Active learning을 적용한

Convolution Recurrent Neural Network를 이용하여

CAPTCHA 이미지 분류하기

빅데이터처리 8조

김동주 (2022157475)

권용성 (2021261614)

이승재 (2021216028)

Ismailov Rustam (2020170247)

**Introduction**

디지털 시대가 되고 다양한 장치들이 발전함에 따라 수많은 사건들이 디지털 데이터로 저장되고 있다. 인터넷 상에서 여러 사람들의 다양한 목적에 의해 작성되는 텍스트 데이터들, 여러가지 공정 또는 센서들에 의해서 수집되는 이미지 데이터 등 다양한 종류의 장치들로 인해 여러 형태의 데이터가 생성되고 있다. 대량의 데이터를 분석하면 기존에는 볼 수 없었던 새로운 특징이나 앞으로 일어날 일을 예측하는데 도움을 받을 수 있다. 그러나 이같은 일을 하기 위해서는 수집되는 데이터의 특성도 같이 입력되어야 한다. 실시간으로 방대한 데이터가 만들어지고 있지만 이것이 나타내는 의미는 사람이 직접 입력해주어야 하는 경우가 있다. 생성되는 데이터의 양을 생각하면 사람이 일일이 의미 (라벨)을 입력하는 것은 불가능에 가깝다. 이러한 문제를 해결하기 위해 기계가 스스로 데이터를 인식하여 의미나 이름을 달아주게 하는 머신러닝 분야가 발전해오고 있다.

보통 데이터를 분류하고 라벨을 달아주는 일은 지도학습이라는 기계학습 방법이 주로 사용되고 있다. 지도학습 방법은 가지고 있는 모든 데이터들의 특성을 기계에 학습시키는 방법이다. 그러나 지도학습을 이용하여 좋은 성능의 기계를 얻기 위해는 많은 양의 라벨링 되어있는 데이터를 필요로 한다. 많은 양의 라벨링 되어있는 데이터를 사람이 직접 만들기에는 너무 큰 비용이 필요하다. 만약 기계가 라벨링이 필요한 데이터를 직접 골라 사람에게 부탁하면 사람의 수고가 줄어들 수 있을 것이다. 이러한 문제를 해결한 방법 중 액티브 러닝 (Active Learning)이 있다. 액티브 러닝은 기계가 분류할 수 있는 데이터는 직접 분류하고, 분류하기 어려운 즉, 정보량이 많은 데이터는 사람에게 직접 부탁하는 방법이다. 액티브 러닝에서는 데이터의 불확실성을 측정하여 분류하기 어렵고 정보량이 많은 데이터를 판단한다. 데이터의 불확실성을 판단하는 방법은 다양한데, 본 연구에서는 대표적으로 쓰이는 마진 샘플링 (Margin Sampling), 리스트 컨피던스 (Least Confidence), 엔트로피 샘플링 (Entrooy Sampling)을 이용하여 불확실성을 고려했다.

분류에 사용한 데이터는 인터넷 상에서 어뷰징을 막기 위해 사용되는 CAPTCAH (Completely Automated Public Turing test to tell Computers and Humans Apart) 이미지이다. CAPTCHA 이미지는 왜곡된 단어로 이루어져 있으며, 사람이라면 명확히 읽을 수 있지만 컴퓨터의 경우 읽기 어려운 단어이다. 우리는 이미지 분류에 주로 사용되는 CRNN (Convolution Recurrent Neural Network) 모델을 이용하여 데이터를 분류하고 앞서 언급한 액티브 러닝 샘플링을 적용하였고, 불확실성을 고려하지 않은 무작위 샘플링을 같이 적용한 모델과 비교하였다.

**Data**

Introduction에서 언급하였듯이 CAPTCHA 이미지 데이터는 인터넷 상에서 불법으로 이득을 취하는 자동 프로그래밍과 같은 어뷰징을 막기 위해 만들어졌다. 해당 이미지는 사람이라면 쉽게 어떤 숫자나 알파벳으로 이루어져 있는지 인식할 수 있지만, 프로그램의 경우 이와 같은 일은 쉽지 않다. Fig 1과 같이 CAPTCHA 이미지는 아무 의미없는 숫자나 알파벳의 조합으로

