

박사학위 논문

딥러닝 기반 텍스트 임베딩을 활용한
직무 추천 모델 연구

- 이력서 데이터를 중심으로 -

A Study on Job Recommendation
Model Using Deep Learning-based
Text Embedding

- Focusing on the Resume Data -

2023년 6월

승실대학교 대학원

프로젝트경영학과

전 정 우

박사학위 논문

딥러닝 기반 텍스트 임베딩을 활용한
직무 추천 모델 연구

- 이력서 데이터를 중심으로 -

A Study on Job Recommendation
Model Using Deep Learning-based
Text Embedding

- Focusing on the Resume Data -

2023년 6월

승실대학교 대학원

프로젝트경영학과

전 정 우

박사학위 논문

딥러닝 기반 텍스트 임베딩을 활용한
직무 추천 모델 연구

지도교수 안 태 호

이 논문을 박사학위 논문으로 제출함


2023년 6월

숭실대학교 대학원

프로젝트경영학과


전 정 우


전 정 우 의 박 사 학 위 논 문 을 인 준 함

심 사 위 원 장 김 광 용 

심 사 위 원 박 종 우 

심 사 위 원 신 호 철 

심 사 위 원 배 성 민 

심 사 위 원 산 태 호 

2023년 6월

승실대학교 대학원

목 차

국문초록	ix
영문초록	xi
제 1 장 서론	1
1.1 연구의 배경	1
1.2 연구의 목적 및 방법	3
1.3 연구의 구성	5
제 2 장 이론적 배경	6
2.1 추천 시스템	6
2.1.1 직무 추천	11
2.1.2 채용공고 추천	12
2.1.3 인재 추천	13
2.2 딥러닝	15
2.2.1 심층 신경망(DNN)	19
2.2.2 합성곱 신경망(CNN)	21
2.2.3 순환 신경망(RNN)	22
2.2.4 하이퍼파라미터(Hyperparameter)	24
2.3 자연어 처리(NLP)	29
2.4 텍스트 임베딩(Text Embedding)	31
2.4.1 워드투벡터(Word2Vec)	32
2.4.2 도큐먼트투벡터(Doc2Vec)	32
2.5 1차원 합성곱 신경망(1D-CNN)	33

2.6 장단기 메모리(LSTM)	34
2.7 트랜스포터(Transformer)	35
2.8 어텐션 매커니즘(Attention Mechanism)	37
2.9 성능 평가 방법	38
2.9.1 예측 모델 성능평가	39
2.9.2 분류 모델 성능평가	40
제 3 장 연구 모형	42
3.1 직무 추천 모델	42
3.2 데이터 정의와 수집	43
3.3 데이터 분석	47
3.3.1 데이터 특성 및 분포 분석	47
3.3.2 상관관계 분석	50
3.4 데이터 전처리	52
3.5 학습 데이터셋 구축	58
3.6 피처 엔지니어링(Feature Engineering)	58
3.7 하이퍼파라미터(Hyperparameter)	60
3.8 딥러닝 모델 학습	61
3.8.1 합성곱 신경망(CNN)	61
3.8.2 장단기 메모리(LSTM)	63
3.8.3 트랜스포머(Transformer)	64
3.9 국가직무능력표준(NCS) 직무 분류체계 매핑	65
3.10 파인튜닝(Fine-Tuning)	66
3.11 모델의 성능측정	68
3.11.1 학습 성능의 측정	68

3.11.2 예측 성능의 측정	68
제 4 장 실험 및 결과분석	69
4.1 실험 환경 및 도구	69
4.2 파인튜닝 선정 실험	69
4.2.1 데이터 관점	69
4.2.2 분류기 관점	71
4.2.3 Layers 관점	71
4.3 하이퍼파라미터 선정 실험	72
4.3.1 가중치 초기화 기법 선정	72
4.3.2 임베딩 차원 선정	73
4.3.3 학습 배치 크기 선정	74
4.3.4 학습률	74
4.3.5 최적화 기법	75
4.4 딥러닝 모델 실험	76
4.4.1 합성곱 신경망(CNN)	77
4.4.2 장단기 메모리(LSTM)	80
4.4.3 트랜스포머(Transformer)	82
4.5 실험 결과 분석	84
4.5.1 하이퍼파라미터의 영향 분석	84
4.5.2 딥러닝 모델의 비교 분석	84
4.5.3 추론 모델의 비교 분석	87
제 5 장 결론	88
5.1 연구의 요약 및 결론	88

5.2 연구의 한계 및 향후 연구	90
참고문헌	91

표 목 차

[표 1-1] 직무 추천 모델 연구의 절차 및 방법	5
[표 2-1] 추천 시스템의 장단점	7
[표 2-2] 경력자-직무 행렬	10
[표 2-3] 활성화 함수의 특징과 수식	17
[표 2-4] 손실 함수의 종류와 수식	18
[표 2-5] 주요 최적화 기법	19
[표 2-6] 가중치 초기화 기법	26
[표 2-7] 하이퍼파라미터의 종류와 특징	28
[표 2-8] 통계적 기반 임베딩 기법	30
[표 2-9] 신경망 기반 임베딩 기법	30
[표 2-10] 모델링 목적과 변수 유형에 따른 평가 방법	39
[표 2-11] 예측 모델 성능평가 방법	39
[표 2-12] 분류 모델 성능평가 방법	41
[표 3-1] 프로세스 및 세부 내용	43
[표 3-2] 이력서 데이터 구성	45
[표 3-3] JOB_CODE 생성	46
[표 3-4] 데이터 특성분석	48
[표 3-5] 데이터 분포 분석 예시	50
[표 3-6] Cramer's V 계수	51
[표 3-7] 결측치 처리 항목	52
[표 3-8] 이상치 처리 프로세스	52
[표 3-9] 이상치 처리 내역	53
[표 3-10] 데이터 코드화 방법	54

[표 3-11] 데이터 전처리 내역	55
[표 3-12] 전처리 완료 데이터 예시	56
[표 3-13] 학습 데이터셋 데이터 개수	58
[표 3-14] 변수별 Feature Tuning	59
[표 3-15] 하이퍼파라미터 항목	61
[표 3-16] NCS 직무 분류체계 매핑 데이터 예시	66
[표 3-17] Fine-Tuning 종류	67
[표 3-18] Fine-Tuning 실험 방법	67
[표 4-1] 데이터 관점 Fine-Tuning	70
[표 4-2] Xavier_uniform, Xavier_normal	73
[표 4-3] 임베딩 차원	74
[표 4-4] 학습 배치 크기 선정	74
[표 4-5] 학습률	75
[표 4-6] RMSProp, Adam 정확도 비교	76
[표 4-7] 딥러닝 모델 실험의 주요 차이점	77
[표 4-8] CNN 실험 결과	77
[표 4-9] LSTM 실험 결과	80
[표 4-10] Transformer 실험 결과	82
[표 4-11] 하이퍼파라미터 선정 결과	84
[표 4-12] 1차 실험 모델의 성능평가	85
[표 4-13] 2차 실험 모델의 성능평가	85
[표 4-14] 1차 실험 추론 모델의 성능평가	87
[표 4-15] 2차 실험 추론 모델의 성능평가	87

그 립 목 차

[그림 2-1] 추천 시스템의 분류	7
[그림 2-2] 내용 기반 필터링 기법	8
[그림 2-3] 사용자 기반 협업 필터링 기법	9
[그림 2-4] 아이템 기반 협업 필터링 기법	9
[그림 2-5] 신경망의 학습 절차	16
[그림 2-6] 심층 신경망(DNN) 구조	20
[그림 2-7] LeNet-5의 구조	22
[그림 2-8] Multi-layer 퍼셉트론 구조	23
[그림 2-9] RNN을 활용한 감성인식 방법	23
[그림 2-10] 은닉층과 각 은닉층의 노드 수	25
[그림 2-11] 드롭아웃 기법 예시	25
[그림 2-12] 1D CNN Structure	34
[그림 2-13] LSTM 구조	34
[그림 2-14] 트랜스포머 모델 구조	36
[그림 3-1] 직무 추천 시스템 연구 모형	42
[그림 3-2] 이력서 데이터 정의	44
[그림 3-3] 자체 코드 매핑 테이블	54
[그림 3-4] 표준 코드화	55
[그림 3-5] CNN 모델 고등학교 Feature Dense Layers	62
[그림 3-6] LSTM 모델 대학교 Feature Dense Layers	63
[그림 3-7] Transformer 모델 언어 Feature Dense Layers	64
[그림 3-8] NCS 분류 체계 예시	65
[그림 3-9] 취업포털 직무 분류와 NCS 직무 분류체계의 매칭 테이블 ·	66

[그림 4-1] CNN 1차 실험 그래프	78
[그림 4-2] CNN 2차 실험 그래프	79
[그림 4-3] LSTM 1차 실험 그래프	80
[그림 4-4] LSTM 2차 실험 그래프	81
[그림 4-5] Transformer 1차 실험 그래프	83
[그림 4-6] Transformer 2차 실험 그래프	83

국문초록

딥러닝 기반 텍스트 임베딩을 활용한

직무 추천 모델 연구

- 이력서 데이터를 중심으로 -

전정우

프로젝트경영학과

승실대학교 대학원

현대 산업사회는 제4차 산업혁명을 통한 기술의 급속한 발전으로 산업 구조의 변화를 가속시켰다. 그로 인해 새로운 직종의 발생에 따라 직무의 세분화 및 직무의 복잡도가 높아졌으며, 이러한 사회로의 진출을 앞둔 청년 구직자에게 다양한 역량의 필요성을 증대시켰으며, 전공에 기반을 둔 직무와 산업이 요구하는 직무의 차이로 인한 반복적인 미스매칭은 구직 의욕 상실 및 취업 포기 증가의 원인이 되었다. 한국경제연구원(2021)의 조사결과에 따르면 청년 취업자 중 전공과 직무의 불일치는 52.3%로 OECD 29개 국가 중에서 2위이며, 기업들은 이로 인한 조기 퇴사 현상을 겪고 있는 것이 현실이다.

일자리 미스매치는 구직자의 전공과 기술이 일자리에서 요구하는 전문성과 일치하지 않는 현상이며, 직무 미스매치는 구직자의 직무 수행 관련 역량과 기술이 해당 직무에서 요구하는 것과 일치하지 않는 현상으로, 일자리 미스매치의 해결은 취업포털을 통한 일자리 정보 공유로 상

당 부분 가능하지만, 직무 미스매치의 해결은 구인 기업이 제공한 채용 정보와 직무 정보를 구직자 스스로 수집하여 직무 분석 및 직무 경험을 학습해야 하는 문제가 있다.

본 연구의 목적은 구직자의 이력서를 기반으로 한 직무 추천 모델들을 작성하고, 가장 우수한 성능을 나타내는 모델을 확인하며, 추천된 직무의 활용도를 높이는 방안을 제안하는 것이다. 이를 위하여 첫째로 대용량 이력서의 학습 데이터 구축과 데이터의 특징(Feature)을 파악하였으며, 둘째로 이력서 학습 데이터셋을 텍스트 임베딩 기법과 딥러닝 모델을 이용하여 파인튜닝을 통한 직무 추천 모델 설계 및 하이퍼파라미터 설정을 구성하였다. 마지막으로 직무 추천 시스템의 사용자 만족도를 높이기 위하여, 추천된 직무를 NCS(National Competency Standard)의 직무 분류 체계와 조합될 수 있도록 하였다.

1차 실험은 CNN, LSTM, Transformer를 적용한 각각의 직무 추천 모델에 이력서 데이터를 52개의 특징(Feature)으로 구성된 학습 데이터셋과 하이퍼파라미터 설정을 사용하여 성능을 평가하였다. 학습 모델은 성능 차이를 발견하지 못했으며, 추론 모델은 CNN이 가장 우수하였다.

2차 실험은 1차 실험에서 사용한 52개 특징에서 결측치가 30% 이상인 특징을 삭제한 34개 특징으로 구성된 학습 데이터셋과 파인튜닝 기법을 적용한 딥러닝 모델의 성능을 평가하였다. 평균적으로 5% 수준의 학습 모델의 성능 상승이 있었으며, 특히 Transformer 모델의 Epoch 값의 상승은 큰 의미가 있다.

본 연구를 통해 학습 데이터의 최적화 및 파인튜닝을 적용한 딥러닝 모델 설계는 직무 추천 모델의 성능 향상에 큰 영향을 미치는 것으로 확인되었으며, Epoch 값의 상승률을 보았을 때 Transformer 모델에 양질의 이력서 데이터를 적용하여 추가적인 모델 학습을 진행한다면, 직무 추천 모델의 성능이 더욱 향상될 것으로 기대한다.

ABSTRACT

A Study on Job Recommendation Model Using Deep Learning-based Text Embedding - Focusing on the Resume Data -

JEON, JUNG-WOO

Department of Project Management
Graduate School of Soongsil University

The modern industrial society has accelerated changes in the industrial structure through rapid technological advancements in the Fourth Industrial Revolution. This has led to the emergence of new occupations and increased complexity in job roles. As a result, there is a growing need for diverse skills among young job seekers preparing to enter such a society. The mismatch between the specialized knowledge acquired through education and the job requirements has become a significant cause of declining motivation and increased job abandonment among job seekers. According to a survey conducted by the Korea Economic Research Institute(2021), the rate of mismatch between major and job roles among young

employed individuals is 52.3%, ranking second among 29 OECD countries. Companies are also experiencing early attrition due to this phenomenon.

Job mismatch can be classified into two categories: skill mismatch and task mismatch. Skill mismatch refers to a misalignment between the job seeker's major and skills and the expertise required for the job. Task mismatch, on the other hand, refers to a situation where the job seeker's competencies and skills related to job performance do not match the requirements of the specific job. While addressing skill mismatch can be partially achieved through job information sharing via employment portals, resolving task mismatch poses a challenge as job seekers need to gather and analyze job-related information provided by employers to match their skills and job requirements.

The purpose of this study is to develop job recommendation models based on job seekers' resumes, identify the most effective model, and propose ways to enhance the usability of recommended job roles. To achieve this, the researchers first built a large-scale dataset of resumes and examined the features of the data. Then, they designed a job recommendation model using text embedding techniques and deep learning models, fine-tuned the model through hyperparameter tuning, and constructed a dataset for training the model with resume data. Lastly, to increase user satisfaction with the job recommendation system, the recommended job roles were combined with the National Competency Standard (NCS) job classification system.

In the first experiment, CNN, LSTM, and Transformer models were applied to each job recommendation model using resume data with 52 features as the training dataset, along with hyperparameter settings. The performance of the training models did not show significant differences, but the inference model using CNN performed the best.

The second experiment evaluated the performance of the deep learning model with fine-tuning using a training dataset composed of 34 features obtained by removing features with missing values exceeding 30% from the 52 features used in the first experiment. On average, there was a 5% improvement in the performance of the training models, and the increase in Epoch for the Transformer model was particularly significant.

Through this study, it was confirmed that optimizing the training data and designing deep learning models with fine-tuning have a significant impact on improving the performance of job recommendation models. Additionally, based on the observed increase in Epoch, it is expected that further improvement in the performance of the job recommendation model can be achieved by applying high-quality resume data to the Transformer model through additional model training.

제 1 장 서 론

1.1 연구의 배경

제4차 산업혁명과 기술 발전은 산업구조 변화를 가속화하고, 산업 분야별로 새로운 직종과 직무가 세분되는데 주요 요인이 되었다. 인간과 인공지능이 융합하는 지능정보사회로 진출을 앞둔 구직자는, 전공에 기반을 둔 직무와 산업이 요구하는 직무 간에 차이로 인하여 의사결정의 어려움을 겪고 있다. 이것은 일자리 미스매칭으로 이어져 사회적·경제적 손실이 발생하고 있다(장석인, 2017).

청년층(15~29세) 취업자의 최근 일자리와 전공과의 관련성 조사 결과에 따르면 전공과 직무의 불일치는 52.3%로 나타났다. 이는 취업자의 절반 이상이 전공과 관련 없는 일자리에서 일하고 있다는 것으로 실제 고용시장의 심각한 문제점으로 지적되고 있다(한국경제연구원, 2021).

취업포털 사람인이 1,124개 기업을 대상으로 '1년 이내 조기 퇴사' 현황에 대한 설문을 조사한 결과, 기업 10곳 중 8곳(84.7%)이 '1년 이내 조기 퇴사자가 있다'라고 응답했으며, 조기 퇴사 사유 중에서도 '직무가 적성에 안 맞아서'가 45.9%로 가장 많았다(사람인, 2022).

정부에서는 “노동시장에서 구인·구직 간 미스매치가 발생하고, 이중구조와 양극화 등 구조적 문제가 심화되고 있다.”라고 밝혔으며, 고용환경 개선과 미스매치 문제 해결을 위하여 많은 예산 집행과 다양한 정책을 추진하고 있다(머니투데이, 2022).

고용노동부 2023년 총예산 30.3조 중 직업훈련(9.2%, 2.8조)과 고용서비스(5.9%, 1.8조)에 배정된 예산은 4.6조(15.1%)로, 2020년 총예산 33.6조 중 3.6조(10.9%)로 증가하였다(고용노동부, 2023). 특히 증가한 예산의 대부분은 직원훈련과 고용서비스에 배정하여 일자리 창출뿐만 아니라 일

자리 미스매치 문제를 해결하기 위하여 노력하고 있다. 하지만, 2023년 3월 기준 미취업자는 49만 7,000명으로 1년 동안 4만 5,000명(9.9%)이 증가하였으며, 구직자는 반복적인 일자리 미스매칭으로 구직 의욕 상실 및 일자리 포기자가 증가한 것을 알 수 있다(경향신문, 2023).

정책과 예산을 지원하는 정부 기관, 일자리 매칭을 위한 이력서를 보유하고 있는 취업 전문기관, 구직자의 학습·경험을 함양시키는 교육기관은 각자의 분야에서 최선의 노력을 하고 있다. 하지만, 미스매치 문제 해결을 위한 노력과 책임은 최종적으로 구직자에게 있다. 미스매치 문제 해결을 위해서 일자리 미스매치(수요-기업)와 직무 미스매치(공급-구직자)로 구분하여 문제의 원인 분석이 필요하다.

일자리 미스매치는 구직자의 능력과 기술이 일자리 요구사항과 맞지 않는 현상으로 구직자의 경험 부족, 역량 부족, 교육 및 훈련의 미비, 구인 기업 현황 등이 복합적 원인이라 할 수 있으며, 교육 및 훈련 강화, 일자리 요구사항 파악을 위한 정보 제공, 취업 지원을 위한 정보 공유 플랫폼 제공을 통하여 많은 부분 해소되고 있다고 볼 수 있다. 특히 저출산·고령화로 인한 노동 인구 감소는 생산 인력의 감소로 이어져, 구직자에게 다양한 취업 기회를 제공할 것이다.

직무 미스매치는 직무 수행과 관련된 역량과 기술이 해당 직무에서 요구하는 것과 맞지 않는 현상으로 특정 직무에 대한 이해 부족, 관련 경험 부족, 기술적인 역량의 부족, 조직의 불명확한 역할 수행 등을 원인으로 볼 수 있으며, 구직자가 스스로 직무 분석과 역할 수행을 위한 학습 경험을 사전 탐색하기에는 역부족인 상황이다. 이러한 문제점을 해소하기 위해서는 일자리 예산 중 미스매치 해소를 위한 예산을 확대해 나가고, 구인 업체와 훈련기관이 함께 참여하는 인력양성 프로그램 개발이 필요하며 구직자 스스로 직무를 탐색하고 학습 계획을 수립할 수 있는

지원 시스템이 필요하다(박재홍, 2020).

앞서 살펴본 통계지표와 미스매치 현상을 통해 일자리 미스매치보다 직무 미스매치의 해소를 위한 연구가 필요하다는 것을 알 수 있으나, 기존 직무 추천 연구들은 딥러닝 모델의 유효성 확인 및 정확도(Accuracy), 정밀도(Precision), 재현율(Recall) 등의 평가지표 산출에 집중되어 있어 추천된 직무를 효과적으로 활용할 수 있는 방법론과 직무 추천 시스템 구축을 위한 실용적 연구에는 한계가 있다.

본 연구에서는 경력자의 직무 정보를 포함하고 있는 이력서 데이터와 국가직무능력표준(National Competency Standards, NCS)의 직무 분류체계 데이터를 학습 데이터로 구성하고, 파인튜닝(Fine-Tuning)을 통한 모델 설계와 CNN(Convolutional Neural Network), LSTM(Long Short-Term Memory), Transformer를 임베딩 기법으로 적용한 3종의 직무 추천 모델의 성능평가 및 추천된 직무가 NCS 직무 분류체계와 매칭될 수 있도록 하는 직무 추천 시스템에 대한 연구를 진행한다.

1.2 연구의 목적 및 방법

본 연구는 구직자에게 적합한 직무 추천에 있어 경력자의 이력서 데이터는 구직자의 직무 선택의 기준 지표를 포함하고 있다는 가설로부터 시작한다. 즉 구직자와 유사한 프로파일을 보유하고 있는 경력자의 이력서 데이터를 기반으로 직무를 추천하였을 때 가장 높은 만족도와 직무 추천 성과를 내포한다는 것이다. 이러한 직무 추천 방법은 전통적인 방법으로 서 구직자의 직무 선택 멘토링에 있어 해당 분야의 전문가 및 선배가 지식과 경험을 바탕으로 구직자의 직무 컨설팅을 진행하는 것과 맥락을 같이 하며, 협업 필터링을 통한 추천 시스템을 이론적 배경으로 하고 있다.

본 연구의 목적은 대규모 데이터에서 복잡한 패턴을 학습하는 데 우수

한 성능이 검증된 딥러닝 모델(CNN, LSTM, Transformer)을 사용하여, 이력서 데이터와 NCS의 직무 분류체계 데이터를 기반으로 구직자에게 적합한 직무를 추천하는 모델의 성능을 확인하는 것이 목적이다. 또한, 독립 변수인 이력서 정보와 종속 변수인 추천 직무 간의 관계를 학습하여 새로운 이력서에 대한 적합한 직무를 추천하는 것에 있다.

본 연구의 방법은 파인튜닝을 통해 설계한 직무 추천 모델을 CNN, LSTM, Transformer의 특성을 반영하여 학습하고, 최종적으로는 각각의 직무 추천 모델의 성능을 검증하여 이력서를 기반으로 한 직무 추천 시스템에 가장 우수한 성능을 나타내는 모델을 확인하기 위한 것이다. 이를 위하여 첫 번째는 대용량 이력서 데이터의 학습데이터 구축 과정과 데이터의 특징(Feature)을 명확히 하고, 두 번째는 이력서 학습 데이터를 텍스트 임베딩 기법과 딥러닝 모델(CNN, LSTM, Transformer)을 적용하여, 최적의 학습 환경 탐색과 성능평가를 통한 최고의 모델을 선정하는 것이다. 세 번째는 직무 추천 시스템의 사용자 만족도를 높이기 위하여 추천된 직무를 NCS 직무 분류체계와 매칭 할 수 있도록 하는 것이다. 또한, 성능 검증에 머물지 않고 NCS 직무 분류체계와 매칭된 직무를 추천할 수 있도록 학습데이터를 구성하여 추천 직무가 실제 시스템에 적용되어 활용될 수 있도록 한다.

본 연구에 사용한 이력서 데이터는 비식별화를 통한 가명처리가 완료된 인적사항, 경력, 희망직무, 학력을 포함하고 있으며 개인정보보호를 위하여 자기소개서와 기업 현황은 제외하였다.

[표 1-1]은 연구단계에 따른 절차와 방법을 도식화한 것이다.

[표 1-1] 직무 추천 모델 연구의 절차 및 방법

연구단계	연구 절차	적용 방법
데이터 분석	데이터 확보, 수집	이력서 데이터 수집
		NCS 직무 분류체계 데이터 수집
	데이터 분석	데이터 특성분석
		데이터 분포분석
		상관관계분석
데이터 전처리	데이터 전처리	데이터 결측치 및 이상치 처리
		코드화 및 수치화 진행
	학습데이터 생성	데이터 증강 및 리샘플링 진행
		데이터 분할 진행
모델개발	임베딩 모델 선정	CNN, LSTM, Transformer
	Feature engineering	Scaling, Normalization
		소범주형, 다범주형, 자연어 변수
		Fully_Connet_Layer, 출력층 설정
	Fine-Tuning	이력서 데이터 관점 Fine-Tuning
		임베딩 모델 관점 Fine-Tuning
		Layer층 관점 Fine-Tuning
	모델 학습	하이퍼파라미터 조정
		Compile 설정
성능평가	직무 추천 모델 성능평가	Accuracy, AUC
	직무 추천 모델 선정	Valiation_Loss 기준 모델 선정
	직무 추론 모델 평가	F1_Score, Recall, Percision

1.3 연구의 구성

본 연구는 총 5장으로 구성되어 있다. 제1장은 본 연구를 진행하게 된 배경과 목적 그리고 연구 방법 등에 관하여 기술하였고, 제2장은 본 연구와 관련된 추천 시스템과 임베딩 기법의 선행연구와 딥러닝 기반의 텍스트 임베딩과 성능평가 방법을 정리하였다. 제3장에서는 직무 추천 모델에 대한 실험 프로세스와 실험 내용을 설명하였으며, 제4장에서는 실험 방법에 따라 진행한 실험 과정과 결과를 제시하였다. 마지막으로 제5장에서는 본 연구의 결론과 한계, 향후 연구 방향에 대하여 제안하였다.

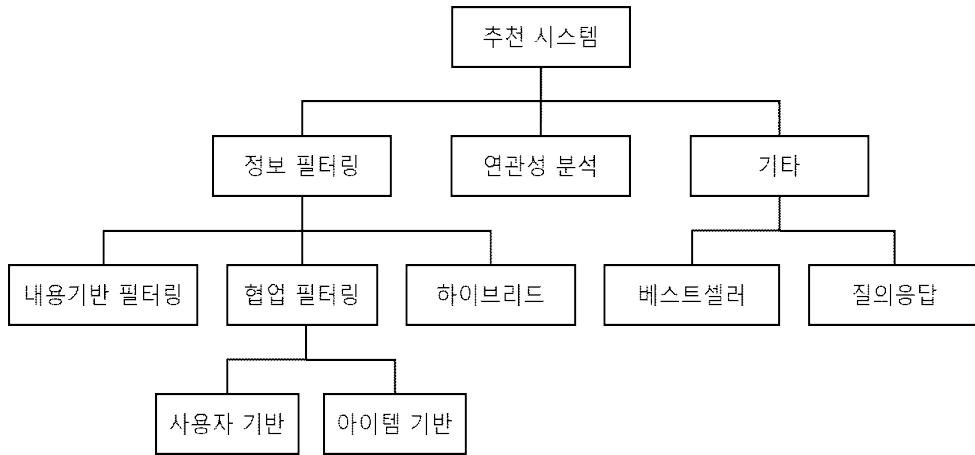
제 2 장 이론적 배경

2.1 추천 시스템

인터넷의 발전으로 인한 디지털 콘텐츠 급증으로 대량의 정보를 처리하는 데 어려움을 겪는 정보 과부하(Information Overload) 문제를 해소하기 위한 추천 시스템(Recommendation Systems)은 사용자에게 가장 관련성 높고, 관심 있는 정보 또는 상품을 선별하여 제공함으로써, 개인화된 경험을 제공하고 사용자 만족도를 높일 수 있다. 또한, 사용자의 선호에 맞는 상품을 노출해 구매를 유도하거나 콘텐츠를 홍보하는 데 효과적이기 때문에 온라인 쇼핑몰, 콘텐츠 플랫폼 등에서 매출 증대에 영향을 준다(Das et al., 2013).

