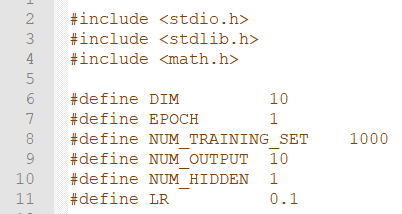
**HW\_2**

전자통신공학과 201670702 신민정

**Code설명**

***Header&Parmeter setting***

****

학습에 필요한 parameter들을 지정해 줍니다.

여기서 EPOCH과 Learning Rate(이하 LR)은 hyper parameter로, 변경하며 학습을 수행하고 최적의 parameter를 선택합니다.

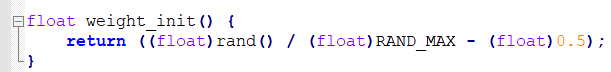
***File Pointer***

**시계이(가) 표시된 사진

자동 생성된 설명**

Data를 read,write할 포인터를 지정해 줍니다.

***Weight initialization***

******

***실내이(가) 표시된 사진

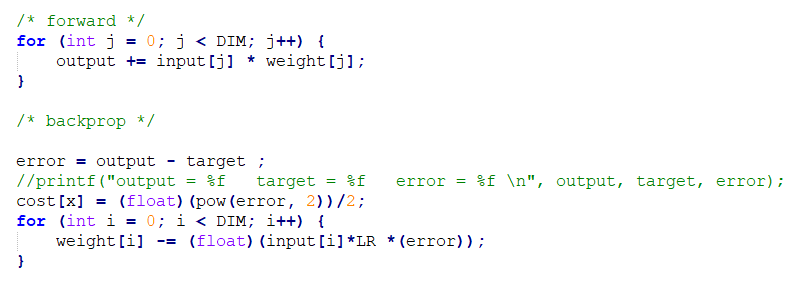
자동 생성된 설명***

Weight를 -0.5~0.5사이로 초기화 시켜줍니다.

***Training***

***쥐고있는, 남자, 방, 재생이(가) 표시된 사진

자동 생성된 설명***

******

Epoch을 돌면서 forward와 backpropagation을 진행합니다.

Training의 목적은 가중치를 수정하여 cost값을 최소로 만드는 것입니다.

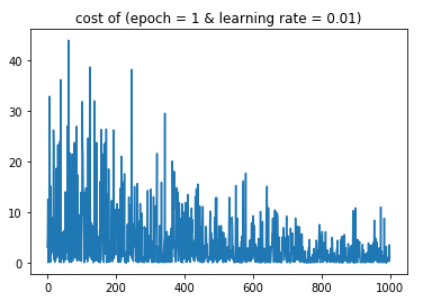
Cost는 MSE(Mean Square Error)로 설정하였습니다. 간단한 단일 perceptron이기 때문에 regulization은 하지 않았습니다.

**결과**

Parameter인 epoch과 learning rate를 변경해 가면 weight와 cost를 비교해 보았습니다.

스크린샷이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명

지도, 텍스트이(가) 표시된 사진

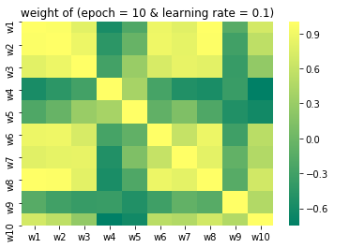
자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명

지도, 텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 지도이(가) 표시된 사진

자동 생성된 설명



**Epoch = 1 & Learning Rate = 0.1**그리기이(가) 표시된 사진

자동 생성된 설명

**Epoch = 10 & Learning Rate = 0.1**텍스트이(가) 표시된 사진

자동 생성된 설명

**Epoch = 1 & Learning Rate = 0.01**개체, 시계이(가) 표시된 사진

자동 생성된 설명

**Epoch = 10 & Learning Rate = 0.01**텍스트이(가) 표시된 사진

자동 생성된 설명

**결과해석&고찰**

총 네가지 경우를 살펴보았다.

*1. Epoch = 1 & Learning Rate = 0.1*

*2. Epoch = 10 & Learning Rate = 0.1*

*3. Epoch = 1 & Learning Rate = 0.01*

*4. Epoch = 10 & Learning Rate = 0.01*

cost와 weight그래프를 비교해 보았습니다.

*1. Epoch = 1 & Learning Rate = 0.1 & 2. Epoch = 10 & Learning Rate = 0.1 :*

*(*learning rate가 같고 epoch이 다른 경우)

이 두 경우에는 약 400epoch에서 cost가 0으로 수렴하고 weight들도 어느 일정값에 모두 수렵하였습니다. Cost/weight가 어느정도 수렴하였을 때, 학습을 멈추어 overfitting을 방지해야 하는데, *2*의 경우는 overfitting의 위험이 있습니다.