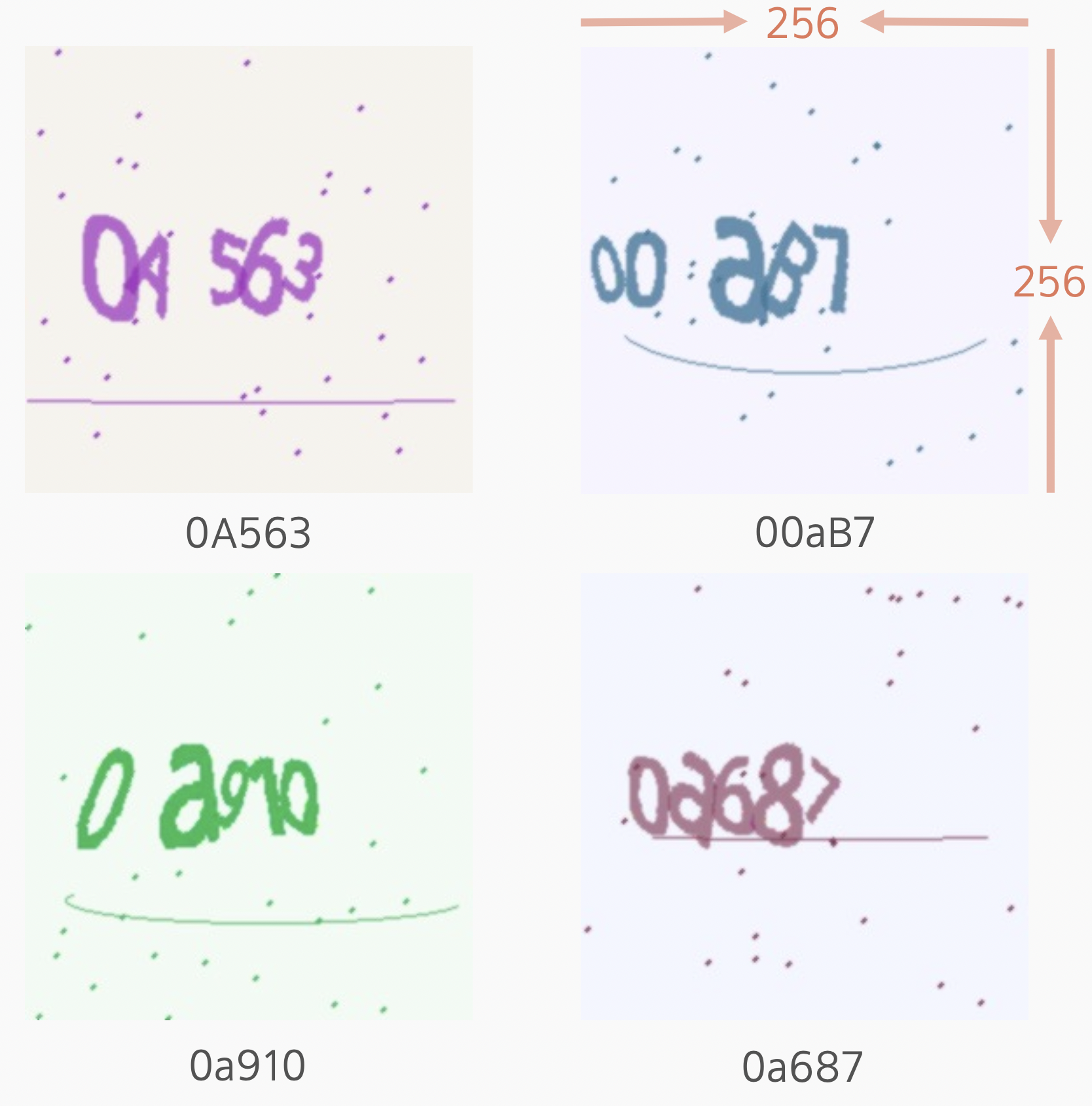


Figure 1. CAPTCHA 데이터 예시. 숫자 (0~9), 소문자 알파벳 (a~z), 대문자 알파벳 (A~Z) 중 5개의 랜덤한 조합으로 이루어져있다. 글자 외에도 의미없는 점들과 선이 같이 배치되어있다. 데이터 셋 안에는 동일한 라벨을 가진 데이터는 존재하지 않는다. 이미지의 크기는 256\*256으로 되어있다.

이루어져 있다. 그림 밑에 쓰여있는 라벨과 비교할 때 그림 안의 문자를 읽을 때 어려움이 없을 것이다. 또한 아무 의미 없는 점들과 선이 포함된 경우도 있다. 우리가 사용한 데이터는 그림과 같이 총 5개의 문자가 이미지 상에 들어가 있으며 숫자 (0~9), 소문자 알파벳 (a~z), 대문자 알파벳 (A~Z) 총 62개의 문자가 무작위로 배치된다.

학습에 이용한 CAPTCHA 이미지 데이터의 총 82,327개이다. 액티브러닝을 사용하지 않은 베이스라인 모델 (CRNN)에서는 트레이닝 셋과 테스트 셋을 약 3:1 비율로 분리했다. 결과적으로 61,745개의 트레이닝 셋과, 20,582개의 테스트 셋으로 분리했다. 액티브 러닝을 적용했을 때도 같은 비율의 트레이닝 셋과 테스트 셋을 사용하였으며, 초기 트레이닝 셋의 크기는 1,000개를 사용하였다.

**Method**

**Convolution Recurrent Neural Network**

모델은 CNN (Convolution Neural Network) 구조와 RNN (Recurrent Neural Network) 구조를 합한 CRNN (Convolution Recurrent Neural Network) 을 사용하였다. CRNN은 보통 이미지 데이터 학습에 사용되며, 이미지 내에 시계열 정보를 인식하기 위해 설계되었다. Fig 2의 convolution layer의 일부는 pre-train된 ResNet-18로 대체하는 전이학습 (transfer learning) 기법을 사용하였다. CAPTCHA 데이터 특성상 동일한 클래스가 없이 종류가 매우 많아 학습에 들어가는 연산 비용이 매우 크기 때문에 전이학습을 적용했다. 또한 수 만에서 수천만장의 이미지로 미리 학습된 높은 성능을 갖는 ResNet이나 VGG 신경망의 특징 추출 능력을 그대로 이용하고 싶었기 때문에 전이 학습 기법을 사용하였다. 미리 학습된 ResNet-18에서 마지막 세 개의 출력 계층 레이어만을 변경하여 CAPTCHA 이미지 데이터로 재학습 시켰다.