추천 시스템은 여러 연구자에 의해 발전해왔지만, 사용자의 행동과 취향을 분석하여 추천을 제공하는 ‘협업 필터링(Collaborative Filtering)’은 개인화된 추천, 새로운 항목에 대한 추천, 커뮤니티 지향적인 추천 등 다양한 분야에 적용할 수 있는 우수한 성능의 기법으로써, Goldberg et al.(1992)의 논문에서 그 개념이 처음으로 소개되었다. 이를 통해 사용자들은 개인적인 관심사를 반영한 추천을 받을 수 있고, 새로운 항목을 발견하며, 다른 사람들과의 연결과 상호작용을 경험할 수 있다.

[그림 2-1] 추천 시스템의 분류에 따라 아이템 자체의 정보만을 기준으로 선호도를 예측하는 내용 기반 필터링(Content-Based Filtering)과 사용자의 평가 정보를 기준으로 선호도를 예측하는 협업 필터링(Collaborative Filtering) 그리고, 두 가지를 결합한 하이브리드(Hybrid) 형태의 3가지 접근법으로 분류할 수 있다. 이 세 가지 접근법은 각각의 특징과 장단점을 고려하여 적합한 방식을 선택할 수 있다.

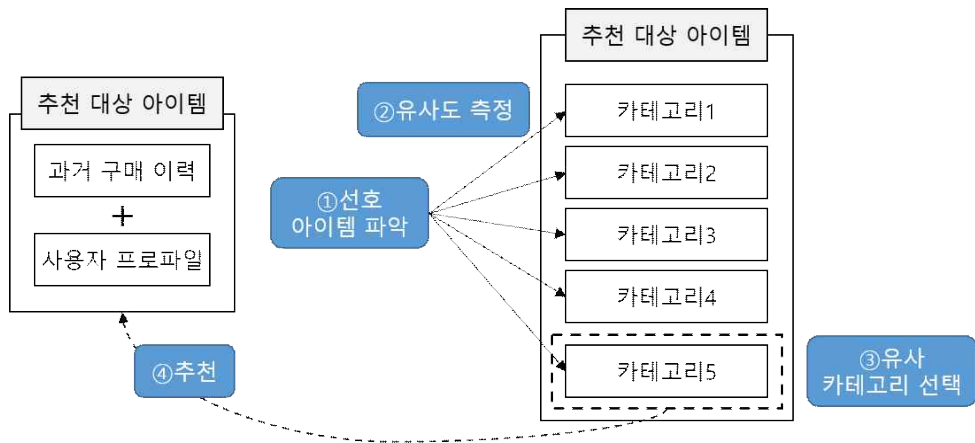


[그림 2-1] 추천 시스템의 분류

[표 2-1]은 추천 시스템의 장단점을 정리한 것이며, 본 연구에서는 협업 필터링의 이론적 배경을 중심으로 직무 추천 연구를 진행하였다.

[표 2-1] 추천 시스템의 장단점

방법	장점	단점
협업 필터링	<ul style="list-style-type: none"> - 새로운 상품이나 사용자 정보를 바로 반영 가능 - 다양한 데이터를 활용하여 맞춤형 추천이 가능 - 사용자 간 유사성을 기반으로 개인화된 추천이 가능 	<ul style="list-style-type: none"> - 콜드스타트(Cold Start) 문제 - 데이터 부족으로 정확도 감소 - 사용자와 상품 간의 상호작용이 적은 제품에 대한 추천에 취약함
내용 기반 필터링	<ul style="list-style-type: none"> - 사용자가 선호하는 특징에 기반하여 추천(정확도 높음) - 콜드스타트 문제 해결 가능 - 상품 내용, 속성을 활용하여 추천이 가능 	<ul style="list-style-type: none"> - 사용자의 선호도가 없는 새로운 제품 추천이 어려움 - 사용자의 행동에 의존하지 않기 때문에, 타 사용자들이 선호하는 제품을 추천하지 못함
하이 브리드	<ul style="list-style-type: none"> - 협업 필터링과 내용 기반 필터링의 장점을 조합하여 정확도를 높일 수 있음 - 다양한 데이터를 활용할 수 있으며, 새로운 사용자나 상품에 대한 추천도 가능 	<ul style="list-style-type: none"> - 두 알고리즘의 결합 방식에 따라서 성능 차이 발생 - 구현이 복잡하여 시스템 구축이 어려울 수 있음



[그림 2-3] 내용 기반 필터링 기법

[그림 2-2]의 내용 기반 필터링 기법은 아이템의 내용을 분석하여 아이템 간의 유사성을 계산하고 사용자의 선호도와 비교하여 개인화된 추천을 제공하는 방식이며, 사용자에게 관심 있는 아이템을 추천하고, 새로운 아이템에 대해서도 추천할 수 있다(Wu et al., 2000). 또한, 내용 기반 필터링은 분석이 비교적 간단하며, 메타데이터를 활용하여 아이템 간의 유사성을 파악하는 것이 가능하다. 이를 통해 다양한 분야의 아이템, 특히 텍스트 기반의 콘텐츠에 대한 추천이 가능하며, 사용자의 개인적인 취향과 관심사를 고려한 개인화된 추천 서비스를 제공할 수 있다(Pazzani & Billsus, 2007).

협업 필터링은 추천 기법에 따라 사용자 기반 협업 필터링과 아이템 기반 협업 필터링으로 구분할 수 있다.

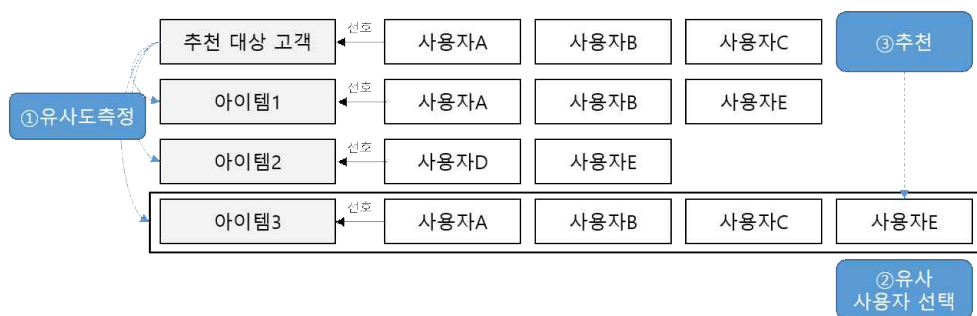
사용자 기반 협업 필터링(User-based Collaborative Filtering)은 사용자들 간의 유사도를 기반으로 추천을 제공하는 기법으로 [그림 2-3]에 추천 과정을 도식화하였다. 이 기법은 사용자들의 과거 구매 이력과 같은 데이터를 사용하여 각 사용자 간의 유사도를 측정하고, 그 결과를 바탕으로 가장 유사한 사용자들이 선택한 아이템을 추천 대상 고객에게 추

천한다. 유사도 측정 방법으로는 두 벡터 간의 공분산(Covariance)과 각 벡터의 분산(Variance)을 계산하여 -1부터 1까지의 값을 가지는 상관계수(Correlation Coefficient)를 구하는 피어슨 유사도(Pearson Similarity)를 일반적으로 사용한다.



[그림 2-4] 사용자 기반 협업 필터링 기법

[그림 2-3]의 사용자 기반 협업 필터링 기법을 일자리 소개 서비스에 적용한다면, 각 사용자의 이력과 능력을 바탕으로 유사도를 측정하여 유사한 능력을 가진 사용자들끼리 연결하는 역할이 가능하다. 또한, 직무 추천 서비스에 적용한다면 사용자들의 이력을 바탕으로 유사한 사용자들이 선호하는 직무를 추천해주는 역할이 가능할 것이다.



[그림 2-5] 아이템 기반 협업 필터링 기법

[그림 2-4] 아이템 기반 협업 필터링(Item-based Collaborative Filtering)은 대상 아이템과 유사한 아이템을 찾아서 대상 아이템을 구매한 사용자들에게 추천하는 기법이다. 이 기법은 아이템 간의 유사도를 측정하고, 대상 아이템과 유사한 아이템들을 찾아내는 단계와 대상 아이템을 구매한 사용자 중에서 대상 아이템과 유사한 아이템들을 구매하지 않은 아이템을 추천하는 단계로 이루어진다.

유사도 측정 방법으로는 두 벡터 간의 각도에 코사인값을 곱하여 내적(Dot Product)을 계산하고, 각 벡터의 크기(Norm)를 곱하여 -1부터 1까지의 값을 가지는 상관계수(Correlation Coefficient)를 구하는 코사인 유사도(Cosine Similarity)를 일반적으로 사용한다. 예를 들어, 상품을 판매하는 매장에서 매장 직원이 고객의 요구사항을 파악하고 해당 제품과 유사한 다른 제품을 추천해주는 방식으로 일반화되어 있으며, 또한 온라인 쇼핑몰에서도 해당 제품과 유사한 다른 제품을 추천하는 데 활용된다.

협업 필터링에서는 추천 대상 고객의 선호도는 사용자의 아이템 만족도인 행렬값으로 예측할 수 있다. [표 2-2]은 경력자의 직무에 대한 만족도를 점수로 나타낸 행렬값의 예시이다.

[표 2-2] 경력자-직무 행렬

	직무 1	직무 2	직무 3	직무 4
경력자 A	4	-	5	-
경력자 B	4	2	1	-
경력자 C	3		2	-
경력자 D	4	4	-	-
구직자 E (신규 사용자)	-	-	-	-

[표 2-2]의 경력자-직무의 행렬에서 각 직무의 정보에 따라 '직무1'과 같이 많은 평가를 받은 아이템은 추천 가능성이 크고, '직무4'와 같이 아

직 아무런 평가를 받지 못한 아이템은 추천 가능성이 작을 수 있다. 또한, '구직자 E'와 같이 새로운 사용자가 등장하는 경우에도 이를 콜드스타트(Cold Start) 문제라고 하며, 아이템 기반 협업 필터링에서는 새로운 사용자의 선호도 정보가 없기 때문에 추천이 불가능하다.

따라서, 새로운 아이템이나 사용자의 등장으로 인한 콜드스타트 문제를 해결하기 위해서는 아이템의 특성 정보를 활용하거나, 하이브리드 추천 시스템(Hybrid Recommender Systems)을 사용하는 등의 방법을 활용할 수 있다. 마찬가지로, 새로운 사용자의 경우에는 사용자 프로파일 정보를 수집하거나, 내용 기반 추천(Content-Based Recommendation)을 활용하는 등의 방법을 사용하여 추천 성능을 향상시킬 수 있다.

2.1.1 직무 추천

정보통신 기술과 플랫폼 서비스의 발전은 산업의 정보 비대칭 문제를 해결하는 데 공헌하였으며 대표적으로 취업 정보를 제공하는 플랫폼 기업들은 구직자의 이력서와 기업의 채용공고 정보를 기반으로 미스매치 문제를 완화시키는 노력을 하고 있다. 이러한 플랫폼 기반의 추천 시스템은 구직자와 기업 간의 상호작용을 촉진하고, 구직자에게 적합한 일자리를 찾을 수 있는 기회를 제공하며, 구직자의 역량과 기업의 요구사항을 분석하여 최적의 추천을 도모한다. 하지만, 직무 분야에 적용된 추천 시스템 역시 근본적인 문제는 여전히 가지고 있다. 즉, 직무 추천 시스템은 신규 구직자 및 추천 직무의 경우 과거 정보가 부족하여 추천 성능이 떨어지는 Cold-Start 문제가 발생하거나 복잡하고 방대한 데이터를 다루는 것에 대한 한계점이 있으며, 콘텐츠의 특성을 대부분 수동적으로 추출해야 하는 문제가 있다(Lakshmi & Lakshmi, 2014).

직무 추천 시스템은 정보통신 기술 발전에 따라 정보 전달에 기반을 두

어 지능형 알고리즘을 기반으로 구직자와 구인기업 간의 상호작용이 원활히 제공될 수 있도록 하는 것이다. 하지만, 기존 연구는 학습 데이터 수집 및 컴퓨터 자원의 한계로 학력, 자격증, 어학성적 등 수치화가 가능한 정형 데이터 중심의 연구를 진행하였다. 이후, 빅데이터 전처리 및 자연어 처리 기법을 통한 구직자의 경력, 사회봉사, 아르바이트 등 사회적 경험을 활용하는 연구로 확장되었다.

협업 필터링과 딥러닝 모델에 MBTI(Myers-Briggs Type Indicator) 성격유형을 반영한 결과와 반영하지 않을 결과를 비교한 연구가 진행하였으며, 장예화 외(2021)는 정형 데이터 중심의 역량 정보만으로 정보를 전달하는 취업 정보 서비스 한계를 개선하기 위해 16가지 성격유형(MBTI)을 반영한 직무 추천 방법론을 제안했다. 이를 위해, 성격유형(MBTI)에 대한 성격 사전을 구축하고, 이를 이력서 데이터의 자기소개글에 적용하여, 구직자의 성격유형(MBTI)을 도출했다.

2.1.2 채용공고 추천

취업포털 사이트의 상단에 노출되는 채용공고는 구직자의 수요에 근거한 것이 아니라, 광고비를 부담하는 구인 기업의 수요에 따라 제공된다. 취업 포털의 주 수입원은 채용공고의 노출 위치 및 크기이며, 매출 증가를 위하여 유료 광고의 빈도를 높이는 방향으로 사이트를 개선해왔다. 따라서 구직자들은 취업포털 및 취업 정보가 넘쳐남에도 불구하고 여전히 자신에게 필요한 채용공고와 기업정보 수집에 어려움을 겪고 있으며, 광고 기반의 채용공고들이 많아지면서 원하는 채용 정보를 얻기가 더욱 힘들어졌다.

직무 미스매치 문제를 해결하기 위한 방법으로 광고를 기반으로 한 채용 정보 제공이 아닌, 구직자 맞춤형 추천 시스템을 활용하면 구직자가

선호하는 채용 정보를 쉽게 제공하여 구직자가 기업의 채용공고를 찾는 데서 겪는 어려움을 해소할 수 있을 것이다. 하지만, 기업 추천 시스템과 비교하여 하이브리드(Hybrid) 기법을 적용한 기업/직무 추천 시스템은 약간의 성능 개선만 있었으며, 그 원인은 두 가지로 분석할 수 있다. 첫 번째는 기업/직무 추천 시스템에 비해 기업 추천 시스템의 성능이 높지 않아서 하이브리드 기법을 적용해도 기업/직무 추천 시스템의 성능이 향상되지 않았기 때문이다. 두 번째는 기업 추천 시스템에서 추천된 기업과 기업/직무 추천 시스템에서 추천된 서비스 사이에 공통된 기업이 적어서 값에 대한 영향이 크지 않았기 때문이다(박수상, 2016).

2.1.3 인재 추천

구직자의 프로파일과 기업의 채용 정보의 요구사항을 추천에 활용하여 구직자에게는 일자리를 추천하고, 기업에게는 인재를 추천해주는 인재 매칭 선행연구를 조사하였다. 구직자의 이력서 정보를 모두 사용하지 않고 일자리 추천에 영향을 미치는 구직자의 전문 기술 정보만을 추출하여 워드 임베딩을 얻는 방식을 사용하면 추천 성능에 영향을 줄 수 있다. 해당 연구는 전문기술에 대한 용어추출 및 표현방식에 따라 벡터 공간모델 기반의 워드 임베딩 방식(Word2Vec)과 빈도 기반의 TF-IDF(Term Frequency-Inverse Document Frequency) 방법을 사용하여 각각 성능을 비교하였다. 정보통신 분야의 기술에만 국한하여 일자리 매칭 커뮤니티 링크드인에서 50명의 프로파일을 수집하고 Catho에서 3,877개의 일자리 프로파일을 수집하여 벡터 공간모델 기반의 워드 임베딩 방식의 두 가지 방식인 CBOW(Continuous Bag of Word)과 Skip-Gram을 실험한 결과, Skip-Gram 임베딩 방식을 활용한 추천이 가장 좋은 성능을 보였다고 보고하였다(Valverde-Rebaza et al, 2018).

TF-IDF 및 bag-of-words와 같은 워드 임베딩 방식 모델은 자연어 처리에 효과적이지만 단어의 출현 빈도에만 초점을 두기 때문에 맥락을 이해하는 데는 부족하다고 지적하며, 맥락을 이해하도록 하기 위해서는 GloVe(Global Vectors for Word Representation), ELMo(Embeddings from Language Model), Word2vec 등과 같은 임베딩 모형을 활용하여 벡터화할 것을 권고하고 있다(Le & Mikolov, 2014).

구직자의 반 구조화된(Semi-Structured) 이력서를 채용공고의 직무 설명과 매칭하여 구직자에게 일자리를 추천하는 삼 신경망(Siamese Neural Network)을 적용한 일자리 추천 모델이 제안되었으며, 이처럼 기존의 일자리 추천 연구에서는 구직자의 이력서 정보와 채용 정보를 전처리 과정에서 다양한 벡터화 기법을 적용하는 연구가 진행되고 있다(Maheshwary & Misra, 2018).

양방향 척도 학습 모델은 상호작용 이력이 있는 취업공고와 구직자 이력서 사이의 적합도를 학습하여 특정 구직자의 이력서에 적합한 일자리 또는 특정 취업공고에 적합한 구직자를 매칭하는 추천 시스템 개발에 사용되며, 신규 구직자나 신규 직무에 대한 추천이 가능하여 협업 필터링의 Cold-Start 문제를 극복할 수 있는 장점이 있다. 이러한 추천 시스템의 연구를 위하여 구직자 ID와 취업공고 ID 대신 구직자 프로필과 취업공고문서의 텍스트를 bag-of-words 벡터로 변환하여 동시에 입력한 후, 심층 신경망을 거쳐 최종 출력층에서 적합도 여부를 학습하였다(송희석, 2020).

최근 머신러닝 및 딥러닝 기술이 가장 많이 적용되고 있는 분야 중 하나가 자연어 처리(NLP, Natural Language Processing) 분야이다. 기존 자연어 처리 분야 연구에서는 워드 임베딩 모델을 활용하여 모델의 입력하기 위한 벡터화(Vectorization)를 주요 연구로 진행하였으나(Le &

Mikolov, 2014), 컴퓨터비전(Computer Vision) 및 이미지 처리(Image Processing) 분야에서 활용하여 좋은 성과를 내는 전이 학습 모델을 자연어 처리 분야에서도 전이 학습된 언어 모델을 이용하는 것이 성능향상에 도움이 된다고 보고하고 있다(유소엽 & 정옥란, 2019).

또한 FAIR(Facebook AI Research) 논문에서 제안된 것과 같이 실험을 통해 랜덤으로 모델을 초기화를 시키는 것보다 사전 학습된 모델의 초기값을 사용하였을 때 더 빠르게 높은 정확도에 도달한다는 실험 결과를 제시하였다(He et al., 2019)

오소진(2022)은 채용 정보와 구직자 프로필과 같은 문서 내 단어의 숨은 의미를 이해할 수 있는 전이 학습 모델을 적용하면, 기존 일자리 추천에서 사용되는 단순한 워드 임베딩을 활용한 학습 모델보다 더 높은 성능을 얻을 수 있을 것이라는 아이디어에 착안하여 BERT(Bidirectional Encoder Representations from Transformers) 기반의 전이 학습 모델을 제안하였다. 해당 연구를 통하여 전이 학습을 이용한 BERT 기반의 양방향 인제매칭 모델은 문맥적인 의미와 관계를 고려하여 의미적 관계 파악의 어려움과 콜드스타트 문제를 해결하는 데 효과적이라 주장하였다.

2.2 딥러닝(Deep Learning)

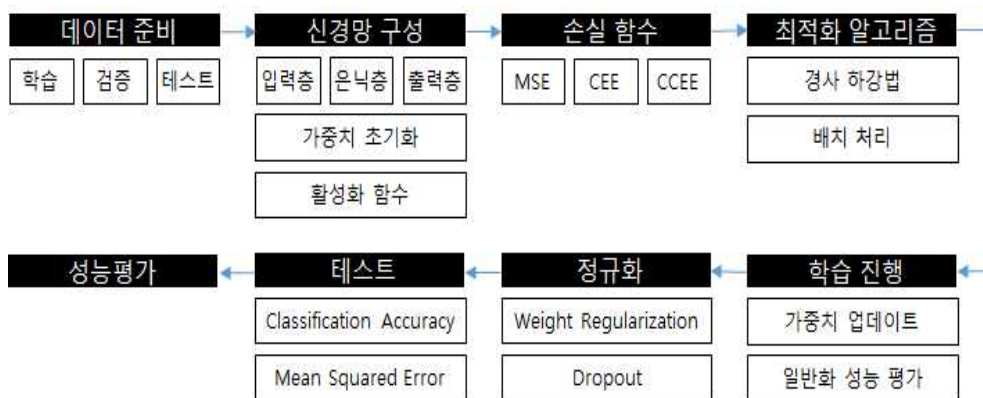
딥러닝(Deep Learning)은 기계학습 기법의 일종으로 심층 신경망(Deep Neural Network, DNN)이라고도 한다. 심층 신경망은 인공 신경망(Artificial Neural Network, ANN)의 일종이지만 은닉층을 추가하여 비선형적인 문제에 대한 해결 방법을 제시하여 주목받게 되었다. 딥러닝은 뉴런 네트워크가 뇌의 동작을 모방하여 이를 알고리즘으로 만든 것으로 네트워크를 구성하기 위해 층(Layer)으로 연결되는 많은 뉴런이라는 노드를 포함한다. 최소한의 층으로 입력층과 출력층은 가져야 하며, 더

많은 정보를 얻기 위해서는 여러 은닉층을 두어 심층 신경망으로 구성할 수 있다(우영춘 외, 2019).

신경망(Neural Network)은 뇌가 동작하는 원리를 모방하는 것으로, 학습을 통하여 연결 강도를 조절하여 성능을 개선한다. 신경망에서 연결 강도를 조절하는 과정은 학습조건에 의하여 제어되며, 이 학습조건은 신경망의 성능에 중요한 역할을 한다. 학습조건은 인간이 경험을 통하여 학습하는 원리를 모방하는 것으로 연결 강도를 신경망의 성능을 개선하는 방향으로 조절한다(백용선 & 김용선, 2010).

신경망의 학습 목적은 입력과 출력 데이터 사이의 대응 함수를 학습하여, 주어진 입력에 대한 정확한 출력을 예측할 수 있도록 가중치와 편향을 조정하는 모델을 만드는 것이다. 이를 위하여 신경망의 학습에는 손실 함수(Loss Function), 최적화 알고리즘(Optimization Algorithm), 배치 처리(Batch Processing), 활성화 함수(Activation Function), 가중치 초기화(Weight Initialization), 정규화(Regularization), 드롭아웃(Dropout) 등의 기법을 활용하여 신경망의 성능을 개선한다.

[그림 2-5]는 신경망 학습 목적에 따른 학습 절차를 정리하였다.



[그림 2-5] 신경망의 학습 절차

신경망의 학습을 위한 알고리즘의 필수적인 요소는 활성화 함수, 손실 함수, 최적화 기법이다. 신경망은 입력값을 가중치와 선형대수 연산을 하고 결과의 값을 출력 형태로 변환하는데 가중 합산한 값을 활성화 함수에 입력하여 출력으로 변환한다. 활성화 함수는 수학적으로 선형 결합한 입력값들을 다양한 형태의 비선형 또는 선형 결합으로 변환하는 역할을 한다(이창기 외, 2014).

최종 출력층이 아닌 은닉층에서 학습의 효율 및 성능을 위해 사용하는 활성화 함수로는 소프트맥스(Softmax) 함수도 사용하지만 하이퍼볼릭 탄젠트(Hyperbolic Tangent, tanh), 렐루(Rectified Liar Unit, ReLU) 등 그 특성에 따라 다양한 함수가 있다. 또한, 학습하고자 하는 목표의 최종 정답 유형에 따라 사용하는 마지막 출력층의 분류를 위한 활성화 함수로는 이진 분류를 통한 예측의 경우 시그모이드(Sigmoid) 함수, 다중 분류를 위해서는 통상적으로 소프트맥스 함수를 사용한다(조운환 외, 2016).

분류 문제에 있어서 최종 출력층에서 주로 사용하는 대표적인 활성화 함수인 시그모이드와 소프트맥스, 그리고 은닉층에서 주로 사용하는 활성화 함수인 TANH와 ReLU의 특징과 수식을 [표 2-3]에 나타내었다.

[표 2-3] 활성화 함수의 특징과 수식

함수	특징	수식
Sigmoid	S자 모양의 Sigmoid 곡선을 가지는 미분 가능한 함수	$f(x) = \frac{1}{1 + e^{-x}}$
Softmax	세 개 이상으로 분류하는 다중 클래스 분류에서 사용되는 활성화 함수	$y_k = \frac{e^{a_k}}{\sum_{i=1}^n e^{a_i}}$
TANH	Sigmoid의 대체제로 사용될 수 있는 활성화 함수로 Sigmoid와는 출력의 범위에서 차이가 있다.	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
ReLU	Sigmoid와 TANH가 갖는 기울기 소실 문제를 해결하기 위한 함수	$f(x) = \max(0, x)$

신경망 학습은 최종 출력층의 예측값과 정답값과의 차이를 줄이기 위해 순전파(Forward Propagation)와 역전파(Back Propagation)를 연결하고, 손실 함수의 그래디언트(Gradient) 계산을 통하여 가중치를 업데이트하는 과정을 반복하는 것이며, 이때 예측값과 정답값의 차이를 수치화해주는 함수를 손실 함수(Loss Function)라고 한다.

손실 함수의 선택은 주어진 문제 유형에 따라 다른 손실 함수를 사용할 수 있는데 그 특징을 [표 2-4]에 나타내었다

[표 2-4] 손실 함수의 종류와 수식

손실 함수	특징	수식
MSE (Mean Squared Error)	예측값과 실제값 차이의 제곱 평균. 즉 오차 제곱의 평균	$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$
Cross-Entropy Error	예측값과 실제값 차이를 줄이기 위한 엔트로피	$cross\ entropy = \sum_{k=1}^i t_k \log_e y_k$
Categorical Cross Entropy Error	분류가 3개 이상인 데이터를 대상으로 사용하는 손실 함수	$CCEE = -\frac{1}{N} \sum_{j=1}^N \sum_{i=1}^C t_{ij} \log(y_{ij})$

역전파는 순전파 단계에서 계산된 출력값과 실제 값의 오차를 역방향으로 전파하여 각 노드의 그래디언트를 계산하고 가중치를 업데이트하는 과정을 반복하므로써 신경망의 예측 오차가 최소화 될 수 있도록 한다. 즉, 손실 함수를 통하여 계산된 차이값을 줄여가는 역전파 학습은 최적화 방법(Optimizer)의 적용에 따라 신경망 모델의 성능에 영향을 미친다. 주로 사용하는 최적화 방법은 기울기의 반대 방향으로 일정 크기만큼 이동하며 연산을 반복하여 손실 함수의 값을 최소화하는 가중치(W)와 편향(b)을 찾는 기법인 경사 하강법 계열의 방법과 Momentum 최적화 기법, RMSProp 기법, Adam 기법 등이 있다(Yazan & Talu, 2017). 이와 관련된 대표적인 최적화 기법은 [표 2-5]와 같다.

