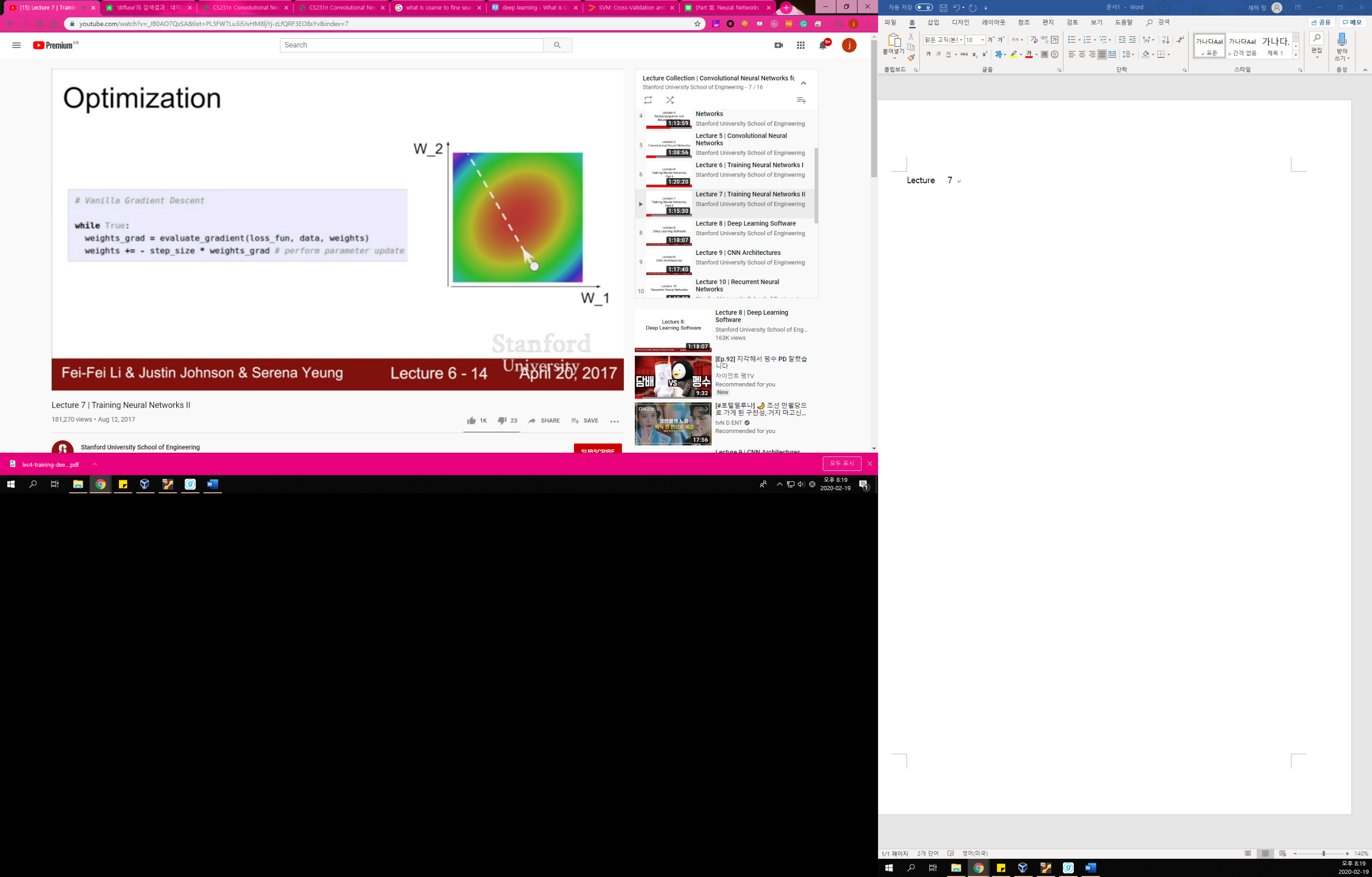