CRNN 모델로 CAPTCHA 이미지를 분류한 후 출력되는 softmax 값을 이용하여 앞서 언급한 액티브 러닝을 적용했다. softmax 값을 이용하여 각 샘플링 방법들은 불확실성을 측정하고, 가장 높은 불확실성을 나타내는 데이터들을 이용하여 재학습하는 방식으로 이용했다.

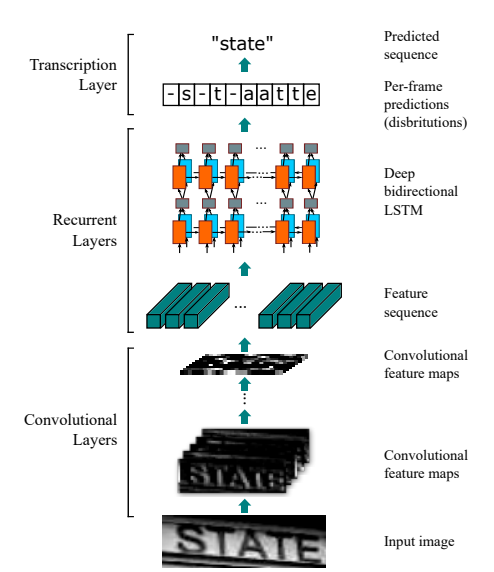


Figure 2. CRNN[1] 네트워크 아키텍처. 아키텍처는 세개의 부분으로 이루어져 있다: 1) 입력 이미지로부터 feature 시퀀스를 추출하는 convolution layers; 2) 각 frame마다 라벨의 분포를 예측하는 recurrent layers; 3) 최종 label 시퀀스 내의 프레임 단위 예측을 번역하는 transcription layer.

**Least Confidence**

Least Confidence는 액티브 러닝에서 불확실성을 판단하는 샘플링 기법중 하나이다. 불확실성이란, 쉽게 말하자면 분류를 애매하게 한 정도를 말한다. 모델은 데이터들을 각 클래스 (정답)에 대한 softmax 값에 따라 분류, 라벨링을 하게 되고, 하나의 클래스만 고른다. 보통 가장 높은 softmax 값을 고르지만, 두번째로 높은 softmax와 차이가 얼마 나지 않는다면 과연 모델이 잘 분류를 한 것일까? 모델이 확실하게 분류를 한 것인지를 판단하기 위해 불확실성이라는 것을 측정하고, 이러한 아이디어를 간단하게 구현한 방법 중 하나가 Least Confidence이다. 이 샘플링 방법은 이름 그대로 제일 작은 확신을 가진 데이터들을 고르는 알고리즘으로 되어있다. 수식(1)과 같이 각 데이터마다 얻게 된 확률 (softmax) 값 중에 제일 작은 값을 갖는 데이터를

(1)

고르게 되어있다. 좀 더 자세하게 설명하면, 모델 가 계산하는 사후확률 에서 하나의 데이터 (data point)에 대한 조건부 확률 (확신도)를 구한다. 그 중 가장 작은 확신도를 갖는 예제를

Table 1. Least Confidence의 예시. 이 쿼리 전략에서 학습자는 가장 “확실하게” 예측한 라벨에 대해 가장 확신도가 낮은 예제를 선택한다. Data point d1은 라벨 A를 0.9의 확신도로, d2는 라벨 B를 0.5의 확신도로 가지고 있다. 따라서 학습자는 d2의 실제 라벨을 알고 싶어한다. 이 방법은 가장 그럴듯한 라벨에 대한 확신도를 사용하고, 다른 라벨에 대한 확률은 고려하지 않는다.

|  |  |  |  |
| --- | --- | --- | --- |
| data point | Label A | Label B | Label C |
| d1 | 0.9 | 0.09 | 0.01 |
| d2 | 0.2 | 0.5 | 0.3 |

선택하게 된다. 실제 출력되어 나오는 확률 값을 이용한 예가 Table 1에 나와있다. 두개의 데이터 포인트 d1과 d2를 넣고 모델을 이용해 세개의 클래스 (Label A, B, C)에 대한 확신도를 나타낸 것이다. 모델은 d1의 경우 A, B, C 클래스에 대해 0.9, 0.09, 0.01 의 확신도를 계산하였고, d2의 경우 0.2, 0.5, 0.3의 값을 출력했다.각 데이터 포인트에서 모델이 분류한 클래스에 대한 확신도는 각각 0.9, 0.5이다. 이중 낮은 확신도는 0.5이고, 이 값이 출력된 d2 데이터가 분류하기 애매한 데이터가 되고 재 훈련 데이터 셋으로 샘플링된다.