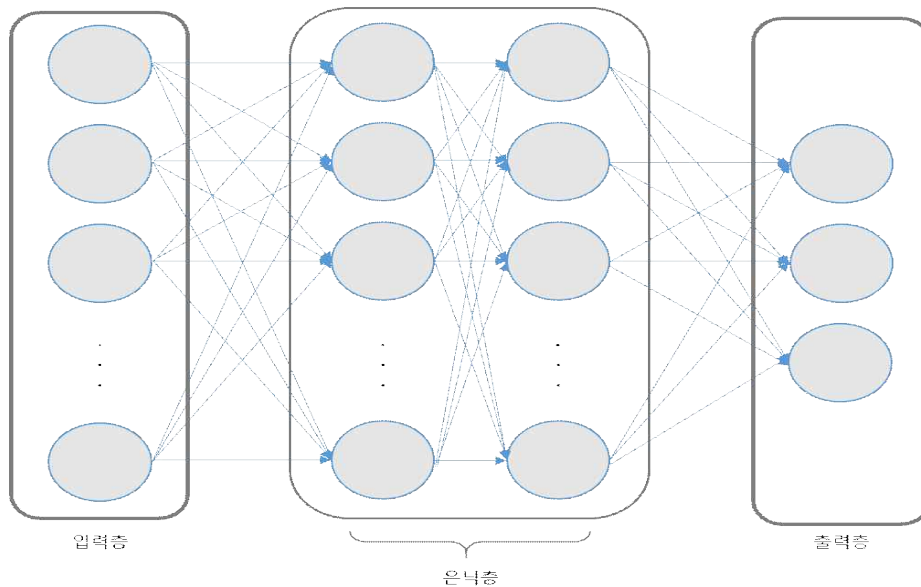
[표 2-5] 주요 최적화 기법

최적화 기법		특징
경사 하강법	배치 (BGD)	<ul style="list-style-type: none"> 전체 데이터의 오차 계산으로 최소값에 안정적으로 수렴 반복될 때마다 모든 학습데이터를 사용하므로 학습이 오래 걸린다는 단점 존재
	확률 (SGD)	<ul style="list-style-type: none"> 한번 반복마다 한 개의 Sample로 경사 하강하는 것이기 때문에 항상 전역 최소값에 도달 백터화 과정에서 속도가 느림. 데이터를 개별적으로 하나씩 처리하는 방식이기 때문에 매우 비효율적
	미니배치 (MBGD)	<ul style="list-style-type: none"> 한번 반복마다 $n(1 < n < m)$개의 데이터를 사용. BGD와 SGD의 장점을 합친 알고리즘
momentum		<ul style="list-style-type: none"> Gradient Descent에서 Gradient의 가중 평균치를 가중치로 업데이트 속도가 빠르고 SGD가 과적합 되는 것을 방지하여 기울기가 지역 최소값이 되지 않도록 구현
RMSProp		<ul style="list-style-type: none"> Root Mean Square Propagation의 약자. 기울기 강하의 속도를 증가시키는 알고리즘. 지수 이동 평균 적용
Adam		<ul style="list-style-type: none"> 모멘텀과 RMSProp을 섞어놓은 최적화 알고리즘. 딥러닝에서 가장 많이 사용되는 최적화 알고리즘 중 하나임.

2.2.1 심층 신경망(DNN)

심층 신경망(Deep Neural Network, DNN)은 비선형 모듈을 통해 분류 또는 예측하는 데 필요한 특성을 자동으로 탐색하고 은닉층이 깊어질수록 더 고차원적인 레벨의 특성을 탐색한다. 딥러닝은 데이터를 고차원 벡터화하여 데이터의 패턴과 특징을 추출하고 이를 토대로 신경망 학습 과정을 통하여 다양한 문제를 효과적으로 해결한다. 심층 신경망은 [그림 3-3]과 같이 입력층(Input Layer)과 출력층(Output Layer) 사이에 여러 개의 은닉층(Hidden Layer)으로 이루어진 인공 신경망으로 입력층으로부터 층이 깊어질수록 더 고차원적인 레벨의 특성을 탐색한다(Lecun et al., 2015).

인공 신경망은 인간의 대뇌를 토대로 만들어진 구조로, 비선형 모듈을 통해 분류(Classification) 또는 예측(Regression)과 같은 작업을 수행하기 위해 필요한 특성을 자동으로 탐색한다(Zhang, Yihua, 2021).



[그림 2-6] 심층 신경망(DNN) 구조

심층 신경망 모델의 특징은 모델 자체가 입력데이터에 의해 내부 파라미터가 자동으로 조절되는 자가 조정 모델이라는 것이다(박상현, 2017).

심층 신경망 모델은 순환 신경망(Recurrent Neural Network, RNN)보다 많은 은닉층을 가진 신경망 구조를 의미하며, 많은 수의 은닉층을 통해 다양한 은닉층과 비선형 함수 근사, 표현 학습, 자동 특징 학습, 대량의 데이터 처리 등으로 비선형 문제를 해결하는 데 중요한 역할을 한다.

Huang et al.(2004)은 비선형 관계인 14개의 입력변수를 이용하여 신용등급을 비교적 정확하게 예측해 냈다. 특히, 전통적인 심층 신경망들은 일반적인 인공 신경망으로 은닉층을 쌓아 설계했지만, 최근의 심층 신경망 연구는 학습 구조에 순환 신경망(RNN)을 적용하여 연구하고 있다.

하나의 예로 자연어를 학습시키는 방법의 하나인 언어 모형화 기법을 적용할 때 심층 신경망 구조를 활용한 연구가 있다. 심층 신경망은 표준 오류 역전파 알고리즘으로 학습할 수 있는데, 합성곱 신경망

(Convolutional Neural Network, CNN)은 영상처리나 이미지 처리에서 좋은 성과가 있었고 성공적인 적용 사례에 대한 지식 또한 잘 축적되어 있다(Vargas et al., 2017).

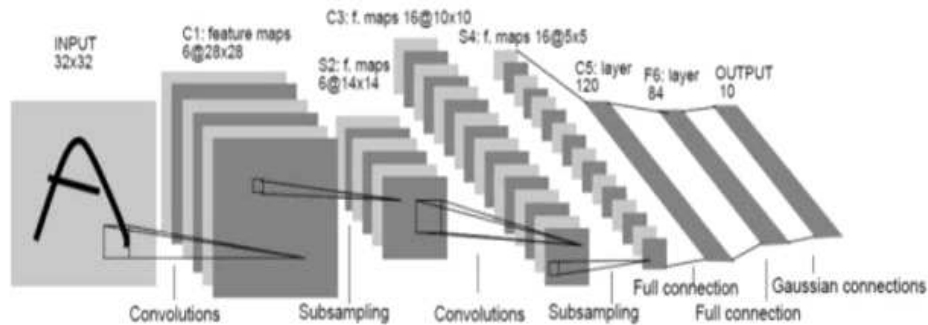
2.2.2 합성곱 신경망(CNN)

합성곱 신경망(Convolutional Neural Network, CNN)은 사물이나 물체를 시각적으로 인식할 때 이미지를 세부적인 작은 구역들로 나누어 부분적인 특징을 인식한 후 이것들을 결합해서 전체 이미지를 인식하는 것에 착안하여 개발된 방법론이다. 합성곱 신경망은 분석할 데이터를 입력하는 1개 이상으로 구성할 수 있는 합성곱 계층(Convolution layer)을 거쳐 정보를 요약하는 풀링층(Pooling Layer)이 있으며 완전 연결층(Fully Connected Layer)을 거쳐 출력층으로 정보가 최종적으로 출력된다. 입력층에 입력된 이미지는 합성곱층을 통과하면서 필터링되어 핵심적인 특징이 추출된다.

CNN 모델은 이미지 내에서 특징점의 위치가 달라질 때도 효과적으로 대응할 수 있다. 예를 들어 이미지의 각도가 바뀌거나 회전이 되거나 할 경우 일반적인 기계학습으로는 전혀 다른 패턴으로 인식이 되는 경우가 많으나 CNN에서는 부분적인 특징으로부터 풀링층을 거쳐서 이미지의 특징들을 잘 추출할 수 있으므로 일반적인 기계학습보다 뛰어난 성능을 발휘한다(이모세 & 안현철, 2018).

CNN 모델은 이미지 인식 분야뿐 아니라 텍스트 문장을 단어 벡터들의 순열로 표현하여 CNN 모형에 입력하여 효과적으로 분류할 수 있다는 것이 선행연구를 통해 입증되었다(하만석, 2019).

[그림 2-7]은 1998년에 얀 르쿤이 제안한 ‘Le-Net-5’의 모델 구조이다.

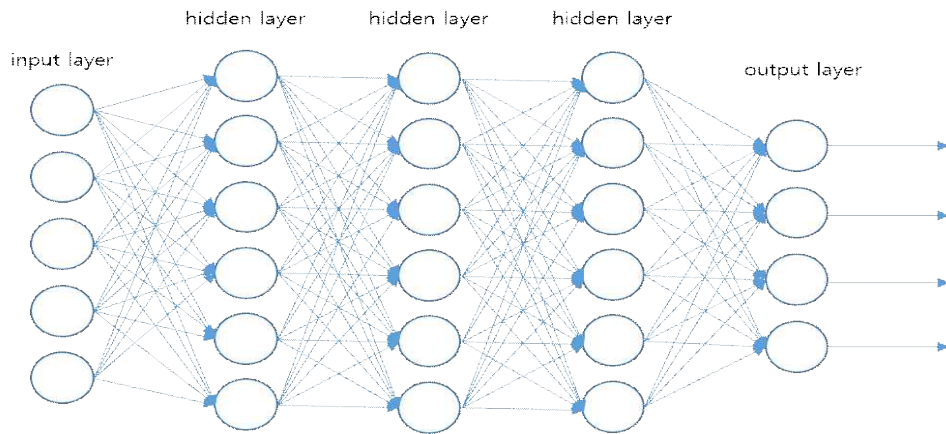


[그림 2-7] LeNet-5의 구조

2.2.3 순환 신경망(RNN)

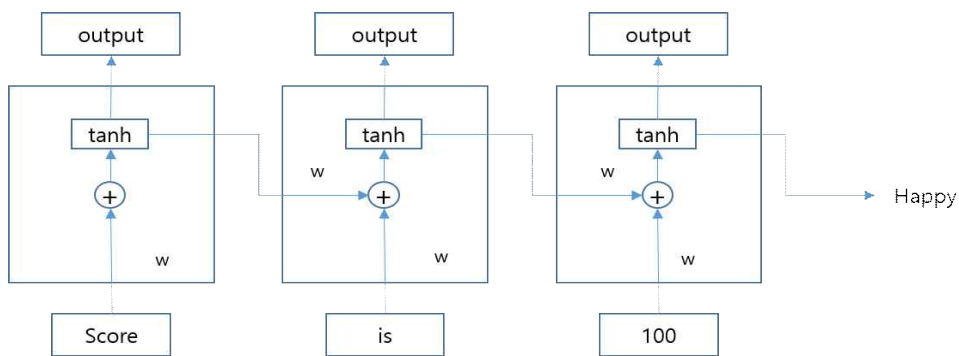
순환 신경망(Recurrent Neural Network, RNN)은 비선형 문제를 학습하는 능력을 갖추고 있으며, 시퀀스 데이터를 이진 분류로 처리하는 데 특화된 인공 신경망 구조이다. 또한, 이전 단계의 출력을 현재 단계의 입력으로 사용하여 예측을 수행함으로써 시간적인 의존성을 고려한다.

RNN은 내부에 퍼셉트론을 포함하고 있으며, 퍼셉트론은 입력값과 가중치를 곱한 다음, 이를 모두 합한 선형 결합을 계산한다. 이후, 계산된 선형 결합에 활성화 함수를 적용하여 출력값을 결정하며, 일반적으로는 시그모이드 함수와 같은 비선형 활성화 함수를 사용한다. 퍼셉트론은 단층 퍼셉트론과 다층 퍼셉트론(MLP, Multi-Layer Perceptron)으로 구분할 수 있으며, [그림 2-8]과 같이 여러 개의 퍼셉트론(Rosenblatt, 1957)을 여러 층으로 쌓은 MLP(Multi-Layer Perceptron)의 형태로 구성할 수 있다. 퍼셉트론은 학습 과정을 통해 가중치를 조정하여 입력값에 대한 적절한 출력을 학습하며, 오차 역전파(Back Propagation) 알고리즘을 사용하여 가중치를 업데이트하고, 반복적인 학습을 통해 최적의 가중치를 찾아내는 것을 목표로 한다. 또한, 자연어 처리, 음성 인식, 시계열 데이터 분석 등 다양한 분야에서 유용하게 활용되고 있다.



[그림 2-8] Multi-layer 퍼셉트론 구조

다층 신경망은 텍스트처리 영역에서 우수한 성능을 보여주고 있지만, 전결합 네트워크(Fully Connected Network)는 엄청난 파라미터 개수가 포함되어 있어 과적합(Overfitting)이 쉽게 발생한다(김광석, 2020).



[그림 2-9] RNN을 활용한 감성인식 방법

[그림 2-9]는 RNN을 활용하여 문장에서 감성 인식을 처리하는 방법을 도식화하였다.

2.2.4 하이퍼파라미터(Hyperparameter)

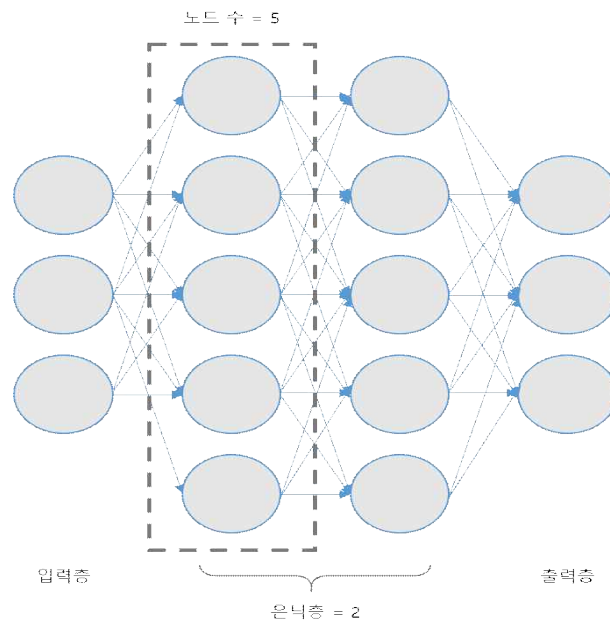
딥러닝 모델의 성능향상을 위해 모델의 학습조건을 설정하는 값으로서 경험적인 지식이나 임의 선정, 혹은 반복적인 실험의 통행을 최상의 모델을 구성하기 위한 값이다(Reimers & Gurevych, 2017).

하이퍼파라미터는 신경망 네트워크 구조와 관련된 하이퍼파라미터와 학습 알고리즘과 관련된 하이퍼파라미터로 나눌 수 있다.

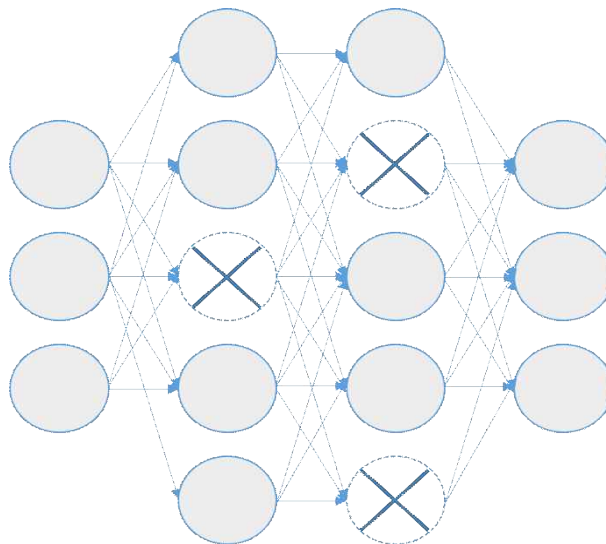
신경 네트워크 기법은 은닉층의 수와 노드(유닛) 수의 조정, 드롭아웃 기법의 적용, 신경망의 가중치 값에 대한 초기화 등이 있으며, 학습 알고리즘 기법은 학습률, 배치 크기, 정규화 강도 등이 있다(Reimers & Gurevych, 2017; Hossain et al., 2020).

은닉층의 개수는 [그림 2-10]에서 볼 수 있듯이 입력층과 출력층 사이에 있는 은닉층의 개수를 의미하고 노드의 개수는 각 층에 있는 뉴런, 유닛이라고도 불리는 뉴런의 개수를 의미하는데 노드의 개수와 은닉층의 개수는 학습 시 발생할 수 있는 과소 적합(Under Fitting)이나 과적합(Over Fitting)에 영향을 주는 것으로 알려져 있다(조경우 외, 2021).

드롭아웃 기법은 과적합을 방지하기 위한 정규화 기법의 하나이다. 학습데이터에 과적합 되도록 학습된 모델을 실제로 사용하기 위해서는 일반적인 형태의 모델이 될 필요가 있는데 이럴 경우, 드롭아웃 기법을 적용하면 [그림 2-10]에서 보이는 바와 같이 은닉층과 각 층의 노드를 주어진 비율로 배제하여 학습하게 된다. 드롭아웃 기법을 적용할 때 설정하는 비율이 낮을 경우 노드나 은닉층의 수를 많이 배제할 수 없어서 효과가 작아지고, 큰 비율을 적용할 경우 과소 적합의 우려가 있는 것으로 알려져 있다(최희열 & 민윤홍, 2015).



[그림 2-10] 은닉층과 각 은닉층의 노드 수



[그림 2-11] 드롭아웃 기법 예시

가중치 초기화 기법은 각 은닉층에서 사용하는 활성화 함수에 따라 가중

치 초기화 기법을 사용하여 가중치를 적절하게 초기화하는 기법이다. 일반적으로 가중치 초기화 시에는 균등분포의 난수를 이용한다(Kumar, 2017; Kwak et al., 2020).

정규분포란 가우시안 분포(Gaussian Distribution)라고도 불리며, 자연 현상에서 관찰되는 값들이 보통 이 분포를 따르는 것으로 알려져 있다. 정규분포는 평균과 표준편차를 기반으로 모양이 결정되며, 평균을 중심으로 좌우 대칭의 종 모양을 이룬다. 균등분포는 확률 변수가 동일한 확률로 일정한 범위에서 발생하는 분포다. 예를 들어, 동전 던지거나 주사위 던지기와 같이 결과값이 균등하게 발생하는 경우 이 분포를 따른다. 균등분포는 간단한 분포로, 직사각형 모양을 이룬다. 많이 알려진 가중치 초기화 기법을 [표 2-6]에 나타내었다.

[표 2-6] 가중치 초기화 기법

구분	특징	입력 파라미터
세이버 (Xavier) 초기화	<ul style="list-style-type: none"> • Uniform 균등분포 / Normal 정규분포 방식이 존재 • (이전 층의 뉴런갯수 ÷ 다음 층의 뉴런갯수)를 이용하여 분포 결정 • 특정 층이 과도하게 주목받는 것을 방지 • Sigmoid와 같은 S자 형태의 활성화 함수에 유용(ReLU에는 불리) 	<ul style="list-style-type: none"> • glorot_normal • glorot_uniform (default)
He 초기화	<ul style="list-style-type: none"> • Uniform / Normal 분포 두 방식이 존재 • 이전 층의 뉴런 갯수를 이용하여 분포 결정 	<ul style="list-style-type: none"> • he_uniform • he_normal

다음으로는 학습 알고리즘과 관련한 하이퍼파라미터들이 있는데 학습률(Learning Rate), 최적화(Optimizer) 기법, 학습 반복수(Epoch), 학습 배치(Batch)의 크기, 비용함수(Cost Function) 등이 있다.

학습률은 기울기 값을 통해 가중치를 얻고자 할 때 어느 정도 많이 이동하여 새로운 기울기 값을 찾을 것인지를 결정하는 변수이다. 학습률을 너무 작게 설정하면 학습 속도가 느려지고, 반대의 경우는 학습이 체대

로 이뤄지지 않는다(Takase et al., 2018).

최적화는 기울기 값을 찾고자 할 때 적용하는 기법으로 최적화 기법에 모델의 성능이 영향을 받는다. 대표적인 최적화 기법으로는 SGD(Stochastic Gradient Descent), RMSProp(Tieleman & Hinton, 2012), Adam(Kingma & Ba, 2014) 등의 기법이 있으며 각 기법을 개선한 다양한 기법들이 존재 한다(Yazan & Talu, 2017).

학습 반복 수란 모델이 전체 데이터를 몇 번 학습할 것인가를 결정하는 수이며 한 번의 학습은 전체 데이터에 대해 순전파(Feed Forwarding)와 역전파(Back Propagation)를 거친 것을 의미한다. 반복적으로 학습을 많이 하면 오차가 낮아질 것으로 예측할 수 있지만, 검증 단계에서는 오차가 감소하다가 어느 순간부터 다시 증가할 수도 있다. 이 현상을 과적합(Over Fitting)이라고 하는데 이러한 이유로 학습 반복 횟수의 경계 지점을 찾는 것이 중요하며, 일반적으로 조기 종료(Early Stopping) 기법을 적용하여 과적합을 방지한다(Caruana et al., 2001).

학습 배치의 크기는 한 번의 배치(일괄 처리)마다 입력되는 데이터 샘플의 크기를 말하는데 전체 데이터를 한 번에 학습하기에는 하드웨어 시스템에 부담이 되거나 시간상의 제약으로 전체 데이터를 일정 크기로 나눠 학습할 때 설정하는 값이다. 여기서 나뉜 일정 크기가 학습 배치 크기가 되는데 보통 배치를 미니 배치라고도 표현하며 나뉜 데이터를 의미한다. 학습 배치의 크기는 어떤 크기로 데이터를 나눌지 결정하는 크기이며, 미니 배치는 같은 크기로 전체 데이터를 나눈 데이터들이다. 또한, 학습 배치의 크기로 나뉜 미니 배치를 전부 다 학습하여 전체 데이터를 1 epoch 학습하는데 실행한 횟수를 이터레이션(Iteration)이라고 한다. 이를 간략한 수식으로 표현하면 다음과 같다.

전체 데이터셋 = 학습 배치의 크기 * 미니 배치

1 반복 = 배치 크기 * 이터레이션

비용함수와 관련한 하이퍼파라미터는 정답값과 예측값의 차이를 구하는 여러 가지 방법들을 의미하는데, 대표적인 방법으로는 MSE(Mean Squared Error)를 최소화하는 최소자승법(Least Mean Square)이나 크로스엔트로피(Cross Entropy) 등이 있다(Kline & Berardi, 2005).

임베딩 차원은 특별히 텍스트 데이터를 입력데이터로 사용할 때 중요한 하이퍼파라미터이다. 케라스 라이브러리에서 임베딩 층(Embedding Layer)을 사용할 때 입력되어야 하는 값이며 임베딩 차원의 수에 따라 텍스트의 의미가 잘 내포되었는지가 결정되는 하이퍼파라미터 값이다(이시영, 2021). [표 2-7]에 하이퍼파라미터의 기법과 특징을 정리하였다.

[표 2-7] 하이퍼파라미터의 종류와 특징

구분	기법	특징
신경 네트워크 관련	은닉층/노드의 수	• 신경망의 은닉층 개수와 은닉층에 포함된 노드의 개수를 조정하여 성능을 높이는 기법
	드롭아웃	• 은닉층의 수와 각 은닉층의 노드를 주어진 비율로 배제하고 학습하는 기법
	가중치 초기화	• 은닉층의 활성화 함수에 따라 가중치의 초기값을 균등분포 난수로 초기화하는 기법
학습 알고리즘 관련	학습률	• 기울기 값으로 가중치를 얻고자 할 때, 어느 정도 이동하여 새로운 기울기 값을 찾을 것인지를 결정하는 변수
	최적화	• 기울기 값을 찾고자 할 때 적용하는 기법으로 최적화 기법의 선정은 모델의 성능에 영향 미침
	학습 반복수	• 전체 데이터셋을 몇 번 학습할 것인지를 결정하는 변수 • 조기 종료(Early Stopping) 기법을 적용하여 반복 학습을 중단하여 과적합 방지
	학습 배치의 크기	• 데이터셋을 어떤 크기로 나누어 학습할지를 결정하는 크기
	비용함수	• 정답값과 예측값의 차이를 구하는 함수를 의미
임베딩 관련	임베딩 차원	• 학습 대상 텍스트를 몇 차원으로 임베딩할 것인지를 결정 • 차원 수에 따라 임베딩의 품질이 결정

2.3 자연어 처리(NLP)

자연어 처리(Natural Language Processing, NLP) 연구 분야는 인간 언어분석과 표현(Representation)을 자동화하기 위하여 상대적으로 낮은 수준의 복잡성을 가지며 선형 분리 문제에 적합한 Shallow Models(선형 회귀, 로지스틱 회귀, 서포트 벡터 머신, 나이브 베이즈 등)을 중심으로 진행되어 왔다. 하지만, Shallow Models를 사용한 기계학습에 기반을 둔 자연어 처리 시스템은 사람의 개입을 통한 특징 추출에 의존하여 시간이 많이 소요되고, 추출된 특징이 불완전하였다.

자연어 처리를 위한 임베딩(Embedding)은 기계가 문장 속 단어의 의미를 문맥으로 구분하고 이해할 수 있도록, 단어를 0과 1의 수치로 표현하는 방법으로 벡터화(Vectorization)라고도 한다(Sohrabi et al., 2018).

Collobert et al.(2011)은 간단한 딥러닝 프레임워크를 제시했는데, 이 프레임워크는 개체명 인식(Named Entity Recognition, NER), 의미역 결정(Semantic Role Labeling, SRL), 품사 태깅(POS Tagging) 같은 일부 자연어 처리에 사용되고 있으며, 그 이후 합성곱 신경망(CNN), 순환 신경망(RNN), 재귀 신경망(Recurrent Neural Network, RNNR) 등 다양한 딥러닝 기반 알고리즘에서도 사용되고 있다.

자연어 처리 기법의 발전 흐름에 따라 단어의 출현 빈도와 분포를 파악하는 데 장점이 있는 통계적 기반 임베딩과 단어 간의 의미적 유사성 및 문맥 정보를 반영하는데 뛰어난 성능을 보이는 신경망 기반 임베딩으로 구분할 수 있다. 통계적 기반 임베딩은 단어의 출현 빈도와 분포를 기반으로 학습하기 때문에 많은 통계적 정보를 활용할 수 있다. 이는 대규모 텍스트에서 단어 간의 유사성을 파악하는 데에 큰 도움이 되며, 비교적 적은 계산 리소스로도 효율적인 임베딩을 생성할 수 있는 장점이 있다. [표 2-8]은 통계적 기반 임베딩 기법에 대하여 정리한 것이다.

[표 2-8] 통계적 기반 임베딩 기법

임베딩 기법	설명
TDM (Term-Document Matrix)	<ul style="list-style-type: none"> · 문서 집합을 표현하는 행렬로 단어 빈도수 정보 포함 · 장점 : 문서 간의 비교 분석, 텍스트마이닝 작업에 활용 · 단점 : 희소성 문제, 단어의 순서 정보 무시, 단어 구분 불가
TF-IDF (Term Frequency-Inverse Document Frequency)	<ul style="list-style-type: none"> · TF-IDF가 높은 단어가 문서에서 중요도가 높다고 간주 문서의 핵심어 추출 및 검색 결과 우선순위 결정에 이용 · 장점 : 단어 중요도 고려, 희소성 문제 완화 · 단점 : 단어 순서 정보 무시로 문맥적 의미 불가
One-hot Encoding	<ul style="list-style-type: none"> · 범주형 변수를 이진 벡터로 표현하는 방법 · 장점 : 범주형 변수 수치화 가능, 독립적 처리 · 단점 : 벡터 차원 증가 및 단어 간 유사도 파악 불가

신경망 기반 임베딩은 텍스트나 이미지와 같은 다차원 데이터를 저차원 공간에 표현하는 방법이며, 이러한 저차원 표현은 원본 데이터의 중요한 특징을 보존하면서 데이터를 밀집 벡터로 변환한다.

