

KENTECH X 에너지 동아리 탐구 자료 제출 양식

본 양식은 에너지 동아리 탐구 자료 작성 시 사용하는 가이드입니다. 이 템플릿은 전체 원고(텍스트, 표, 그래픽)의 작성 방법을 상세히 기술하고 있습니다. 학생들은 이 파일을 활용하여 자신의 연구를 체계적으로 전달할 수 있습니다.

양식의 활용

1. 양식의 표시된 텍스트에 원고를 바로 작성할 수 있습니다.
2. 텍스트 안에 그래픽을 삽입하거나 그림, 도식 또는 표로 삽입하려면 새 줄을 만들고 원하는 위치에 그래픽을 삽입합니다. 크기를 맞게 조정해야 하는 경우 그래픽 프로그램에서 그래픽 크기를 다시 조정하고 템플릿에 아트웍을 다시 붙여넣습니다 (단일 열 아트웍의 경우 최대 너비 3.3인치(8.5cm), 이중 열 그래픽의 경우 최대 너비 7인치(17.8cm)).
3. 원고를 제출하기 전에 모든 페이지에 페이지 번호가 있는지 확인하세요.
4. 이 지침과 해당 제출물에 필요하지 않은 섹션을 삭제합니다.
5. 다른 이름으로 저장(파일 메뉴)을 선택하고 .dotx 템플릿 파일이 아닌 문서 파일로 저장합니다.
6. 파일 이름에는 학교, 제목, 작성자가 표기되어야 합니다. (예: 켄텍고등학교_탐구제목_작성자)
7. 자신으로부터 비롯되지 않은 모든 아이디어는 출처가 표기되어야 합니다.

손글씨 숫자를 인식하기 위한 다층 퍼셉트론의 입력과 가중치

최지원¹, 서은상^{*2}

¹해룡 고등학교 ²해룡 고등학교

ABSTRACT: 단일 퍼셉트론만으로 해결할 수 없는 복잡한 문제를 어떻게 다층 퍼셉트론으로 구현할 수 있는지 이해한다. 문서를 이해하고 손글씨를 통한 검색을 용이하게 하기 위해서는 다층 퍼셉트론을 통한 손글씨 인식 모델의 구현이 필요하다. 이를 구현해내기 위해서는 선의 굵기, 잉크의 농도, 글자의 가로세로 비율 등의 입력값을 설정해야한다. 실험 결과 모델은 대부분의 텍스트 이미지를 예측하였으며, softmax 함수를 통해 개별 예측 신뢰도 또한 확인할 수 있었다.

상대를 대할 때의 표정 등 다양한 기준이 필요하므로 비선형적 문제라 할 수 있다.

손글씨 인식 모델은 종이 문서의 손글씨를 자동으로 텍스트로 변환하고, 그를 기반으로 한 검색을 용이하게 한다. 손글씨 인식 모델을 구현하기 위해서는 선의 굵기, 검은색 픽셀 수에 따른 잉크의 면적, 해당 이미지 안에서 글자가 차지하는 크기, 잉크의 농도(진한 정도), 글씨의 가로 세로 비율 등의 입력값이 필요하다.

서론 (Introduction)

인공 신경망은 생물학적 뉴런처럼, 입력된 입력값과 그 각각의 가중치를 곱해 더한 가중합이 임계치를 도달하는 경우, 최종적으로 결과값을 출력한다. (예를 들어 2개의 입력값이 있을 때 첫 번째 입력과 두 번째 입력 각각의 가중치가 0.5일 때 가중합은 $(1 \times 0.5 + 1 \times 0.5 = 1)$ 이다. 만약 모든 입력이 0에 가깝다면 각각 가중치를 곱해도 가중합이 0에 가까워지기 때문에 정확한 임계치에 도달하기 어렵고, 따라서 정확한 출력이 어려워진다. 이러한 상황을 대비하여 퍼셉트론에서는 임계선을 조절하여 보다 정확한 결정을 내릴 수 있게 하는 편향(bias)이 존재한다. 활성화 함수는 퍼셉트론의 출력을 결정하는 여러 종류의 함수인데, 예를 들어 활성화 함수 중 계단 함수는 입력값이 참일 때 1, 거짓일 때 0을 출력하고(인공신경망의 기초: 퍼셉트론의 이해) 시그모이드 함수는 0과 1사이의 출력값을 출력한다. 보고서의 실험에서는 ReLU 함수와 softmax 함수를 활성화 함수로 사용하였다. ReLU 함수는 가중합이 음수일 때는 출력하지 않고, 양수일 때 출력한다. 인공신경망의 마지막 판단 단계를 담당하는 softmax 함수는 실제 숫자와 일치하는지에 관한 가중합 점수를 확률로 바꾼다.

