

Type something...

01) Linear Regression

## 01) Linear Regression

# Full training code

```
x_train = torch.FloatTensor([[1], [2], [3]])
y_train = torch.FloatTensor([[2], [4], [6]])

W = torch.zeros(1, requires_grad=True)
b = torch.zeros(1, requires_grad=True)

optimizer = optim.SGD([W, b], lr=0.01)

nb_epochs = 1000
for epoch in range(1, nb_epochs + 1):
    hypothesis = x_train * W + b
    cost = torch.mean((hypothesis - y_train) ** 2)

    optimizer.zero_grad()
    cost.backward()
    optimizer.step()
```

### 한번만

1. 데이터 정의
2. Hypothesis 초기화
3. Optimizer 정의

### 반복!

1. Hypothesis 예측
2. Cost 계산
3. Optimizer 로 학습

1. `x_train, y_train` define `torch.FloatTensor(array)` : 데이터셋을 가져오는 단계
2. Weight, bias 초기화 `torch.zeros(size, require_grad = True)` `require_grad`: backpropagation 진행
3. optimizer 설정 `optim.SGD([W, b], lr = 0.01)`
4. epochs 만큼 반복
5. hypothesis 예측 `hypothesis = x_train * W + b`
6. cost 계산하기 `cost = torch.mean((hypothesis - y_train) ** 2)` , MSE 사용
7. optimizer 학습 `optimizer.zero_grad()` :누적 값을 zero로 초기화 시키기, `cost.backward()` :backpropagation 진행, `optimizer.step()` : 가중치 업데이트