

# 1장. 회귀 분석 : 전복의 고리 수 추정 신경망

부산대학교 정보컴퓨터공학부

15학번 이연걸

## 1. 개요

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

## 2. 파라미터별 학습결과 및 결론

### 2.1 학습횟수를 기준으로 변화시킨 경우

#### 2.1.1 모델 1-1

Table 1 모델 1-1

구분	내용
조건	- 학습률(learning rate) : 0.001 - 매개변수 인수값(mb_size) : 10 - 학습량(epoch) : 10
데이터 정확도	80.3%
결과화면	<pre>LEARNING_RATE = 0.001 abalone_exec(epoch_count=10, mb_size=10, report=1)  Epoch 1: loss=33.698, accuracy=0.554/0.811 Epoch 2: loss=8.181, accuracy=0.820/0.810 Epoch 3: loss=7.539, accuracy=0.813/0.805 Epoch 4: loss=7.438, accuracy=0.810/0.805 Epoch 5: loss=7.358, accuracy=0.811/0.805 Epoch 6: loss=7.289, accuracy=0.810/0.806 Epoch 7: loss=7.232, accuracy=0.810/0.806 Epoch 8: loss=7.181, accuracy=0.810/0.806 Epoch 9: loss=7.134, accuracy=0.812/0.804 Epoch 10: loss=7.097, accuracy=0.811/0.803  Final Test: final accuracy = 0.803</pre>

### 2.1.2 모델 1-2

**Table 2 모델 1-2**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.001</li> <li>- 매개변수 인수값(mb_size) : 10</li> <li>- 학습량(epoch) : 100</li> </ul>
데이터 정확도	82.2%
결과화면	<pre>LEARNING_RATE = 0.001 abalone_exec(epoch_count=100, mb_size=10, report=10)</pre> <p>Epoch 10: loss=7.104, accuracy=0.810/0.815  Epoch 20: loss=6.842, accuracy=0.809/0.815  Epoch 30: loss=6.709, accuracy=0.810/0.815  Epoch 40: loss=6.613, accuracy=0.812/0.815  Epoch 50: loss=6.525, accuracy=0.811/0.818  Epoch 60: loss=6.450, accuracy=0.813/0.817  Epoch 70: loss=6.378, accuracy=0.814/0.818  Epoch 80: loss=6.308, accuracy=0.816/0.819  Epoch 90: loss=6.240, accuracy=0.816/0.821  Epoch 100: loss=6.177, accuracy=0.817/0.822</p> <p>Final Test: final accuracy = 0.822</p>

### 2.1.3 모델 1-3

**Table 3 모델 1-3**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.001</li> <li>- 매개변수 인수값(mb_size) : 10</li> <li>- 학습량(epoch) : 1000</li> </ul>
데이터 정확도	84.0%
결과화면	<pre>LEARNING_RATE = 0.001 abalone_exec(epoch_count=1000, mb_size=10, report=100)</pre> <p>Epoch 100: loss=5.982, accuracy=0.820/0.820  Epoch 200: loss=5.519, accuracy=0.828/0.827  Epoch 300: loss=5.231, accuracy=0.834/0.831  Epoch 400: loss=5.056, accuracy=0.835/0.837  Epoch 500: loss=4.946, accuracy=0.837/0.838  Epoch 600: loss=4.878, accuracy=0.839/0.839  Epoch 700: loss=4.833, accuracy=0.840/0.840  Epoch 800: loss=4.805, accuracy=0.840/0.840  Epoch 900: loss=4.786, accuracy=0.840/0.841  Epoch 1000: loss=4.772, accuracy=0.841/0.840</p> <p>Final Test: final accuracy = 0.840</p>

## 2.1.4 모델 1-4

**Table 4 모델 1-4**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.001</li> <li>- 매개변수 인수값(mb_size) : 10</li> <li>- 학습량(epoch) : 10000</li> </ul>
데이터 정확도	83.3%
결과화면	<pre> LEARNING_RATE = 0.001 abalone_exec(epoch_count=10000, mb_size=10, report=1000)  Epoch 1000: loss=4.692, accuracy=0.842/0.834 Epoch 2000: loss=4.640, accuracy=0.842/0.833 Epoch 3000: loss=4.620, accuracy=0.843/0.833 Epoch 4000: loss=4.609, accuracy=0.844/0.833 Epoch 5000: loss=4.603, accuracy=0.843/0.833 Epoch 6000: loss=4.600, accuracy=0.844/0.833 Epoch 7000: loss=4.598, accuracy=0.843/0.833 Epoch 8000: loss=4.595, accuracy=0.844/0.832 Epoch 9000: loss=4.595, accuracy=0.843/0.833 Epoch 10000: loss=4.595, accuracy=0.844/0.833  Final Test: final accuracy = 0.833 </pre>

## 2.1.5 결론 (학습량 조정의 경우)

학습량을 조정시켜보았을 때, 확실히 학습량이 많을수록 데이터 정확도가 올라가는 성향을 보였다. 다만, 학습량이 1000번이상 진행되었을 경우에는 데이터 정확도가 진전이 없었다. 따라서, 학습량과 정확도는 비례관계가 있지만 어느정도 한계에 다다르면, 영향을 주지 못한다고 볼 수 있다.

