# 단백질 시퀀스 분류를 위한 딥러닝 방법 비교 연구

A Comparative Study on the Deep Learning Methods of Protein Sequence Classification

이화여자대학교

바이오인포매틱스 연계전공

이름: 조서영

학번: 1976371

연락처: stellamore99@ewhain.net

2022년 12월

# 단백질 시퀀스 분류를 위한 딥러닝 방법 비교 연구

조서영
이화여자대학교 바이오인포매틱스 연계전공
stellamore99@ewhain.net

# A Comparative Study on the Deep Learning Methods of Protein Sequence Classification

Seoyoung Jo
Ewha Womans University
Bioinformatics

**요 약**

본 논문에서는 아미노산 시퀀스를 기반으로 해당 시퀀스가 어떤 단백질의 시퀀스인지 분류하기 위해 딥러닝 방법들을 이용한다. 딥러닝 모델 학습을 위해 UniProt의 단백질 임베딩을 사용하였으며, 비교에 사용된 딥러닝 모델은 RNN, LSTM, Transformer이다. 학습 후 테스트 시퀀스로 분류 정확도를 모델별로 비교해 본 결과 RNN 기반 모델이 좋은 성능을 보임을 확인할 수 있었다.

## 1. 서 론

인간 게놈 시퀀싱의 시대가 열리면서 방대한 양의 유전자, 단백질 시퀀스가 모이고 있다. 이 시퀀스들을 잘 해석하고 이용하는 것이 생명공학자들에게 있어 중요한 과업이 되었고, 그 중에서도 생명을 구성하며 생명 유지에 중요한 역할을 하는 단백질에 대한 연구가 활발히 진행되고 있다. 단백질을 이루는 기본 단위는 아미노산이므로 단백질은 아미노산의 서열로 나타낼 수 있다. 정보가 적은 새 단백질의 아미노산 서열을 알고 있을 때 해당 서열을 이용해 단백질이 어느 상위 그룹에 속하는지, 단백질의 역할이 무엇일지를 알아내는 것은 단백질 연구에 있어 중요한 부분이라고 할 수 있다. 아미노산 서열만으로 어떤 그룹에 속하는 단백질인지 추측할 수 있다면 연구의 범위를 좁힐 수 있으므로 시간과 비용을 절약할 수 있기 때문이다. 따라서 많은 연구들에서 아미노산 서열을 이용한 단백질 분류를 시도해왔다. SVM(Support Vector Machine), Decision Tree와 같은 비교적 전통적인 머신 러닝 방법에서부터 최근에 이르러서는 Neural Network를 이용한 방법들도 많이 쓰이고 있다. Neural Network를 깊게 쌓은 딥러닝 방법들을 이용한 단백질 분류도 활발히 진행되고 있으며 좋은 성과를 내고 있다.

본 연구는 이러한 딥러닝 방법들을 사용하여 단백질 시퀀스 분류를 시도하고 여러 딥러닝 모델을 비교해 보고자 한다. 단백질의 아미노산 서열은 순서가 중요한 Time series 정보라는 점에서 문자열 정보와 비슷하다. 해당 특징에서 착안하여 문자열 정보 처리에 주로 쓰이는 딥러닝 모델인 RNN(Recurrent Neural Network), LSTM(Long Short Term Memory), Transformer를 사용한다. 단백질의 아미노산 서열 임베딩 정보를 이용하여 지도 학습 분류 문제로 세 개의 모델을 학습시킨 후, 테스트 데이터를 모델에 입력하여 어떤 모델의 분류 정확도가 가장 정확한지 비교하고자 한다.

## 2. 관련 연구

국내외에서 아미노산 서열을 이용한 단백질 분류 연구는 활발히 있어 왔다. SVM을 이용해 단백질의 1차 아미노산 서열로부터 기능적 분류를 하는 연구 [1], Decision Tree를 이용해 단백질의 구조를 분류하는 연구 [2], 단백질 분류에 Neural network를 사용한 연구 [3] 등이 있다. 뿐만 아니라 비교적 최근에 들어서 딥러닝의 부흥과 함께 딥러닝을 이용한 연구도 많은데, 딥러닝을 이용해 단백질 구조를 분류하는 연구 [4]가 그 예이다. 문자열 혹은 텍스트 분류 방식에서 영감을 받아 단백질 분류를 진행한 연구 [5]도 있었던 것으로 보아 전통적으로 컴퓨터 공학 분야에서 문자열 처리에 사용되던 방법들이 생명과학 분야의 시퀀스 처리에 접목되어 연구되는 경우도 있음을 확인할 수 있었다.

