

# 1장. 회귀 분석 : 전복의 고리 수 추정 신경망

부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공  
201724579 정현모

## 1. 개요

하이퍼 파라미터 및 모델 파라미터를 튜닝하여 성능 개선을 위해 작업을 하였다. 기본적으로 전복의 고리 수 추정을 위한 신경망 모델의 실험을 진행하였다. 하이퍼 파라미터는 모델 구조나 학습 과정에 영향을 미치는 각종 상숫값 (학습률, 학습횟수, 미니배치 크기)을 의미하며, 개발자가 미리 정해주어야하고, 최적의 값을 찾을 수 있어야 좋은 결과를 기대할 수 있다.

## 2. 코드 변경

학습 과정에 영향을 미치는 하이퍼 파라미터를 `epoch_count`, `mb_size`, `learning_rate` 이렇게 3개로 잡았으며, 하이퍼 파라미터 변화에 따른 결과값을 한눈에 보기 위해 코드를 수정하였다.

코드 수정 내용은, `abalone_exec()` 함수에 인자로 `learning_rate`를 추가하였고, 학습결과를 한눈에 보기 위해 그래프로 도출하였다. 그래프로 학습결과를 확인하기 위해 최종 정확도를 `return`하도록 `abalone_exec()` 함수와 `train_and_test()` 함수를 수정하였다.