**Margin Sampling**

위에서 언급한 Least Confidence 방법은 각 데이터 포인트마다 가장 높은 확신도만 취하기 때문에 매우 심플한 방법이라 할 수 있다. Margin Sampling 또한 간단한 방법이지만, 다른 아이디어에서 출발하였다. 불확실성을 판단하는 방법으로써, 정답이 애매한 데이터를 찾아야한다. 모델이 가장 정답이라 생각하는 라벨을 두개 골라 비교하는 방법이다. Least Confidence 방법에서는 가장 정답이라 생각하는 라벨의 확률만 이용하여 제일 낮은 값을 선택하였지만, 이 방법에서는 가장 높은 두개의 확률을 가지고 모델이 헷갈려하는 정도를 정의하여 사용했다. Margin Sampling에서는 각 데이터 포인트에서 출력된 확률 값 (softmax)의 2번째로 높은 값까지 이용한다. 가장 높은 값과 두번째로 높은 확률 값의 차이를 margin이라 정의하고, margin이 가장 낮은 데이터 포인트를 불확실성이 가장 높은 것으로 판단한다. 식

(2)

(2)에서 대괄호 안에 있는 식이 margin을 나타낸다. margin이 낮을 수록 가장 유력한 라벨과 두번째로 유력한 라벨의 확률이 비슷하다는 것을 의미하고, 이는 모델이 두개의 라벨을 비슷한 확률로 정답이라 분류한다 생각할 수 있다. Table 2에 Margin Sampling의 예시가 있다. 두개의 데이터 포인트 (d1, d2) 중 불확실성이 높은 것을 고르는 예에서, 각각의 margin을 구하여 표 오른쪽 margin 열에 입력되어있다. d1의 경우 Label A일 확률이 0.9로 가장 높고, Label B가 0.09로 두번째로 높다. 따라서 이때 d1의 margin은 0.81이 된다. d2의 경우 Label B일 확률 0.5, Label C일 확률이 0.3으로 margin은 0.2이다. 따라서 d1와 d2 중 margin이 낮은 데이터 포인트는 d2이고, d2의 불확실성이 가장 높다고 말할 수 있다.

Table 2. Margin Sampling의 수식과 예시. Least confidence에서 최고 확률의 라벨 이외의 것을 고려하지 않는다는 점을 보완하기 위해 고안된 방법이다. 가장 확률이 높은 라벨과 두 번째 라벨의 확신도의 차이가 가장 적은 라벨을 선택하는 쿼리 전략이다. d1은 0.9-0.09=0.81, d2는 0.5-0.3=0.2의 점수를 받고, 학습자는 역시 d2에 대한 실제 라벨을 요구한다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| data point | Label A | Label B | Label C | Margin |
| d1 | 0.9 | 0.09 | 0.01 | 0.81 |
| d2 | 0.2 | 0.5 | 0.3 | 0.2 |

**Entropy Sampling**

Entropy Sampling은 정보 엔트로피 개념을 이용한 샘플링 방법이다. 위에서 언급한 Least Confidence나 Margin Sampling의 경우, 각각 한개, 두개의 확률을 이용하여 불확실성을 판단하였다. Entropy Sampling에서는 각 데이터 포인트에서 얻어낸 모든 확률들을 이용한다. 각 데이터 포인트에서 얻어낸 확률들 (확률분포)이 어떤 모양으로 나타나있는지를 본다. 정보 엔트로피 (Shannon Entropy)는 확률 분포의 모양에 따라 다른 값을 갖는다. 확률 분포가 고른 (uniform) 경우 엔트로피는 높은 값을 갖고, 반대로 고르지 않을 경우 (한 라벨에 대해 높은 값을 갖고, 나머지 라벨에 대해 낮은 값을 갖을 때) 엔트로피 값은 낮아진다. 따라서 모델이 모든 라벨에 대해 분류를 정확히 하지 못하고 애매한 값을 뱉는다면 엔트로피 값을 높아지게 되고, 우리는 이 때 불확실성이 높다고 여길 수 있다. 식 (3)을 보면

(3)

대괄호 안에 정보 엔트로피 형태가 들어있고, 이 값이 최대가 되는 데이터 포인트를 뱉는 형태로 되어있다. Table 3을 보면 두가지 데이터 포인트 (d1, d2)에 대한 각 라벨의 확률값이있고 Entropy Sampling을 이용하는 예시가 있다. d1의 확률을보면 Label A에서 0.9로 가장 높은 값을 갖고, 나머지 라벨에 대해 비교적 낮은 값으로 나타났다. d2의 경우, Label B에서 0.5로 가장 높은 값을 가졌지만 나머지 라벨에서 얻어진 확률 값들과 그리 큰 차이를 보이지 않는다. d1와 d2의 경우를 비교하면, 비교적 d2의 경우가 d1 보다 고른 확률분포를 가진 것으로 볼 수 있다. 이때 각각의 엔트로피를 계산하면 d2의 경우가 크다는 것을 알 수 있고, Entropy Sampling 방법에 의해서 d2의 불확실성이 크다는 것을 알 수 있다.

