

## 1장. 회귀 분석 : 전복의 고리 수 추정 신경망

### 1. 개요

회귀 분석 방법을 이용한 전복의 고리수 추정 신경망 모델 실험에서 성능 개선을 위한 튜닝을 진행하였다. 간단히 하이퍼 파라미터를 변경하여 결과를 개선할 수가 있다. 파라미터란 가중치나 편향처럼 모델이 학습을 통해 최적의 값을 찾는 변수를 말하며, 하이퍼 파라미터는 모델이 학습하면서 최적의 값을 자동으로 찾는 것이 아니라 사람이 직접 지정해 주어야 하는 변수를 의미한다. 하이퍼 파라미터에는 학습률, 학습 횟수, 미니배치 크기 등이 있는데, 이를 조절하여 결과를 개선할 수 있다. 각각의 값을 변경하여 나온 결과를 비교해 어떤 조건에서 성능 개선이 가능할지 살펴보고자 한다.

### 2. 파라미터별 학습결과

학습률(learning rate), 미니배치 크기(mb\_size), 에포크 횟수(epoch\_count)의 세가지 변수가 있기 때문에 두가지 변수는 고정한 후 하나의 변수만 변경해보며 그 영향을 살펴보고자 한다.

#### 2.1 학습률(learning rate)이 변하는 경우

##### (1) 모델 1

구분	내용
조건	- 학습률(learning rate) : 0.001 - 에포크 횟수(epoch_count) : 100 - 미니배치 크기(mb_size) : 100
데이터 정확도	80.8%
결과 화면	<pre>LEARNING_RATE = 0.001 abalone_exec(epoch_count = 100, mb_size = 100, report = 20)  Epoch 20: loss=7.730, accuracy=0.815/0.814 Epoch 40: loss=7.483, accuracy=0.808/0.807 Epoch 60: loss=7.356, accuracy=0.808/0.807 Epoch 80: loss=7.260, accuracy=0.808/0.807 Epoch 100: loss=7.184, accuracy=0.808/0.808  Final Test: final accuracy = 0.808</pre>

##### (2) 모델 2

구분	내용
----	----

조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 에포크 횟수(epoch_count) : 100</li> <li>- 미니배치 크기(mb_size) : 100</li> </ul>
데이터 정확도	81.8%
결과 화면	<pre>LEARNING_RATE = 0.01 abalone_exec(epoch_count = 100, mb_size = 100, report = 20)</pre> <p>Epoch 20: loss=6.967, accuracy=0.808/0.811  Epoch 40: loss=6.736, accuracy=0.811/0.812  Epoch 60: loss=6.565, accuracy=0.811/0.815  Epoch 80: loss=6.412, accuracy=0.815/0.816  Epoch 100: loss=6.271, accuracy=0.818/0.818</p> <p>Final Test: final accuracy = 0.818</p>

### (3) 모델 3

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.1</li> <li>- 에포크 횟수(epoch_count) : 100</li> <li>- 미니배치 크기(mb_size) : 100</li> </ul>
데이터 정확도	82.6%
결과 화면	<pre>LEARNING_RATE = 0.1 abalone_exec(epoch_count = 100, mb_size = 100, report = 20)</pre> <p>Epoch 20: loss=5.804, accuracy=0.825/0.831  Epoch 40: loss=5.259, accuracy=0.834/0.828  Epoch 60: loss=5.056, accuracy=0.837/0.838  Epoch 80: loss=4.950, accuracy=0.838/0.840  Epoch 100: loss=4.910, accuracy=0.840/0.826</p> <p>Final Test: final accuracy = 0.826</p>

## 2.2 에포크 횟수(epoch\_count)가 변하는 경우

### (1) 모델 4

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 에포크 횟수(epoch_count) : 10</li> <li>- 미니배치 크기(mb_size) : 100</li> </ul>
데이터 정확도	80.6%

결과 화면	<pre> LEARNING_RATE = 0.01 abalone_exec(epoch_count = 10, mb_size = 100, report = 2)  Epoch 2: loss=8.270, accuracy=0.819/0.813 Epoch 4: loss=7.525, accuracy=0.808/0.808 Epoch 6: loss=7.388, accuracy=0.808/0.806 Epoch 8: loss=7.286, accuracy=0.808/0.806 Epoch 10: loss=7.208, accuracy=0.809/0.806  Final Test: final accuracy = 0.806 </pre>
-------	---