신경망 기반 임베딩은 NPLM(Neural Probabilistic Language Model)이 발표된 이후부터 더욱 주목받기 시작했으며, NPLM은 문맥 단어를 입력으로 받아 현재 단어의 등장 확률을 예측하며, 이를 위해 단어를 밀집 벡터로 표현하여 단어 간의 의미적 관계를 반영하는 데에 뛰어난 성능을 발휘한다.

[표 2-9]는 신경망 기반 임베딩 기법에 대하여 정리한 것이다.

[표 2-9] 신경망 기반 임베딩 기법

임베딩 기법	설명
Word2Vec (Mikolov et al., 2013)	의미적인 성질이 유사한 단어들은 벡터 공간상에서 유클리디안(Euclidean)나 코사인 유사도(Cosine Similarity)에 가까운 벡터들로 표현되는 단어 간 유사도를 반영하고 단어를 벡터화할 수 있는 방법
FastText (Kuyumcu et al., 2019)	단어를 개별 단어가 아닌 N-gram의 Characters(Bag-of Characters)를 적용하여 임베딩 함으로써 하나의 단어를 여러 개로 잘라서 벡터로 계산하는 방식
ELMo (Peters et al., 2018)	ELMo(Embeddings from Language Model) / 새로운 워드 임베딩 방법론으로 사전 훈련된 언어 모델(Pre-trained Language Model)을 사용

2.4 텍스트 임베딩(Text Embedding)

텍스트 임베딩(Text Embedding)은 텍스트 마이닝(Text Mining)이나 자연어를 처리하는 분야에서 활용되고 있는 기법으로 하나의 단어 또는 문장을 인공 신경망을 적용하여 숫자 형태의 벡터로 변환하는 작업을 의미한다. 기존에는 단어 자체를 아스키코드(ASCII)나 유니코드(Unicode)로 처리해서 사용했지만, 이러한 숫자 코드만을 가지고는 단어의 실제적인 의미를 추론하기는 어렵다. 예를 들어 ‘왕’과 ‘여왕’이 관련이 있는 단어이고 ‘왕’의 성별은 남성이고 ‘여왕’의 성별은 여성이라는 사실을 기존의 숫자 코드만을 가지고는 알아내기 어렵다(Li et al., 2015).

텍스트 임베딩의 연구는 텍스트 마이닝의 문제점을 개선하기 위하여 진행된 것이라 할 수 있으며, 텍스트 마이닝은 자연어를 컴퓨터가 이해할 수 있는 벡터(Vector)로 변환되도록 설계된 알고리즘을 통하여 의미 있는 정보를 추출하고 분석하는 분야다. 응용 분야로는 문서 분류(Document Classification), 문서 군집(Document Clustering), 정보 추출(Information Extraction), 문서 요약(Document Summarization) 등이 있다. 하지만, 대용량 텍스트의 경우 차원이 높고 희소성 문제가 있어서 단어의 의미를 잘 파악하지 못하고 문맥을 고려하지 못하는 한계가 있다.

텍스트 임베딩은 이러한 텍스트 마이닝의 한계를 극복하기 위해 개발된 기술이며, 단어나 문장을 저차원의 실수 벡터로 표현하여 의미와 문맥을 더욱 잘 이해할 수 있도록 한다. 또한, 사전 훈련된 임베딩 모델을 사용하여 단어의 의미적 유사성과 관련성을 반영하는 벡터 표현을 생성하고, 이를 활용하여 텍스트 마이닝에 적용하여 성능을 향상할 수 있다.

텍스트 임베딩은 NPLM을 비롯한 다양한 알고리즘과 모델을 통해 발전해왔으며, Word2Vec은 NPLM의 개선된 형태로, 단어를 예측하기 위해 문맥 단어를 사용하는 Skip-Gram과 CBOW(Continuous Bag of Words) 모델로 확장되었다.

2.4.1 워드투벡터(Word2Vec)

하만석(2019)은 Word2Vec는 신경망 기반의 텍스트 임베딩 방법의 하나로, 주변 단어를 통해 그 단어를 유추할 수 있다는 개념에서 착안한 알고리즘이라 하였다. 또한, Turney & Pantel(2010)는 기본적으로 비슷한 단어들은 동일한 문맥에서 등장하고 비슷한 분포를 가진다는 가정을 하였으며, 김우주 외(2016)는 이러한 가정하에 Word2Vec는 신경망을 이용하여 단어 간 유사도를 표현할 수 있도록 각 단어 자체의 의미를 벡터로 표현하는 방법론으로 이를 응용하여 텍스트에 내재한 복잡한 개념을 표현하거나 단어 간의 연관 관계를 추론할 수 있고 하였다.

Word2Vec은 CBOW 기법과 Skip-Gram 기법으로 개별 단어를 벡터로 표현한다. CBOW는 주변의 단어들로부터 가운데 단어의 출현을 예측하는 방법이며, Skip-Gram은 가운데 단어로 주변 여러 단어를 예측하는 모델을 만들어 학습시킨 뒤 단어들에 입력된 가중치를 해당하는 단어의 연속적인 벡터값으로 사용한다(Mikolov et al., 2013).

2.4.2 도큐먼트투벡터(Doc2Vec)

Doc2Vec은 개별 단어의 관계를 학습하는 Word2Vec과 유사하지만, 범위를 넓혀서 문장 또는 문서 자체를 학습하는 방법으로 Doc2Vec은 각각의 문서 또는 문장을 하나의 벡터로 변환하며, 변환된 각각의 벡터들은 문장 또는 문서 내부의 단어들을 예측하기 위해 기계학습에 사용된다.

Word2Vec은 개별 단어의 관계를 학습하는데 주안점을 둔 알고리즘이고 Doc2Vec은 문장, 문단, 문서처럼 더 큰 블록에 대한 연속적인 표현을 비지도 학습 방식으로 모델을 생성하는 알고리즘이다(Le & Mikolov, 2014).

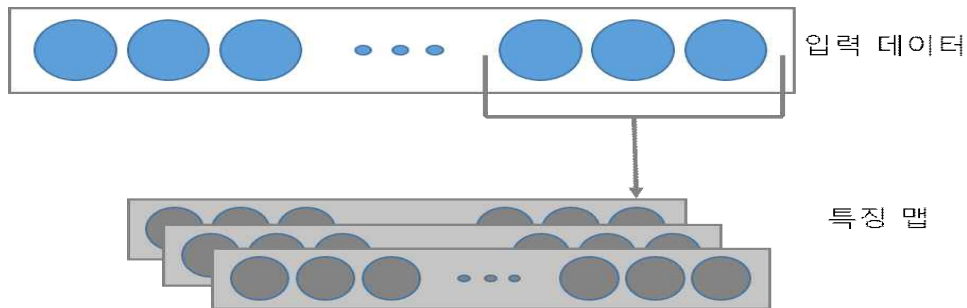
Word2Vec은 개별 단어의 순서와 뜻을 파악하기에 어려움이 있지만,

Doc2Vec은 개별 단어의 순서와 뜻을 포함해서 벡터로 변환할 수 있다. 즉, 각각의 문서에 같은 단어를 사용하지만, 단어의 순서가 다르면 다른 벡터를 만들어 내고, 비슷한 뜻을 가진 각각의 문서들을 벡터 공간상에서 가까운 거리에 위치하도록 벡터를 만들게 된다. 이렇게 만들어진 벡터를 이용하여 문서 분류를 하면 Word2Vec 모델을 적용했을 때보다 더 좋은 성능을 얻을 수 있다(유용민, 2018).

2.5 1차원 합성곱 신경망(1D-CNN)

합성곱 순환 신경망(Convolutional Neural Network, CNN)은 인간의 시신경을 모방하여 만든 심층 학습 구조 중 하나로서 이미지 처리와 패턴 인식에 주로 사용되는 신경망 구조이다. 주요 구성 요소로는 합성곱 계층(Convolutional Layer), 풀링 계층(Pooling Layer), 완전 연결 계층(Fully Connected Layer) 등이 있다.

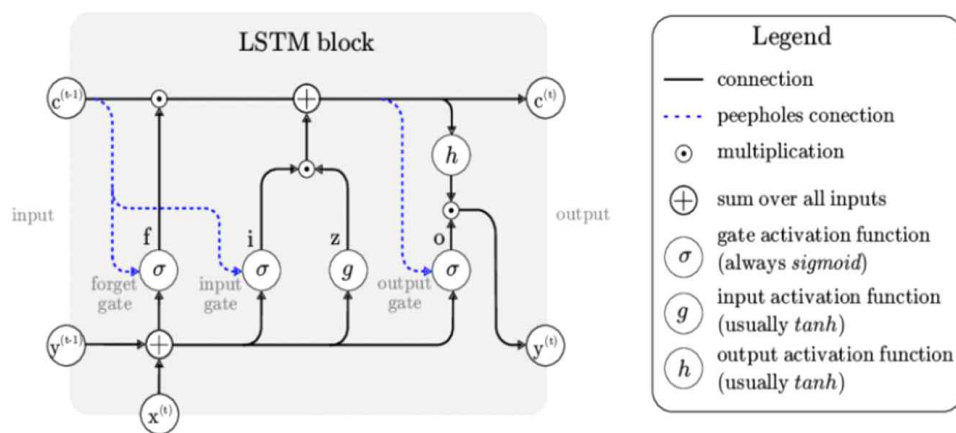
인간의 시신경 뉴런들은 작은 수용영역(Receptive Field)을 가지고 있는데 여러 수용영역이 겹쳐서 이를 합치면 전체 시야를 감싸는 구조를 갖게 된다. 수용영역은 입력 이미지의 특정 부분에 대한 반응 영역으로서, 계층 구조를 통해 저수준에서 고수준 특징을 추출할 수 있다. 수용영역 크기는 작업과 요구되는 특징에 따라 조정된다. 이와 비슷하게 1차원 합성곱 신경망(1D-CNN)은 일반적인 CNN에 스펙트럼 데이터와 같은 1차원의 데이터와 필터가 사용된다. 1D CNN의 합성곱층은 [그림 3-7]과 같이 필터(Filter)의 형태가 1차원으로 이루어져 있다. 필터가 이동하면서 한 수용영역 안의 내용과 합성곱 층의 뉴런을 연결한다. 입력된 스펙트럼에서 많은 특징을 잡아가며 회귀 분석을 진행하게 된다. 1D CNN을 사용하면 순환 신경망과 같은 다른 심층 학습 모델을 사용할 때보다 연산량이 적지만 비슷하게 높은 성능을 유도할 수 있는 이점이 있다.



[그림 2-12] 1D CNN Structure

2.6 장단기 메모리(LSTM)

LSTM(Long Short-Term Memory)은 RNN 기반의 순환 신경망으로써 시계열처럼 순서를 갖는 데이터 분석에 적합한 알고리즘이다.



[그림 2-13] LSTM 구조

LSTM은 Hochreiter & Schmidhuber(1997)에 의해 제안된 알고리즘으로, RNN에서 발생하는 기울기 소멸(Vanishing Gradient) 문제를 해결하고자 장기 의존성(Long-Term Dependency)을 모델링할 수 있도록 설계된 딥러닝 모델이다. Vanishing Gradient는 과거의 데이터 값을 현재로

가져와 계산하게 되는데 현시점에서 멀수록 값이 소실하게 되는 현상이다. 따라서 RNN은 계산하고자 하는 시점이 먼 과거의 어떠한 값으로 인해 영향을 받게 되는지 산출하기 쉽지 않다. 이를 보완한 방법이 LSTM이며, LSTM의 은닉층에 Gate를 추가하여 과거의 데이터를 예측값 산출에 소실되지 않도록 하는 기법이다.

LSTM은 크게 Forget Gate, Input Gate, Cell State, Output Gate로 구분할 수 있다. 우선 Forget Gate에서는 이전 시점에서 전달된 것과 현시점의 Input을 고려하여 어떤 값을 망각하게 될지를 선택한다. Input Gate에서는 Cell State에 인풋을 할지 안 할지를 선택하며, 하게 된다면 어느 강도로 인풋을 하게 될지를 정하게 된다. 그리고 산출된 Forget Gate와 Input Gate를 사용하여 Cell State 을 갱신하며, 최종적으로 Output을 계산한다. LSTM을 사용하여 다양한 예측 분야에 적용하고 있으며, 실제로 금융시장의 주가 예측에 활용하고 있었으며, 신경망 DNN을 활용하여 예측한 값보다 큰 정확도를 나타내고 있다(최영웅, 2022).

2.7 트랜스포머(Transformer)

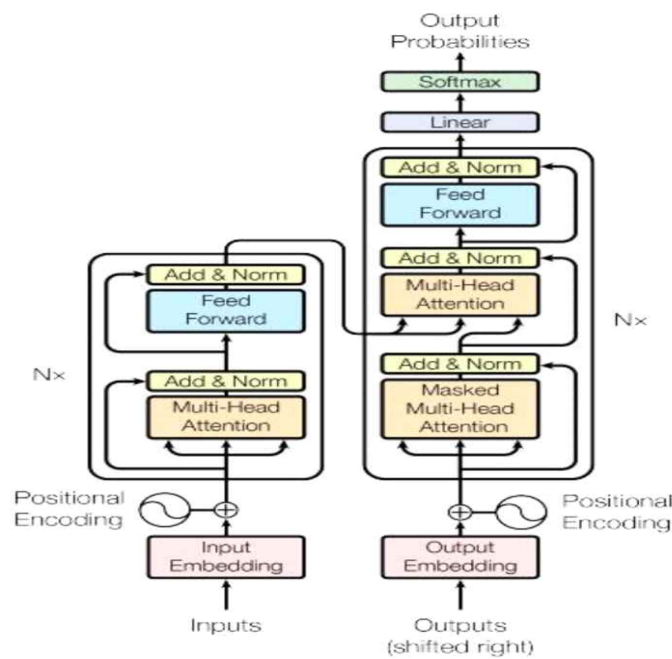
트랜스포머(Transformer)는 입력과 출력 시퀀스 사이의 양방향 정보 교환을 가능하게 하여, 문장의 의미를 파악하고, 번역이나 요약과 같은 작업을 수행한다. Transformer의 주요 구성 요소로는 어텐션(Attention), 셀프 어텐션(Self Attention), 인코더(Encoder), 디코더(Decoder)가 있다.

어텐션은 입력 시퀀스의 각 위치에서 다른 위치의 정보에 가중치를 부여하여 중요한 정보를 집중하도록 하는 메커니즘이며 셀프 어텐션은 입력 시퀀스 내 다른 위치의 정보 간의 관련성을 계산하는 데 사용하여 문장 내의 단어 간의 의미적인 연결을 파악한다.

인코더는 입력 시퀀스를 처리하는 부분으로, 다수의 층으로 구성되어

있다. 각 층은 어텐션과 피드포워드 신경망(Fully Connected Neural Network)으로 구성되어 입력 시퀀스의 특징을 추출한다. 디코더는 출력 시퀀스를 생성하는 부분으로서 인코더와 비슷한 구조를 가지고 있지만, 더불어 인코더의 출력에 셀프 어텐션을 추가로 활용하여 입력과 출력 간의 상호 의존성을 고려한다.

Transformer는 LSTM보다 병렬 처리가 용이하며, 긴 시퀀스 처리에도 효과적이며, 층이 쌓이는 구조로 인해 깊은 네트워크를 구축할 수 있고, 복잡하고 풍부한 문맥 정보를 다룰 수 있다. 특히, 번역, 요약, 질문 응답 등의 자연어 처리 분야에서 혁신적인 모델 주목받고 있으며, 다양한 언어 처리 작업에서 우수한 성능을 보여준다.



[그림 2-14] 트랜스포머 모델 구조

[그림 2-14]는 트랜스포머 모델 구조를 보여주며, 트랜스포머 모델은

기본적으로 인코더-디코더 구조로 되어 있으며, 큰 특징 중 하나는 셀프 어텐션을 수행한다는 것이다. 트랜스포머 모델은 기계 번역 시스템에서 악명 높은 문제인 상호 참조를 셀프 어텐션을 통해 해결하였다(오소진, 2022).

2.8 어텐션 매커니즘(Attention Mechanism)

Transformer와 LSTM은 모두 어텐션 매커니즘(Attention Mechanism)을 활용하여 시퀀스 데이터를 처리하고, 주변 단어 간의 상호작용을 모델링하며 문맥을 이해하는 능력을 향상시킨다. 어텐션은 각 시퀀스 요소에 가중치를 할당하고, 이를 통해 관련 있는 정보에 더 많은 영향력을 부여한다.

LSTM과 같은 순환 신경망 모델은 순차적으로 데이터를 처리해야 하는 특성 때문에 동시에 여러 작업을 수행하지 못하여, 병렬처리의 효율이 떨어지게 된다. 이를 해결하기 위해 CNN(합성곱 신경망)이 사용되기도 했는데, CNN을 사용하는 경우에도, 입력 계층의 길이가 길어질수록 입력 원소와 출력 원소 사이의 의존성(Dependency) 학습이 어려워진다는 문제가 있다. 또한, 입력 문장이 긴 상황에서 자연어 처리나 번역의 품질이 떨어지는 현상이 나타났고, 이러한 현상을 보정하기 위해 바로 중요한 단어에 집중하는 어텐션 메커니즘이 자연어 기계 번역을 위한 Seq2Seq(Sequence-To-Sequence) 모델에 처음 도입되었다(Sutskever et al., 2014; Cho et al., 2014).

어텐션 메커니즘은 Seq2Seq 모델이 디코딩 과정에서 현재 스텝에서 가장 관련된 입력 부분에 집중할 수 있도록 해줌으로써 기계 번역의 품질을 크게 향상했다. 또한, 다양한 작업에서 강력한 시퀀스 모델 및 변환 모델의 필수적인 부분이 되었으며, 입력과 출력의 원소들 사이의 거리와

무관하게 의존성을 학습할 수 있게 되었다(Bahdanau et al., 2015; Kim et al., 2017).

대부분의 어텐션은 기존의 신경망과 함께 사용됐으나 Vaswani, A., et al.(2017)의 연구에서는 기존 순환 신경망에서 발생하는 문제들을 해결하고 셀프 어텐션(Self-Attention)을 이용하는 Transformer를 제안하였는데, 이 연구에서는 반복(Recurrence)을 제거함으로써 오직 어텐션 메커니즘만을 이용해서 입력과 출력 사이의 전역 의존성(Global Dependency)을 학습하게 된다. 반복 작업의 제거로 더 손쉽게 병렬 처리가 가능하고 훈련 시간을 단축하였다.

셀프 어텐션은 독해, 추상적 요약, 텍스트 수반, 학습과제 독립적인 문장 표현을 포함한 다양한 과제에서 성공적으로 사용되었다(Cheng et al., 2016; Parikh et al., 2016; Paulus et al., 2017; Lin et al., 2017).

2.9 성능평가 방법

모델의 성능평가는 실제 값과 예측값의 오차를 구하는 과정으로써 오차가 작을수록 모델이 예측을 정확히 수행했다고 할 수 있다. 하지만 실제값과 예측값이 완전히 일치하는 것은 불가능하기 때문에 일정 수준의 오차를 허용하여 얼마나 정확한 예측을 수행하는지 판단한다.

모델 평가는 과적합(Overfitting)을 방지하고 최적의 모델을 찾기 위해 수행되며, 모델의 목적이나 목표 변수의 유형에 따라 다양한 평가 지표를 사용할 수 있다. 일반적으로는 Training data와 Validation data 간의 차이를 확인하여 과적합 여부를 판단한다. Training data는 좋은 성능을 보이지만, Validation data에서 성능이 크게 저하되면 모델이 과적합된 상태라고 판단할 수 있다. 예측 모델과 분류 모델에 따른 평가 지표를 통해 모델의 성능을 수치로 측정하고 비교할 수 있으며, 모델 간의 비교

를 통하여 오차가 작은 모델이 더 정확한 예측을 수행한다고 판단할 수 있다. [표 2-10]은 모델링 목적에 따른 평가 방법을 정리하였다.

[표 2-10] 모델링 목적과 변수 유형에 따른 평가 방법

모델링의 목적	변수 유형	관련 모델	평가 지표
예측/회귀(Prediction)	연속형	-선형 회귀	MSE, RMSE, MAE, MAPE
분류(Classification)	범주형	-로지스틱 회귀 -의사결정나무 -서포트벡터머신	정확도, 정밀도, 재현율, F1-score

2.9.1 예측 모델 성능평가

예측/회귀 모델의 평가 지표로는 평균 제곱 오차(Mean Squared Error, MSE), 평균 제곱근 오차(Root Mean Squared Error, RMSE), 평균 절대 오차(Mean Absolute Error, MAE), 평균 절대 비율 오차(Mean Absolute Percentage Error, MAPE) 등이 일반적으로 사용된다.

이러한 평가 지표는 예측값과 실제값의 오차를 측정하여 모델의 성능을 평가하며, 오차 값은 작을수록 모델의 성능이 좋다는 것을 의미한다.

[표 2-11]은 예측 모델의 성능평가 방법에 대하여 정리한 것이다.

[표 2-11] 예측 모델 성능평가 방법

지표	정의
평균 제곱 오차 MSE(Mean Squared Error)	실제 값과 예측값의 차이를 제곱해 평균한 것으로 오차의 제곱을 평균화하여 평가
평균 제곱근 오차 RMSE(Root Mean Squared Error)	MSE 값의 제곱근이며, 오차의 크기를 실제 값의 단위와 일치시켜 해석하기 쉽게 함.
평균 절대 오차 MAE(Mean Absolute Error)	실제 값과 예측값의 차이를 절대 값으로 변환해 평균한 것으로 오차의 크기를 평균하여 평가함.
평균 절대 비율 오차 MAPE(Mean Absolute Percentage Error)	실제 값과 예측값의 차이를 실제 값의 백분율로 변환하여 평균함. (상대적 오차를 통해 MSE, RMSE의 단점을 보완)

2.9.2 분류 모델 성능평가

분류 모델의 평가 방법은 예측 모델 평가(회귀 모형)와 같이 실제값과 예측값의 차이에 기반하지만, 업무 특성과 요구사항에 맞게 더 중요한 평가 지표를 선택하여 결정한다.

분류 모델 평가에는 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1-Score, ROC-AUC, 오차 행렬(Confusion Matrix)과 같은 지표가 일반적으로 사용된다. 정확도(Accuracy)는 전체 예측 중 올바르게 예측한 비율을 나타내는 지표이며, 모델의 전체적인 성능을 평가하는 데 사용된다. 정확도가 높을수록 모델의 예측이 정확하다고 해석할 수 있다. 정밀도(Precision)는 정답이라고 예측한 샘플 중 실제로 정답인 샘플의 비율을 나타내는 지표이며, 모델이 정답으로 예측한 것 중 실제로 정답인 것을 얼마나 정확하게 예측하는지를 나타낸다. 정밀도는 잘못된 정답 예측을 최소화하는 데 중요한 역할을 한다. 재현율(Recall)은 실제로 정답인 샘플 중 모델이 정답으로 정확하게 예측한 샘플의 비율을 나타내는 지표이며, 모델이 실제 정답인 것을 얼마나 잘 찾아내는지 나타낸다. 재현율은 놓치면 안 되는 정답 샘플을 놓치지 않도록 하는 데 중요한 역할을 한다. F1-Score는 정밀도와 재현율의 조화 평균으로 계산되는 지표이며, 정밀도와 재현율은 서로 상충하는 관계를 나타낸다. F1 스코어는 이 둘을 균형 있게 고려한 평가 지표로 사용되며, 모델의 정확도와 재현율을 동시에 고려하는 데 유용하다. ROC(Receiver Operating Characteristic) 곡선은 재현율과 특이도의 관계를 시각적인 그래프로 나타낸 것이며, 이렇게 표현한 ROC 곡선의 아래쪽 영역을 AUC(Area Under Curve)라 한다. ROC 곡선은 모델의 분류 임계값을 변화시켰을 때, 재현율과 거짓 양성 비율(False Positive Rate) 사이의 관계를 나타낸다. AUC-ROC는 분류 모델의 성능을 평가하고, 클래스 간의 분리도를

측정하는 데 사용된다. AUC-ROC 값이 1에 가까울수록 모델의 성능이 우수하다고 해석할 수 있다. 오차 행렬(Confusion Matrix)은 분류 모델의 예측 결과와 실제 레이블 사이의 관계를 나타낸다. 주로 2x2 형태로 구성되는 행렬이며, 예측한 클래스와 실제 클래스에 따라 True Positive(TP), False Positive(FP), False Negative(FN), True Negative(TN)로 분류된다. 오차 행렬을 통해 모델의 성능을 분석하고, 정확도, 정밀도, 재현율 계산할 수 있다. [표 2-12]는 분류 모델의 성능평가 방법에 대하여 정리한 것이다.

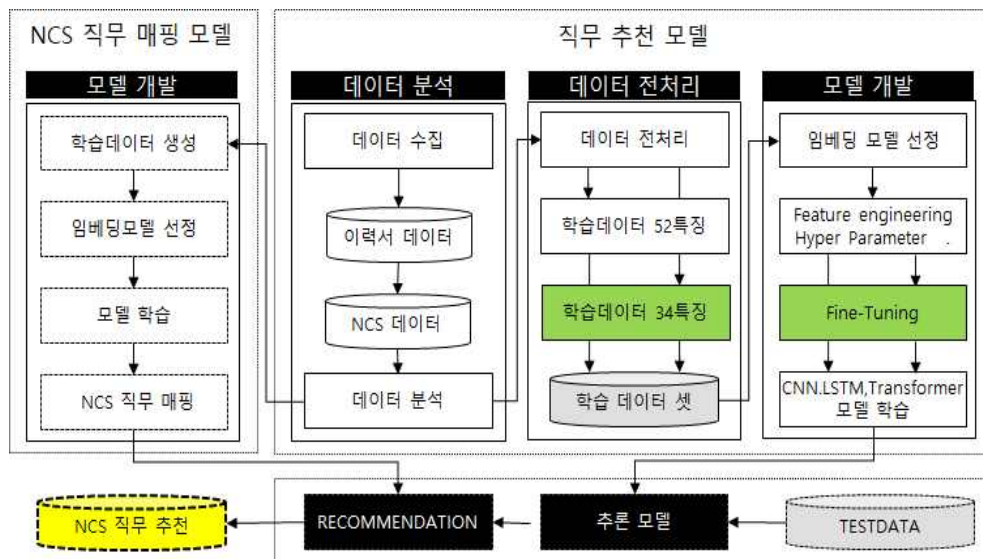
[표 2-12] 분류 모델 성능평가 방법

지표	정의
정확도 (Accuracy)	· 전체 예측 중 올바르게 예측한 비율 · (올바른 예측 수) / (전체 예측 수)
정밀도 (Precision)	· 예측한 양성 클래스 중 실제 양성 클래스의 비율 · (진짜 양성 수) / (진짜 양성 수 + 거짓 양성 수)
재현율 (Recall)	· 실제 양성 클래스 중 예측한 양성 클래스의 비율 · (진짜 양성 수) / (진짜 양성 수 + 거짓 음성 수)
F1-Score	· 정밀도와 재현율의 조화 평균 · $2 * (\text{정밀도} * \text{재현율}) / (\text{정밀도} + \text{재현율})$
AUC-ROC	· 수신자 작동 특성 곡선 아래 영역 · ROC 곡선 아래 면적을 계산하여 0~1 사이의 값을 가짐
오차 행렬 (Confusion Matrix)	· 예측 결과의 실제 클래스와의 관계를 나타내는 행렬 · 모델의 예측 결과와 실제 레이블을 비교하여 생성됨

제 3 장 연구모형

3.1 직무 추천 모델

본 연구에서는 구직자에 맞춤형 직무 추천 시스템을 구축하기 위해 텍스트 임베딩을 적용하여 자연어로 구성된 이력서 데이터를 Embedding Vector로 추출하여 분석하였다. 이를 통해 직무 간 유사성 및 이를 활용한 딥러닝 기반 직무 추천 모델을 연구하였다.



