로 별도로 제시합니다. 이 접근의 목표는 특정 task에 적합한 데이터를 직접 수집·처리해 보면서, 딥러닝에서 데이터가 갖추어야 할 조건을 경험적으로 이해하는 것입니다. 데이터셋 구축에는 많은 시간과 노력이 소요되므로, 이를 성공적으로 수행할 경우 높은 평가를 받을 수 있습니다.

수행 과정에서는 딥러닝 학습이 가능할 정도의 충분한 데이터를 확보하고, 각 데이터에 대해 목표 task에 맞는 정확한 annotation을 기록해야 합니다. 예를 들어, object detection 문제에서는 이미지 속 물체의 위치를 모두 확인하고 명시해야 합니다. 실제 딥러닝이 가능할 정도의 데이터 확보는 소수의 샘플로는

불가능합니다. 예컨대 대표적인 간단한 데이터셋인 MNIST조차 60,000장의 이미지와 그에 대한 라벨링을 필요로 합니다.

또한 데이터의 종류에 따라 전처리 과정이 필수적인 경우가 있으며, 이러한 방법을 스스로 탐색·적용해야 하는 어려움이 있습니다. 예를 들어, 라벨 분포가 불균형한 데이터셋에서는 모델 학습이 원활하지 않을 수 있습니다.

그럼에도 불구하고, 데이터셋을 구축하는 과정은 다양한 딥러닝 알고리즘 발전의 중요한 토대가 되었던 만큼 매우 가치 있는 작업이며, 그 자체로 높게 평가받을 수 있습니다. 특히 프로젝트 수행 과정에서 **데이터셋을 직접 수집·정제하는 등 많은 노력과 정성을 기울인 경우에는 Novelty 항목에서 가산점**을 부여할 예정입니다.<sup>2</sup>

### 3 Report

보고서는 **\*\*CVPR (Computer Vision and Pattern Recognition conference)\*\***에서 사용되는 양식을 따릅니다. 해당 Author Kit은 ETL의 프로젝트양식 폴더에 업로드되어 있습니다. Word 또는  $\text{\LaTeX}$ 를 사용하여 작성한 뒤, 반드시 PDF 형식으로 제출해야 합니다. 보고서 분량은 최대 10페이지를 넘지 않도록 합니다.

보고서에는 다음 내용이 반드시 포함되어야 합니다:

- 아이디어 및 알고리즘 설명
- 코드 구현 과정
- 사용한 데이터에 대한 설명
- 트레이닝 과정 및 결과

그 외에도 추가적으로 설명하고 싶은 내용이 있다면 자유롭게 포함할 수 있습니다. 프로젝트 수행 과정에서 다양한 논문이나 공개된 연구 자료를 참고할 수 있으며, 실제로 이를 적극적으로 활용하기를 권장합니다.

단, **참고한 논문이나 활용한 오픈 소스의 출처는 반드시 보고서에 명시해야 합니다. 출처를 기재하지 않을 경우 0점 처리될 수 있습니다.**

---

<sup>2</sup>단, 가산점을 받기 위해서는 실제 학습이 가능할 정도의 규모와 품질을 갖춘 데이터셋을 구축해야 합니다.

## 4 Grading

프로젝트 점수: 최대 100점 (Novelty: 20 / Code: 40 / Report: 20 / Presentation: 20)

### 1. Novelty: 20 points

- 아이디어, 연구 주제에 대한 종합적인 평가를 반영함 (20 points)

### 2. Code: 40 points

- 실행 여부 (30 points)
- 주석 및 설명: 코드에 대한 간략한 설명 (10 points)

### 3. Report: 20 points

- 논리 (10 points)
- 양식 (10 points)
  - 양식 준수: 보고서의 양식 준수 여부 (5 points)
  - 실험 결과 표 혹은 그래프 (5 points)

### 4. Presentation: 20 points

- 발표 자료 구성 및 내용 (10 points)
- 발표 능력: 간결하고 명확하게 (10 points)

프로젝트에 부여되는 점수는 위의 기준을 따라 부여될 예정이며, 특히 프로젝트에 많은 노력을 들이는 경우 이를 감안하여 평가를 할 예정입니다. Guidelines에서 추가적으로 언급한 것과 같이, 프로젝트를 수행하기 위해 데이터셋을 만드는 등 많은 시간과 노력이 들어가는 작업을 하면 이에 대한 노력을 감안하여 Novelty 측면에서 높은 평가를 줄 예정입니다.

## 5 Helpful tips

- Feel freely to use Google, ARXIV, and GitHub. But remember to contribute the the works.

- We will estimate all what you did on the project, even if your result does not reach on goal described above. So do your best!
- Please write down all the procedures as much as you can in report. It will help us to evaluate the efforts you put to the works.

## 6 Submission form and date

추후 별도 공지 예정.