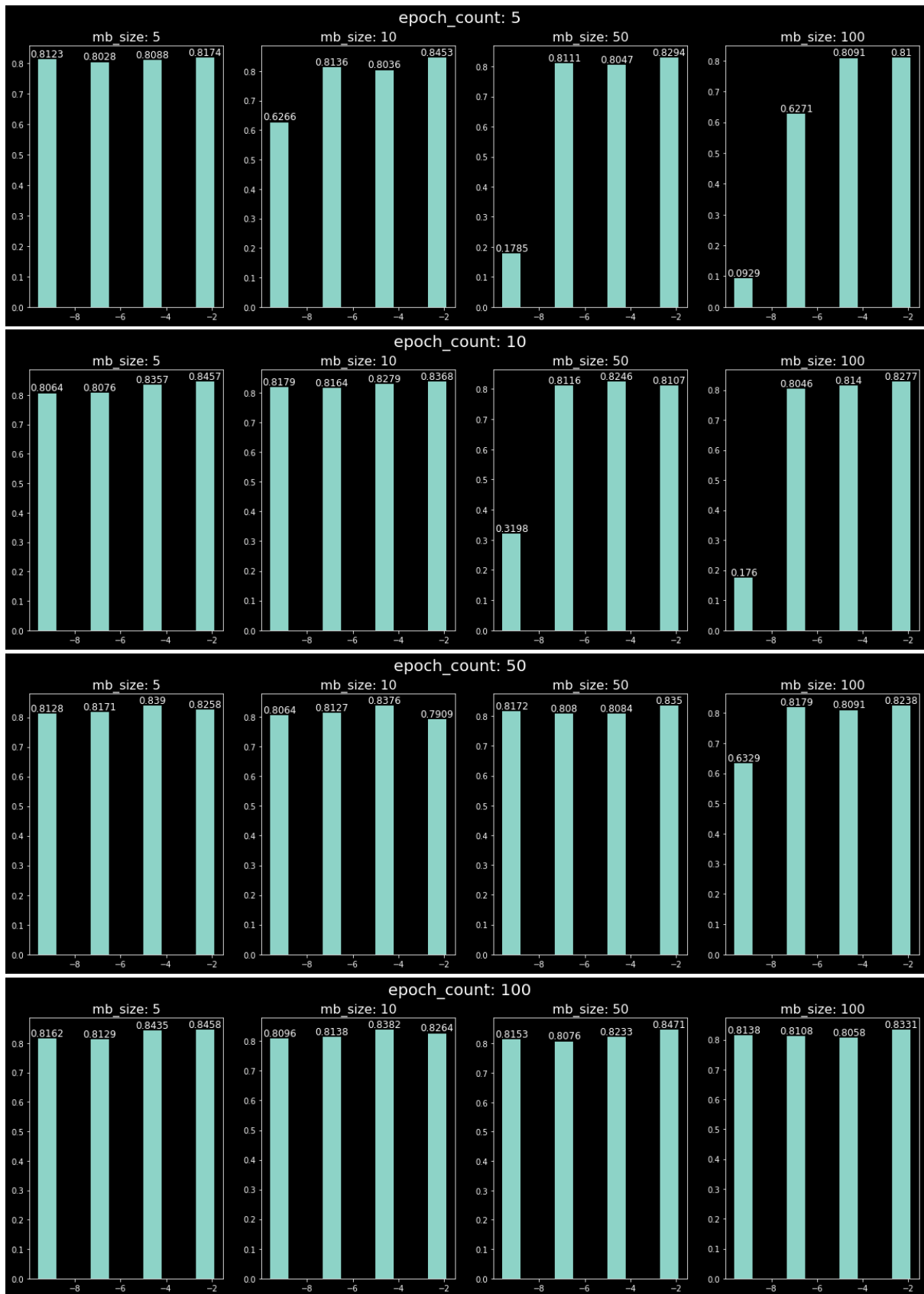
## 3. 파라미터별 학습 결과

아래 X축은 `learning_rate`로 `[0.0001, 0.001, 0.01, 0.1]`을 사용하였고, 그래프에서 보기 편하도록 `np.log()`를 사용해 간격을 맞춰주었다. Epoch가 높을 수록 대체로 학습 정확도가 좋았으며, 미니배치 사이즈는 10에서 대체로 정확도가 좋았으며, `learning_rate`는 어느정도 클수록 학습 정확도가 좋았다.

학습 정확도는 아래와 같을 때 가장 최대가 되었다.

<code>epoch_count</code>	5
<code>mb_size</code>	10
<code>learning_rate</code>	0.1
<code>final_accuracy</code>	0.8561468646

# 1장. 회귀 분석 : 전복의 고리 수 추정 신경망



# 1장. 회귀 분석 : 전복의 고리 수 추정 신경망

## 4. 수정한 코드

```
import matplotlib.pyplot as plt

epoch_counts = [5, 10, 50, 100]
mb_sizes = [5, 10, 50, 100]
learning_rates = [0.0001, 0.001, 0.01, 0.1]

maximum = {"epoch_count": -1,
           "mb_size": -1,
           "learning_rate": -1,
           "final_accuracy": 0}

for epoch_count in epoch_counts:
    fig = plt.figure(figsize=(20,6))
    fig.suptitle("epoch_count: " + str(epoch_count), fontsize=20)
    for i, mb_size in enumerate(mb_sizes):
        plt.subplot(1, len(mb_sizes), i + 1)
        plt.title("mb_size: " + str(mb_size), fontsize=16)
        final_accuracy = []
        for j, learning_rate in enumerate(learning_rates):
            final_accuracy.append(
                abalone_exec(epoch_count = epoch_count,
                             mb_size = mb_size,
                             learning_rate = learning_rate))
        if final_accuracy[j] > maximum["final_accuracy"]:
            maximum["final_accuracy"] = final_accuracy[j]
            maximum["epoch_count"] = epoch_count
            maximum["mb_size"] = mb_size
            maximum["learning_rate"] = learning_rate

    # print(learning_rates, final_accuracy)
    plt.bar(np.log(learning_rates), final_accuracy)
    for i, v in enumerate(np.log(learning_rates)):
        plt.text(v, final_accuracy[i], round(final_accuracy[i], 4),
                 fontsize = 12,
                 color='white',
                 horizontalalignment='center',
                 verticalalignment='bottom')

    plt.show()

print("maximum accuracy")
for key, value in maximum.items():
    print(key, value)
```