Table 3. Entropy Sampling의 수식과 예시. 이 방법은 모든 가용한 라벨 확률값을 전부 사용하는 방법으로, 정보이론에서의 entropy 개념을 사용한다. Entropy를 구하는 식을 각 data point에 적용하고, entropy(uncertainty)가 가장 큰 예제를 쿼리하는 방식이다. d1의 entropy는 0.115, d2의 entropy는 0.447로, d2에 대한 실제 라벨을 요구하게 된다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| data point | Label A | Label B | Label C | Entropy |
| d1 | 0.9 | 0.09 | 0.01 | 0.115 |
| d2 | 0.2 | 0.5 | 0.3 | 0.447 |

**Result**

본 연구에서는 CAPTCHA 이미지 데이터를 분류하는 작업을 액티브 러닝을 통해 해결했다. 이미지 데이터를 분류하는 딥러닝 모델 중 하나인 Convolution Recurrent Neural Network (CRNN)을 사용하였다. 액티브 러닝에서는 다양한 샘플링 방법을 이용하였다. 본 보고서 Method 부분에서 다룬 Least Confidence, Margin Sampling, Entropy Saampling을 이용하였고, 각 방법들을 비교하기 위해 무작위 샘플링을 이용한 결과도 얻었다. 본 보고서의 Data에서 언급하였듯이, 총 82,327 장의 CAPTCHA 이미지를 이용해 학습하였다. Figure 3에 각 방법으로 학습한 모델의 accuracy와 loss가 있다. CRNN으로만 학습한 결과가 baseline으로 표시되어있고, 각 샘플링 방법으로 학습한 결과도 표시되어있다. 기본 지도학습인 CRNN의 경우 accuracy가 액티브 러닝에 비해 빠르게 증가한 것을 보이고 있다. loss 또한 제일 먼저 낮은 값으로 수렴하였다. 그러나 액티브 러닝에 비해 큰 노이즈를 가지 는 것으로 보였다. 학습이 완료되었을 때 baseline과 액티브 러닝 accuracy와 loss 값은 모두 오차 범위 내로 비슷한 것으로 보였다. Figure 4에 같은 결과이지만 액티브 러닝의 결과만을 보기 위해 baseline을 제거한 결과가 있다. 무작위 샘플링을 포함하여 4개의 샘플링 방법 모두 오차 범위 내에서 비슷한 속도로 수렴하였고, 비슷한 accuracy와 loss를 보였다. CAPTCHA 이미지의 특성상 중복되는 클래스가 존재하지 않는다. 따라서 클래스의 수가 매우 많았던 문제에 의해 정보를 많이 담은 데이터를 선별하는 액티브 러닝이 일반 지도학습보다 뚜렷한 성능을 보이지 못했다고 추측했다.

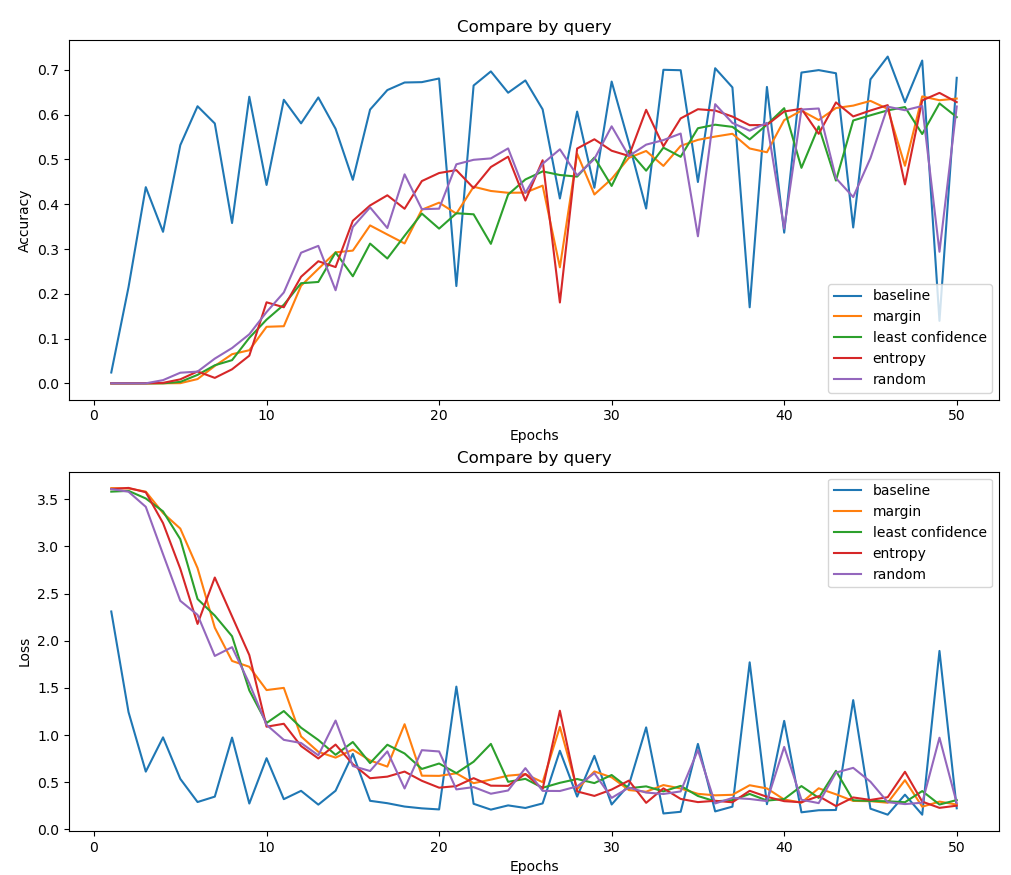


Figure 3. 초기 훈련 데이터셋 1000개를 이용해 모델을 한 번 훈련시켰다. 그 뒤 매 에폭마다 unlabeled pool의 훈련 데이터 2000개중 각각 다른 쿼리 전략을 사용해 500개씩 데이터를 선정해 라벨링을 하였다. 쿼리 사이즈와 풀 사이즈를 고정시켜 놓고 각각 다른 쿼리 전략에 대한 테스트 정확도 및 손실을 그래프로 표현한 것이다