퍼셉트론의 구현의 간단한 예시로서 and, nor 논리 게이트를 들 수 있다. and 게이트는 입력값 모두가 참(1)일 때만 임계치에 도달하여 참(1)을 출력하고, 하나라도 참이 아닌 거짓일 경우(0) 결과값으로 거짓(0)을 출력해낸다. 단순한 직선으로 해결할 수 없는 비선형적 문제는 여러 퍼셉트론을 조합한 다층 퍼셉트론으로 구현할 수 있다. 여기서 단순한 직선으로 해결할 수 없는 비선형적 문제란, 한 가지 기준으로 참과 거짓을 판별할 수 없는 문제를 말한다. 예를 들어 숫자만으로 한 사람의 키를 판별하는 문제와 달리 한 사람의 사회성을 판별하는 데에는 목소리 크기, 말투,

본 보고서에서는 인공지능 기반 알고리즘의 기계 학습 모델인 손글씨 숫자 인식 다층 퍼셉트론을 구현하기 위한 입력값과 가중치를 Python을 통해 조정하고자 한다. 이때 0~9 사이의 손글씨 숫자를 인식할 수 있는 모델을 구현하도록 한다.

이론적배경 (Background)

1. 신경망

인간 뇌는 뉴런 약 1000억 개를 가지고 있다. 뉴런은 혈액의 아미노산으로부터 신경전달물질을 만들어 주위의 다른 세포들에게 신호를 전달한다. 이때 하나의 신경세포는 여러 개의 다른 신경세포로부터 전달받은 입력신호에 반응하고, 그 값이 임계값에 도달했을 경우 다음 신경 세포에도 일정한 크기의 출력신호를 전달한다. 입력은 가지돌기가, 출력은 축삭돌기를 담당한다. 가지돌기의 신호는 세포체와 축삭돌기를 통과해 축삭 말단에서 다시 다른 뉴런의 가지 돌기로 전달되는데, 서로 다른 뉴런의 축삭말단과 가지돌기 사이를 시냅스라 부른다. 축삭말단에서 가지돌기 사이의 틈을 통과하기 위해서는 화학적 형태의 신호가 필요한데, 이때의 화학적 형태가 도파단, 세로토닌과 같은 신경 전달 물질이다. 신호는 축삭말단의 소낭들이 터져 가지돌기의 수용체로 전달된다.(유튜브 15분 만에 정리하는 뇌과학(신경과학) 입문 [북툰 과학다큐])

2. 인공신경망과 퍼셉트론

인공신경망은 인간의 뉴런구조를 착안한 문제 해결 능력을 가진 인공지능 모델이다. 인공신경망은 생물이 뇌에서 신호를 전달하는 방식, 즉 뉴런의 자극전달 방식을 모방한 모델로서 1940년대부터 그 연구가 시작되었다. 2025년 시장 규모 미화 약 25.85억 달러에서 시작하여 2034년 미화 약 142.01억까지의 성장세가 예측되었기 때문에, 앞으로의 인공지능 발전 방향을 모색하기 위해서는 인공신경망에 대한 명확한 이해와 연구가 필요하다. 인공신경망의 가장 기본적인 구성

요소는 퍼셉트론인데, 여기서 퍼셉트론이란 생물학적 뉴런의 구조를 단순화하여, 것처럼 작동하도록 설계된 수학적 알고리즘이다.

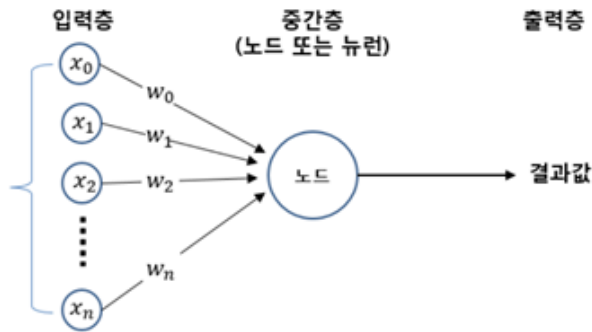


Figure 1. 퍼셉트론의 구조조

Figure 1과 같이 퍼셉트론은 1000억개의 생물학적 뉴런이 여러 입력을 받아 특정 기준치에 도달하면 다른 뉴런으로 출력해 전달하는 방식을 수학적 구조로 묘사한 모델이기 때문에 인공지능망의 기본 구성 단위라 볼 수 있다. 입력값은 가지돌기에 들어오는 입력을, 퍼셉트론이 가중합이 임계치를 도달하는지의 여부를 판단하는 과정은 세포체와 축삭돌기를 통과하는 과정이 구현되었다고 볼 수 있다.