이와 같은 선행 연구들을 바탕으로 본 논문에서는 문자열 처리에 주로 사용되는 딥러닝 모델들을 단백질 시퀀스 분류에 적용하고 비교한다. 이를 통해 딥러닝 모델들이 아미노산 서열을 처리하는 데 있어 얼마나 효과적인지를 확인하고, 모델들 중 어떤 모델이 가장 좋은 성능을 보이는지도 확인해보고자 한다.

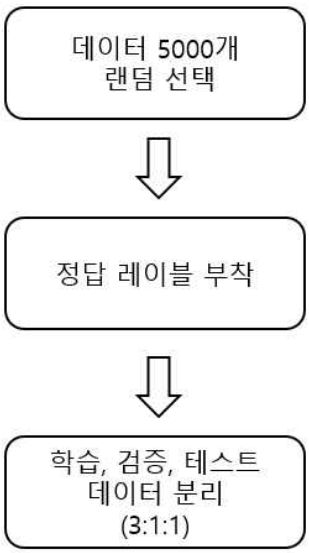
## 3. 데이터

### 3.1 데이터셋

본 연구에서는 공공 단백질 데이터베이스인 UniProt에서 제공하는 데이터를 사용한다. 그 중에서도 단백질 별로 아미노산 서열이 임베딩 되어 제공되는 'Protein Embeddings'를 사용한다. 임베딩이란 문자열, 아미노산 서열과 같은 정보를 정해진 크기의 벡터 안에 숫자로 나타내는 것을 말한다. 임베딩을 거친 벡터들은 숫자로 나타나므로 벡터 간 연산이 자유롭다. 또한 벡터 공간 내에서 어떤 벡터들이 가까운지를 확인해서 벡터 간의 유사도를 쉽게 확인할 수 있다. 딥러닝 모델에 인풋으로 넣기 전에 아미노산 서열을 숫자로 나타내는 과정이 필요하므로 임베딩 데이터를 사용했다.

사용할 데이터는 Caenorhabditis elegans, Homo sapiens, Mus musculus 각 종의 여러 단백질의 서열을 임베딩한 데이터이다. 모든 벡터의 크기는 1024이다. 해당 데이터를 이용해 서열을 학습시키고 새로운 단백질 벡터가 입력으로 들어왔을 때 어떤 종의 단백질인지 분류한다.

### 3.2 데이터 전처리




[그림 1]

딥러닝 모델에 넣기 위해서는 데이터 전처리가 필요하다. 그림 1에서 볼 수 있듯 단계는 크게 세 단계로 나뉜다. 우선 각 종의 데이터에서 임베딩 벡터 5000개씩을 각각 랜덤 추출하고, 각 데이터에 정답 레이블을 부착한다. 해결하고자 하는 문제가 단백질-종 분류 지도학습 문제이므로 각 데이터가 어떤 종에 속하는지 정답을 모델에 알려줘야 하기 때문에 정답 레이블을 부착해야 한다. 그 후 5000개의 데이터를 학습:검증:테스트=3:1:1의 비율로 나눈다. 검증 데이터는 딥러닝 모델의 과적합 문제를 피하기 위해, 테스트 데이터는 모델 학습의 적절성과 정확도를 평가하기 위해 필요하다.

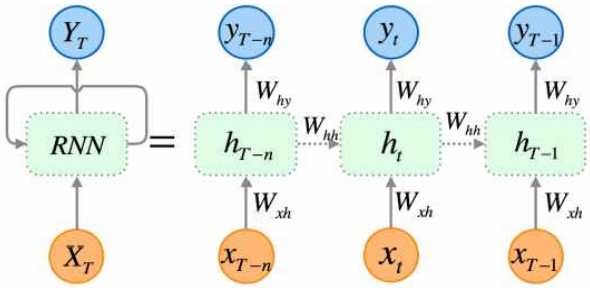
## 4. 딥러닝 모델

본 연구에서는 크게 RNN, LSTM, Transformer 세 가지의 모델을 학습시키고 결과를 비교한다. 각 모델에 사용되는 학습, 검증, 테스트 데이터는 모두 같다.