[그림 3-1] 직무 추천 시스템 연구 모형

본 연구는 총 7단계의 실험 프로세스로 구성하였으며 52개 항목으로 구성된 이력서 데이터를 딥러닝 기반의 임베딩 기법을 이용하여 독립 변수화한 후 종속 변수인 “JOB_CODE”를 통해 직무를 추천하는 것이다.

이를 위하여 데이터 전처리 및 학습 데이터셋 구축하고 Feature Engineering을 진행한다. 이후 CNN, LSTM, Transformer 모델들을 사

용하여 직무 추천 모델 학습하고 각 모델을 비교하여 정확도와 정밀도가 높은 모델을 선정하여 최종적으로 직무 추천 모델을 선정하였다.

[표 3-1] 프로세스 및 세부 내용

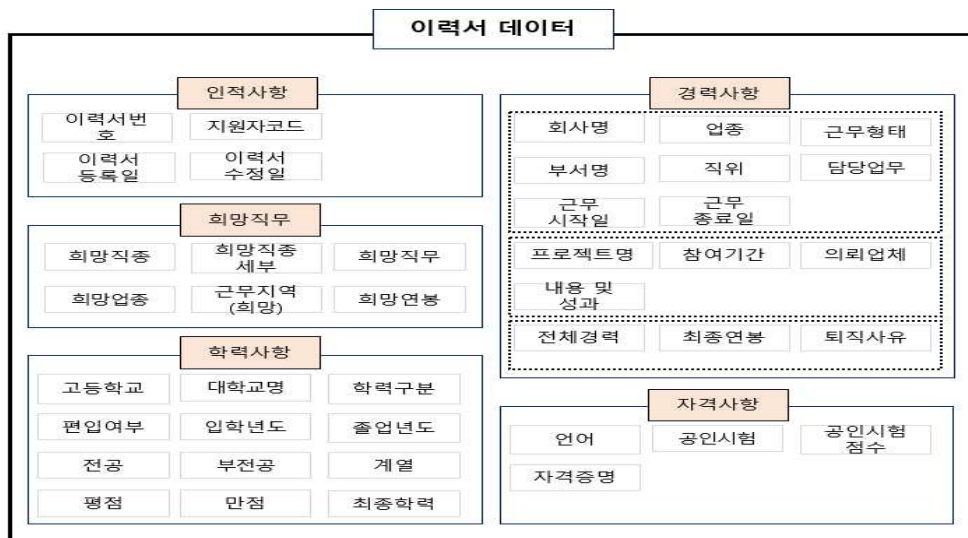
idx	프로세스	세부 내용
1	데이터 수집	실험을 위해 글로벌 채용 정보 사이트에서 사용자 정보, 경력정보, 학력정보, 직무정보, 자격정보를 포함하는 직무 지원 기록 데이터를 수집하였으며, 취업포털의 직무 분류 체계와 매핑하기 위해 NCS 직무 분류 체계의 데이터를 수집함
2	데이터 분석	해당 단계는 데이터 특성분석, 데이터 분포 분석, 상관관계 분석을 진행하여, 종속 변수 및 독립 변수에 가장 영향력이 있는 컬럼을 찾아서 전처리 및 학습 단계에서의 가중치를 다르게 할수 있도록 데이터 분석을 진행함
3	데이터 전처리	데이터 분석 단계에서 데이터를 분석 후 분석에 맞는 결측치 처리, 이상치 처리, 통합 전처리 과정을 거쳐 데이터 정제를 진행함
4	학습 데이터셋 구축	Resampling, 1:1 Negative Sampling을 활용하여 데이터 증강함
5	Feature Engineering	정규화 및 Scaling으로 입력데이터를 변환하며, 변수별로 나눠서 Feature Tuning을 진행함. 소범주형 변수, 다범주형, 자연어변수로 구분하여 해당 변수별로 기법을 다르게 사용하여 Feature 설정함. 이후 완성된 Feature들을 하나로 결합하고 완전 연결층(Fully Connected Layer)를 연결하여 최종 출력값을 추론할 수 있는 벡터 공간을 탐색 할 수 있도록 유도함
6	딥러닝 기반 모델 설정	데이터셋에 적합한 모델을 만들고, 모델을 이용하여 새로운 데이터를 어떻게 분류할지 예측한다. 본 예측 모델의 사용된 딥러닝 모델은 CNN, RNN, 트랜스포머, BERT를 사용하여 예측모델을 설계하여 실험을 진행함
7	모델 성능평가	본 예측 모델의 성능평가는 학습 성능의 측정 방법, 예측 성능의 측정 방법 두 가지로 나눠 측정한다. 두 가지의 성능측정은 텍스트 임베딩 기법을 이용한 딥러닝 학습과 빈도기반 텍스트 분석 방법에 동일하게 적용함

3.2 데이터 정의와 수집

취업포털로부터 연구목적으로 받은 총 370,824개의 이력서 데이터 중, 이상치 및 결측치를 제거한 구직자 298,465명의 이력서 데이터를 52개의 항목별로 전처리 완료한 학습데이터를 기준으로 연구를 진행하였다.

취업포털이 보유한 이력서 데이터를 익명화 처리하여 사용하였으며, 데이터 항목은 [그림 3-2]와 같이 인적사항, 희망직무 및 업종, 학력 사항, 경력 사항, 자격 사항 등 5가지로 구성하였다.

‘이력서 번호, 지원자 코드, 이력서 등록일, 이력서 수정일’은 인적사항으로 구분되며, ‘희망직종, 희망직종 세부, 희망직무, 희망근무 지역, 희망업종, 희망연봉’은 희망직무로 구분된다. ‘고등학교, 대학교, 학력구분, 편입여부, 입학연도, 졸업연도, 전공, 부전공, 계열, 평점, 만점, 최종학력’은 학력 사항으로 구분되며, ‘회사명, 업종, 근무형태, 부서명, 직위, 담당업무, 근무 시작일, 근무 종료일, 프로젝트명, 참여기간, 의뢰업체, 내용 및 성과, 전체경력, 최종연봉, 퇴직사유’는 경력 사항으로 구분된다. ‘언어, 공인시험, 공인시험점수, 자격증명’은 자격 사항으로 구분된다.



[그림 3-2] 이력서 데이터 정의

[표 3-2]는 이력서 데이터의 항목과 데이터 구성을 정리한 것이다.

[표 3-2] 이력서 데이터 구성

Idx	항목명	데이터 구성	샘플 데이터
1	이력서 번호	구직자의 이력서 번호	RID123456
2	지원자	구직자의 지원자 코드	CID123456
3	등록일	이력서 등록일	2000-01-01
4	수정일	이력서 수정일	2010-01-01
5	희망직종	희망 직무 키워드	화학·화공
6	희망직종 세부직종	희망 직무 세부 키워드	석유화학,에너지,화공,정유
7	희망직무	직무 키워드_선택	재료·화학·섬유·의복
8	희망(경력) 업종	희망 업종	제조 서비스 - 전기, 가스, 수도, 에너지,
9	전체경력	전체 경력 사항	10년 2개월
10	근무형태	정규직/계약직/인턴 등 형태	정규직
11	희망연봉	희망연봉	6000 ~ 6500
12	최종연봉	최종연봉	6000 ~ 6500
13	근무지역	근무지역	울산,전남,충남
14	최종학력	대학교졸업/석사 등	대학교졸업
15	고등학교명	졸업한 고등학교	** 고등학교
16	학교명	최종학력 대학교	** 대학교
17	학력구분	최종학력	대학교졸업
18	편입여부	편입 여부	편입
19	입학년도	입학년도	1992
20	졸업년도	졸업년도	2000
21	전공	전공명	화학공학
22	부전공	부전공명	컴퓨터공학과
23	계열	상관계열/공학계열 등 전공계열	공학계열(화공)
24	평점	학점 평점	3.17
25	만점	학점 만점	7
26	업종	이전 근무 회사 업종	정보통신 - 웹에이전시
27	회사명	이전 근무 회사	(주)이미지소스
28	근무시작일	이전 근무 회사 근무 시작일	2002-04-10
29	근무종료일	이전 근무 회사 근무 종료일	2003-03-03
30	부서명	이전 근무 회사 근무 부서명	기획마케팅팀
31	직위	이전 근무 회사 직위	대리(주임연구원)
32	담당업무	이전 근무 회사 담당 업무	웹 프로젝트 제안서 작성 및 기획

33	퇴직사유	이전 근무 회사 퇴직 사유	학업
34	프로젝트명	이전 근무회사 진행 프로젝트명	스마트카드 단말기 개발
35	의뢰업체	이전 근무회사 진행 프로젝트의뢰업체	** 테크놀로지스
36	참여기간	이전 근무회사 진행 프로젝트 참여 기간	2002-10-01~2002-12-31
37	내용 및 성과	이전 근무회사 진행 프로젝트의 내용 및 성과	웹사이트로 스마트카드 단말기 및 플랫폼 개발
38	언어	사용 가능 언어	러시아어
39	공인시험	보유하고있는 국가공인시험 종류	JLPT
40	점수	보유 공인 자격증 시험 점수	865점
41	자격증명	보유 국가 자격증명	자동차운전면허 1종 보통

직무 추천 모델의 종속 변수는 ‘희망직종’, ‘희망직종 세부직종’이며, 두 가지의 데이터를 함께 사용하기 위해 두 가지의 데이터를 군집화한 뒤 해당 내용을 Labeling 하여 코드값을 부여하였다. 이렇게 Labeling 된 데이터를 ‘JOB_CODE’라고 지정하여 종속 변수로 사용하였다. 즉, ‘희망직종’, ‘희망직종 세부직종’ 컬럼을 제외한 모든 컬럼이 원인변수가 되어 ‘JOB_CODE’라는 결과값을 만들어 낸다.

[표 3-3]의 ‘JOB_CODE’는 취업포털의 직무체계를 활용하여 총 125개로 Labeling 하였으며, NCS 직무 분류체계와 대응하여 사용할 수 있도록 구성하였다.

[표 3-3] JOB_CODE 생성

희망직종	희망직종 세부직종	Labeling
IT·게임	HTML코딩	1
IT·게임	게임기획	2
IT·게임	게임디자인	3
....
문화·예술·신문·방송	아나운서 ·리포터 · VJ · 성우	125

3.3 데이터 분석

데이터 수집이 완료되면 이후 데이터 분석 단계를 진행한다. 본 예측 모델의 데이터 분석은 기본적인 데이터 특성분석, 데이터 분포분석, 상관관계분석으로 진행하였다.

기본적인 데이터 특성분석은 각 항목의 데이터 형태, 데이터 유형, 데이터 세부 유형, 결측치 수 및 데이터 Rate를 분석하며, 데이터 분포분석은 도수분포표를 활용하여 각 항목의 데이터 구성과 분포를 확인하였다. 분포분석을 통해 데이터의 분포는 작지만, 값이 큰 값을 이상치로 판단하여 전처리단계에서 처리하였으며, 다음으로는 상관관계분석, 회귀 분석을 통해 종속 변수에 가장 많이 영향을 주는 칼럼을 확인하여 모델 학습 단계에서 가중치를 많이 줄 수 있게 설계하였다.

3.3.1 데이터 특성 및 분포분석

데이터 특성분석 결과는 [표 3-4]로 정리하였으며, 데이터 특성분석 종류는 각 항목의 데이터 형태, 데이터 유형, 데이터 세부 유형, 결측치 수 및 데이터 Rate를 분석하였다.

데이터 형태는 각 항목을 정형 데이터와 비정형 데이터로 구분하여 분석하고, 데이터 유형은 수치형 데이터와 범주형 데이터로 구분하여 분석하였다. 수치형 데이터는 수치를 값으로 가지는 변수이므로 수학적으로 활용할 수 있다. 범주형 데이터는 어떤 대상의 그룹을 나눌 때 사용한다. 데이터 세부 유형은 수치형 데이터의 연속형 데이터, 이산형 데이터로 분류하였고, 범주형 데이터는 순서형 데이터, 명목형 데이터로 분류하였다. 연속형 데이터는 더 작은 단위로 나눌 수 있다는 점이며, 예로 시간은 일 단위, 초 단위로 나눌 수 있고 사람의 나이는 년, 개월로 나눌 수 있다. 순서형 데이터는 순서 관계를 가지는 데이터이며, 예로 평점, 학점

등이 있다. 명목형 데이터는 범주를 분류할 수 있는 데이터를 의미한다.
 결측치 수 및 데이터 Rate는 370,824의 데이터 중 항목별로 결측치가 존재하여 결측치 수를 파악하고 결측치를 제외한 데이터 수를 데이터 Rate로 도출한다.

[표 3-4] 데이터 특성분석

Idx	컬럼이름	데이터 형태	데이터 유형	세부유형	결측수	RATE
1	이력서번호	정형 데이터	범주형	명목형	0	100.0%
2	지원자	정형 데이터	범주형	명목형	0	100.0%
3	등록일	정형 데이터	범주형	명목형	0	100.0%
4	수정일	정형 데이터		명목형	0	100.0%
5	희망직종	비정형데이터	범주형	명목형	0	100.0%
6	희망직종 세부 직종	정형 데이터	범주형	명목형	0	100.0%
7	희망직무	비정형데이터	-	-	26078	92.97%
8	희망(경력)업종	정형 데이터	범주형	명목형	115104	68.96%
9	전체경력	정형 데이터	수치형	연속형	0	100.0%
10	근무형태	정형 데이터	범주형	명목형	96593	73.95%
11	희망연봉	정형 데이터	수치형	연속형	0	100.0%
12	최종연봉	정형 데이터	수치형	연속형	34362	90.73%
13	근무지역	정형 데이터	범주형	명목형	0	100.0%
14	최종학력	정형 데이터	범주형	명목형	141	99.96%
15	고등학교명	정형 데이터	범주형	명목형	45	99.99%
16	학교명	정형 데이터	범주형	명목형	0	100.0%
17	학력구분	정형 데이터	범주형	명목형	141	99.96%
18	편입여부	정형 데이터	범주형	명목형	234186	36.85%
19	입학년도	정형 데이터	범주형	명목형	253	99.93%
20	졸업년도	정형 데이터	범주형	명목형	245	99.93%
21	전공	정형 데이터	범주형	명목형	5	100.0%
22	부전공	정형 데이터	범주형	명목형	213603	42.40%
23	계열	정형 데이터	범주형	명목형	122032	67.09%
24	평점	정형 데이터	수치형	연속형	1356	99.63%
25	만점	정형 데이터	범주형	명목형	8	100.0%
26	업종	정형 데이터	범주형	명목형	41280	88.87%

27	회사명	정형 데이터	범주형	명목형	58640	84.19%
28	근무시작일	정형 데이터	수치형	연속형	58666	84.18%
29	근무종료일	정형 데이터	수치형	연속형	61265	83.48%
30	부서명	비정형데이터	-	-	58706	84.17%
31	직위	정형 데이터	범주형	명목형	58736	84.16%
32	담당업무	비정형데이터	-	-	41357	88.85%
33	퇴직사유	비정형데이터	-	-	97333	73.75%
34	프로젝트명	비정형데이터	-	-	130082	64.92%
35	의뢰업체	정형데이터	범주형	명목형	130144	64.90%
36	참여기간	정형데이터	범주형	명목형	130059	64.93%
37	내용 및 성과	비정형데이터	-	-	130059	64.93%
38	언어	정형 데이터	범주형	명목형	123501	66.70%
39	공인시험	정형 데이터	범주형	명목형	123501	66.70%
40	점수	정형 데이터	수치형	연속형	123501	66.70%
41	자격증명	비정형데이터	-	-	123501	66.70%

데이터 분포분석은 데이터별로 Unique 값을 찾고, 해당 값이 이상치인지 확인하기 위해 진행한다. 즉, 데이터가 불균형을 방지하여 이상치가 없는 데이터를 모델 학습시킨다. 해당 단계에서 개인정보, 날짜, 비정형 데이터 등 불균형한 데이터를 제거하였다.

불균형 데이터는 “희망직종”, “직위”, “학력구분”, “계열”, “만점” 필드로 구성된다. “희망직종”은 호텔/콘도/리조트, 여행사, 관광 3가지 데이터가 각각 5, 3, 2개 밖에 없어서 해당 3가지 데이터는 기타 직종으로 넣어 사용한다. “직위”는 전무, 면접 후 결정, 회장 3가지 데이터가 각각 161, 73, 24개 밖에 없어서 해당 3가지 데이터는 기타 직위로 넣어 사용한다. “학력구분”은 고등학교 졸업 예정이라는 데이터가 47개밖에 없어서 기타 학력으로 넣어 사용한다. “계열”은 종교학 계열이라는 데이터가 156개 밖에 없어 기타 계열로 넣어 사용한다. “만점”은 0, 선택이라는 데이터가 각각 29, 8개 밖에 없어 본 데이터에서 삭제하였다.

[표 3-5]는 이력서 데이터에 포함된 희망직종을 기준으로 한 분포 분석

을 정리한 것이며, idx 1번 ‘경영,기획,회계,사무’가 데이터 편향의 실제적인 사례라고 할 수 있다. 데이터는 양적 다양성과 질적 대표성을 가져야 하며, 데이터의 다양성은 인공지능 모델을 다양한 패턴과 특징을 학습하도록 하여 일반화 능력을 향상시킬 수 있으나, 데이터가 특정 부분에 편향되어 있다면 해당 부분에서만 잘 작동하고 다른 상황에서는 부정확한 예측을 할 수 있다.

[표 3-5]의 데이터 분포 분석은 딥러닝 모델의 성능 평가에 있어 데이터 편향을 확인하는 매우 중요한 과정이며, 데이터 분포 분석 결과에 따라 새로운 데이터를 수집하고 모델을 업데이트함으로써 새로운 패턴과 특징을 학습하고 성능을 개선할 수 있다.

[표 3-5] 데이터 분포 분석 예시

idx	희망직종	index	처리방법
1	경영,기획, 회계, 사무	140,559개	-
2	재료,화학,섬유	58,728개	-
3	영업,판매TM	45,422개	-
4	IT게임	38,801개	-
....
13	여행사	3개	기타직종
14	관광	2개	기타직종

3.3.2 상관관계분석

본 연구 모델의 상관관계분석은 Cramer's V 계수를 활용하여 진행하였다. 해당 기법은 범주형 변수 간 상관관계를 파악하는 경우 사용되며, 비교 대상 범주 대상이 3개 이상일 경우 효과적이다. 또한, 해당 기법은 카이 제곱 독립성 검정의 효과 크기 측정이며 두 카테고리형 필드가 얼마나 강력하게 연관되는지를 측정한다.

0.2보다 크고 0.6보다 작으면 필드들이 적당하게 연관된다. 0.6보다 큰

경우는 필드들이 강력하게 연관된 경우이다.

종속 변수인 “JOB_CODE”를 기준으로 독립변수들의 모든 필드를 넣어서 도출하였다. 전체적으로 0.2보다 큰 값은 없었으며, 각각의 필드들은 종속 변수에 영향을 주지 않는다는 점을 도출할 수 있었다. 하지만 몇몇 필드들은 약한 상관관계가 있는 필드들이 있었으며, 그 필드는 “여자학교 여부”, “남자학교 여부”, “경력 존재 여부”, “부서명 존재 여부”, “계열”, “업종” 등 있었다. 각각의 계수는 0.1, 0.09, 0.09, 0.09, 0.04, 0.03으로 측정되었다.

[표 3-6] Cramer's V 계수

Idx	컬럼이름	label	계수	Idx	컬럼이름	label	계수
1	이력서등록월	aa	0	24	편입여부	be	0
2	이력서수정월	ab	0	25	입학년도	bf	0.01
3	이력서등록요일	ac	0	26	졸업년도	bg	0
4	이력서수정요일	ad	0	27	대학기간	bh	0
5	이력서DIFF	ae	0.01	28	계열	bi	0.04
6	희망연봉	af	0.01	29	평점	bj	0
7	최종학력	ag	0.02	30	경력존재여부	bk	0.09
8	고등학교명	ah	0.01	31	경력일수	bl	0
9	고등학교세부유형	ai	0	32	근무형태	bm	0.02
10	일반고여부	aj	0	33	업종	bn	0.03
11	특목고여부	ak	0	34	부서명존재여부	bo	0.09
12	특성화고여부	al	0.01	35	직위	bp	0.01
13	자율고여부	am	0.01	36	담당업무	bq	0.01
14	공립여부	an	0	37	퇴직사유	br	0.01
15	사립여부	ao	0	38	근무성과	bs	0
16	남자학교여부	aq	0.09	39	최종연봉	ca	0.01
17	여자학교여부	ar	0.1	40	근무지역1	cb	0.01
18	남여공학여부	as	0.01	41	근무지역2	cc	0.01
19	고등학교LOC	at	0	42	근무지역3	cd	0.01
20	대학교명	ba	0	43	언어	ce	0.01
21	대학교LOC	bb	0	44	공인시험	cf	0
22	대학설립구분	bc	0.01	45	언어성적	ch	0
23	대학구분	bd	0	46	JOB_CODE	index	1

3.4 데이터 전처리

데이터 전처리는 1단계 결측치 처리, 2단계 이상치 처리, 3단계 데이터 코드화로 나누어 진행하며 학습 데이터셋 구축을 위한 이전 단계이다.

[표 3-7]의 1단계 결측치 처리는 Null 값이 데이터에 있을 시 딥러닝 모델에 입력이 되었을 때 문제가 될 수 있어, 결측치를 모두 제거하거나 값을 넣어주는 방법을 사용하여 처리하였다. 결측치가 없는 경우 이력서 데이터를 그대로 사용하며, 결측치가 있는 경우에도 데이터의 타입이 정수(Integer) 경우 숫자 0으로 결측치를 입력하였으며, 데이터 타입이 문자열일 경우 Unknown(UNK)으로 결측치를 입력하였다.

[표 3-7] 결측치 처리 항목

	숫자(0)	UNK
결측치 처리 항목	등록일, 수정일, 근무형태, 희망연봉, 최종연봉, 근무지역, 최종학력, 편입여부, 입학년도, 졸업년도, 계열, 평점, 업종, 회사명, 근무시작일, 근무종료일, 전체경력, 부서명, 직위, 담당업무, 퇴직사유, 내용 및 성과, 언어, 공인시험, 점수	전공 부전공 자격증

[표 3-8] 이상치 처리 프로세스

idx	이상치 처리	세부 내용
1	불필요한 문자 삭제	영문, 한글, 숫자등 다양한 데이터들이 혼재되어 있는 상태에서 필요한 문자열만 추출하여 원본데이터에 입력한다.
2	데이터 타입 변경	원본 데이터의 데이터 타입을 필요한 형태로 변경함. 즉, 날짜 데이터인 경우 String값으로 정의가 되어 있을 때 데이터 타입을 변경하여 DATETIME 타입으로 변경하여 날짜 데이터로 활용한다.
3	컬럼 삭제	원본 데이터 중 데이터의 모수가 없거나 혹은 불필요한 데이터일 경우 컬럼을 삭제하여 진행한다.

[표 3-8]의 2단계 이상치 처리는 각 데이터의 특성에 따라 ① 불필요한 문자 삭제 , ② 데이터 타입 변경, ③ 컬럼 삭제, 총 3가지의 방법을 활용하였으며 [표 3-9]는 이상치를 처리한 내역을 정리하였다.

[표 3-9] 이상치 처리 내역

Idx	컬럼이름	불필요한 문자 삭제	데이터 타입 변경	컬럼 삭제
1	등록일		O	
2	수정일		O	
3	고등학교명	O		
4	학교명	O		
5	학력구분			O
6	입학년도		O	
7	졸업년도		O	
8	전공	O		
9	부전공	O		
10	계열	O		
11	평점	O		
12	만점	O		
13	회사명	O		
14	근무시작일		O	
15	근무종료일		O	
16	부서명	O		
17	직위	O		
18	담당업무	O		
19	퇴직사유	O		
20	프로젝트명			O
21	의뢰업체			O
22	참여기간			O
23	내용 및 성과	O		
24	공인시험	O		
25	점수	O		
26	자격증	O		

[표 3-10]의 3단계 데이터 코드화는 각 데이터의 특성에 따라 자체코드화, 표준코드화, 수치계산, 기타의 총 4가지 방법을 활용하였다. 데이터 코드화는 자연어 처리 기반의 전이학습 모델을 적용한 텍스트 임베딩으로 처리하는 것도 가능하지만, 본 연구에서는 직무 추천 모델의 경량화를 통한 학습 속도 개선을 위하여 데이터 전처리에 포함하였다.

[표 3-10] 데이터 코드화 방법

코드화 방법	설명
자체코드화	데이터를 군집화하여 자체 코드 테이블을 생성하고, 해당 테이블과 매칭되는 값을 변환하는 방법
표준코드화	국가에서 제공하는 표준 코드 체계를 활용하여 코드 테이블과 매칭되는 값을 변환하는 방법
수치계산	특정 정형 데이터의 값들을 계산하여 숫자의 값으로 변환하는 방법
기타	원 데이터를 그대로 사용하는 경우를 포함하여, 상기 방법들 외에 사용된 전처리 방법

[그림 3-3]의 자체 코드 매핑 테이블은 대부분 명목형 변수로 구성되어 있는 이력서 데이터의 학습 속도를 개선하기 위하여 수치형 변수로 변환하기 위한 것이며, [그림 3-3] 자체 코드 매핑 테이블과 [그림 3-4] 표준 코드화에 따라 [표 3-11] 데이터 전처리 내역에 표기된 데이터 항목에 대하여 전처리를 진행하여 최종적인 학습데이터를 생성하였다.