각각의 샘플링 방법에서 동일한 query, pool size를 이용해 결과를 얻었다. 이 외에도 하나의 샘플링 방법에서 다양한 query size, pool size를 적용한 결과도 확인하였다. Figure 5에 동일한 샘플링 방법에서 pool size를 변경한 학습 결과가 나타나있다. 샘플링 방법은 Margin Sampling을 이용했고, pool size를 1000, 2000, 5000으로 적용한 후 결과를 얻었다. 정확도 그래프에서 붉은색 사각형으로 표시해둔 부분은 남아있는 pool이 없는 부분이다. 따라서 약 50 epochs 까지의 결과를 보면, pool size가 클 수록 느리게 수렴하는 경향을 보였다. 그러나 이와 반대로 손실 그래프에서는 pool size가 클 수록 더 빠르게 수렴한 결과를 보였다.

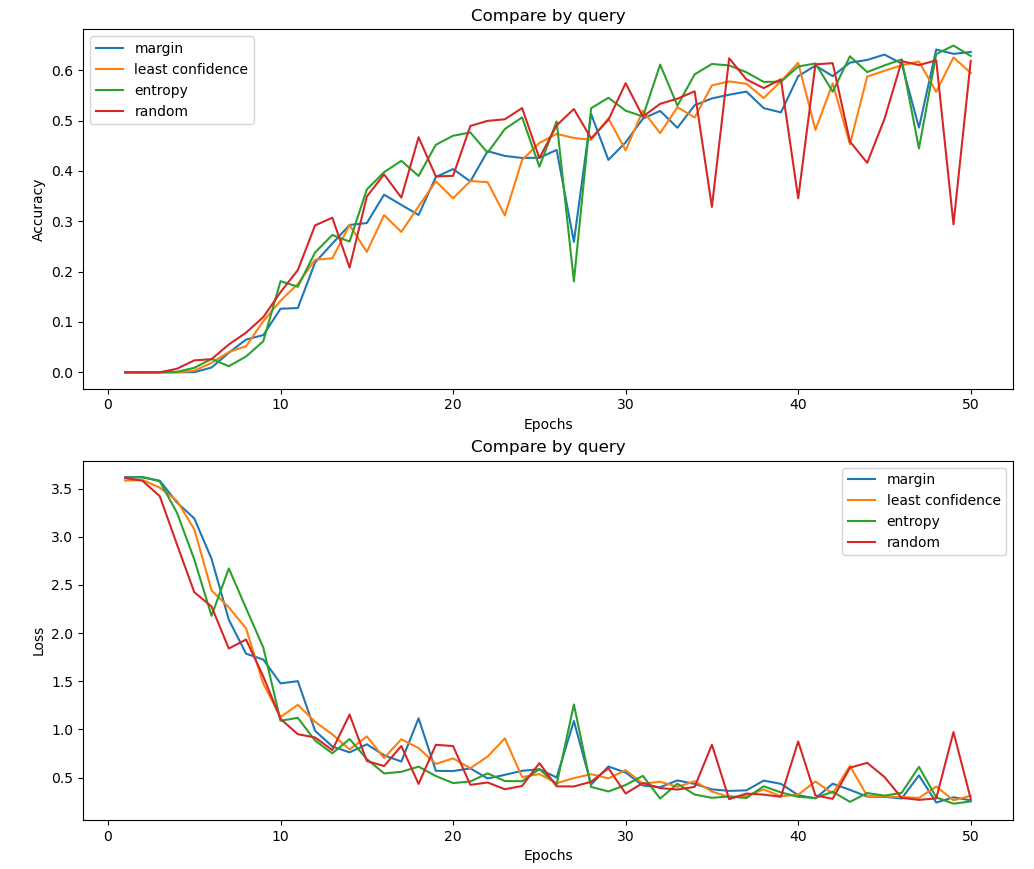


Figure 4. 초기 훈련 데이터셋 1000개를 이용해 모델을 한 번 훈련시켰다. 그 뒤 매 에폭마다 unlabeled pool의 훈련 데이터 2000개중 각각 다른 쿼리 전략을 사용해 500개씩 데이터를 선정해 라벨링을 하였다. 쿼리 사이즈와 풀 사이즈를 고정시켜 놓고 각각 다른 쿼리 전략에 대한 테스트 정확도 및 손실을 그래프로 표현한 것이다

동일한 샘플링 방법에서 query size를 조절하여 학습한 실험도 진행했다. Figure 6을 보면 Margin Sampling 방법, 동일한 pool size에서 query size를 1000, 2000, 5000으로 조절하며 학습한 결과를 나타내었다. 이 실험에서도 마찬가지로 붉은색 사각형을 표시된 지점은 pool 에 남은 데이터가 없으므로 일반 기계학습과 동일하게 학습되었다. 액티브 러닝이 잘 작동하는 50 epochs 까지를 보면 query size가 클 수록 정확도와 손실 그래프가 빠르게 수렴하는 것을 확인할 수 있었다.