### 4.1 RNN (Recurrent Neural Network)

RNN은 주로 일련의 데이터에서 패턴을 감지하는 데 사용되는 신경망 아키텍처의 한 유형이다[6]. 각각의 유닛이 자신의 출력 값을 가중치를 반영하여 다음 유닛에게 입력 값으로 전달함으로써 순차적으로 처리되어야 하는 시계열 데이터를 처리한다[7].

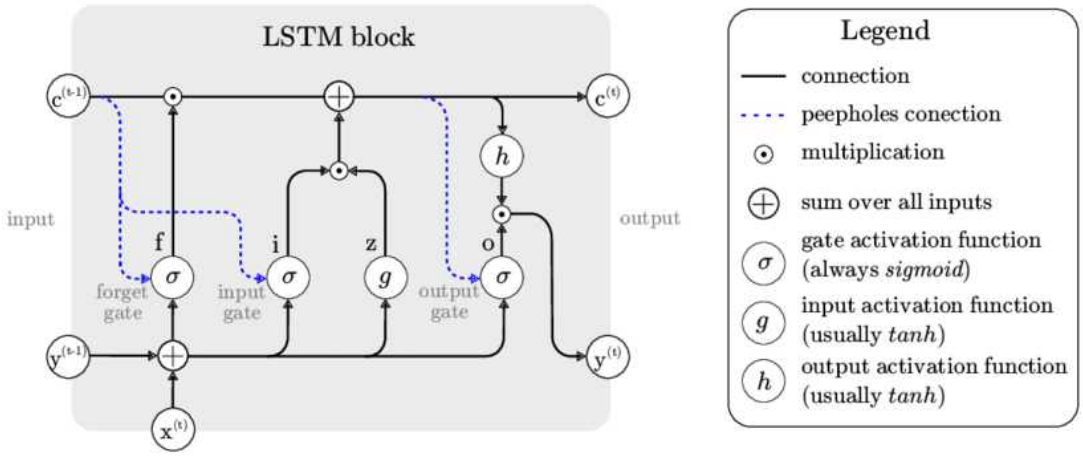
아래의 [그림 2]는 RNN의 기본적인 구조이다.




[그림 2] [8]

### 4.2 LSTM (Long Short Term Memory)

LSTM은 cell state와 게이트를 활용하여 기존의 전통적인 RNN 네트워크의 문제점인 '기울기 소실' 문제를 해결하기 위해 고안되었다. 기울기 소실 문제란 데이터와 데이터 사이의 시간 간격이 멀 경우 학습능력이 크게 저하되는 것을 의미한다. LSTM의 기본구조는 [그림 3]처럼 입력 게이트(input gate) i, 망각 게이트(forget gate) f, 제어 게이트(control gate) c 및 출력게이트(output gate) o의 4개의 게이트를 갖는 LSTM 셀의 구조를 나타낸다[9]. Forget gate, input gate, output gate와 cell을 이용하여 어떤 값을 얼마만큼 반영할지 조절하고, cell이 memory 역할을 해서 long-term memory를 저장한다. 기울기 소실 문제를 완화하였으므로 보다 긴 길이의 입력값에 대해서도 비교적 좋은 학습 결과를 낼 수 있다.




[그림 3] [10]

### 4.3 Transformer

Transformer는 RNN기반 모델인 기본 RNN과 LSTM의 문제점인 순차성을 제거한 모델이다. 기존 RNN 기반 모델들은 데이터를 순차적으로 처리하므로 병렬화를 할 수 없어 계산 속도가 느리다는 단점이 있었다. 이를 해결하기 위해 Transformer는 타겟 간의 연관성을 계산하는 메커니즘인 Attention 메커니즘만을 이용한다. 이를 통해 병렬성을 증가시키고 비약적인 성능 향상을 이룬 모델이다. 구조는 [그림 4]와 같다.

