7 ДӘРІС

Нейрондық желілерді оңтайландыру

Шығын функциясы градиент және параметр жаңарту

CSE8841

02

Оқыту мақсаты

1. **Параметрлер болжамды анықтайды**
2. Шығын қате шамасын өлшейді
3. Орташа шығын дерек бойынша есептеледі
4. Оңтайландыру шығынды азайтады

03

Шығын функциялары

01

MSE регрессия қатесін өлшейді

02

Cross entropy класс ықтималдығын бағалайды

03

Салмақталған шығын теңсіз класты ескереді

04

Regularization күрделілікті шектейді

Тақырып элементтері бір жүйе ретінде қарастырылады.

04

Градиенттік түсу

Градиент ең жылдам өсу бағыты

Параметр қарсы бағытта жаңартылады

Learning rate қадам өлшемін береді

Үлкен қадам тұрақсыздық туғызады

Градиенттік жаңарту

$$g_t = \nabla_\theta L(\theta_t)$$

$$\theta_{t+1} = \theta_t - \eta g_t$$

$$v_t = \beta v_{t-1} + (1-\beta) g_t$$

Learning rate η қадам өлшемін басқарады. Momentum градиенттің жинақталған бағытын қолданып тербелісті азайтады.

06

Momentum

1. **Алдыңғы бағыт жылдамдықта сақталады**
2. Тербеліс әлсірейді
3. Жазық бағытта қозғалыс жеделдейді
4. Коэффициент инерцияны басқарады

Нейрондық желі құрылымы

Кіріс қабаты белгілерді қабылдайды.
Жасырын қабаттар салмақтарды үйренеді.
Шығыс қабаты есептің жауабын береді.



08

Backpropagation

Алға өту аралық мәндерді сақтайды

Шығыннан кері сигнал басталады

Chain rule қабаттар арқылы өтеді

Градиент параметрлерге жиналады

PyTorch-та толық байланысқан желі

```
import torch.nn as nn

model = nn.Sequential(
    nn.Linear(784, 128),
    nn.ReLU(),
    nn.Linear(128, 10)
)
```

Linear қабаты салмақталған қосындыны, ReLU сызықтық емес түрлендіруді орындайды.

Код дәріс материалындағы тәсілге сүйеніп бейімделді

ҚОРЫТЫНДЫ

Негізгі тұжырымдар

1. **Оңтайландыру шығынды градиент арқылы азайтады**
2. Learning rate негізгі гиперпараметр
3. Validation дерегі жалпылауды бақылауға қажет

Өзіндік тексеру: негізгі формуланы түсіндіріп, әдістің бір практикалық қолданылуын атаңыз.