3. 다층 퍼셉트론

단일 퍼셉트론이 뉴런 하나의 입력과 출력을 구현한 모델이라면, 다층 퍼셉트론은 뉴런과 다른 뉴런 사이의 신경 전달 과정을 구현해낸다. 예를 들어 손글씨 숫자 3과 8을 구분하려면 곡선의 교차점, 기울기와 열리고 닫히는 형태 등 여러 비선형적 특징을 종합해야한다.(인공신경망의 기초: 퍼셉트론의 이해) 다층 퍼셉트론의 경우에는 단일 퍼셉트론과 달리 퍼셉트론과 퍼셉트론 사이의 연결이 존재하는데, 이때의 연결은 생물학적 뉴런의 가지돌기와 축삭말단 사이의 시냅스를 구현했다고 볼 수 있다.

4. MNIST

MNIST(Modified National Institute of Standards and Technology)란 손글씨 숫자 이미지로 이루어진 대규모 데이터 베이스를 의미한다. 28 x 28 픽셀 크기(작은 정사각형 784개, 픽셀의 숫자는 밝기값을 나타낸다. 이때 픽셀의 밝기값은 0-255)인 70,000장의 흑백 이미지로 구성되어 있다. 이미지 픽셀값을 입력하면 모델이 숫자를 예측하는데, MNIST의 각 라벨값과 비교하여 학습한다.

재료 및 방법 (Materials and Methods)

구현한 모델 학습에는 TensorFlow와 Keras 라이브러리를 사용하였으며, 데이터는 공개된 MNIST(Mixed National Institute of Standards and Technology) 손글씨 숫자 데이터셋을 활용하였다. 모델은 입력층, 은닉층, 출력층으로 구성하였다. 입력층에서는 28x28 크기의 이미지를 1차원으로 펼쳐 784개의 각 이미지의 픽셀값을 사용하였다. 은닉층은 128개의 입력을 조합해 학습하는 은닉 노드를 가지며, 가중합이 양수일 때 출력하는 ReLU(Rectified Linear Unit) 함수를 활성화 함수로 사용하였다. 출력층은 10개의 노드로 구성되어, 각 노드가 0부터 9까지의 숫자 중 하나를 나타내도록 하였다. 출력층의 활성화 함수로는 Softmax를 사용하여 확률 형태로 결과를 출력할 수 있었다.

아래와 같이 Python 활용해 손글씨 숫자를 인식하기 위한 다층 퍼셉트론 코드를 구현해보았다. (출처: chat Gpt, 2025. 11.11)

손글씨 숫자 인식 (MNIST) 다층 퍼셉트론 실험

실행 환경: Python 3 + TensorFlow 설치

```
import tensorflow as tf
```

```
from tensorflow import keras
```

```
import matplotlib.pyplot as plt
```

1 MNIST 데이터셋 불러오기 (손글씨 0~9)

```
(x_train, y_train), (x_test, y_test) =  
keras.datasets.mnist.load_data()
```

2 데이터 정규화 (0~255 → 0~1 사이로)

```
x_train = x_train / 255.0
```

```
x_test = x_test / 255.0
```

3 모델 만들기 (다층 퍼셉트론 구조)

```
model = keras.models.Sequential([
```

```
    keras.layers.Flatten(input_shape=(28, 28)), # 28x28  
    이미지를 1차원으로 펴기 (784개 입력)
```

```
    keras.layers.Dense(128, activation='relu'), # 은닉층  
    (노드 128개)
```

```
    keras.layers.Dense(10, activation='softmax') # 출력층  
    (0~9 중 하나 예측)
```

```
])
```

4 모델 컴파일 (학습 방법 정하기)

```
model.compile(
    optimizer='adam',          # 가중치 조정 방법
    loss='sparse_categorical_crossentropy', # 다중 분류 손실 함수
    metrics=['accuracy']       # 정확도 계산
)
```

5 학습 (에포크 5회)

```
print(" 학습을 시작")
model.fit(x_train, y_train, epochs=5, batch_size=32,
          verbose=2)
```

6 테스트 데이터로 정확도 평가

```
print("\n 테스트 중")

test_loss, test_acc = model.evaluate(x_test, y_test,
                                     verbose=0)

print(f" 테스트 정확도: {test_acc * 100:.2f}%")
```

