from sklearn.datasets import load\_iris iris=load\_iris()

X=tf.constant(iris.data[:,[0,1,2]],dtype=tf.float32)

**설명1: SepalLengthCm,SepalWidthCm,PetalLengthCm를 입력값 x로 한다.**

y=tf.constant(iris.data[:,3],dtype=tf.float32)

w=tf.Variable(tf.random.normal([3,5]))

b=tf.Variable(tf.random.normal([5]))

u=tf.nn.relu(X@w+b)

**설명2: input 값 X와 가중치값인 w, 상수항 b를 사용하여 비선형함수인 relu 함수를 사용하고** hidden layer를 활성화한다.

ww=tf.Variable(tf.random.normal([5,5]))

bb=tf.Variable(tf.random.normal([5]))

uu=tf.nn.relu(u@ww+bb)

www=tf.Variable(tf.random.normal([5,1]))

bbb=tf.Variable(tf.random.normal([]))

pred\_y=uu@www+bbb

**설명3: 중간 레이어를 2번 거치면서 PetalWidthCm에 대한 예측을 생성한다.**

mse=tf.reduce\_mean(tf.square(y-pred\_y))

**설명4: 실제 값인 y와 오류 값인 pred\_y의 오차를 제곱하여 평균을 구한 값을 mse에 넣는다.**

optimizer = tf.train.GradientDescentOptimizer(learning\_rate=0.001) train\_op=optimizer.minimize(mse)

**설명5: 오류를 최소하하기 위하여 학습률을 0.001로 한 optimizer의 최소 값을 train\_op에 삽입한다.**

costs=[]

tf.global\_variables\_initializer().run()

for i in range(300):

sess.run(train\_op)

costs.append(mse.eval())

plt.plot(costs)