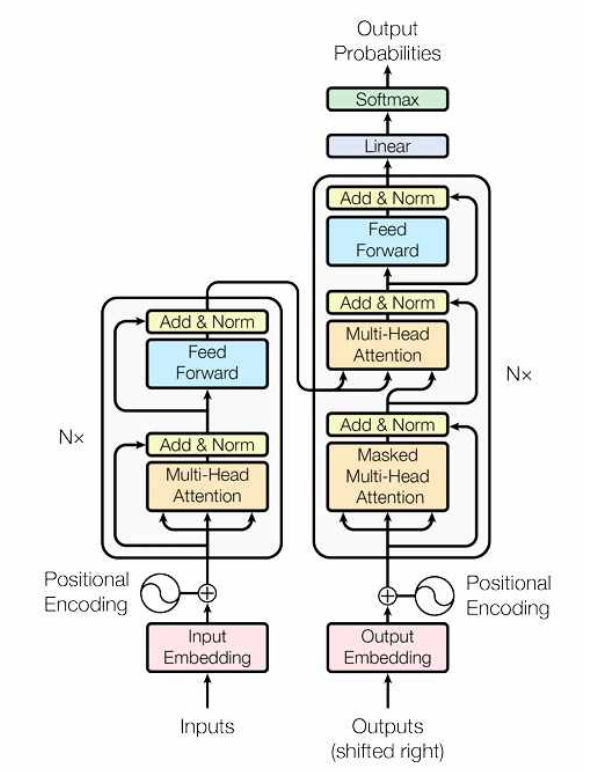



Figure 1: The Transformer - model architecture.

[그림 4] [11]

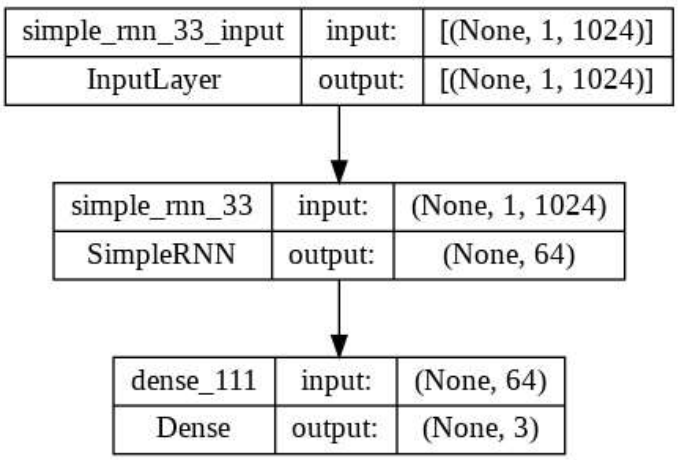
## 5. 실험 및 결과 분석

### 5-1. 실험 환경

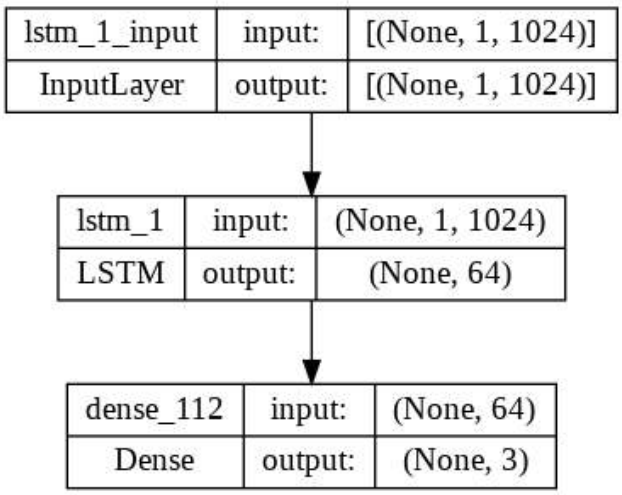
하드웨어 실험 환경은 Google Colab GPU 하드웨어 가속 환경이며, 세 모델에 모두 같은 데이터가 사용되었다. 세 모델의 은닉층 뉴런 수, 옵티마이저, 손실 함수, 에포크 수, 배치 사이즈 또한 모두 같았다. 세 모델에 모두 학습 조기 종료가 적용되어 검증 손실값이 10번의 에포크 동안 감소하지 않으면 학습을 조기 종료했다.