이력서 데이터 자체 코드 매핑 테이블												
코드 값	근무형태	근무지역	최종학력, 학력구분	계열	희망연봉, 최종연봉	회사명	부서명	관입여부	직위	퇴직사유	언어	공인시험
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	기타	기타	기타	기타	면접시험의, 당사규정, 기타	유	유	유	기타	기타	기타	기타
2	정규직	전국	고등학교졸업	공학계열(산업공학)	1000 이하				임시직	학업	영어	TOEFL
3	해외취업	서울	대학(2,3) 졸업	상경계열	1200~1400				사원(연구원)	이직	중국어	HSK
4	병역특례	강원	대학교 졸업	자연과학계열	1400~1600				계장(연구원)	계약만료	일본어	TOEIC
5	인재파견	경기	석사	인문/사회계열	1600~1800				주임(연구원)	회사폐업	스페인어	JLPT
6	계약직	경남	박사	공학계열(전자)	1800~2000				대리(주임연구원)	개인 사유	독일어	JPT
7	인턴	경북		공학계열(전산)	2000~2200				과장(선임연구원)	재직중	러시아어	G-TLP
8	프리랜서	광주		어문계열	2200~2400				차장(수석연구원)	퇴직	기타	TEPS
9		대구		예/체능계열	2400~2600				팀장		프랑스어	IELTS
10		대전		공학계열(기타)	2600~2800				부장(연구소장)			JTRA

[그림 3-3] 자체 코드 매핑 테이블

학교명	학교코	대학구	학교구분	지역	설립구	본분교	단과대학명	단과대학코드	학교별학과코
ICT폴리텍대학	0000579	전문대학	기능대학	경기	사립	본교	단과대구분없음	0999	0225077
ICT폴리텍대학	0000579	전문대학	기능대학	경기	사립	본교	단과대구분없음	0999	0225076
ICT폴리텍대학	0000579	전문대학	기능대학	경기	사립	본교	단과대구분없음	0999	0225078
ICT폴리텍대학	0000579	전문대학	기능대학	경기	사립	본교	단과대구분없음	0999	0236001
ICT폴리텍대학	0000579	전문대학	기능대학	경기	사립	본교	단과대구분없음	0999	0236000
ICT폴리텍대학	0000579	전문대학	기능대학	경기	사립	본교	단과대구분없음	0999	0225075
가야대학교 보건대학원(20002940)	00002940	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0219039
가야대학교 보건대학원(20002940)	00002940	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0220855
가야대학교 보건대학원(20002940)	00002940	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0219040
가야대학교 보건대학원(20002940)	00002940	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0219038
가야대학교 보건대학원(20002940)	00002940	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0220857
가야대학교 보건대학원(20002940)	00002940	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0219041
가야대학교 일반대학원(20002941)	00002941	대학원	일반대학원	경남	사립	본교	단과대구분없음	0999	0219042
가야대학교 항만물류대학(0002867)	0002867	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0215539
가야대학교 행정대학원(20002939)	00002939	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0219037
가야대학교 행정대학원(20002939)	00002939	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0219036
가야대학교 행정대학원(20002939)	00002939	대학원	특수대학원	경남	사립	본교	단과대구분없음	0999	0220856
가야대학교(고령)	0002749	대학	대학교	경북	사립	제2캠퍼스	단과대구분없음	0999	0208646

[그림 3-4] 표준 코드화

[표 3-11] 데이터 전처리 내역

데이터 항목명	전처리	전처리 내용
이력서 등록일	수치계산	이력서 등록일과 수정일 차이계산
이력서 수정일	수치계산	이력서 등록일과 수정일 차이계산
희망 직종	자체코드	자체 코드화 진행
희망 세부직종	자체코드	자체 코드화 진행
희망 업종	자체코드	자체 코드화 진행
전체경력	수치계산	전체 경력기간의 '일 수'를 계산
근무형태	자체코드	정규직 활용하여 자체 코드 테이블 구성 후 자체 코드화 진행 / 근무형태 값이 복수일 경우 제일 좋은 값(정규직)만 유지
희망연봉	자체코드	군집화 후 연봉 정보 추출 / 추출된 데이터를 범주형으로 구성된 자체 코드 테이블을 활용
최종연봉	자체코드	
근무지역1	자체코드	복수의 값은 근무지역1, 근무지역2, 근무지역3으로 분리 / 자체 코드화 진행
근무지역2	자체코드	지역구에 따른 코드테이블 구성
학력구분	자체코드	자체 코드화 진행
고등학교	자체코드	표준 코드 테이블 활용
학교명	자체코드	표준 코드 테이블 활용
편입여부	자체코드	자체 코드화 진행
졸업년도	자체코드	자체 코드화 진행
졸업기간	자체코드	자체 코드화 진행
입학년도	자체코드	자체 코드화 진행

전공/부전공	기타	자체 텍스트 처리 알고리즘 사용
계열	자체코드	자체 코드화 진행
평점/만점	수치계산	백분율 계산
업종	자체코드	자체 코드화 진행
회사명	자체코드	자체 코드화 진행(값이 있으면1, 없으면 0)
부서명	자체코드	자체 코드화 진행(값이 있으면1, 없으면 0)
직위	자체코드	자체 코드화 진행(값이 있으면1, 없으면 0)
담당업무	자체코드	자체 코드화 진행(값이 있으면1, 없으면 0)
퇴직사유	자체코드	자체 코드화 진행
내용 및 성과	자체코드	자체 코드화 진행(값이 있으면1, 없으면 0)
언어	자체코드	자체 코드화 진행
공인시험 점수	수치계산	점수 표준화 진행
공인시험	자체코드	자체 코드화 진행
자격증	기타	자체 텍스트 처리 알고리즘 사용

[표 3-12]는 전처리 방안 설계에 따른 전처리 과정이 완료된 데이터이며, 해당 데이터를 사용하여 학습 데이터셋 구축을 진행하였다.

[표 3-12] 전처리 완료 데이터 예시

idx	항목명	전처리 후 데이터
1	이력서번호	RID121811
2	이력서등록월	0.083333
3	이력서수정월	0.166667
4	이력서등록요일	0.666667
5	이력서수정요일	1
6	이력서DIFF	0.206301
7	희망연봉	0.32
8	최종학력	4
9	고등학교명	97
10	고등학교세부유형	21
11	일반고여부	1
12	특목고여부	0
13	특성화고여부	0
14	자율고여부	0
15	공립여부	1

16	사립여부	0
17	남자학교여부	0
18	여자학교여부	1
19	남여공학여부	0
20	고등학교LOC	3
21	대학교명	74
22	대학교LOC	3
23	대학설립구분	5
24	대학구분	5
25	편입여부	0
26	입학년도	0.987141
27	졸업년도	0.988625
28	대학기간	3
29	전공	컴퓨터공학
30	부전공	UNK
31	계열	0
32	평점	80
33	경력존재여부	0
34	경력일수	0
35	근무형태	0
36	업종	0
37	부서명존재여부	0
38	직위	0
39	담당업무	0
40	퇴직사유	0
41	근무성과	0
42	최종연봉	0.28
43	근무지역1	3
44	근무지역2	3
45	고등학교세부유형	21
46	근무지역3	5
47	언어	0:0:0:0:0:0
48	공인시험	0
49	언어성적	0:0:0:0:0:0
50	자격증명	전자계산기조직응용기사;정보처리기사
52개	JOB_CODE	S2101
52	ANS	1

3.5 학습 데이터셋 구축

학습 데이터셋은 구직자의 적합한 직무뿐만 아니라 부적합한 직무도 포함하여 학습하기 위해서 1:1 Negative Sampling 방식으로 전체 이력서 데이터 370,824개에서 이상치 및 결측치 처리를 완료한 298,465개를 Positive Sample 298,465개, Negative Sample 298,465개를 합쳐서 총 596,930개의 데이터로 구성하였다.

Positive Sample, Negative Sample을 구분하기 위해 ‘ANS’ 컬럼을 설정하여 이진분류 하였다. 구직자가 선호하는 Positive sample 데이터를 ANS = 1로 선정하고 구직자가 선호하지 않은 Negative sample 데이터를 ANS = 0로 선정하였다. 즉, 이력서 데이터 전체를 ANS = 1로 설정하고, 복사한 데이터를 ANS = 0로 설정하여 두 개의 데이터를 하나의 학습 데이터셋으로 구성하였다. 모델의 학습 단계에서는 train_set, validation_set, test_set 6 : 2 : 2로 데이터를 분할 하여 사용하였다.

[표 3-13] 학습 데이터셋 데이터 개수

데이터셋	데이터 수
이력서 데이터	298,465개
학습 데이터 셋 구축 1:1 Negative sampling	전체 : 596,930 긍정 사례 : 298,465개 / 부정 사례 : 298,465개
훈련 데이터	477,544개 (80%)
시험용 데이터	119,386개 (20%)

3.6 피처 엔지니어링(Feature Engineering)

학습 데이터셋 구축 후 모델 학습을 하기 위해 변수별로 feature tuning을 진행하였다. 변수는 소범주형 변수, 다범주형 변수, 자연어 변수로 나뉘서 진행하며, 소범주형 변수는 Category-Encoding, 다범주형 변수는 Embedding, 자연어 변수는 사전학습모델을 이용하였다.

[표 3-14] 변수별 Feature Tuning

	소범주형 변수	다범주형 변수	자연어 변수
추론	Category-Encoding	Embedding	사전학습모델
용도	범주형 변수를 수치형 변수로 변환	텍스트 데이터를 수치형 데이터로 변환	대규모 데이터를 사전에 학습된 모델
Data Sets (Details)	대학설립구분, 대학구분, 대학교LOC, 최종학력, 고등학교세부유형, 고등학교LOC,대학교LOC, 계열,근무형태,업종대분류, 업종소분류, 직위, 퇴직사유, 근무지역	고등학교명, 대학교명, 담당업무, 언어, 공인시험, JOB_CODE	전공, 자격증

소범주형 변수는 사전에 정의된 라벨(Label)이 30개 이하인 변수이며 학습데이터는 Label Encoding 되어 있는 형태로 입력되며, Category Encoding 층을 거쳐 One-hot Encoding 된 형태로 변환되어 딥러닝 모델에 입력하였다. 소범주형 변수의 주요 Layer 층은 Category Encoding 이며, 해당 Encoding은 정수의 특징을 인코딩하는 전처리 계층이다.

다범주형 변수는 사전에 정의된 라벨(Label)이 30개 이상인 변수이며 One-hot Encoding으로 처리할 경우 차원의 수가 너무 커져 데이터 희박성(Sparsity) 문제가 발생할 가능성이 높은 데이터 형태를 의미한다. 이러한 다범주형 변수의 경우 모델 내에서 직무 추천을 위한 적절한 임베딩 공간의 탐색이 필요하다. 특히 본 모델에서 사용되는 다범주형 변수의 경우 범주의 개수가 1,000개 이상인 경우가 다수 존재한다. 이를 위하여 임베딩 층을 설계하여 다범주형 변수를 적절한 벡터 공간에 표현할 수 있도록 하는 한편, 각 다범주형 변수별로 의미 있는 벡터 공간의 탐색을 위해 직무 추천에 필요한 특성(Feature)만을 추출하기 위한 층을 추가로 설계하여 반영하였다. 즉, 다범주형 변수를 먼저 큰 벡터 공간에 표현하는 임베딩 층을 통과시켜 벡터화를 진행하고 임베딩 층의 결과를 직무 추천에 적합한 수치로 변환하기 위한 1D 컨브넷(Conv1D) 구조를

반영하였다.

1D 컨브넷(Conv1D)은 합성곱 연산의 커널(Kernal) 개념을 도입하여 1D 패치(부분 시퀀스)를 추출하여 합성곱 연산을 적용한다. 그에 따라 시퀀스에 있는 지역 패턴을 인식할 수 있어 각 범주가 가지는 시퀀스를 고려하면서 직무 추천에 도움이 되는 지역 패턴의 식별이 가능하다. 이를 통해 원천 다범주형 변수는 임베딩 층을 통과하여 다차원의 벡터 공간에 수치로 표현된다. 다범주형 변수의 주요 Layer 층은 Embedding, Conv1D로 구성되어 있으며, Conv1D 작업 전에 단어를 밀집 벡터로 만드는 역할을 하는 Embedding 작업이 필요하다. Embedding은 정수 인코딩이 되는 단어를 임베딩 된 값으로 출력한다. 자연어 처리 알고리즘인 유니버설 센텐스 인코더(Universal sentence encoder)를 통하여 다범주형 변수 처리방법을 사용하지 못하는 자연어 변수를 처리하고, 신규 전공 또는 전공 간의 결합 등의 데이터 변화에도 강건한 모델을 설계하기 위하여 적용하였다. 특성(Feature) 결합 및 추천을 위한 벡터 공간 탐색 과정은 각 변수 대분류별로 표현된 34개의 수치를 하나로 결합하고 직무 추천을 위한 벡터 공간을 탐색하는 과정이다. 이를 위해 대분류별 34개의 특징을 결합하고 이어 완전 연결층(Fully Connected Layer)을 연결하여 최종 출력값을 추론할 수 있는 벡터 공간을 탐색할 수 있도록 유도한다. 추천을 위한 벡터 공간 탐색은 모델의 출력값의 형태에 따라 상이하게 설계되며, 최종 출력값의 차원보다 더 큰 벡터공간의 탐색을 하여 정보의 손실을 최소화는 딥러닝 모델을 설계하였다.

3.7 하이퍼파라미터(Hyperparameter)

하이퍼파라미터는 최적의 학습 모델을 구현하기 위해 설정하는 변수로 가중치 초기화, 학습률, 최적화, 학습반복, 학습 배치 크기, 임베딩 차원,

은닉층의 개수 등이 있으며, 하이퍼파라미터의 튜닝 기법은 그리드 탐색, 랜덤 탐색, 베이지안 최적화, 휴리스틱 탐색 등이 있다.

하이퍼파라미터는 모델의 매개 변수를 추정하는 프로세스로서 개발자가 임의로 조정할 수 있는 매개변수이며, 하이퍼파라미터의 최적값은 데이터 분석 결과에 의해 결정되는 것이 아니므로 절대적인 최적값은 존재하지 않는다. [표 3-15]는 하이퍼파라미터 설정 변수이다.

[표 3-15] 하이퍼파라미터 항목

Idx	하이퍼파라미터	선정방법
1	가중치 초기화	Xavier 초기화 기법과 he 초기화 기법을 사용
2	학습률	임의의 학습률 선정 후 반복적 미세 조정
3	최적화	RMSProp, Adam 중 최고 성능 최적화 기법 선정
4	학습반복	검증 손실(Validation loss) 값의 2회 초과 증가 시 학습 반복 중단(조기 종료 조건을 통한 학습 반복 횟수의 선정)
5	학습 배치 크기	16, 64, 128, 256, 512의 각각 다른 사이즈로 반복 학습을 통한 최고 성능의 크기 선정
6	임베딩 차원	50, 100, 200, 400의 각각 다른 임베딩 차원을 입력 임베딩으로 선정하여 반복 학습 후 최고 성능의 임베딩 차원 선정
7	은닉층 개수	네트워크 구조(은닉층)을 설정하여 가중치를 조절

3.8 딥러닝 모델 학습

본 연구에서는 그리드 탐색 실험으로 선정된 하이퍼파라미터를 딥러닝 기반 텍스트 임베딩 모델(CNN, LSTM, Transformer)에 적용하여 학습하였다. 활성화 함수는 각 모델의 은닉층에 따라 일부 차이가 있으며, 손실 함수는 이진 분류 Binary Cross Entropy를 이용하였다.

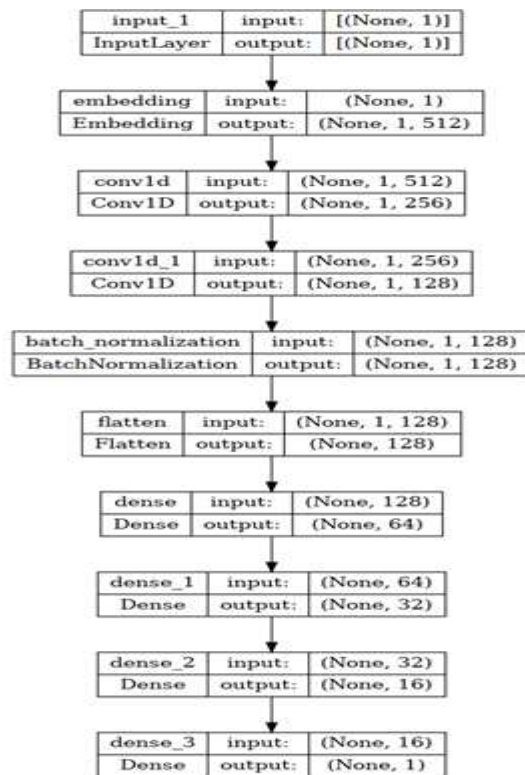
3.8.1 합성곱 신경망(CNN)

CNN 전체 모델 중에서 ‘고등학교명’ 컴럼은 1개의 입력층, 8개의 은닉층, 1개의 출력층으로 구성한다. 입력층은 정수화한 텍스트 데이터를 입

력 받아서 3차원의 임베딩 벡터를 출력하는 임베딩층으로 구성한다.

은닉층 중 첫 번째 층은 256개 노드를 가진 CNN 모델로 학습하고, 두 번째 층은 128개 노드를 가진 CNN 모델로 학습한다.

Batch Normalization 알고리즘을 통해 배치 정규화를 진행했으며, Flatten 함수를 사용하여 1차원 데이터로 가공한 뒤 텐스층(Dense Layer)로 구성한다. 텐스층은 결합층(Fully Connected Layer)이라고도 하는데 입력층의 노드와 출력층의 노드를 모두 연결해 주는 층을 의미한다. 예를 들어, 입력층의 노드가 2개이고, 출력층의 노드가 8개라면 텐스층은 이를 2×8 로 곱하여 16개의 가중치를 생성한다. 이후 활성화 함수인 sigmoid 함수를 거쳐 출력층으로 수치가 산출된다.



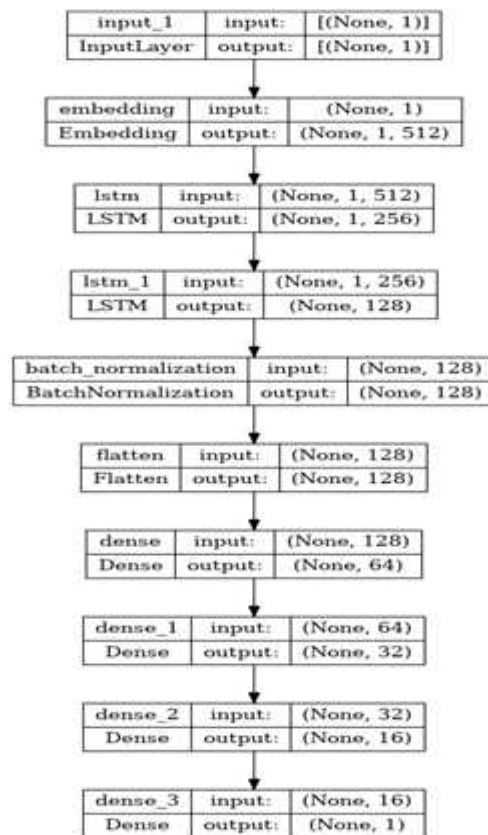
[그림 3-5] CNN 모델 고등학교 Feature Dense Layers

3.8.2 장단기 메모리(LSTM)

LSTM 전체 모델 중에서 ‘대학교명’ 컬럼은 1개의 입력층, 8개의 은닉층, 1개의 출력층으로 구성한다. 입력층은 정수화한 텍스트 데이터를 입력받아서 3차원의 임베딩 벡터를 출력하는 임베딩층으로 구성한다.

은닉층 중 첫 번째 층은 256개 노드를 가진 LSTM 모델로 학습하고, 두 번째 층은 128개 노드를 가진 LSTM 모델로 학습한다.

Batch Normalization 알고리즘을 통해 배치 정규화를 진행하고, Flatten 함수를 사용하여 1차원 데이터로 가공한 뒤 텐스층으로 구성한다. 이후 활성화 함수인 Sigmoid 함수를 거쳐 출력층으로 수치가 산출된다.

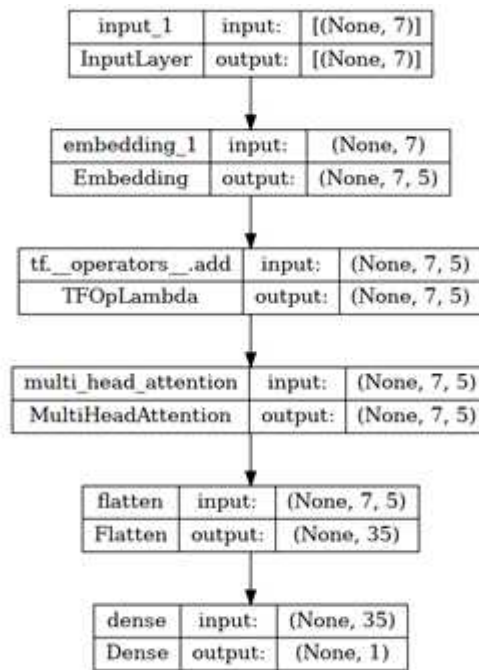


[그림 3-6] LSTM 모델 대학교 Feature Dense Layers

3.8.3 트랜스포머(Transformer)

Transformer 전체 모델 중에서 ‘언어’ 컬럼은 1개의 입력층과 4개의 은닉층, 1개의 출력층으로 구성한다. 입력층은 정수화한 텍스트 데이터를 입력 받아서 3차원의 임베딩 벡터를 출력하는 임베딩층으로 구성한다.

은닉층 중 첫 번째 층을 7개의 노드를 가진 TFOpLambda 모델로 학습하고 두 번째 층은 7개의 노드를 가진 Transformer 모델 중 Multi-Head Attention 모델로 학습한다. Flatten 함수로 1차원 데이터로 가공한 뒤 텐스층으로 구성한다. 활성화 함수인 Sigmoid 함수를 거쳐 출력층으로 수치가 산출된다.



[그림 3-7] Transformer 모델 언어 Feature Dense Layers

3.9 국가직무능력표준(NCS) 직무 분류체계 매핑

NCS(National Competency Standards)의 직무 분류는 직무의 유형을 중심으로 대분류, 중분류, 소분류, 세분류의 순으로 단계적 직무 구성을 나타낸다. 본 연구를 위하여 1,064건의 직무를 수집하였다.



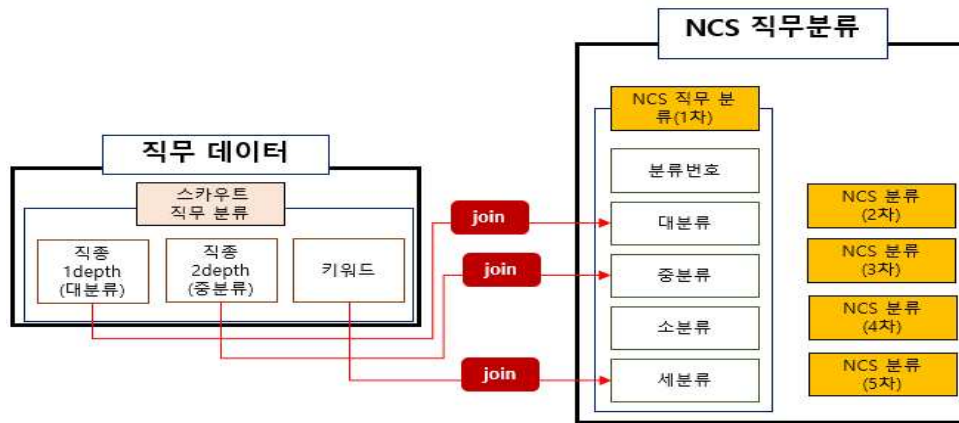
[그림 3-8] NCS 직무 분류 체계 예시

NCS 직무 분류체계는 국가가 산업현장에서 필요한 능력(지식, 기술, 태도)을 표준화한 것으로 이를 활용하여 교육훈련 및 자격증 취득을 통해 현장 중심의 인재를 양성할 수 있도록 지원하고 있다.

취업포털의 직무체계와 NCS 직무 분류체계는 각각 Depth1, Depth2, Depth3 및 대분류, 중분류, 소분류, 세분류로 구성됩니다. [그림 4-10]과 같이 Depth1는 대분류, Depth2는 중분류, Depth3는 세분류에 매핑하였으며, Input값을 취업포털 직무 분류 체계로 입력시키고 Output값은 NCS분류 체계의 데이터를 넣어 데이터를 구성하였다. 케라스에서 지원하는 임베딩 함수를 사용하여 완전 연결층으로 변환한 뒤, 바이너리 엔트로피 손실 함수를 사용하여 이진 분류를 수행하였다. 이렇게 분류된

결과를 기반으로 적합도가 높은 분류체계끼리 매핑하였다.

[그림 3-9]는 취업포털 직무체계와 NCS 직무 분류체계를 매핑하는 구조이며, [표 3-16]은 NCS의 직무 분류체계 매칭 데이터를 구조화한 것이다.



[그림 3-9] 취업포털 직무 분류와 NCS 직무 분류체계의 매칭 테이블

[표 3-16] NCS 직무 분류체계 매핑 데이터 예시

직종Depth1	직종Depth2	Labeling	NCS
IT·게임	HTML코딩	1	N20010207
IT·게임	게임기획	2	N08030101
IT·게임	게임디자인	3	N08030205
....	
문화·예술·신문·방송	아나운서 ·리포터 · VJ · 성우	125	N07030101

3.10 파인튜닝(Fine-Tuning)

직무 추천 모델에서 사용되는 이력서 데이터는 범주형 데이터와 자연어 데이터로 구성되어 있어, 초기 모델 학습뿐만 아니라 추가된 새로운 데이터로 인한 모델 재학습에 대용량의 컴퓨터 자원과 학습 시간이 소모된다. 이러한 문제점 해결하기 위하여 사전 훈련(Pre-Training)된 모델

을 기존에 학습된 모델에 적용하여 성능을 향상시키는 미세 조정 기술이 연구되었다. Fine-Tuning은 보다 적은 데이터로도 높은 성능을 달성할 수 있도록 도와주며, 새로운 작업에 대한 전체 모델을 처음부터 학습하는 것보다 더 적은 시간과 비용이 소모되며, 기존 모델과 비교하여 향상된 성능을 얻을 수 있다. [표 3-17]에 Fine-Tuning 종류를 정리하였다.

[표 3-17] Fine-Tuning 종류

Idx	Fine-Tuning 종류	세부 내용
1	전체모델 파인튜닝 (Full Model Fine-Tuning)	모든 층을 새로운 데이터셋에 맞게 재학습 시키는 방법, 해당 방법은 새로운 작업에 대해 최고의 성능을 보장하지만, 학습에 시간과 자원이 많이 필요하다
2	상위 층 파인튜닝 (Top Layers Fine-Tuning)	미리 학습된에서 최상위 층(분류기)만 새로운 데이터셋에 맞게 재학습하는 방법이다. 이 방법은 전체 모델 파인튜닝에 비해 학습 시간과 자원이 적게 소요되며, 새로운 작업에 대한 높은 정확도를 보장한다
3	부분적 파인튜닝 (Partial Fine-Tuning)	일부 층만 새로운 데이터셋에 맞게 재학습하는 방법. 이 방법은 전체모델 파인튜닝에 비해 더 적은 자원이 필요하며, 기존 모델에서 어떤 층을 사용할지 선택할수 있는 유연성을 제공한다

본 연구에서는 상위층 파인튜닝과 부분적 파인튜닝을 사용하여 성능을 향상시킨다. [표 3-18]은 데이터 관점, 분류기 관점, Layers 관점으로 Fine-tuning을 실험하는 방법에 대하여 정리하였다.

[표 3-18] Fine-Tuning 실험 방법

관점	처리 방법
데이터 관점	1. 불필요한 데이터 제거 2. LDA 모델링을 통한 키워드 추출 및 데이터 추가
분류기 관점	1. Universal-Sentence-Encoder와 XLM 모델 사용 2. 임베딩 모델을 활용한 학습
Layers 관점	1. 부서명, 담당업무, 근무성과 데이터 Concatenate 2. 중간 및 마지막에 입력하여 학습 성능 테스트

3.11 모델의 성능측정

3.11.1 학습 성능의 측정

완전 연결층(Fully Connected Layer)으로 구성된 Input(학습데이터셋) 값과 직무 추천 결과의 따른 적합 확률값을 이용하여 Input(학습데이터셋)에 적용하여 학습시킨 후 모델을 생성한다. 특성 결합을 거쳐 CNN, LSTM, Transformer 모델을 설계하고 최종적으로 모델 Compile 함수를 이용하여 해당 모델에 적용할 Loss Function, Optimizer, Metrics 등을 설정한다. 이후 모델에 적용할 옵션 등을 설정하고 해당하는 구조에 맞춰 Input data를 입력시켜 학습을 진행하며, 구직자의 선호 직무가 맞다면 적합, 선호 직무가 아니면 부적합으로 정의하였는데, 이는 궁극적으로 Binary Classification의 문제이며, 이러한 이유로 이진 분류를 위한 분류 모델을 적용한다.

