exam 개인학습

Lec 9

Type something...

01) Tips
01_Maximum Likelihood Estimation
02_Overfitting
1) Regularization
03_Data Preprocessing

01) Tips

01_Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE)

$$K \sim \mathcal{B}(n, \theta)$$

$$P(K = k) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$
$$= \frac{n!}{k!(n-k)!} \cdot \theta^k (1-\theta)^{n-k}$$

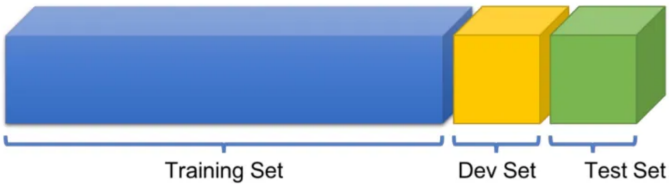
Likelihood (이전 글 참조) 는 확률 분포가 가정된 상황에서 관측값이 가지는 확률값을 의미합니다.
Maximum Likelihood Estimation(MLE)는 관측되는 데이터들을 가장 잘 모델링하는 확률분포의 parameter를 찾는 알고리즘입니다.

- MLE는 관측되는 데이터들을 가장 잘 모델링하는 확률분포의 **parameter 를 찾는 알고리즘이다. 특정 분포를 따른다고 했을 때 미분값이 0인 지점을 통해서 값을 구할 수 있다.**

- 나중에 Tutorial on maximum likelihood estimation 논문을 통해 자세히 알아보자.

02_Overfitting

Overfitting



- overfitting은 training set에 너무 과적합되어있는 것을 말하며 이것을 해결하기 위해 validation set을 따로 두어 overfit를 피할 수 있다.

Overfitting

- More Data
- Less features
- **Regularization**

- Overfittng을 피할 수 있는 방법을 3가지 제시 중이다.

1. More Data: 더 많은 데이터
2. Less Features: parameter를 줄이기
3. Regularization: 규제

1) Regularization

Regularization

- Early Stopping
- Reducing Network Size
- Weight Decay
- Dropout
- Batch Normalization

- 정규화를 하는 방법은 5가지가 있다.

1. Early stopping: 조기 종료→ 일정 수준 이상 정확도가 안높아지면 종료
2. Reducing Network size: 만든 모델의 사이즈를 줄이기
3. Weight Decay: parameter의 크기를 제한 → 더 많은 parameter를 사용하면 규제가 붙음 L1 regularziation, L2 regularization
4. Dropout: 일부 network를 제거
5. Batch Normalization: 배치 사이즈 마다 정규화

Learning Rate

적절한 숫자로 시작해 발산하면 작게, cost가 줄어들지 않으면 크게 조정하자.

```
model = SoftmaxClassifierModel()
```

```
optimizer = optim.SGD(model.parameters(), lr=1e-1)
```

```
train(model, optimizer, x_train, y_train)
```

```
Epoch    0/20 Cost: 1.341573
Epoch    1/20 Cost: 1.198802
Epoch    2/20 Cost: 1.150877
Epoch    3/20 Cost: 1.131977
Epoch    4/20 Cost: 1.116242
Epoch    5/20 Cost: 1.102514
Epoch    6/20 Cost: 1.089676
Epoch    7/20 Cost: 1.077479
Epoch    8/20 Cost: 1.065775
Epoch    9/20 Cost: 1.054511
Epoch   10/20 Cost: 1.043655
Epoch   11/20 Cost: 1.033187
Epoch   12/20 Cost: 1.023091
Epoch   13/20 Cost: 1.013356
Epoch   14/20 Cost: 1.003968
Epoch   15/20 Cost: 0.994917
Epoch   16/20 Cost: 0.986189
Epoch   17/20 Cost: 0.977775
Epoch   18/20 Cost: 0.969660
```

- parameter를 잘 조정해서 발산하지 않게 해보자.

03_Data Preprocessing

Data Preprocessing

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}$$

여기서 σ 는 standard deviation, μ 는 평균값 이다.

```
mu = x_train.mean(dim=0)
```

```
sigma = x_train.std(dim=0)
```

```
norm_x_train = (x_train - mu) / sigma
```

```
print(norm_x_train)
```

```
tensor([[-1.0674, -0.3758, -0.8398],
        [ 0.7418,  0.2778,  0.5863],
        [ 0.3799,  0.5229,  0.3486],
        [ 1.0132,  1.0948,  1.1409],
        [-1.0674, -1.5197, -1.2360]])
```

- data의 값들이 너무 크면 발산을 할 수 있기 때문에 정규화를 진행하여 데이터를 0을 평균으로, 표준편차를 1로 가지게 만들자. 그러면 성능이 더 좋아진다.