### 5-2. 모델 구조 비교

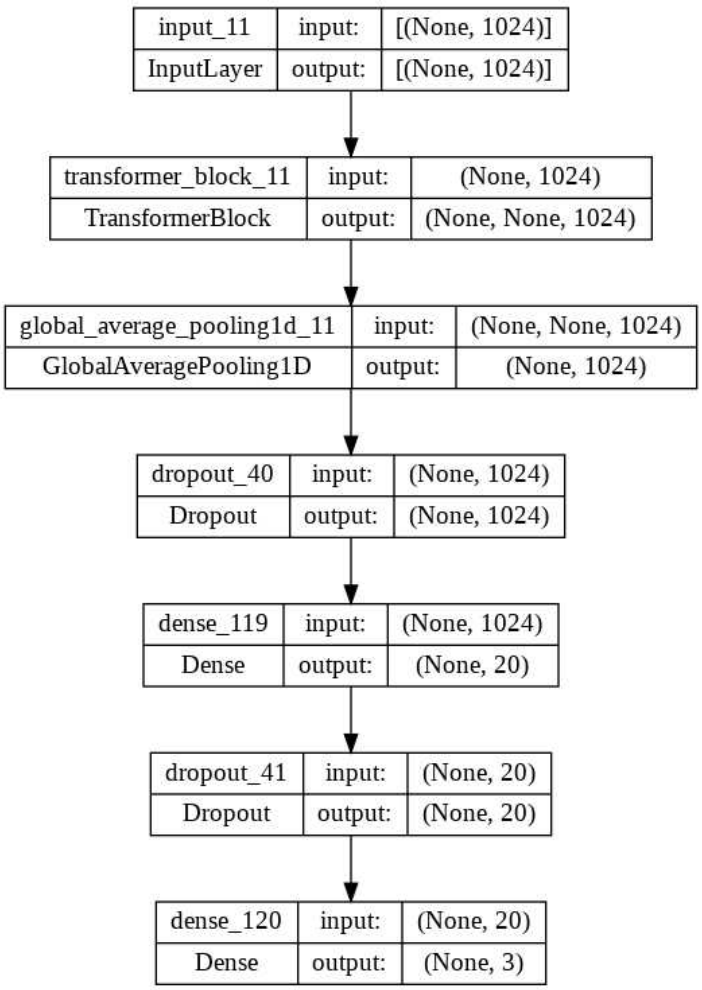
실험에 이용된 RNN, LSTM, Transformer 모델 각각의 구조는 아래 [그림 5, 6, 7]과 같다.