## 2.2 학습률을 기준으로 변화시킨 경우

### 2.2.1 모델 2-1

**Table 5 모델 2-1**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.001</li> <li>- 매개변수 인수값(mb_size) : 10</li> <li>- 학습량(epoch) : 100</li> </ul>
데이터 정확도	81.5%
결과화면	<pre>LEARNING_RATE = 0.001 abalone_exec(epoch_count=100, mb_size=10, report=10)</pre> <p>Epoch 10: loss=7.017, accuracy=0.811/0.805  Epoch 20: loss=6.763, accuracy=0.812/0.806  Epoch 30: loss=6.630, accuracy=0.811/0.809  Epoch 40: loss=6.532, accuracy=0.813/0.807  Epoch 50: loss=6.447, accuracy=0.813/0.808  Epoch 60: loss=6.366, accuracy=0.815/0.811  Epoch 70: loss=6.291, accuracy=0.816/0.810  Epoch 80: loss=6.219, accuracy=0.817/0.812  Epoch 90: loss=6.148, accuracy=0.819/0.811  Epoch 100: loss=6.086, accuracy=0.819/0.815</p> <p>Final Test: final accuracy = 0.815</p>

### 2.2.2 모델 2-2

**Table 6 모델 2-2**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 매개변수 인수값(mb_size) : 10</li> <li>- 학습량(epoch) : 100</li> </ul>
데이터 정확도	84.7%
결과화면	<pre>LEARNING_RATE = 0.01 abalone_exec(epoch_count=100, mb_size=10, report=10)</pre> <p>Epoch 10: loss=6.333, accuracy=0.816/0.822  Epoch 20: loss=5.828, accuracy=0.824/0.832  Epoch 30: loss=5.515, accuracy=0.829/0.836  Epoch 40: loss=5.327, accuracy=0.833/0.838  Epoch 50: loss=5.210, accuracy=0.834/0.841  Epoch 60: loss=5.144, accuracy=0.835/0.839  Epoch 70: loss=5.099, accuracy=0.837/0.837  Epoch 80: loss=5.064, accuracy=0.837/0.838  Epoch 90: loss=5.052, accuracy=0.837/0.848  Epoch 100: loss=5.043, accuracy=0.837/0.847</p> <p>Final Test: final accuracy = 0.847</p>

### 2.2.3 모델 2-3

**Table 7 모델 2-3**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.1</li> <li>- 매개변수 인수값(mb_size) : 10</li> <li>- 학습량(epoch) : 100</li> </ul>
데이터 정확도	84.8%
결과화면	<pre> LEARNING_RATE = 0.1 abalone_exec(epoch_count=100, mb_size=10, report=10)  Epoch 10: loss=5.287, accuracy=0.832/0.851 Epoch 20: loss=5.175, accuracy=0.833/0.800 Epoch 30: loss=5.125, accuracy=0.834/0.837 Epoch 40: loss=5.154, accuracy=0.833/0.851 Epoch 50: loss=5.142, accuracy=0.834/0.848 Epoch 60: loss=5.187, accuracy=0.833/0.839 Epoch 70: loss=5.105, accuracy=0.834/0.824 Epoch 80: loss=5.123, accuracy=0.834/0.832 Epoch 90: loss=5.139, accuracy=0.835/0.851 Epoch 100: loss=5.174, accuracy=0.834/0.848  Final Test: final accuracy = 0.848 </pre>

### 2.2.4 결론 (학습률 조정의 경우)

학습률을 조정시켜보았을 때, 학습률이 높을수록 3%가량 증가한 모습을 확인할 수 있었다.

하지만, 학습량과 마찬가지로 정체가 시작되는 부분이 존재했다.