직무 추천 모델의 성능지표는 수집한 전체 데이터의 80%를 이용하여 평가한다. Train 데이터는 전체 데이터의 80%의 80%를 사용하여 Train 데이터를 구성한다. 즉, 64%가 Train 데이터이며, 16%가 Validation 데이터로 구성된다. 학습이 잘 되었는지를 평가하는 방법으로는 학습 손실과 검증손실을 이용하는 방법이 있다. 분류 모델의 경우 분류가 잘 되었는지를 판단하기 위한 성능평가 방법이 여러 가지 있는데 본 연구에서는 정확률(Accuracy), 정밀도(Precision)를 사용하여 측정하였다.

3.11.2 예측 성능의 측정

직무 추천 모델의 예측 성능지표는 수집한 전체 데이터의 20%를 이용하여 평가한다. 모형의 예측 성능측정은 텐서플로(Tensorflow)와 케라스(Keras) 라이브러리의 측정치 정의를 통해 설정한 값을 활용한다.

제 4 장 실험 및 결과분석

4.1 실험 환경 및 도구

본 연구에 사용된 실험 장비의 운영체제는 Ubuntu 20.04.5 LTS, CPU는 Silver 4210, GPU는 Nvidia Quadro A6000 48GB, Memory는 64GB이며, 실험 도구는 Python 3.10과 Tensorflow 2.8, Keras를 사용하여 전처리 및 모델 학습을 진행하였다.

4.2 파인튜닝 선정 실험

본 연구에서는 상위층 파인튜닝과 부분적 파인튜닝의 방법을 사용하여 학습 모델의 성능을 높이려고 하였으며, 상위층 파인튜닝은 사전 학습된 모델의 상위층을 새로운 작업에 맞게 조정하는 것에 중점을 두었다.

사전 학습된 모델의 하위층은 일반화된 특성을 학습했기 때문에 그대로 사용하고, 상위층은 새로운 작업과 관련된 특성이 반영되도록 조정하였다. 이를 통하여 학습 성능향상에 필요한 세부 정보를 더욱 민감하게 학습하고자 하였다.

부분적 파인튜닝은 사전 학습된 모델의 일부 층만을 파인튜닝하는 것이다. 전체 모델을 다시 학습하는 대신, 일부 층만 새로운 작업에 맞게 업데이트된다. 이를 통하여 파라미터 수를 줄이고 학습 비용을 절감하고자 하였다. 또한, 학습 모델의 설계 및 실험은 데이터 관점, 분류기 관점, Layers 관점으로 나누어 진행하였다.

4.2.1 데이터 관점

데이터 관점의 파인튜닝은 사전 학습된 모델에 새로운 작업에 필요한

데이터를 제공하여 모델을 조정하는 것이다. 직무 추천 모델(CNN, LSTM, Transformer)의 1차 실험을 분석한 결과 구직자의 특징을 충분히 추출하지 못한 것으로 확인되어, 2차 실험에서는 부서명, 담당업무 특징을 추가하여 모델을 조정하였다. 이를 위하여 해당 데이터에 대해 토큰화(Tokenization)하고, 한국어 형태소 분석기(KoNLPy)를 사용하여 명사만 추출하였다. 그 후 잠재 디리클레 할당(Latent Dirichlet allocation, LDA) 모델을 사용하여 추출된 명사를 기반으로 키워드를 추출하였다. 부서명은 1~2개의 키워드로 이루어져 있으므로 이를 추출하여 모델 학습에 사용하였다. 담당업무는 여러 문장으로 이루어져 있으므로 문장의 영향력 있는 키워드 5개를 추출하였다.

[표 4-1]은 데이터 관점의 Fine-Tuning 데이터를 가시화한 것이다.

[표 4-1] 데이터 관점 Fine-Tuning

idx	항목명	전처리 후 데이터
1	이력서번호	RID121811
2	이력서등록년	0.9896193771626296
3	이력서DIFF	0.0706849315068493
4	희망연봉	0.36666666666666666
5	최종학력	2
6	고등학교명	612
7	일반고여부	1
8	고등학교LOC	13
9	대학교명	19
10	대학교LOC	3
11	입학년도	2003-10-01
12	졸업년도	2008-07-01
13	대학기간	240
14	전공	기계공학과
15	부전공	UNK
16	계열	0
17	평점	90

18	경력존재여부	1
19	근무시작일	1998
20	근무종류일	1999
21	경력일수	1
22	근무형태	3
23	업종	20
24	직위	2
25	담당업무	설계;수행;분석;lg;업무;lca;경영;해외;원가;정보
26	퇴직사유	0
27	부서명	개발
28	최종연봉	0.28
29	언어	0;0;0;0;0;0
30	공인시험	0
31	언어성적	0;0;0;0;0;0
32	자격증명	건설기계;일반기계;산업안전;중등실기교사;자동차운전면허
33	31JOB_CODE	S2101
34	ANS	1

4.2.2 분류기 관점

분류기 관점의 파인튜닝은 사전 학습된 모델의 특징 추출기와 분류기 중에서 분류기 부분을 새로운 분류 작업에 맞게 조정하는 것이다.

본 연구에 사용된 이력서 데이터의 특징(Feature)은 소범주형, 다범주형, 자연어 변수로 구분하였으며, 다범주형 변수인 경우 Conv1D(CNN), LSTM, Transformer의 임베딩을 적용하여 최적화 및 성능이 가장 좋은 모델을 선정하였다. 자연어 변수의 경우는 Universal-Sentence-Encoder와 XLM(Cross-lingual Language Model) Transformer 기반 사전 언어 학습 모델을 사용하여 모델 학습을 진행하고 최적화 및 성능이 가장 좋은 모델을 선정하였다.

4.2.3 Layers 관점

Layers 관점의 파인튜닝은 사전 학습된 모델의 일부 계층을 파인튜닝

에 참여시킬지를 선정하는 것으로서, 일부 계층은 고정하고 일부 계층만을 학습할 수 있도록 조정하는 것이다.

Layers 관점은 새로운 컬럼을 마지막에 Concatenate를 하는 방법과 드롭아웃(Dropout) 기법에 대해 0.5~0.8 사이의 최적값을 찾는 방법으로 딥러닝 모델을 구성하였다. Concatenate는 두 개 이상의 입력을 결합하는데 사용되는 연산이며, 중요도가 높은 컬럼을 마지막에 Concatenate를 하면 해당 컬럼의 가중치를 최대한 보존하면서 학습이 가능하다. 이에 따라 마지막에 Concatenate를 하는 경우, 해당 컬럼의 영향력이 높아질 수 있으므로, 해당 내용에 따라 중요도가 높은 컬럼은 마지막에 넣어 가중치 손실을 최소화한다.

2차 실험에서 추가된 “부서명”, “담당업무” 컬럼에 대하여 Concatenate를 마지막에 사용하여 딥러닝 모델을 구성하였다.

드롭아웃은 훈련 데이터에만 과도하게 학습되어 실제 데이터에 대한 일반화 성능이 떨어지는 오버피팅(Overfitting)을 방지하는 방법으로서, 일반화 성능을 향상시킬 수 있도록 0.5~0.8 사이의 최적값을 찾아 딥러닝 모델을 구성하였다.

4.3 하이퍼파라미터 선정 실험

4.3.1 가중치 초기화 기법

Xavier 초기화 혹은 Glorot 초기화라고도 불리는 가중치 초기화(Weight Initialization) 기법은 각 계층의 출력 분산이 입력 분산과 거의 같도록 가중치를 초기화하여 그레이디언트 소실(Gradient Vanishing) 또는 폭주(Gradient Exploding) 문제를 완화하고, 모델의 학습을 안정화한다.

본 연구에서는 Xavier 초기화의 Normal, Uniform 두 가지 방법을 활용하여 실험을 진행한다. Normal 방법은 초기화할 파라미터 값들의 범위

에서 평균은 0으로 유지하고, 분산 값을 루트($2 / (Input + Output)$)으로 조정하는 방법이다. Uniform 방법은 하한값과 상한값을 지정하여 그 사이 범위에서 파라미터 값을 초기화시키는 것이다.

본 연구 모델의 실험 결과는 Xavier_uniform을 사용하여 학습을 진행하였을 때, 0.005 높은 것으로 확인된다.

[표 4-2] Xavier_uniform, Xavier_normal

Idx	Xavier_uniform	Xavier_normal
1	0.8134	0.8096
2	0.8201	0.8151
3	0.8243	0.8233
4	0.8253	0.8249
5	0.8266	0.8261

4.3.2 임베딩 차원

딥러닝에서 임베딩 차원은 학습된 임베딩 공간에서 특징을 표현하는 데 사용되는 차원 수를 나타낸다. 임베딩은 자연어 처리를 위한 신경망, 컴퓨터비전, 추천 시스템 등 많은 딥러닝 모델에서 사용된다. 예를 들어, 자연어 처리를 위한 신경망에서 텍스트 말뭉치의 단어나 토큰은 고차원 공간에서 실수 벡터인 임베딩으로 표현될 수 있다. 임베딩 벡터의 크기, 즉, 임베딩 차원은 각 단어 또는 토큰을 표현하는데 사용되는 특징의 수를 결정하는 하이퍼파라미터이다. 임베딩 차원을 선택하는 것은 딥러닝 모델의 성능에 영향을 미칠 수 있으므로 중요하다. 임베딩 차원이 클수록 특징을 더 복잡하게 표현할 수 있지만, 더 많은 학습데이터와 계산 리소스가 필요할 수 있다. 임베딩 차원이 작을수록 학습 속도가 빨라질 수 있지만, 입력 데이터의 정확도 떨어지거나 표현력이 떨어질 수 있다.

본 연구에서는 임베딩 차원 선정을 64, 128, 256, 512 총 4가지 차원을

활용하여 실험을 진행하였으며, 512의 차원을 적용한 실험이 가장 높은 것으로 확인된다.

[표 4-3] 임베딩 차원

Idx	64	128	256	512
1	0.5092	0.5050	0.5081	0.4993
2	0.5204	0.5107	0.5186	0.5260
3	0.5289	0.5248	0.5297	0.5433
4	0.5373	0.5347	0.5385	0.5566
5	0.5456	0.5443	0.5487	0.5676

4.3.3 학습 배치 크기

배치 크기는 네트워크를 통해 전파될 샘플의 수를 정의한다. 예를 들어 1,000개의 학습 데이터가 있고 배치 크기를 100으로 설정한다는 가정을 하면 알고리즘은 학습 데이터셋에서 처음 100개의 샘플(1 ~ 100)을 가져와 네트워크를 학습한다. 그런 다음 두 번째 100개의 샘플(101~ 200)을 가져와 네트워크를 다시 학습한다. 모든 학습 데이터가 네트워크에 전파될 때까지 이 절차를 계속 수행할 수 있다.

본 연구 모델의 학습 배치 크기 실험은 16, 64, 128, 256, 512 총 5가지의 실험을 진행하였으며, 학습 배치 크기 128개를 적용한 실험이 가장 높은 것으로 확인된다.

[표 4-4] 학습 배치 크기 선정

Idx	16	64	128	256	512
1	0.8134	0.8058	0.7967	0.8135	0.7684
2	0.8222	0.8200	0.8152	0.8152	0.8152
3	0.8243	0.8272	0.8222	0.8175	0.8152
4	0.8244	0.8322	0.8367	0.8226	0.8153
5	0.8271	0.8440	0.8447	0.8402	0.8185

4.3.4 학습률

최적화 도구가 학습 프로세스 중에 모델의 매개 변수를 업데이트하는 단계의 크기를 결정하는 하이퍼파라미터이다. 학습률은 역전파 중에 계산된 추정 오류 기울기에 따라 모델의 매개 변수가 얼마나 변화하는지 제어한다. 학습률이 높으면 옵티마이저(Optimizer)가 최적값을 초과하여 학습이 불안정해지거나 동작이 달라질 수 있으며, 학습률이 낮으면 수렴 속도가 느려지고 학습 시간이 길어질 수 있다. 따라서 딥러닝 모델을 효과적으로 훈련하려면 적절한 학습률을 설정하는 것이 중요하다.

본 연구 모델의 학습률 선정은 $1e-2$, $1e-3$, $1e-4$, $1e-5$, $3e-5$ 총 5가지의 실험을 진행하였으며, 학습률 $1e-2(0.01)$ 을 적용한 실험이 가장 높은 것으로 확인된다.

[표 4-5] 학습률

Idx	0.01 ($1e-2$)	0.001 ($1e-3$)	0.0001 ($1e-4$)	0.00001 ($1e-5$)	0.00003 ($3e-5$)
1	0.7950	0.7276	0.4547	0.7555	0.8122
2	0.8152	0.8152	0.7807	0.8131	0.8138
3	0.8152	0.8152	0.8151	0.8138	0.8138
4	0.8164	0.8162	0.8157	0.8138	0.8138
5	0.8283	0.8246	0.8154	0.8139	0.8138

4.3.5 최적화 기법

최적화 기법은 딥러닝 모델의 학습 과정에서 손실 함수를 최소화하기 위해 모델의 파라미터를 조정하는데 사용되는 방법이며, 모델의 출력과 실제 값 사이의 오차를 계산하여 손실 함수를 얻고, 이 손실 함수를 최소화하는 방향으로 모델의 파라미터를 업데이트 한다.

RMSProp(Root Mean Square Propagation)는 경사하강법의 변형인 확

률적 경사 하강(Stochastic Gradient Descent, SGD)을 기반으로 기울기의 제곱에 대한 이동 평균을 사용하여 학습률을 조절한다. 즉, 큰 기울기에는 작은 학습률이 적용되고, 작은 기울기에는 큰 학습률이 적용된다.

Adam(Adaptive Moment Estimation)은 이전 기울기와 제곱 기울기의 평균을 사용하여 기하급수적으로 감소하는 평균을 유지하고, 이러한 추정치를 사용하여 학습 중에 각 매개 변수에 대한 학습 속도를 조정한다. 이를 통해 옵티마이저(Optimizer)는 각 파라미터에 대한 학습률을 개별적으로 조정할 수 있으므로 기존 최적화 기법에 비해 더 빠르게 수렴하고 더 나은 성능을 얻을 수 있다.

본 모델의 실험은 RMSProp 기법과 Adam 기법 두 종류의 최적화 기법을 사용하여 실험하였다.

[표 4-6] RMSProp, Adam 정확도 비교

Idx	RMSProp	Adam
1	0.7555	0.4229
2	0.8131	0.8095
3	0.8138	0.8138
4	0.8138	0.8138
5	0.8139	0.8139

4.4 딥러닝 모델 실험

본 연구는 CNN, LSTM, Transformer를 사용하여 전체 모델을 1차 실험과 2차 실험으로 구분하여 실험하였다. 1차 실험은 52개 특징(Feature)에서 추출된 179,589,881개의 파라미터와 하이퍼파라미터 설정으로 학습하였으며, 2차 실험은 1차 실험에서 사용한 52개 특징(Feature) 중 결측치가 높은 항목을 제거한 34개 특징(Feature)에서 추출된 348,629,913개의 파라미터와 파인튜닝 기법을 활용하여 학습하였다. 또한, 모든 실험에

는 동일한 최적화 기법(RMSProp)과 손실 함수(Binary_Crossentropy) 및 하이퍼파라미터 설정값을 적용하였으며, 데이터셋은 6:2:2로 분할하여 학습, 검증, 시험을 수행하도록 실험하였다.

[표 4-7]은 딥러닝 모델 실험의 진행에 있어 1차 실험과 2차 실험의 주요 차이점에 정리하였다.

[표 4-7] 딥러닝 모델 실험의 주요 차이점

	입력층	은닉층	출력층	매개변수	기타
1차 실험	31	100	1	179,589,881	하이퍼파라미터
2차 실험	23	135	1	348,629,913	파인튜닝
증감률	-34.78%	25.93%	0.00%	48.49%	

입력층은 결측치 항목을 삭제하여 34.78%로 감소하였으며 은닉층은 복잡한 패턴과 상호작용을 학습할 수 있도록 25.95% 증가되었다. 또한 뉴런(Neuron)의 연결 가중치와 편향을 나타내는 매개 변수(Parameter)는 48.49% 증가되었다.

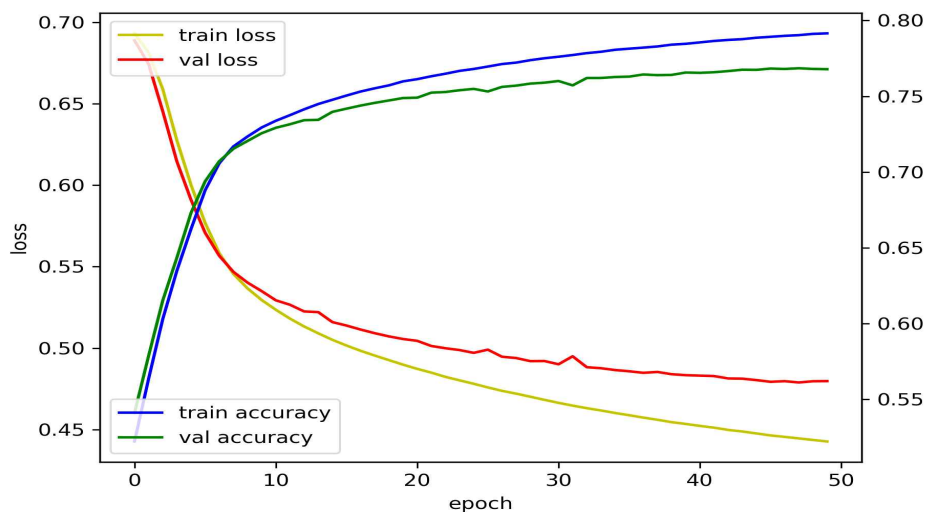
하이퍼파라미터는 1차 실험의 설정값을 그대로 2차 실험에 적용하였으며, 2차 실험에는 파인튜닝 기법을 추가하여 파인튜닝이 CNN, LSTM, Transformer 모델의 성능에 미치는 영향에 대하여 확인하고자 하였다.

4.4.1 합성곱 신경망(CNN)

[표 4-8] CNN 실험 결과

	epoch	Loss	Accuracy	AUC	val_Loss	val_Accuracy	val_AUC
1차	50	0.4446	0.7903	0.8723	0.4797	0.7685	0.8497
2차	44	0.4446	0.8219	0.9123	0.6297	0.7885	0.8997
증감률		0%	3.99%	4.58%	31.23%	2.57%	5.8%

[표 4-8]은 CNN 모델의 1차 실험과 2차 실험의 결과를 정리한 것이다.
CNN 기반의 1차 실험은 Accuracy = 0.7903, Loss = 0.4446, epoch = 50의 학습 모델이 최종 모델로 선정되었다.



[그림 4-1] CNN 1차 실험 그래프

[그림 4-1]의 CNN 학습 곡선을 살펴보면 epoch = 50에서 학습이 중단된 것을 확인할 수 있다. 이것은 train_loss는 지속적으로 감소하지만, 초기 종료(Early Stopping) 조건인 검증손실(val loss)이 epoch=49에서 다시 상승했기 때문임을 알 수 있다.

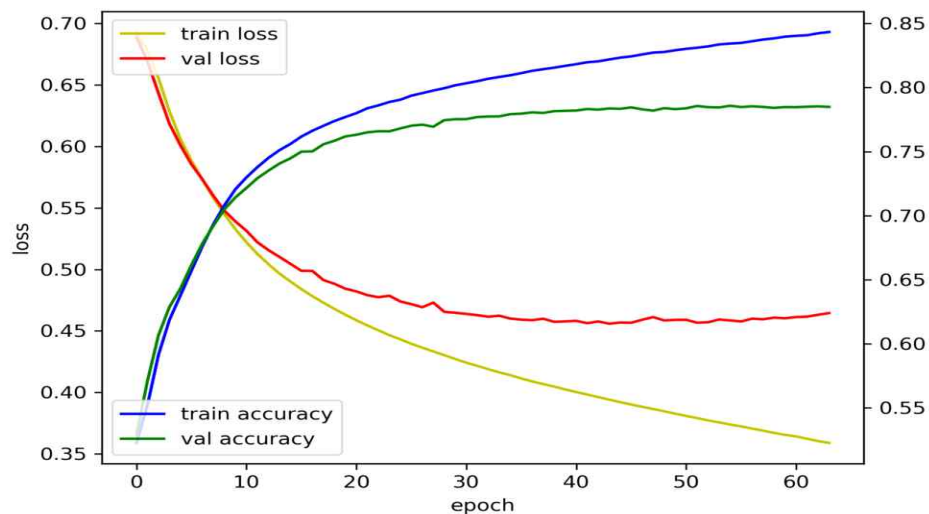
Loss = 0.4446과 val_Loss = 0.4797의 정확도가 0.0351로 근소한 차이를 보이는 것은 Training 데이터뿐만 아니라 Validation 데이터에서도 일관된 성능을 보인다는 의미이다. 즉, 모델이 과적합(Overfitting) 되지 않고 일반화(Generalization) 능력이 높다는 것을 나타낸다.

Accuracy = 0.7903과 AUC = 0.8723에서 AUC의 정확도가 더 높다는

것은 Positive와 Negative 클래스를 구분하는 성능평가에서 Negative 클래스를 구분하는 능력이 높은 것을 나타낸다.

CNN 1차 실험 모델은 [표 5-7]의 실험 결과 지표 간의 균형이 적절하므로 직무 추천 모델에 적용이 가능 할 것으로 보인다.

CNN 기반의 2차 실험은 Accuracy = 0.8219, Loss = 0.4446, epoch=44의 학습 모델이 최종 모델로 선정되었다.



[그림 4-2] CNN 2차 실험 그래프

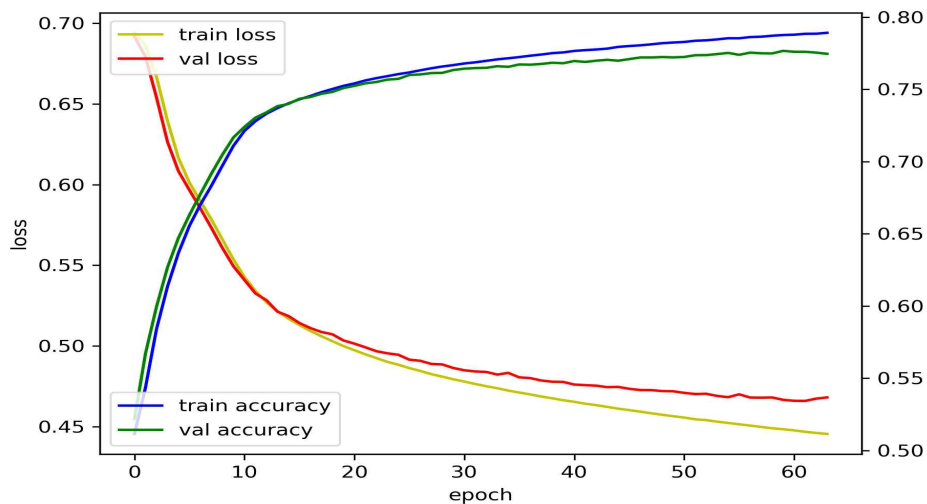
[그림 4-2]의 CNN 학습 곡선을 살펴보면 epoch=64에서 학습을 멈추었고 학습의 중지는 조기 종료 조건인 검증손실이 epoch=45에서 다시 상승했기 때문에 epoch=44의 학습 모델을 최종 모델로 종료하였다. val_Loss를 제외한 모든 지표의 성능이 향상되었으며, train_Loss는 계속 감소하지만, val_Loss는 감소하다가 어느 순간부터 증가하는 대표적인 과적합(Overfitting) 현상을 보여주고 있다. 그 원인으로는 일정 수준 이상이 입력층, 은닉층, 매개 변수의 증가는 CNN 모델의 한계로 추정된다.

4.4.2 장단기 메모리(LSTM)

[표 4-9] LSTM 실험 결과

	epoch	Loss	Accuracy	AUC	val_Loss	val_Accuracy	val_AUC
1차	64	0.4468	0.7885	0.8706	0.4659	0.7760	0.8579
2차	45	0.5068	0.8332	0.9006	0.6359	0.7760	0.8579
증감률		11.84%	5.36%	3.33%	26.73%	0.0%	0.0%

[표 4-9]는 LSTM 모델의 1차 실험과 2차 실험의 결과를 정리하였다. LSTM 기반의 1차 실험은 Accuracy = 0.7885, Loss = 0.4468, epoch=64의 학습 모델이 최종 모델로 선정되었다.



[그림 4-3] LSTM 1차 실험 그래프

[그림 4-3]에서 LSTM 학습 곡선을 살펴보면 epoch = 64에서 학습을 멈추었고 학습의 중단은 조기 종료(Early Stopping) 조건인 검증손실(val loss)이 epoch=63에서 다시 상승했기 때문이다. 조기 종료는 검증손실을

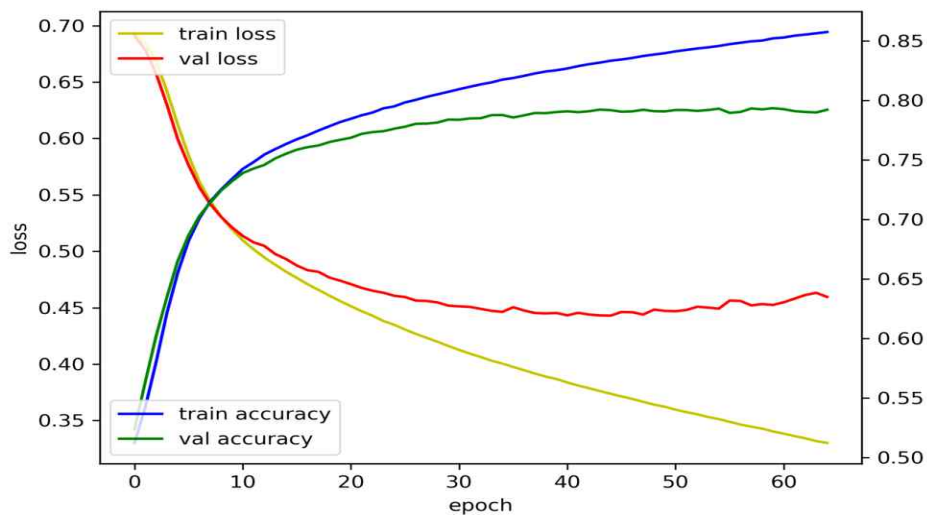
기준으로 patience=2, min_delta=0, mode=auto로 설정하여 val_loss 재상승을 저지하고, 학습 종료를 통하여 과적합(Overfitting)을 방지하였다.

Loss = 0.4468과 val_Loss = 0.4659의 정확도가 0.082로 근소한 차이를 보이는 것은 Training 데이터뿐만 아니라 Validation 데이터에서도 일관된 성능을 보인다는 의미이다. 즉, 모델이 과적합(Overfitting) 되지 않고 일반화(Generalization) 능력이 높다는 것을 나타낸다.

Accuracy = 0.7885과 AUC = 0.8706에서 AUC의 정확도가 더 높다는 것은 Positive와 Negative 클래스를 구분하는 성능평가에서 Negative 클래스를 구분하는 능력이 높은 것을 나타낸다.

LSTM 2차 실험 모델은 [표 4-9]의 실험 결과 지표 간의 균형이 적절하므로 직무 추천 모델에 적용이 가능할 것으로 보인다.

LSTM 기반의 2차 실험은 Accuracy = 0.8332, Loss = 0.5068, epoch=45의 학습 모델이 최종 모델로 선정되었다.