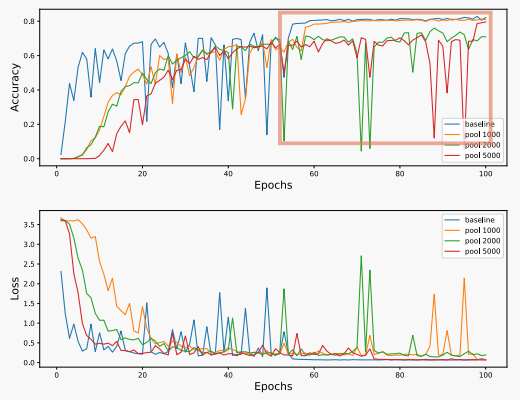


Figure 5. 동일한 샘플링 방법에서 pool size를 조절하며 학습한 결과. 최종 정확도와 손실은 비슷한 값으로 나왔지만, pool size가 작을 수록 정확도 값은 먼저 수렴하는 것으로 보였다. 이와 반대로 pool size가 큰 경우, 손실이 빠르게 수렴하는 것을 보였다.

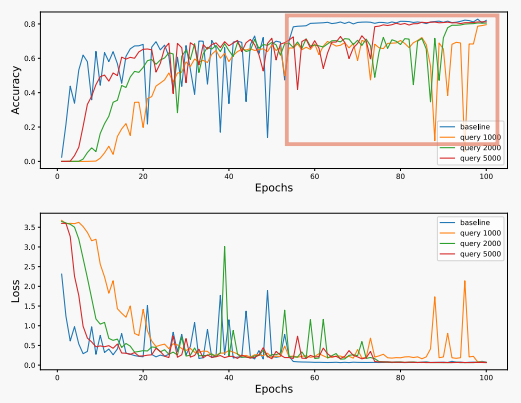


Figure 6. 동일한 샘플링 방법에서 query size를 조절하며 학습한 결과. query size가 클 수록 정확도와 손실 그래프에서 빠르게 수렴하는 결과를 보임.

**Conclusion**

기존 기계학습 모델들은 학습을 할 때 많은 양의 데이터 요구한다. 본 보고서에는 적은 양의 데이터를 이용하여 효율적으로 학습할 수 있는 액티브 러닝에 대하여 다루었다. 액티브 러닝은 모델이 다루기 어려운 데이터를 사람에게 요청하는 방식이다. 따라서 모델이 다루기 어려운, 학습시에 도움이 되는 정보량이 큰 데이터를 판단하는 다양한 방법이 있다. 우리는 본 연구에서 액티브 러닝에서 대표적으로 사용되는 Least Confidence, Margin Sampling, Entropy Sampling을 다루었다. 사용한 데이터는 인터넷 어뷰징을 막기 위해 사용되는 CAPTCHA 이미지를 사용하였고, 이미지 분류에 특화된 CRNN을 사용했다. 앞에서 언급한 샘플링들은 각각 가장 높은 확신도, 가장 높은 것과 두번째로 높은 확률, 출력된 확률 전체를 사용하여 정보량을 파악한다. 그러나 우리의 결과는 오차범위 내에서 비슷한 수렴 속도와 정확도를 얻었다. 이는 CAPTCHA 데이터 특성상 중복되는 라벨이 없고 클래스의 종류가 매우 많기 때문이라 추측했다.

**Reference**

[1] Baoguang Shi, Xiang Bai and Cong Yao.

An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. In [arXiv:1507.0577](https://arxiv.org/abs/1507.05717)

**Appendix**

**A.Details of CRNN**



Figure 8. CRNN의 pytorch 코드, CRNN class 내의 forward function에 의해 해당 cnn과 rnn이 사용된다.

Fig 8에서 resnet18(pretrained=True)을 전이학습의 방법으로 사용한다. Resnet18의 child 부분을 rnn이 좀 더 의미있게 처리할 수 있게 하기 위하여 cnn layer 두개와 fully connected layer로 바꿔준 것을 확인할 수 있다. rnn은 GRU를 사용하였고, 최종 출력으로 [5,batch\_size,num\_classes]의 텐서를 반환하게끔 설정한다.