7 예측해보기 — 테스트 이미지 하나 선택

```
plt.imshow(x_test[0], cmap='gray')

plt.title(f"정답: {y_test[0]}")

plt.axis('off')

plt.show()

prediction = model.predict(x_test[:1])

print(" 모델의 예측 결과:", prediction.argmax())
```

결과 (Results)

손글씨 숫자별 예측 확률은 `model.predict(x_test)` 를 사용해 알아낼 수 있다. 손실값은 `model.evaluate(x_test, y_test_cat)`에서 나오는 `test_loss`로, 모델이 얼마나 틀릴 가능성이 있는지 확인할 수 있다.

보고서에서 구축한 다층 퍼셉트론 모델은 MNIST 손글씨 숫자 데이터셋을 대상으로 학습되었으며, 테스트 데이터에 대한 예측 성능을 확인하였다. 모델의

출력층에는 `Softmax` 활성화 함수를 적용하여 각 클래스별 예측 확률을 산출하였다.

예를 들어, 한 테스트 이미지가 실제 숫자 '3'을 나타내는 경우, 모델은 다음과 같이 각 숫자에 대한 예측 확률을 산출하였다. (출처: chat Gpt, 2025. 11.11)

```
{0: 0.01,
 1: 0.02,
 2: 0.05,
 3: 0.88,
 4: 0.01,
 5: 0.01,
 6: 0.01,
 7: 0.01,
 8: 0.005,
 9: 0.005}
```

가장 높은 확률을 가진 클래스는 '3'으로, 이는 실제 값과 일치하였다. 본 사례를 통해 모델이 높은 확률을 기반으로 정확한 예측을 수행함을 확인할 수 있다.

또 다른 사례로 실제 숫자가 '5'인 이미지에서는 모델이 클래스별 확률을 다음과 같이 산출하였다.

```
{0: 0.01,
 1: 0.02,
 2: 0.02,
 3: 0.05,
 4: 0.05,
 5: 0.70,
 6: 0.05,
 7: 0.05,
 8: 0.0,
 9: 0.0}
```



최종 예측은 '5'로 나타났으며, 70%의 확률로 예측되었다. 확률이 다소 낮은 경우, 다른 클래스와 혼동 가능성을 내포함을 확인할 수 있었다.

이와 같이, Softmax 함수를 통해 산출된 클래스별 확률은 모델의 예측 신뢰도를 정량적으로 평가할 수 있게 하며, 개별 이미지 단위에서의 예측 정확도와 잠재적 오류를 분석하는 데 유용히 쓰였다.

토의(Discussion), 결론(Conclusion)

본 보고서에서 구축한 다층 퍼셉트론 모델은 MNIST 손글씨 숫자 데이터셋을 대상으로 학습되었다. 테스트 결과 대부분의 이미지에서 높은 정확도로 숫자를 예측하였으며, Softmax 출력 확률 분석을 통해 모델의 예측 신뢰도를 확인할 수 있었다. 그러나 일부 이미지에서는 확률이 낮거나 여러 클래스 간 확률 차이가 미미하였기 때문에 숫자 간 혼동 가능성이 존재함 또한 확인할 수 있었다. 이러한 오류는 손글씨의 개인차, 잉크의 굵기 및 글자 형태의 다양성 등으로 인해 발생한 것으로 판단된다. 그러나 사용한 모델이 단순히 다층 퍼셉트론 구조에 국한되어 있어, 이미지의 공간적 특징을 효과적으로 학습하는 데에는 한계가 있었다.

다층 퍼셉트론 기반의 손글씨 숫자 인식 모델을 구현하고, MNIST 데이터셋을 활용하여 모델의 예측 성능을 평가한 결과 모델은 대부분의 테스트 이미지를 정확히 예측하였다. 또한 Softmax 함수를 통한 정량적 확률을 계산함으로써 개별 예측의 신뢰도를 분석할 수 있었다.

단순한 다층 퍼셉트론 모델만으로도 기본적인 손글씨 숫자 인식이 가능함을 확인하였으며, 보다 정교한 신경망 구조와 다양한 데이터셋을 활용한다면 모델의 정확도를 개선할 수 있을 것이다.

REFERENCES

- [1] 한미란, 「인공지능기반 MNIST 손글씨 인식에 대한 연구」, 한국IT정책경영학회, 2019
- [2] 이상구 with 이재화, 함윤미, 박경은, 「인공지능을 위한 기초수학 입문(Introductory Mathematics for Artificial Intelligence)」