(2) 모델 5

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 에포크 횟수(epoch_count) : 100</li> <li>- 미니배치 크기(mb_size) : 100</li> </ul>
데이터 정확도	81.8%
결과 화면	<pre> LEARNING_RATE = 0.01 abalone_exec(epoch_count = 100, mb_size = 100, report = 20)  Epoch 20: loss=6.967, accuracy=0.808/0.811 Epoch 40: loss=6.736, accuracy=0.811/0.812 Epoch 60: loss=6.565, accuracy=0.811/0.815 Epoch 80: loss=6.412, accuracy=0.815/0.816 Epoch 100: loss=6.271, accuracy=0.818/0.818  Final Test: final accuracy = 0.818 </pre>

(3) 모델 6

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 에포크 횟수(epoch_count) : 1000</li> <li>- 미니배치 크기(mb_size) : 100</li> </ul>
데이터 정확도	83.6%

결과 화면	<pre>LEARNING_RATE = 0.01 abalone_exec(epoch_count = 1000, mb_size = 100, report = 200)</pre>
	<pre>Epoch 200: loss=5.749, accuracy=0.825/0.827 Epoch 400: loss=5.223, accuracy=0.835/0.833 Epoch 600: loss=5.017, accuracy=0.838/0.835 Epoch 800: loss=4.933, accuracy=0.839/0.837 Epoch 1000: loss=4.896, accuracy=0.839/0.836  Final Test: final accuracy = 0.836</pre>

## 2.3 미니배치 크기(mb\_size)가 변하는 경우

### (1) 모델 7

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 에포크 횟수(epoch_count) : 100</li> <li>- 미니배치 크기(mb_size) : 10</li> </ul>
데이터 정확도	83.1%
결과 화면	<pre>LEARNING_RATE = 0.01 abalone_exec(epoch_count = 100, mb_size = 10, report = 20)</pre> <pre>Epoch 20: loss=5.764, accuracy=0.825/0.816 Epoch 40: loss=5.235, accuracy=0.834/0.829 Epoch 60: loss=5.024, accuracy=0.837/0.837 Epoch 80: loss=4.933, accuracy=0.839/0.833 Epoch 100: loss=4.899, accuracy=0.839/0.831  Final Test: final accuracy = 0.831</pre>

### (2) 모델 8

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 에포크 횟수(epoch_count) : 100</li> <li>- 미니배치 크기(mb_size) : 100</li> </ul>
데이터 정확도	81.8%

결과 화면	LEARNING_RATE = 0.01 abalone_exec(epoch_count = 100, mb_size = 100, report = 20)
	Epoch 20: loss=6.967, accuracy=0.808/0.811 Epoch 40: loss=6.736, accuracy=0.811/0.812 Epoch 60: loss=6.565, accuracy=0.811/0.815 Epoch 80: loss=6.412, accuracy=0.815/0.816 Epoch 100: loss=6.271, accuracy=0.818/0.818  Final Test: final accuracy = 0.818

### (3) 모델 9

구분	내용
조건	- 학습률(learning rate) : 0.01 - 에포크 횟수(epoch_count) : 100 - 미니배치 크기(mb_size) : 1000
데이터 정확도	80.7%
결과 화면	LEARNING_RATE = 0.01 abalone_exec(epoch_count = 100, mb_size = 1000, report = 20)  Epoch 20: loss=7.857, accuracy=0.817/0.813 Epoch 40: loss=7.575, accuracy=0.808/0.805 Epoch 60: loss=7.456, accuracy=0.808/0.806 Epoch 80: loss=7.365, accuracy=0.808/0.806 Epoch 100: loss=7.292, accuracy=0.807/0.807  Final Test: final accuracy = 0.807

### 3.결론

파라미터별 학습 결과를 살펴보면 다음 표와 같다.

변한 파라미터	학습률			에포크 횟수			미니배치 크기		
	0.001	0.01	0.1	10	100	1000	10	100	1000
정확도	80.8%	81.8%	82.6%	80.6%	81.8%	83.6%	83.1%	81.8%	80.7%

(비교대상이 아닌 파라미터의 경우, 학습률=0.01, 에포크 횟수=100, 미니배치 크기=100의 수치를 사용하였다.)

학습률과 에포크 횟수의 경우, 각각 0.001~0.1, 10~1000으로 변할수록 정확도가 개선되는 경향성을 보여주었다. 하지만 미니배치 크기의 경우, 크기가 커질수록 오히려 정확도가 떨어지는 모습을 보여주었다. 미니배치의 크기가 커질수록 일반화 성능은 반대로 감소하게 된다는 예측을 해볼 수 있다.