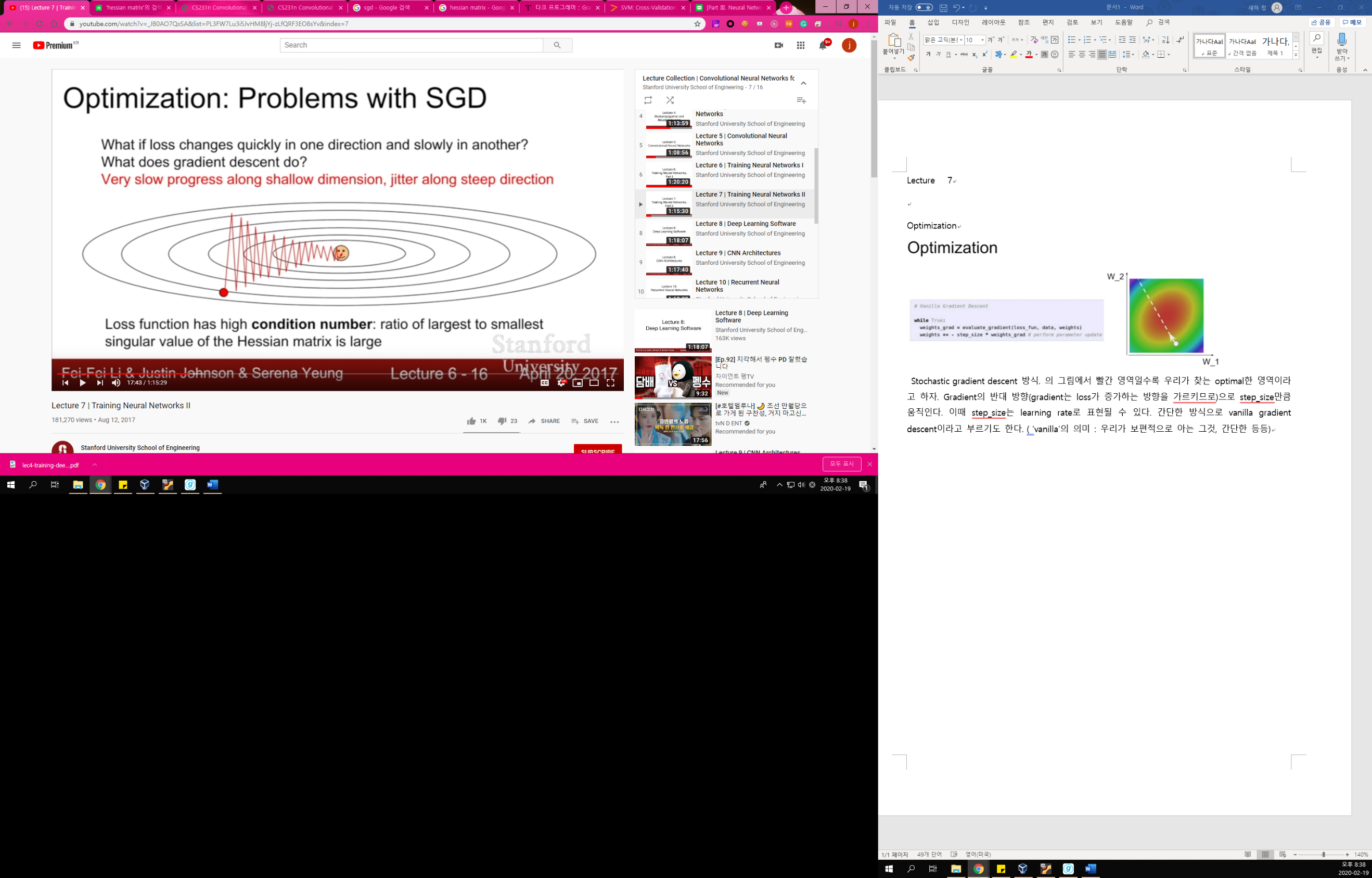
Lecture 7