[그림 5] RNN shape



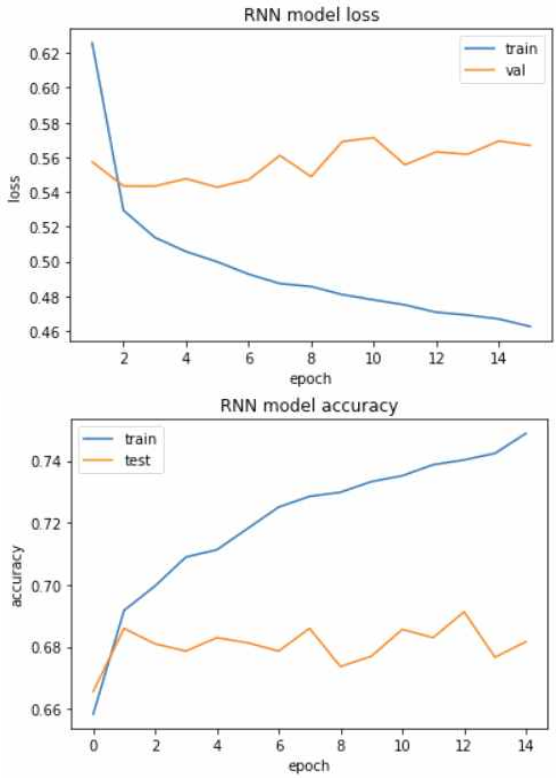

[그림 6] LSTM shape



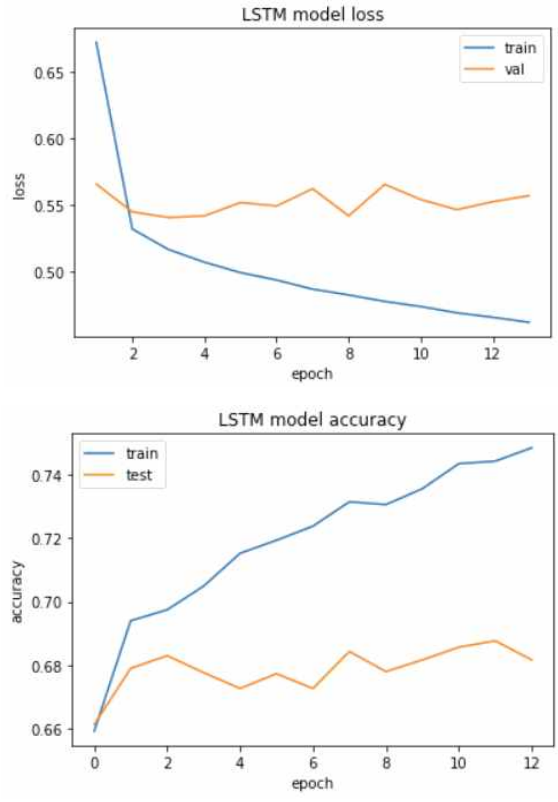

[그림 7] Transformer shape

### 5-3. 학습 결과 및 분석

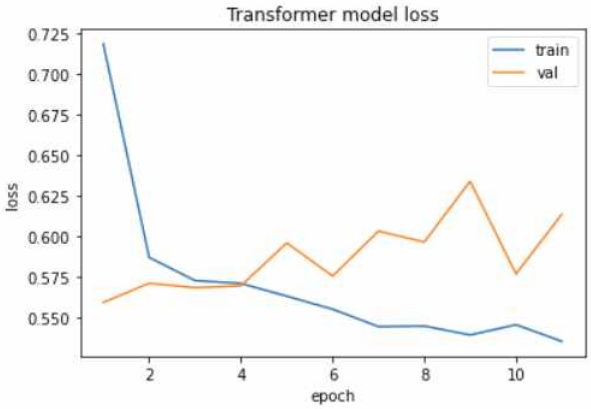
학습 결과 각 모델 별 손실과 정확도 그래프는 [그림 8, 9, 10]과 같다.

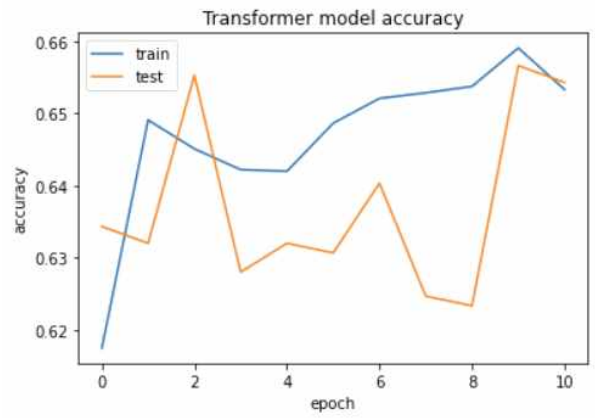



[그림 8] RNN 손실 및 정확도 그래프




[그림 9] LSTM 손실 및 정확도 그래프




[그림 10] Transformer 손실 및 정확도 그래프

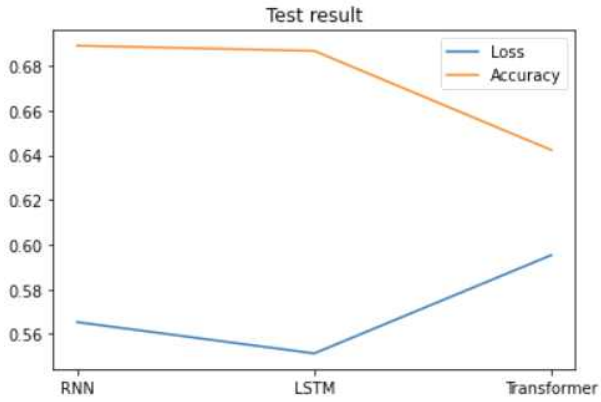
각 모델별 손실과 정확도 그래프를 보면 Transformer에 비해 RNN 기반의 나머지 두 모델의 손실 그래프는 꾸준히 줄어들고 정확도 그래프는 꾸준히 상승하는 경향을 보였다. 따라서 RNN기반 두 모델에서 더 안정적으로 학습이 진행됐음을 알 수 있었다. 또한 RNN과 LSTM은 세부 구조는 다르지만 크게 보았을 때 순차적으로 데이터를 처리한다는 점에서 공통점을 가지고, 이로 인해 학습의 추이도 비슷하다는 것을 알 수 있었다. Transformer의 경우 특히 학습이 불안정했으므로 단백질 분류 문제에 있어 나머지 두 모델에 비해 적합하지 않음을 판단할 수 있었다.