아래는 이외에 사항에 대해 추가로 모델을 테스트한 결과를 표로 나타낸 것이다.

	학습률	에포크 횟수	10	100	1000	10000	10	100	1000	10000	10	100	1000	10000
		미니배치 크기	10	10	10	10	100	100	100	100	1000	1000	1000	1000
데이터 정확도	0.001		0.808	0.817	0.835	0.834	0.814	0.808	0.818	0.837	0.165	0.802	0.807	0.817
	0.01		0.820	0.831	0.831	0.838	0.806	0.818	0.836	0.838	0.805	0.807	0.817	0.837
	0.1		0.847	0.826	0.798	0.827	0.799	0.826	0.839	0.838	0.806	0.819	0.837	0.838

학습률이 커질수록 에포크횟수가 커질수록 데이터 정확도는 증가하는 경향을 보이는데, 미니배치 크기가 커진다고 해서 정확도가 증가하지는 않는다. 오히려 에포크 횟수나 미니배치 크기가 너무 작거나 큰 경우에 대해 데이터 정확도가 증가하지 않고 감소하는 모습이 나타났다. 이 점에 따라 생각해 보면, 정확도가 선형의 값으로 나타나지 않으며, 학습률, 에포크 횟수, 미니배치 크기를 적당히 조합해야 좋은 정확도를 얻게 됨을 알 수 있다.

흥미로운 점은, (에포크 횟수/미니배치 크기)가 같으면 각각의 에포크 횟수, 미니배치 크기가 달라 도 데이터 정확도가 거의 똑같이 나타나는 경우가 많다는 점이다. 예를 들어 표에서도 볼 수 있듯, 학습률이 0.01일 때 에포크 횟수, 미니배치 크기가 각각 10, 100일 때와 100, 1000일 때 정확도가 0.806, 0.807으로 거의 똑같이 나타났다. 두 조건의 비율이 데이터의 정확도에 영향을 어느 정도 미친다는 것을 짐작할 수 있는 결과이다.

실험 결과에 따르면, 정확도는 선형의 값으로 나타나지 않으며, 무조건 학습 횟수를 늘린다고 해서 좋은 결과를 얻게 되는 것도 아님을 알 수 있었다. 실험 과정에서 에포크 횟수가 10000회로 설정했을 때 소요되는 시간이 너무 길었고, 특히 학습률이 0.1일 때 정확도의 결과가 예상하지 못한 수치로 많이 나타나는 것으로 보였기에, 학습률과 에포크 횟수를 이보다 조금 줄여야 정확도를 높일 수 있다고 생각했다.

```
LEARNING_RATE = 0.08  
abalone_exec(epoch_count = 5000, mb_size =200, report = 1000)
```

```
Epoch 1000: loss=4.821, accuracy=0.841/0.836  
Epoch 2000: loss=4.784, accuracy=0.841/0.839  
Epoch 3000: loss=4.788, accuracy=0.841/0.836  
Epoch 4000: loss=4.782, accuracy=0.840/0.838  
Epoch 5000: loss=4.782, accuracy=0.841/0.841
```

```
Final Test: final accuracy = 0.841
```

학습률을 0.08, 에포크 횟수를 5000, 미니배치의 크기를 200으로 두고 다시 한번 모델 학습을 진행해 본 결과, 0.841의 정확도를 얻을 수 있었다. 비록 표에서 볼 수 있었듯이 변경해 본 값 중에서 학습률 0.1, 에포크 횟수를 10, 미니배치의 크기를 10로 설정하는 것이 가장 큰 정확도를 얻을 수 있는 방법이었으나, 에포크 횟수, 미니배치의 크기가 너무 작은 것을 감안하면, 학습률 0.08, 에포크 횟수 5000, 미니배치의 크기 200으로 설정하는 것이 적절한 선택이라는 생각이 든다.

이번 실험에서 성능 개선을 위해 여러번 하이퍼 파라미터를 변경해보았으나, 정확도가 대부분 80~84%로 나타나며, 눈에 띄게 데이터의 정확도가 상승하지는 않았다. 이를 통해 여러 조건에서도 전복의 고리수 추정 신경망 모델이 비교적 안정적인 일반화 성능을 가지고 있음을 유추해 볼 수 있다.