[그림 4-4] LSTM 2차 실험 그래프

[그림 4-4]의 LSTM 학습 곡선은 [그림 4-2]의 CNN 학습 곡선과 동일한 패턴을 확인할 수 있으며 [표 4-9]의 실험 결과 역시 val_Loss 증가로 인한 대표적인 과적합(Overfitting) 현상을 보여주고 있다.

4.4.3 트랜스포머(Transformer)

[표 4-10] Transformer 실험 결과

	epoch	Loss	Accuracy	AUC	val_Loss	val_Accuracy	val_AUC
1차	39	0.4443	0.7898	0.8726	0.4670	0.7759	0.8579
2차	100	0.4779	0.7637	0.8464	0.4879	0.7864	0.8386
증감률		7%	-3.3%	-3.0%	4.44%	1.35%	-2.2%

[표 4-10]은 Transformer의 1차와 2차 실험의 결과를 정리하였다.

Transformer 기반의 1차 실험은 Accuracy = 0.7898, Loss = 0.4443, epoch = 39의 학습 모델이 최종 모델로 선정되었다.

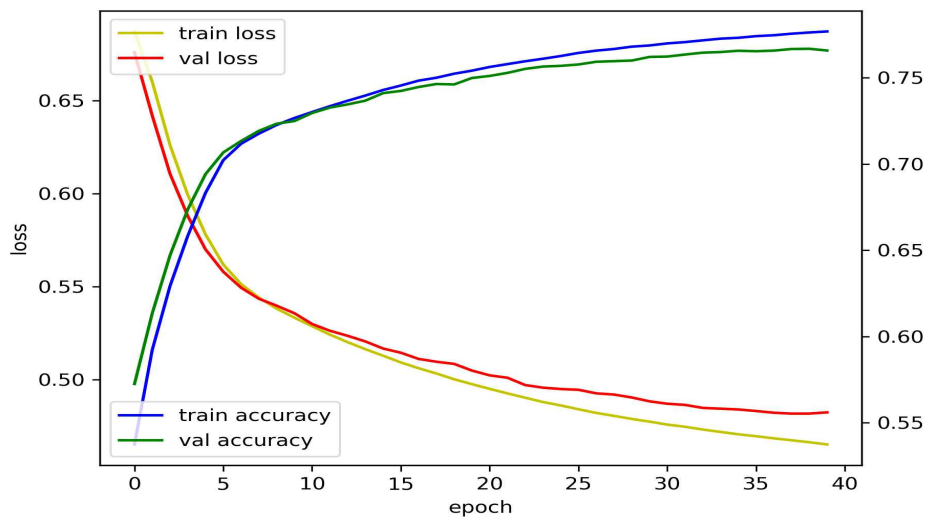
[그림 4-5]에서 Transformer 학습 곡선은 epoch = 39에서 검증손실(val loss)의 증가로 조기 종료(Early Stopping)된 것을 나타낸다.

Transformer의 epoch = 39는 CNN, LSTM 모델과 대비하여 낮은 수치이다. 물론 epoch가 높을수록 모델 학습 성능이 향상되는 것은 명확하지 않지만, 1차 실험에서 동일하게 사용한 입력층, 은닉층, 하이퍼파라미터가 Transformer 모델에는 적합하지 않을 수 있다는 추정이 가능하다.

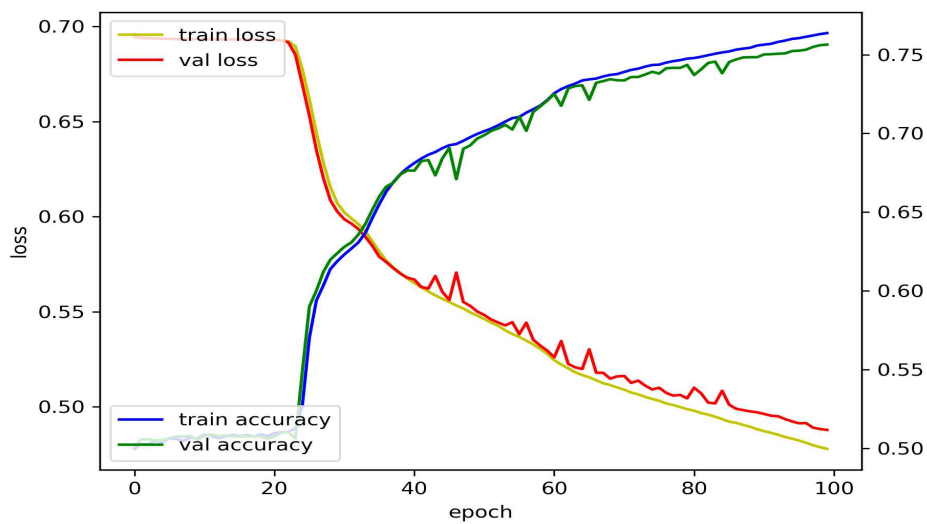
Loss = 0.4443과 val_Loss = 0.4670의 정확도가 0.0227로 근소한 차이를 보이는 것은 과적합(Overfitting) 되지 않고 일반화(Generalization) 능력이 높다는 것을 나타낸다. 또한, Transformer는 CNN, LSTM의 모델 중 Loss = 0.4443으로 가장 낮고, AUC = 0.8726와 val_AUC = 0.8579의 평균이 가장 높은 결과를 보였다. 하지만 Transformer, CNN, LSTM의

성능 지표의 차이가 근소하여 상대적인 차별성은 도출하지 못했다.

Transformer 기반의 2차 실험은 Accuracy = 0.7637, Loss = 0.4779,
epoch = 100의 학습 모델이 최종 모델로 선정되었다.



[그림 4-5] Transformer 1차 실험 그래프



[그림 4-6] Transformer 2차 실험 그래프

[그림 4-6]의 2차 실험은 설정값인 epoch=100까지 진행되어 종료되었으므로 Epoch의 설정값을 증가하여 추가 학습을 한다면 성능지표가 더욱 상승할 것으로 예상하며, Accuracy 상승은 동일했으나 train_loss와 val_loss가 안정적으로 성능 향상되었다. 이것은 1차 실험 성능보다 2차 실험 성능이 개선된 것을 확인할 수 있으며 그 원인으로는 파인튜닝과 특징(Feature)을 52개에서 34개로 감소한 것이 원인으로 추정된다.

4.5 실험 결과분석

4.5.1 하이퍼파라미터의 영향 분석

딥러닝 모델의 하이퍼파라미터는 모델 성능에 중요한 영향을 미치므로 하이퍼파라미터의 조정을 통해 모델의 정확도, 학습 속도 등을 개선할 수 있다. 본 연구 실험에는 하이퍼파라미터 조정의 항목은 가중치 초기화, 임베딩 차원 설정, 학습 배치 크기 선정, 학습률, 최적화 기법을 사용하여 실험한다. 먼저, 가중치 초기화 기법은 Xavier_uniform을 사용하여 학습되며, 임베딩 차원 설정은 512차원, 학습 배치 크기 선정 128개, 학습률 1e-3, 최적화 기법, RMSProp를 사용하여 학습한다,

[표 4-11] 하이퍼파라미터 선정 결과

항목	선정 기법
가중치 초기화 기법	Xavier_uniform
임베딩 차원 설정	512
학습 배치 사이즈 선정	128
학습률	1e-3
최적화 기법	RMSProp

4.5.2 딥러닝 모델의 비교 분석

딥러닝 모델의 비교 분석은 모델의 성능을 평가하기 위해 정확도, 재현

율, 정밀도 등의 지표를 사용한다. 이러한 지표는 모델이 얼마나 잘 예측하는지에 대한 정보를 제공한다.

[표 4-12] 1차 실험 모델의 성능평가

	CNN	LSTM	Transformer
Loss	0.4446	0.4468	0.4443
Accuracy	0.7903	0.7885	0.7898
AUC	0.8723	0.8706	0.8726
val_Loss	0.4797	0.4659	0.4670
val_Accuracy	0.7685	0.7760	0.7759
val_AUC	0.8497	0.8579	0.8579

[표 4-13] 2차 실험 모델의 성능평가

	CNN	LSTM	Transformer
Loss	0.4446	0.5068	0.4779
Accuracy	0.8219	0.8332	0.7637
AUC	0.9123	0.9006	0.8464
val_Loss	0.6297	0.6359	0.4879
val_Accuracy	0.7885	0.7760	0.7864
val_AUC	0.8997	0.8579	0.8386

실험 모델의 성능평가에 대한 설명을 통해 각 모델의 학습 곡선과 성능향상을 비교하고 실험 결과를 분석하였다. 실험에서는 CNN, LSTM, 그리고 Transformer 모델을 사용하였으며, 특징 개수가 52개인 1차 실험과 특징 개수가 34개인 2차 실험을 단계별로 진행하였다.

1차 실험에서는 CNN과 LSTM 모델이 비슷한 성능을 보였으며, Transformer 모델이 약간 더 우수한 성능을 나타냈다.

2차 실험에서는 CNN과 LSTM 모델은 학습 초기에 낮은 정확도와 높은 손실로 시작하여, 학습이 진행됨에 따라 성능이 향상되는 경향을 보였다.

반면에 Transformer 모델은 학습 초기부터 높은 정확도와 낮은 손실을 보여주며, epoch=100으로 학습을 조기에 종료했음에도 불구하고 높은 성능을 유지하였다.

Transformer 모델은 epoch=100 이상으로 추가적인 학습을 진행시 높은 성능을 도출할 수 있다고 예상된다.

모델의 성능평가에서 AUC 지표는 모델의 이진 분류 능력을 평가하는데 중요한 역할을 했다. CNN과 Transformer 모델은 AUC가 높은 성능을 보였는데, 이는 모델이 Positive와 Negative 클래스를 잘 구분하는 능력이 높다는 것을 의미한다. 따라서, CNN과 Transformer 모델이 전반적으로 더 우수한 성능을 보였다. 또한, Transformer 모델은 학습 초기부터 안정적인 성능을 보여주어서 조기 종료 기법을 적용하더라도 높은 성능을 유지할 수 있다.

실험 결과를 종합해보면, 1차와 2차 실험에서 특징 개수의 차이가 모델의 성능에 큰 영향을 미쳤습니다. 특징 개수가 적은 2차 실험에서는 모델의 성능이 상승되었으며, 불필요한 데이터를 처리하고 구직자의 개인적인 Feature를 추가하여 모델이 따라서, 모델 개발 시 특징 개수를 적절히 선택하고 최적의 특징 조합을 고려하는 것이 중요하다. 또한, 실험에서는 CNN, LSTM, 그리고 Transformer 모델을 비교하였는데, 세 모델 모두 일정 수준의 성능을 보였다. 하지만, Transformer 모델이 가장 우수한 성능을 나타내었으며, 이는 Transformer의 셀프 어텐션 메커니즘이 시퀀스 데이터 처리에 효과적이기 때문일 것으로 예상된다. 따라서, 텍스트 데이터와 같은 시퀀스 데이터를 다루는 경우에는 Transformer 모델을 고려해볼 가치가 있다.

마지막으로, 최적화 기법으로는 RMSProp을 사용하였는데, 이는 모델의 학습을 효과적으로 수행하기 위해 사용되는 알고리즘이다. 하지만, 다른

최적화 기법을 사용하거나 하이퍼파라미터를 조정함으로써 모델의 성능을 더욱 개선할 수 있을 것이다. 따라서, 최적화 알고리즘과 하이퍼파라미터 설정에 대한 추가적인 실험과 조정을 수행하여 모델의 성능을 더욱 향상시킬 수 있을 것으로 기대한다.

4.5.3 추론 모델의 비교 분석

모델 학습 후 추론 모델을 만들어 재현율, 정밀도, F1_Score를 평가 지표로 하여 테스트 데이터를 활용하여 성능평가를 진행한다. 성능평가에 대해 어떤 지표가 가장 중요한지는 모델에 따라 다르므로, 상황에 따라 적절한 지표를 선택하여 모델의 성능을 평가한다.

본 연구 모델의 직무 추천 시스템은 재현율을 사용해서 평가한다. 재현율은 실제 Positive인 데이터 중 모델이 Positive로 예측한 데이터의 비율을 나타낸다. 다시 말해, 모델이 얼마나 실제 Positive 데이터를 빠짐없이 예측할 수 있는지를 나타내는 지표이다. 재현율은 Transformer 모델이 83%로 가장 높았다. Transformer는 실제 정답값을 빠짐없이 찾아내는 데 성공했다는 의미이다.

[표 4-14] 1차 실험 추론 모델의 성능평가

	CNN	LSTM	Transformer
Precision	0.55	0.26	0.29
Recall	0.50	0.80	0.83
F1_Score	0.52	0.39	0.43

[표 4-15] 2차 실험 추론 모델의 성능평가

	CNN	LSTM	Transformer
Precision	0.24	0.13	0.26
Recall	0.93	0.91	0.94
F1_Score	0.39	0.18	0.43

제 5 장 결론

5.1 연구의 요약 및 결론

본 연구는 구직자들의 전공에 기반을 둔 직무와 산업에서 요구하는 직무 간의 차이로 발생하는 반복적인 미스매칭 중 직무 미스매치에 대한 연구로 직무 미스매치를 취업포탈의 이력서 데이터와 NCS 직무 분류 데이터를 활용하여 딥러닝 기반 직무 추천 시스템을 개발하고, 성능평가를 통한 최적의 모델 선정과 실제 시스템에 적용할 수 있는 연구를 제안하는 것이다.

직무 미스매치 문제를 해소하기 위한 딥러닝 기반의 직무 추천 모델은 협업 필터링을 이용하였으며, 이력서 데이터를 학습 데이터로 구성하는데 있어 2가지 가설을 설정하였다. 첫째, 경력자의 이력서 데이터는 구직자의 직무 선택의 기준 지표를 포함하고 있다. 둘째, 구직자와 유사한 프로파일을 보유하고 있는 경력자의 이력서 데이터를 기반으로 직무를 추천하였을 때 가장 높은 만족도와 직무 추천 성과를 내포한다는 것이다.

이에 따른 단계별 연구 진행을 위하여 1단계로 경력자 중심의 이력서 데이터를 기반으로 52개의 특징과 34개의 특징을 갖는 2종의 학습 데이터셋을 구축하였으며, 2단계로 텍스트 임베딩 기법과 딥러닝 모델을 활용하여 과인튜닝을 통한 직무 추천 모델을 설계하고, 그리드 탐색 실험으로 최적의 하이퍼파라미터를 선정하였다. 마지막 3단계로 직무 추천 시스템의 사용자 만족도를 높이기 위해 추천된 직무를 NCS 직무 분류 체계와 조합하였다.

본 연구는 딥러닝 모델(CNN, LSTM, Transformer)의 성능평가를 통한 최적의 모델 선정을 위하여, 52개의 특징과 하이퍼파라미터를 중심으로 하는 1차 실험과 34개 특징과 과인튜닝을 중심으로 하는 2차 실험의

성능을 측정하였다.

1차 실험의 학습모델 성능평가 결과로는 직무 추천 모델의 성능 차이를 확인하지 못하였으나, 상대적으로는 CNN 모델의 정확률(Accuracy)이 0.7903으로 가장 우수하였으며, 추천 모델 성능은 Transformer 모델의 재현률(Recall)이 0.83으로 가장 우수하였다. 또한, 자연어 처리에 강점이 있다고 알려진 Transformer 모델이 예상과는 다르게 다소 미흡한 성능을 보였으며, CNN 모델이 상대적으로 가장 좋은 성능을 나타냈다. CNN 모델이 LSTM, Transformer 모델과 비교하여 상대적으로 높은 성능평가 결과가 나타난 원인은 CNN 모델을 기준으로 선정한 하이퍼파라미터 설정값을 LSTM, Transformer 모델에 동일하게 적용하였기 때문이며, 이러한 결과를 통하여 딥러닝 모델의 우수성보다 하이퍼파라미터 선정이 모델 성능에 미치는 효과가 크다는 것을 확인하였다.

2차 실험의 학습모델 성능평가 결과는 1차 실험과 비교하여 평균 5% 수준의 상승이 있었으며, 성능 향상의 주요 요인으로는 특징(Feature)을 52개에서 34개로 축소하여 학습 데이터셋을 최적화한 것과 파인튜닝 과정으로 모델 설계를 진행한 것이라 할 수 있다. 특히, 파인튜닝을 통해 재설계된 Transformer의 Epoch는 1차 실험에서 epoch=39로 중단되었지만, 2차 실험에서는 epoch=100까지 도달하였다. 이러한 결과로 파인튜닝을 통한 모델 설계와 양질의 이력서 데이터 확보는 직무 추천 모델의 성능과 직접적인 관계가 있다는 것을 확인하였다. 또한, NCS 직무 분류체계와 조합되는 직무 추천은 취업포털 뿐만 아니라, 민간 교육기관에서도 구직자 및 교육생에게 추천된 직무와 조합되는 직무 정보를 제공하므로 궁극적으로는 직무 미스매치 문제 해결에 도움이 될 것으로 기대한다.

5.2 연구의 한계 및 향후 연구

이력서를 기반으로 한 직무 추천 연구는 민감한 정보인 이력서 데이터를 활용하여 구직자에게 적합한 직무를 추천하기 위한 것이다.

본 연구를 진행하면서 도출된 대표적인 문제점은 이력서 데이터의 수집과 관리, 그리고 학습 모델의 컴퓨터 자원의 확보라고 할 수 있다. 이력서 데이터를 수집 및 관리하는 관련 기업이 아니라면, 개인정보보호법을 준수하면서 학습 데이터로 활용하기 위한 충분한 데이터의 수집에 한계가 있으며, 수집된 이력서 데이터의 개인정보 유출 등의 관리 위험이 존재하여, 연구에 어려움이 있다. 또한, 새로운 직무에 대한 정확성과 완결성이 미흡한 학습 데이터는 직무 추천 모델의 성능에 영향을 미치므로 지속적인 연구를 위해 새로운 이력서 데이터를 수집하고 정제하여 데이터 품질을 지속적으로 유지하는것에 어려움이 있다.

그리고, 구직자의 경험, 기술, 선호도 등은 시간이 지남에 따라 변화하므로 이력서를 기반으로 하는 추천 시스템은 구직자의 동적 변화에 대응하기에는 한계가 있으며, 추가되는 이력서 데이터를 고려한 학습 모델의 학습 주기를 면밀히 고려해야 한다.

향후 연구에서는 Transformer의 Epoch 학습을 지속할 수 있는 환경을 구성하고, 사용성 검증이 필요할 것이며, 직무 추천과 연계된 직무 역량 강화를 위한 학습 콘텐츠 추천 모델로의 확장 연구도 고려해 볼 수 있을 것이다.

참고문헌

[국내문헌]

- 김광석. (2020). *RNN 기반 모바일 결제 데이터를 이용한 사용자 구매 예측 방법*. 성균관대학교 석사학위논문.
- 김우주, 김동희, & 장희원. (2016). Word2vec 을 활용한 문서의 의미 확장 검색방법. *한국콘텐츠학회논문지*, 16(10), 687-692.
- 박상현. (2017). *토픽모델링과 인공신경망에 기반한 온라인 쇼핑물 리뷰 데이터 분류 및 응용*. 경희대학교 대학원 석사학위논문.
- 박수상. (2016). *협업적 필터링을 활용한 추천 채용 시스템의 설계와 구현*. 서울대학교 석사학위논문.
- 박재홍. (2020). *구인-구직 간 일자리 미스매치 실태와 개선방안에 관한 연구*. 부산대학교 대학원 석사학위논문.
- 백용선, & 김용수. (2010). 선택적 학습률을 활용한 학습법칙을 사용한 신경회로망. *한국지능시스템학회 논문지*, 20(5), 672-676.
- 송희석. (2020). 인재매칭을 위한 내용기반 척도학습모형의 설계. *Journal of Information Technology Applications & Management*, 27(6), 141-151.
- 오소진. (2022). *BERT 기반의 전이학습 모델을 적용한 양방향 인재매칭 시스템*. 한남대학교 박사학위논문.
- 우영춘, 이성엽, 최완, 안창원, & 백옥기. (2019). 디지털 헬스케어 데이터 분석을 위한 머신 러닝 기술 활용 동향. *[ETRI] 전자통신동향분석*, 34(1), 0-0.
- 유소엽, & 정옥란. (2019). BERT 모델과 지식 그래프를 활용한 지능형

- 챗봇. *한국전자거래학회지*, 24(3), 87-98.
- 유용민. (2018). *Doc2vec과 문서 군집기법을 적용한 카테고리 자동생성*. 인하대학교 석사학위논문.
- 이모세, & 안현철. (2018). 효과적인 입력변수 패턴 학습을 위한 시계열 그래프 기반 합성곱 신경망 모형: 주식시장 예측에의 응용. *지능정보연구*, 24(1), 167-181.
- 이시영. (2021). *뉴스 기사 텍스트 임베딩을 이용한 딥러닝 기반 기업성과 예측 모델 연구*. 숭실대학교 박사학위논문.
- 이창기, 김준석, & 김정희. (2014). 딥 러닝을 이용한 한국어 의존 구문 분석. *제 26 회 한글 및 한국어 정보처리 학술대회*, 87-91.
- 장석인. (2017). 제 4 차 산업혁명 시대의 산업구조 변화 방향과 정책과제. *국토연구원*, 424, 22-30.
- 장예화, 이병현, 정재호, & 김재경. (2021). MBTI 성격유형을 반영한 심층 신경망 기반 직무 추천 서비스. *인터넷전자상거래연구*, 21(4), 99-113.
- 조경우, 정용진, & 오창현. (2021). 미세먼지 농도 예측을 위한 딥러닝 알고리즘별 성능 비교. *한국향행학회논문지*, 25(5), 409-414.
- 조운환, 서영덕, 박대준, & 정제창. (2016). DNN 에서 효율적인 학습을 위한 활성화 함수에 대한 고찰. *대한전자공학회*, 2016(11), 800-803
- 최영웅. (2022). *Diff LSTM 알고리즘 기반의 미래 생활인구 예측 정확도 향상기법*. 성균관대학교 석사학위논문.
- 최희열, & 민윤희. (2015). Dropout 알고리즘에 대한 이해. *정보과학회지*, 33(8), 32-38.

- 하만석, & 안현철. (2019). 정형 데이터와 비정형 데이터를 동시에 고려하는 기계학습 기반의 직업훈련 중도탈락 예측 모형. *한국콘텐츠학회 논문지*, 19(1), 1-15.
- Kwak, M. J., Park, K. T., Park, J. W., & Kang, B. S. (2019). A development of longitudinal and transverse springback prediction model using artificial neural network in multipoint dieless forming of advanced high strength steel. *한국소성가공학회 학술대회 논문집*, 189-189.
- Zhang, Yihua. (2021). *구직자의 성격을 반영한 DNN기반 직무 추천 시스템*. 경희대학교 석사학위논문.
- 경향신문. (2023). 청년 50만명, 구직·취준 않고 ‘그냥 쉬었다’...역대 최대. <https://m.khan.co.kr/economy/economy-general/article/202303201715021#c2b>.
- 고용노동부. (2023). 2023년 일자리 예산 30.3조원, 미래 경쟁력 확보와 고용취약계층 노동시장 진입 중점 편성. https://www.moel.go.kr/news/enews/report/enewsView.do?news_seq=14478.
- 머니투데이. (2022). 이정식 고용장관 "일자리 미스매치·양극화 문제 해결 위해 노력". <https://news.mt.co.kr/mtview.php?no=2022100509285250903>.
- 사람인. (2022). MZ 세대, 10명 중 3명은 1년 안에 회사 떠난다. https://www.saramin.co.kr/zf_user/hr-magazine/view?hr_idx=953.
- 한국경제연구원. (2021). 청년 대졸자 고용률 75.2% OECD 37개국 중 31위. http://www.keri.org/web/www/news_02?p_p_id=EXT_BBS&p_p_lifecycle=0&p_p_state=normal&p_p_mode=view&_EXT_BBS_struts_a

ction=%2Ftext%2Fbbs%2Fview_message&_EXT_BBS_messageId=356280#3.

[국외 문헌]

- Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.
- Caruana, R., Lawrence, S., & Giles, C. (2000). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. *Advances in Neural Information Processing Systems*, 13.
- Cheng, J., Dong, L., & Lapata, M. (2016). Long Short-Term Memory-Networks for Machine Reading. In *2016 Conference on Empirical Methods in Natural Language Processing*, 551-561. Association for Computational Linguistics.
- Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder - Decoder Approaches. *Syntax, Semantics and Structure in Statistical Translation*, 103.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(ARTICLE), 2493-2537.
- Das, M., De Francisci Morales, G., Gionis, A., and Weber, I. (2013), Learning to question : Leveraging user preferences for shopping

- advice, In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 203–211.
- Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4, 133–151.
- He, K., Girshick, R., & Dollár, P. (2019). Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4918–4927.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Hossain, M. D., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020). LSTM-based network attack detection: performance comparison by hyper-parameter values tuning. In *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* 62–69. IEEE.
- Huang, Z., Chen, H., Hsu, C. J., Chen, W. H., & Wu, S. (2004). Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems*, 37(4), 543–558.
- Kim, D. G., Park, Y. S., Park, L. J., & Chung, T. Y. (2019). Developing of new a tensorflow tutorial model on machine learning: focusing on the Kaggle titanic dataset. *IEMEK Journal of Embedded Systems and Applications*, 14(4), 207–218.

- Kingma, D., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.
- Kline, D. M., & Berardi, V. L. (2005). Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing & Applications*, 14, 310–318.
- Kumar, S. K. (2017). On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*.
- Lakshmi, S. S., & Lakshmi, T. A. (2014). Recommendation systems: Issues and challenges. *International Journal of Computer Science and Information Technologies*, 5(4), 5771–5772.
- Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 1188–1196. PMLR.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X., & Chen, E. (2015). Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- Lin, Z., Feng, M., dos Santos, C., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). A structured self-attentive sentence embedding. In *International Conference on Learning Representations*. International Conference on Learning Representations, ICLR.
- Maheshwary, S., & Misra, H. (2018). Matching resumes to jobs via

- deep siamese network. In *Companion Proceedings of the The Web Conference 2018*, 87–88.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Parikh, A., Täckström, O., Das, D., & Uszkoreit, J. (2016). A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2249–2255.
- Paulus, R., Xiong, C., & Socher, R. (2017). A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.
- Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. *The adaptive web: methods and strategies of web personalization*, 325–341.
- Reimers, N., & Gurevych, I. (2017). Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*.
- Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.
- Sohrabi, B., Vanani, I. R., & Shineh, M. B. (2018). Topic modeling and classification of cyberspace papers using text mining. *Journal of Cyberspace Studies*, 2(1), 103–125.

- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27.
- Takase, T., Oyama, S., & Kurihara, M. (2018). Effective neural network training with adaptive learning rate based on training loss. *Neural Networks*, 101, 68–78.
- Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSE: *Neural Networks for Machine Learning*, 4(2), 26–31.
- Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37, 141–188.
- Valverde-Rebaza, J. C., Puma, R., Bustios, P., & Silva, N. C. (2018). Job Recommendation Based on Job Seeker Skills: An Empirical Study. In *Text2Story@ ECIR*, 47–51.
- Vargas, M. R., De Lima, B. S., & Evsukoff, A. G. (2017). Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on cOmputational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 60–65. IEEE.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Wu, Y. H., & Chen, A. L. (2000, April). Index structures of user

profiles for efficient web page filtering services. In *Proceedings 20th IEEE International Conference on Distributed Computing Systems*, 644–651. IEEE.

Yazan, E., & Talu, M. F. (2017). Comparison of the stochastic gradient descent based optimization techniques. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1–5. IEEE.