5월 1주차 산학협력 딥 러닝 스터디

강의의 전체적 목표는 머신러닝에 대한 기본적인 이해이다. 기본적인 알고리즘을 이해해야 한다. 그것을 바탕으로 텐서플로우와 파이썬을 이용해 실습을 할 예정이다.

머신러닝은 무엇일까?

일종의 소프트웨어이다. Explicit program은 제한점이 많다. 이에 따라서 1959년 자동으로 데이터를 기반으로 학습하여 프로그램을 구성하는 것을 머신러닝이라고 한다.

이때의 supervised learning이라고 부르는 학습이 존재하는데, 이미 레이블이 달려있거나 이미 관련 정보가 정해진 러닝이다.

Unsupervised learning은 이미 레이블이 달려있지 않거나 일일히 분류를 할 수 없는 스스로 학습하는 러닝이다.

Training data set을 통해서 특정 레이블이 달려있는 데이터셋을 통해서 머신러닝을 사전 학습을 시키게 된다.

이런 supervised learning을 결과값에 따라서 다양하게 나누어진다. Regression, classification, multi labled classification으로 나뉘게 된다. Regression은 특정 값을 예측하게 되는 모델이다. Classification은 p/f와 같이 binary로 예측하는 모델이다. 마지막 모델은 여러 개의 계층으로 나누는 것이다.

Tensorflow은 data flow graph를 통해서 numerical한 계산을 하며 파이썬을 사용한다. 텐서플로우 이하 tf에서는 그래프를 전체적으로 그리는 것으로 시작한다. 각각의 노드에 값 또는 연산을 넣고 그래프를 그린 후에 session을 run 시킴으로써 출력을 진행하게 된다. 이때에 노드를 placeholder이라는 노드를 만들게 되면, feed\_dict라는 것을 실행시키며 해당 노드에 값을 대입시키며 실행 시킬 수 있다. tensor에서는 차원을 rank라고 하며 shape을 통해서 각각의 차원에 얼마만큼의 element를 가지고 있는지 표현한다. 또한 데이터타입을 float32, int32를 사용하는게 대부분이다.

Regression을 사용하게 되면 train 과정을 필수적으로 거치게 된다. 다음의 과정에서 학습을 하며 예측을 하게 된다. 특정 데이터를 통해서 예측을 하는 것이다. 이때의 예측에서는 가설을 세우게 된다. 이런 가설 중에서 linear로 예측하는 모델을 짜게 되면 linear regression이 되는 것이다. 이때의 선을 h(x) = Wx+b라는 선을 생성하게 되는데, 해당 선이 적당한 선인지 계산하는 것을 cost function 또는 lost function이라고 한다. 다양하게 계산을 진행하는데, (h(x) -y)^2을 통해서 cost를 계산하게 된다. 각각의 값에 대해서 (h(x) -y)^2을 더하고 평균값을 구하게 된다. 이때의 cost를 가장 작게 하는 것이 학습의 목표가 된다.

이때 linear regression을 tensorflow를 통해서 구현하게 되면, variable을 통해서 weight를 통해서 기울기를 구성하고, bias를 통해서 y절편을 구성한다. 이때, cost를 reduce\_mean으로 구한다, 이때 세션을 구성하고 variable을 initialize한다. 이후에 해당 cost를 minimize를 진행한다. 학습이 일어나면 w와 b를 조절하게 된다. 이때 placeholder을 사용하면 학습 데이터의 편집이 좀더 간편해진다.

Cost를 최소화하는 것이 regression의 목적인데, 이때 사용하는 알고리즘이 gradient descent algorithm이다. 이때 해당 알고리즘은 어디서 시작하든 상관이 없다. 계속해서 경사도를 구하게 되면 가장 낮아지는 점을 찾을 수 있다. 해당 기울기를 계속해서 조금씩 조정하여 cost를 통해서 미분을 통해 기울기의 최소값을 찾게 된다. 이러한 알고리즘은 어떤 모양의 function을 가지고 있는지 확인한 후에 진행 할 수 있어야 한다.

이러한 알고리즘을 구현하기 위해서는 matplotlib이라는 library가 필요하다. Cost와 W가 변하는 것을 계산하며 조금씩 W의 값을 조정해주는 것이다. 이러한 조정은 cost의 기울기를 기준으로 실행하게 된다. 텐서플로우에서는 미분말고도 optimizer을 통해서 최소점을 구할 수도 있다.

이러한 regression에서 여러 개의 값이 주어지는 경우에 어떤 식으로 구성하는지를 배워보자. 이때에는 선형적으로 구성해준다. Linear matrix 형식으로 구성해준다. 결과에 대한 각각의 데이터셋을 instance라고 부르는데, 많은 수의 instance가 존재하면 matrix를 쓰게 되면 더욱 간단하게 데이터의 계산을 정리를 할 수 있다. X가 nxm일 때 n은 instance개수, m은 var개수이다. 이에 비해 w는 var개수x1이다. 이러한 matrix를 통해서 간단하게 계산이 가능해지는 것이다.

원래의 wx+b꼴의 함수에서 변수 x를 즉 데이터셋을 행렬로 계산하는 것만 바뀌었다. 이에 따라서 데이터를 대입할 때에 좀더 간단하게 대입하기 위해서 nxm vetor을 쓰는 것이 편해보인다. 또한 너무 많은 데이터를 입력하게 될 경우에 일일히 수기로 입력하는 것은 굉장히 어려움으로 numpy를 통해서 dataframe을 구성하여 넣는 것이 대부분이다. 이때 전처리 과정을 통해서 데이터를 고치고 간단하게 만들어 넣게 된다. 너무 큰 파일일 경우에 queue runners를 이용하게 된다. 특정한 queue에 데이터를 올려놓고 자르고 쓰는 것이 보통이다. 그러한 자른 데이터를 batch사이즈를 통해서 특정 양만큼 루프를 돌며 값을 구성하게 된다. 학습시키며 shape을 신경써서 구성해야한다.