### 5-4. 테스트 결과 및 분석

학습이 완료된 세 모델에 테스트 데이터를 입력하여 분류 정확도와 손실값을 도출하고 비교해보았다. 결과는 [표 1], [그림 11]과 같다.

| Test result | 손실 | 정확도 |
|---|---|---|
| RNN | 0.5653 | 0.6890 |
| LSTM | 0.5513 | 0.6867 |
| Transformer | 0.5953 | 0.6423 |

[표 1] 테스트 결과




[그림 11] 테스트 결과

결과를 보면 학습이 불안정했던 Transformer에서 loss값이 크고 accuracy가 낮았음을 확인할 수 있다. 이는 나머지 두 모델에 비해 Transformer의 단백질 분류 성능이 떨어진다는 것을 의미한다. RNN, LSTM의 경우 70% 정도의 비교적 높은 정확도로 분류를 해내고 있음을 확인할 수 있었다.

## 7. 결론 및 향후 연구

이 논문에서는 벡터로 임베딩된 세 종의 단백질 시퀀스를 이용하여 RNN, LSTM, Transformer의 세 가지 모델을 학습시켰다. 학습된 모델을 바탕으로 테스트를 수행하여 학습된 모델들이 새로운 데이터가 들어왔을 때에도 해당 단백질 시퀀스가 어느 종에 속하는 단백질인지를 잘 분류하는지 확인해보았다. 실험 결과 세 모델 모두 60% 이상의 정확도를 보였고, 그 중에서도 RNN과 LSTM은 Transformer에 비해 높은 학습 안정성과 테스트 정확도를 보였다. 따라서 RNN 기반 모델이 해당 분류 문제에 있어 더 적합한 모델임을 알 수 있었다.

본 연구는 얕은 layer를 사용한 모델을 이용해 학습했으며, 비교적 적은 수의 데이터로 모델을 학습했다는 한계를 가진다. 따라서 더 높은 컴퓨팅 파워와 자원을 가용해 방대한 양의 데이터, 더 큰 모델을 학습한다면 더 좋은 결과를 낼 수 있을 것으로 기대된다. 그리고 LSTM과 RNN 모델이 더 좋은 성능을 보였는데, 일반적으로 RNN 기반 모델은 Attention 메커니즘을 함께 사용했을 때 더 좋은 성능을 보인다는 점을 감안하면 Attention 메커니즘을 함께 사용한 모델의 분류 정확도가 더 높을 것이라는 추측도 가능하다. 향후 더 큰 RNN 기반 모델에 Attention 메커니즘을 결합한 모델을 이용해 단백질 분류 연구를 진행한다면 더 좋은 성능을 보일 수 있을 것이라고 기대된다.

## 참고 문헌


[1] Cai, C. Z., et al. “SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence.“ Nucleic acids research 31.13 (2003): 3692-3697.

[2] Çamoğlu, Orhan, et al. “Decision tree based information integration for automated protein classification.“ Journal of Bioinformatics and Computational Biology 3.03 (2005): 717-742.

[3] Weinert, Wagner Rodrigo, and Heitor Silvério Lopes. “Neural networks for protein classification.“ Applied Bioinformatics 3.1 (2004): 41-48.

[4] Nanni, Loris, et al. “iProStruct2D: Identifying protein structural classes by deep learning via 2D representations.“ Expert Systems with Applications 142 (2020): 113019.

[5] Cheng, Betty Yee Man, Jaime G. Carbonell, and Judith Klein-Seetharaman. “Protein classification based on text document classification techniques.“ Proteins: Structure, Function, and Bioinformatics 58.4 (2005): 955-970.

[6] Schmidt, Robin M. (2019). “Recurrent Neural Networks (RNNs): A gentle Introduction and Overview”, arXiv:1912.05911 [cs.LG]

[7] K. Rao, H. Sak & R. Prabhavalkar. (2017, December). Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 193-199). IEEE. DOI : 10.1109/ASRU.2017.8268935

[8] Gawde, Rishikesh. (2021). Image Caption Generation Methodologies.

[9] 장진수, 이민준, and 이태로. “RNN 을 이용한 제 2 형 당뇨병 예측모델 개발.“ 디지털융복합연구 17.8 (2019): 249-255.

[10] Van Houdt, Greg & Mosquera, Carlos & Nápoles, Gonzalo. (2020). A Review on the Long Short-Term Memory Model. Artificial Intelligence Review. 53. 10.1007/s10462-020-09838-1.

[11] Vaswani, Ashish, et al. “Attention is all you need.“ Advances in neural information processing systems 30 (2017).