|  |  |
| --- | --- |
| 교육 제목 | **딥러닝, 경사하강법** |
| 교육 일시 | 2021년 11월 30일 화요일 |
| 교육 장소 | 영우글로벌러닝 2층 |
| **교육 내용** | |
| 오전 | 경사 하강법은 오차를 수정하는 기법  대부분의 데이터는 입력 변수가 여러개인 다중 회귀를 이용하며, 다중 회귀는 최소 제곱법을 쓰지 못하기 때문에 경사 하강법은 매우 중요합니다.  https://blog.kakaocdn.net/dn/lvj7U/btqDY2FNxkA/P151C8ekNcEnIWfzlf1E91/img.png  경사 하강법은 먼저 기울기 a와 오차간의 관계를 알아야 함  **<단일 회귀 경사 하강법 적용>**  https://blog.kakaocdn.net/dn/bU5Ii6/btqDYu3xsJe/jWneHQkNDOi3ycrIiyfNJ1/img.png  **a와 오차의 관계함수를 이용해 그 기울기가 가장 작을때의 점 m을 찾는 것이 바로 경사 하강법. 기울기 a와 똑같이 y절편 b도 오차와의 관계가 동일하기 때문에 b도 경사 하강법을 이용해 구한다.**  '**학습률'**을 이용합니다. 1보다 작은 학습률을 곱해줘서 화살표가 아래로 향해 m을 효율적으로 찾게끔 해주는 것  https://blog.kakaocdn.net/dn/cux92s/btqDZYv4UEG/PSlKiPpiMmG8XUmAzu5gsK/img.png  . |
| 오후 | 공부 시간 x와 성적y의 리스트를 만들기  data = [[2,81],[4,93],[6,91],[8,97]]  x = [i[0] for i in data]  y = [i[1] for i in data]  # 그래프로 나타내기  plt.figure(figsize=(8,5))  plt.scatter(x,y)  plt.show()  # 리스트로 되어 있는 x와y값을 넘파이 배열로 바꾸기(인덱스를 주어 하나씩 불러와 계산이 가능하게 하기 위함)  x\_data = np.array(x)  y\_data = np.array(y)  # 기울기 a와 절편b의 값 초기화  a = 0  b = 0  # 학습률 정하기  lr = 0.05  # 몇 번 반복될지 설정(0부터 세므로 원하는 수+1)  epochs = 2001  # 경사 하강법 시작  for i in range(epochs):  y\_pred = a\*x\_data+b  error = y\_data-y\_pred  # 오차 함수를 a,b로 미분한 값  a\_diff = -(1/len(x\_data))\* sum(x\_data\* (error))  b\_diff = -(1/len(x\_data))\* sum(y\_data-y\_pred)  # 학습률을 곱해 기존의 a,b값을 각각 업데이트  a = a-lr \* a\_diff  b = b-lr \* b\_diff    #100번 반복될 때마다 현재 a값, b값을 출력  if i % 100 == 0:  print('epochs=%.f, 기울기=%.04f, 절편=%.04f' % (i,a,b))    # 앞서 구한 기울기와 절편을 이용해 그래프를 다시 그리기  y\_pred=a\*x\_data+b  plt.scatter(x,y)  plt.plot([min(x\_data), max(x\_data)],[min(y\_pred),max(y\_pred)])  plt.show  https://blog.kakaocdn.net/dn/cjnUVc/btqD0hRPKts/y9FcgkrFukAz63KaRjrxKK/img.pnghttps://blog.kakaocdn.net/dn/cneRba/btqD1OBl1kU/rhWkjE9XsTUBMtUcUpStMK/img.png  epochs가 100단위로 반복될 때마다 a=2.3, b=79 로 수렴  경사 하강법이 최소 제곱법과 마찬가지로 최소의 오차를 찾아서 최선의 회귀선을 그려주는 것임을 알 수 있다 |