**B.Details of query method**

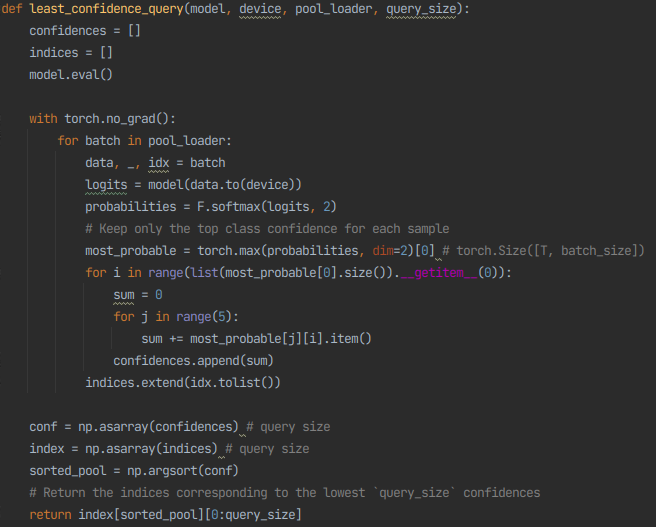


Figure 9. least confidence의 파이토치 코드,가장 큰 confidence들 중 작은 data point를 찾는 쿼리 전략이다.

Fig 9에서 training 되고 있는 model에 대한 unlabeled data의 output을 구한다. 그리고 파이토치의 max 라이브러리를 사용하여 class들 중 제일 높은 확신도의 텐서를 구한다.

Least confidence 샘플을 구하기 위해 uncertainty가 높은 순서대로 sorting을 하고 사용자가 미리 지정해 둔 query\_size(=500) 만큼 labeling을 human oracle에게 요구한다.

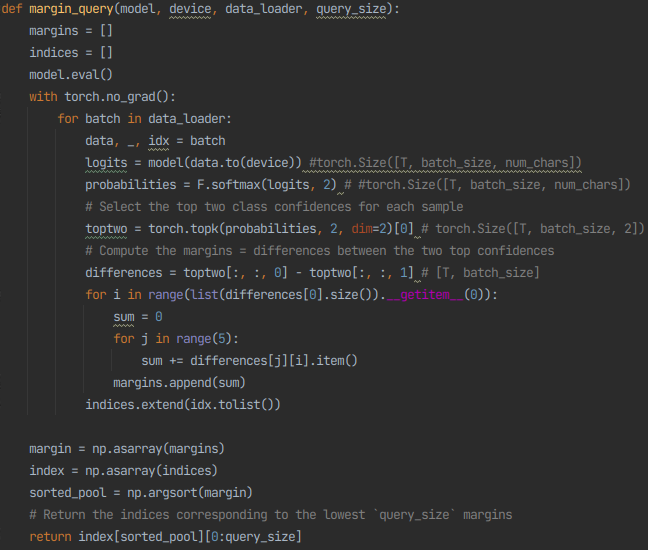


Figure 10. margin sampling의 파이토치 코드, top two의 confidence를 이용한 쿼리 전략이다.

Fig 10에서 training 되고 있는 model에 대한 unlabeled data의 output을 구한다. 그리고 파이토치의 topk 라이브러리를 사용하여 class들 중 제일 높은 확신도와 그 다음으로 높은 확신도의 텐서를 구한다.

Margin을 구하기 위해 margin sampling의 공식대로 그 둘을 빼준 뒤 uncertainty가 높은 순서대로 sorting을 하고 사용자가 미리 지정해 둔 query\_size(=500) 만큼 labeling을 human oracle에게 요구한다.

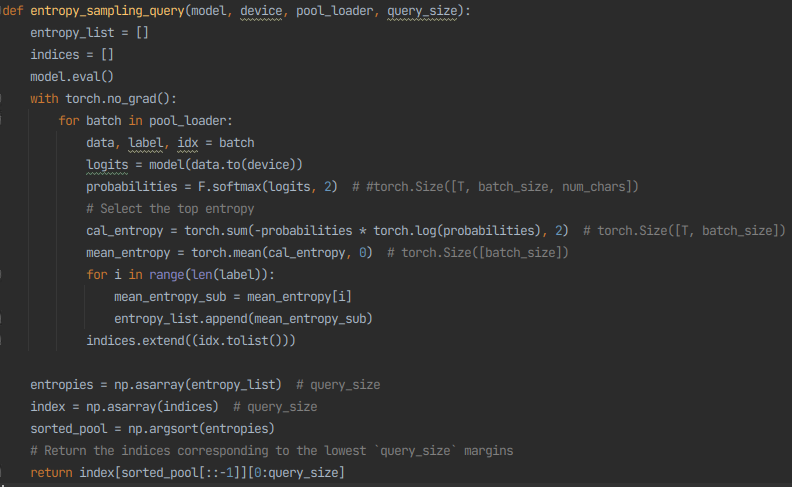


Figure 11. entropy sampling의 파이토치 코드, top two의 confidence를 이용한 쿼리 전략이다.

Fig 11에서 training 되고 있는 model에 대한 unlabeled data의 output을 구한다. 그리고 파이토치의 sum 라이브러리를 사용하여 엔트로피 공식에 의거하여 텐서들의 엔트로피를 구한다.

CAPTCHA의 다섯 자리 각각의 엔트로피를 사용하기 보다는 전체적인 엔트로피에 의거하여 uncertainty를 구하는 것이 타당하다고 생각해서 파이토치의 mean 라이브러리를 사용하여 모든 자릿수의 엔트로피 평균을 구한다.

그 뒤 엔트로피가 높은 순서대로 sorting을 하고 사용자가 미리 지정해 둔 query\_size(=500) 만큼 labeling을 human oracle에게 요구한다.