## 2.3 매개변수 인수값을 기준으로 변화시킨 경우

### 2.3.1 모델 3-1

**Table 8 모델 3-1**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 매개변수 인수값(mb_size) : 1</li> <li>- 학습량(epoch) : 100</li> </ul>
데이터 정확도	84.5%
결과화면	<pre>LEARNING_RATE = 0.01 abalone_exec(epoch_count=100, mb_size=1, report=10)</pre> <p>Epoch 10: loss=5.167, accuracy=0.835/0.833  Epoch 20: loss=5.162, accuracy=0.834/0.777  Epoch 30: loss=5.125, accuracy=0.836/0.852  Epoch 40: loss=5.152, accuracy=0.835/0.822  Epoch 50: loss=5.117, accuracy=0.835/0.850  Epoch 60: loss=5.097, accuracy=0.836/0.851  Epoch 70: loss=5.125, accuracy=0.836/0.830  Epoch 80: loss=5.122, accuracy=0.835/0.852  Epoch 90: loss=5.114, accuracy=0.834/0.845  Epoch 100: loss=5.117, accuracy=0.835/0.845</p> <p>Final Test: final accuracy = 0.845</p>

### 2.3.2 모델 3-2

**Table 9 모델 3-2**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 매개변수 인수값(mb_size) : 10</li> <li>- 학습량(epoch) : 100</li> </ul>
데이터 정확도	82.7%
결과화면	<pre>LEARNING_RATE = 0.01 abalone_exec(epoch_count=100, mb_size=10, report=10)</pre> <p>Epoch 10: loss=6.002, accuracy=0.819/0.807  Epoch 20: loss=5.554, accuracy=0.827/0.817  Epoch 30: loss=5.260, accuracy=0.832/0.818  Epoch 40: loss=5.090, accuracy=0.835/0.829  Epoch 50: loss=4.988, accuracy=0.837/0.835  Epoch 60: loss=4.919, accuracy=0.838/0.828  Epoch 70: loss=4.881, accuracy=0.839/0.838  Epoch 80: loss=4.839, accuracy=0.840/0.841  Epoch 90: loss=4.831, accuracy=0.840/0.837  Epoch 100: loss=4.802, accuracy=0.841/0.827</p> <p>Final Test: final accuracy = 0.827</p>

### 2.3.3 모델 3-3

**Table 10 모델 3-3**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 매개변수 인수값(mb_size) : 100</li> <li>- 학습량(epoch) : 100</li> </ul>
데이터 정확도	82.1%
결과화면	<pre>LEARNING_RATE = 0.01 abalone_exec(epoch_count=100, mb_size=100, report=10)</pre> <p>Epoch 10: loss=7.106, accuracy=0.808/0.810  Epoch 20: loss=6.866, accuracy=0.809/0.813  Epoch 30: loss=6.738, accuracy=0.811/0.814  Epoch 40: loss=6.641, accuracy=0.811/0.815  Epoch 50: loss=6.549, accuracy=0.815/0.812  Epoch 60: loss=6.480, accuracy=0.814/0.816  Epoch 70: loss=6.404, accuracy=0.815/0.818  Epoch 80: loss=6.331, accuracy=0.815/0.820  Epoch 90: loss=6.267, accuracy=0.816/0.819  Epoch 100: loss=6.203, accuracy=0.818/0.821</p> <p>Final Test: final accuracy = 0.821</p>

### 2.3.4 모델 3-4

**Table 11 모델 3-4**

구분	내용
조건	<ul style="list-style-type: none"> <li>- 학습률(learning rate) : 0.01</li> <li>- 매개변수 인수값(mb_size) : 1000</li> <li>- 학습량(epoch) : 100</li> </ul>
데이터 정확도	80.9%
결과화면	<pre>LEARNING_RATE = 0.01 abalone_exec(epoch_count=100, mb_size=1000, report=10)</pre> <p>Epoch 10: loss=11.491, accuracy=0.790/0.803  Epoch 20: loss=7.744, accuracy=0.818/0.819  Epoch 30: loss=7.523, accuracy=0.811/0.812  Epoch 40: loss=7.423, accuracy=0.810/0.811  Epoch 50: loss=7.342, accuracy=0.810/0.810  Epoch 60: loss=7.272, accuracy=0.810/0.810  Epoch 70: loss=7.210, accuracy=0.810/0.810  Epoch 80: loss=7.157, accuracy=0.811/0.810  Epoch 90: loss=7.110, accuracy=0.811/0.810  Epoch 100: loss=7.067, accuracy=0.811/0.809</p> <p>Final Test: final accuracy = 0.809</p>

### 2.3.5 결론 (매개변수 인수값 조정의 경우)

매개변수 인수값을 조정시켜보았을 때, 매개변수 인수값이 증가할수록 데이터 정확도가 떨어지는 경향을 보였다. 이는 매개변수 인수값이 적을수록 epoch당 계산횟수가 많기 때문에 정확도가 상대적으로 올라가지않았나 생각되며 이 때문에 학습에 걸리는 시간도 오래걸렸다.

## 2.4 결론

하이퍼 파라미터에 따라 정확도가 어떻게 달라지는지 살펴보았다. 하이퍼 파라미터를 변경하여 초기 80.3%의 모델을 최대 84.8%까지 정확도 향상을 이뤄낼 수 있었다. 따라서, 정확도를 높일수 있는 적절한 하이퍼 파라미터값들을 조합하면 학습에 도움이 될 것이다.