*1. Epoch = 1 & Learning Rate = 0.1 & 3. Epoch = 1 & Learning Rate = 0.01:*

(epoch이 같고 learning rate가 다른 경우 )

이 두 경우 cost/weight의 수렴속도가 다른 것을 알 수 있습니다. Learning rate가 작을수록 학습속도가 느리다. 학습시간이 충분하다면 learning rate를 아주 작게하여 발산을 방지하고 objective function의 global minimum을 찾는 것이 최선인것처럼 보이나, 이 경우 global minimum이 아닌 local minimum에 빠질 수 있기 때문에 learning rate를 단순히 작게하는 것이 최선은 아닙니다.

Epoch을 너무 크게하여 학습하다보면 overfitting에 빠질 위험이 있고 epoch을 작게하여 학습하면 학습이 덜 된 불완전한 model이 나올 수 있습니다.

learning rate는 너무 크다면 overshotting되어 버리고, 너무 작다면 global minimum에 빠질 위험이 있습니다.

따라서 parameter를 조정할 때 epoch과 learningrate에 주의를 기울여야합니다.

다양한 시도를 하며 parameter를 튜닝하는 방법도 있지만, parameter를 지정해주고 최적을 computer가 찾아주는 **gridsearch**를 이용할 수 도 있습니다. (gridsearch는 epoch,learning rate뿐만 아니라 regulization과 activation function 설정시에도 아주 유용합니다)

또한 **EarlyStopping, ModelCheckpoint, ReduceLRonPlateau**를 이용할 수도 있습니다.

loss또는 accuracy가 어느정도 수렴했다면 학습을 중단하는 ***EarlyStopping***으로 epoch&learning rate의 문제들를 해결할 수 있습니다.

또한 컴퓨팅 파워나 메모리의 문제로 학습이 중단되었을 때 ***ModelCheckpoint***을 사용하여 학습을 이어나갈 수 있습니다.

Objective function의 global minimum을 찾아나갈 때, initialization의 영향으로 local minimum에 빠질 위험이 있는데, ***ReduceLROnPlateau***을 사용하여 해결할 수 있습니다.

***[ModelCheckpoint*** : 모델이 학습하면서 정의한 조건을 만족했을 때 Model의 weight 값을 중간 저장. 학습시간이 오래걸린다면, 모델이 개선된 validation score를 도출해낼 때마다 weight를 중간 저장함으로써, 혹시 중간에 memory overflow나 crash가 나더라도 다시 weight를 불러와서 학습을 이어나갈 수 있기 때문에, 시간을 save해 줄 수 있다.

***EarlyStopping*** : model의 성능 지표가 설정한 epoch동안 개선되지 않을 때 조기 종료

***ReduceLROnPlateau*** : Local Minima에 빠져 더이상 학습률이 개선되지 않고 정체되었을 때, learning rate를 늘리거나 줄여주는 방법으로 빠져나온다.]

지도, 텍스트이(가) 표시된 사진

자동 생성된 설명

이 그래프는 델타학습규칙으로 update되는 weight들의 그래프 입니다.

W1은 이상적으로 수렴이 잘 되는 것을 볼 수 있고, w10의 경우는 초기에 많은 흔들림이 있는 것을 그래프로 확인할 수 있습니다.

이는 weight의 초기화에 따라 차이가 나는 것입니다. W1같은 경우에는 초기에 random하게 initializing된 값이 최적의 weight로 잘 수렴할 수 있도록 설정되어있음을 알 수 있고, w10의 경우는 그 반대일 것이라고 예상할 수 있습니다. 시계이(가) 표시된 사진

자동 생성된 설명

초기값에 따라 수렴정도에 차이가 있지만, 학습이 진행됨에 따라 일정값으로 수렴하는 것으로 보아 적절하게 학습되었음을 알 수 있습니다. 가중치 갱신이 1000보다 적었다면 weight들이 최적으로 학습되지 못했을 것입니다.

Weight가 최종적으로 수렴하는것도 충요하지만, weight initialization이 갖는 의의도 고려해보아야 합니다.

Weight initialization으로 weight가 vanishing되거나 exploding되는 것을 방지할 수 있습니다.

아래 표는 Weight initialization으로 test error를 줄일 수 있다는 내용의 논문에서 발췌해온 그래프 입니다.

지도, 텍스트이(가) 표시된 사진

자동 생성된 설명

Weight initialization의 방법에는 표준 정규분포, Xavier, he initialization등 다양하며, 최근에는Xavier/he initialization을 많이 사용하는 추세입니다.

아래는 Xavier initialization의 우수성을 나타내는 그래프 입니다.

텍스트, 지도이(가) 표시된 사진

자동 생성된 설명

**전제code**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**스크린샷이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**