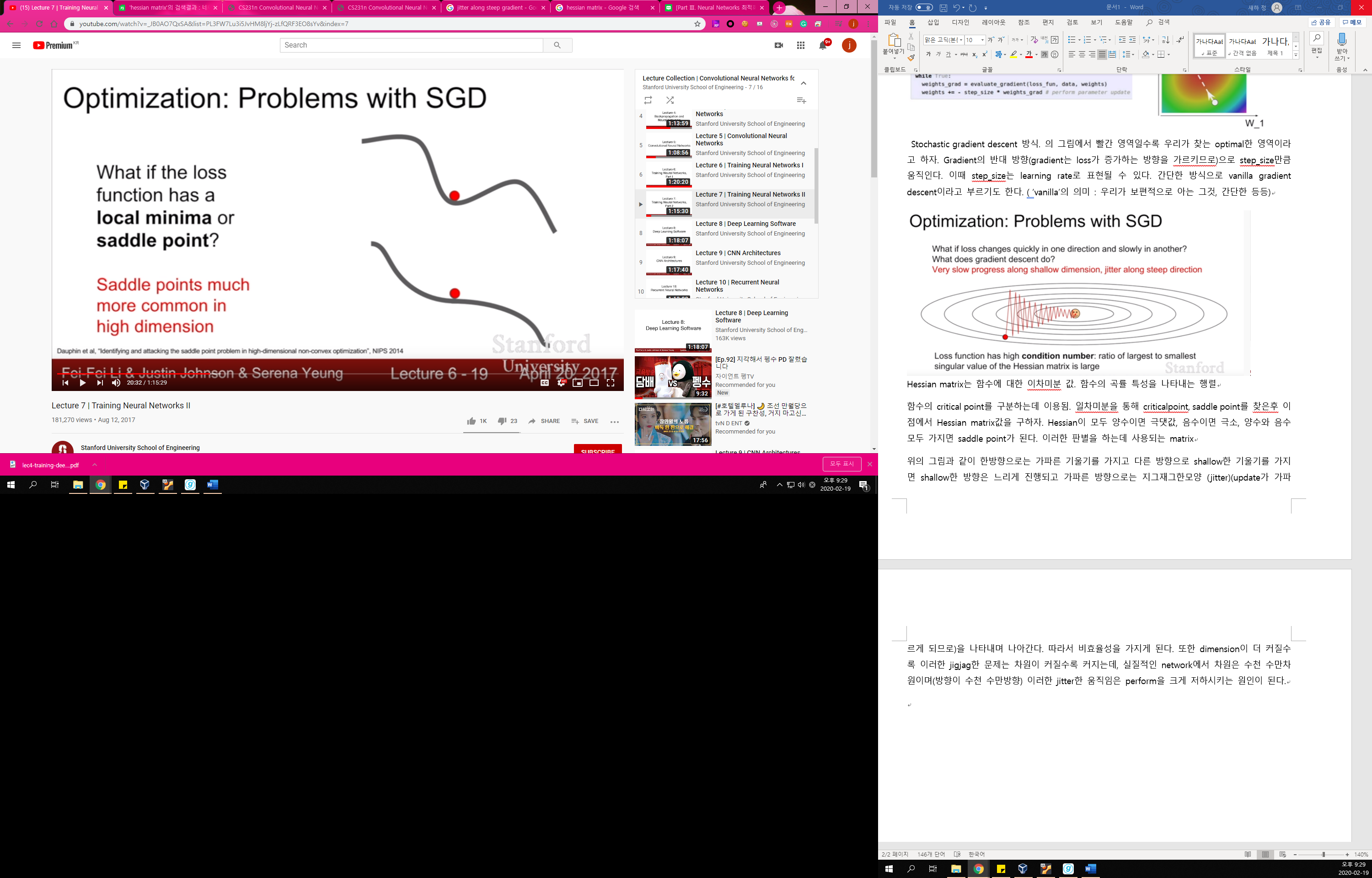
Optimization

 Stochastic gradient descent 방식. 의 그림에서 빨간 영역일수록 우리가 찾는 optimal한 영역이라고 하자. Gradient의 반대 방향(gradient는 loss가 증가하는 방향을 가르키므로)으로 step\_size만큼 움직인다. 이때 step\_size는 learning rate로 표현될 수 있다. 간단한 방식으로 vanilla gradient descent이라고 부르기도 한다. ( ‘vanilla’의 의미 : 우리가 보편적으로 아는 그것, 간단한 등등)

Hessian matrix는 함수에 대한 이차미분 값. 함수의 곡률 특성을 나타내는 행렬

함수의 critical point를 구분하는데 이용됨. 일차미분을 통해 criticalpoint, saddle point를 찾은후 이점에서 Hessian matrix값을 구하자. Hessian이 모두 양수이면 극댓값, 음수이면 극소, 양수와 음수 모두 가지면 saddle point가 된다. 이러한 판별을 하는데 사용되는 matrix

위의 그림과 같이 한방향으로는 가파른 기울기를 가지고 다른 방향으로 shallow한 기울기를 가지면 shallow한 방향은 느리게 진행되고 가파른 방향으로는 지그재그한모양 (jitter)(update가 가파르게 되므로)을 나타내며 나아간다. 따라서 비효율성을 가지게 된다. 또한 dimension이 더 커질수록 이러한 jigjag한 문제는 차원이 커질수록 커지는데, 실질적인 network에서 차원은 수천 수만차원이며(방향이 수천 수만방향) 이러한 jitter한 움직임은 perform을 크게 저하시키는 원인이 된다.

loss function 이 그림과 같이 생겼고 update가 local minima나 saddlepoint에서 멈췄다면 더 이상 update가 진행되지 않을 수 있다. 왜냐하면 gradient값이 0이기 때문에 -방향으로 움직일수 없으므로 gradients stuck현상이 발생하기 때문이다. Local minima에서 어떤 방향으로 움직이던 loss가 증가한다면, saddlepoint에서는 어떤 방향으로는 loss가 증가하고 어떤방향으로는 loss가 감소한다. 이러한 문제는 high dimension에서 빈번히 일어나고 따라서 saddlepoint는 high dimension에서 더 큰 문제가된다.(local minima보다) 또한 saddle point근처에서는 gradient가 0에 가깝기 때문에 slow하게 진행되거나 거의 향상되지 않으므로 big problem!

