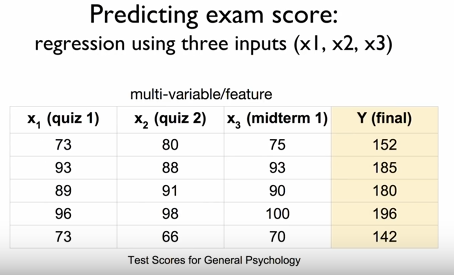
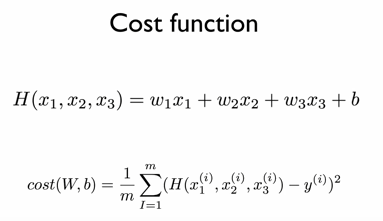
**4장. multi-variable linear regression**

(\*) 한 개의 입력이 아닌 여러 개(X1, X2, X3)의 입력으로 Y를 예측하는 것

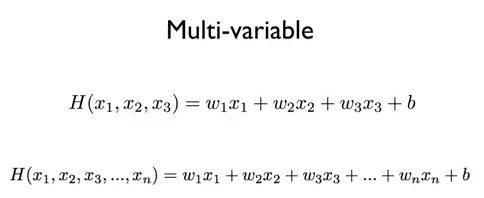


위 예제의 3개 입력에 대한 가설과 cost함수는 아래 수식으로 표현된다.

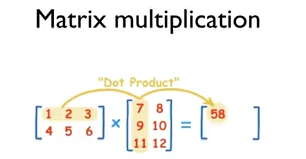


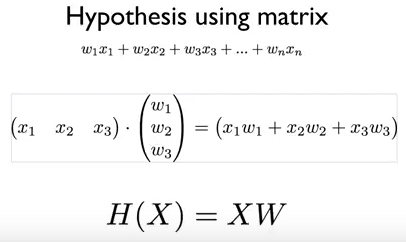
* 학습해야 할 W값이 3개로 많아짐

(\*) 입력이 n개인 경우, 가설은 아래와 같은 수식으로 표현할 수 있다



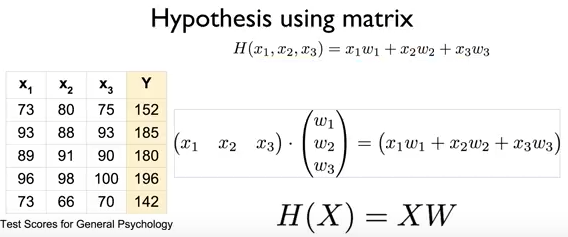
(\*) 입력이 많아지는 경우의 수식을 위와 같이 나열하지 않고, 매트릭스 곱셈으로 표현할 수 있다.



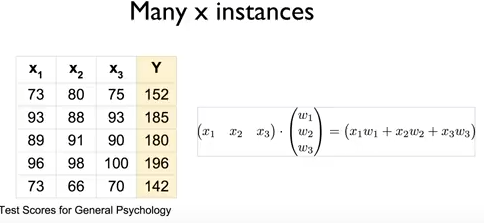


(\*) 입력값과 Weight값이 매트릭스라는 것을 나타내기 위하여 대문자로 표시하며, X\*W 순서로 쓴다.

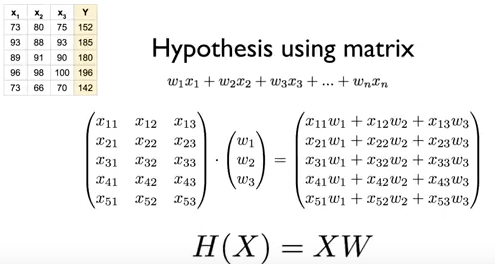
(\*) 즉, 위의 예제문제는 아래와 같은 매트릭스 곱셈으로 표현할 수 있다.



(\*) 실제로 학습해야 할 데이터(instance)가 복수개(아래 예제는 5개)인 경우 각각의 매트릭스 곱셈으로 학습하여도 되지만, 복수개의 데이터(instance)도 매트릭스 곱셈으로 표현하면 한번 연산으로 결과값을 계산해 낼 수 있게 된다.



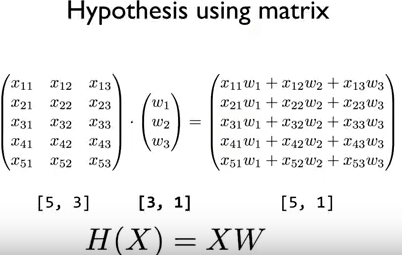
(\*) instance가 늘어나도 W는 변함없다.

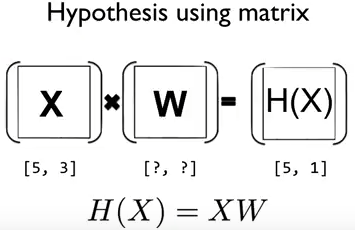


(\*) 입력값 매트릭스의 차원은 [instance갯수, 입력값갯수]

출력값 매트릭스의 차원은 [instance갯수, 예측값(Y)개수] 으로 미리 알수 있다

* Weight 매트릭스의 차원은 [입력값갯수, 예측값(Y)개수]로 계산해 낼 수 있다.

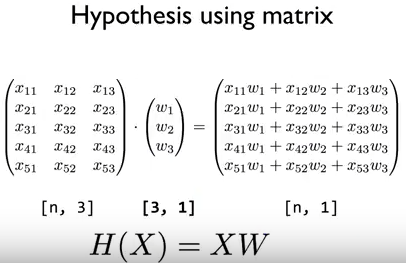




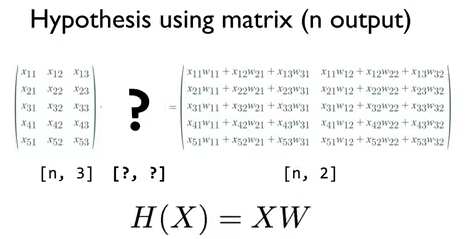
(\*) 위 Weight 매트릭스는 몇 차원일까요? [3, 1]

(\*) 일반적으로 instance의 개수 n개는 확정된 값이 아니며,

실제 구현할때 Numpy에서는 -1 로 표시하고, Tensorflow에서는 none 으로 표시한다.



(\*) 매트릭스 곱셈을 사용하면, instance 개수가 n개인 경우와 출력이 n개인 경우 모두에 확장하여 사용할 수 있다



(\*) 실제로 TensorFlow 등에서 구현할 때는 XW 순서대로 매트릭스곱셈을 해야 한다. 순서에 주의하자