



## صورت جلسات گروه بهینه‌سازی دکتر خونساری

جلسه

تاریخ: ۱۴۰۲/۴/۱۷

خلاصه جلسه:

در این جلسه خانم فریدی مسئله‌ای را ارائه کردند و در نهایت دکتر خونساری نکاتی را بیان کردند.

### ارائه خانم فریدی:

- سوال ۶.۶ از additional Exercises بوید
- Maximum likelihood estimation of an increasing nonnegative signal
- توضیح مسئله
  - سیگنال ورودی و ویژگیهای آن
  - کاربرد مسئله
  - توضیح likelihood function
  - توضیح objective و شرایط برای دو حالت مسئله
  - توضیح کد و نمودار

### نکات دکتر خونساری:

- وقتی نسبت به دیتا یک اطلاعات پیشینه داریم به ما کمک میکند که بهینه‌سازی بهتری داشته باشیم. وقتی اطلاعاتی نداریم دچار overshoot و undershoot متعدد می‌شویم.
- در بهینه‌سازی constrained در واقع در یک لایه از فضای کلی جستجو را می‌گردیم. و باعث میشود که حدس‌های عاقلانه تری داشته باشیم. همچنین نکته دوم این است که در دیدگاه compressive sensing میتوانیم از نظر محاسباتی نیز saving داشته باشیم.
- در قسمت penalty function, regularization, تابع هدف دارای ترم ارور و یک ترم نورم یا پنالتی است. که به تابع objective اضافه می‌کنیم با یک ضریب لاندا. ترم پنالتی اضافه شده به ما میگوید که تابع هدف قرار است به شکل خاصی حرکت کند. این شکل خاص با آن norm بدست می‌آید.
- سلول‌های ژوپیترونوت بوک کوچک باشد که فهم و توضیح آن آسان تر باشد.

- توضیح  $\text{penalty function}$ : داده های یادگیری ماشین دارای  $\text{feature}$  هایی هستند. در سیستم های جدید یادگیری ماشین تعداد فیچرها بسیار زیاد هستند. همچنین تعداد داکيومنت ها هم بسیار زیاد است. در نتیجه اطلاعات موجود حجم زیادی دارند و انجام یک بهینه سازی روی داده ها و یادگیری یک مدل و کمینه کردن ارور به محاسبات زیادی نیاز دارد. اگر در یادگیری ماشین نسبت تعداد داده ها ( $m$ ) به میزان ابعاد آن ها ( $d$ ) زیاد باشد میتوان از روش های ساده بهینه سازی برای حل مسئله استفاده کرد ولی وقتی تعداد داده کم است و مثلاً وقتی  $m \sim d$ ، ممکن است  $\text{overfit}$  رخ دهد. یک روش برای جلوگیری از  $\text{overfit}$  اضافه کردن داده (مثلاً با  $\text{bootstrap}$ ) و روش دیگر اضافه کردن  $\text{constraint}$  است. یکی از روش های اضافه کردن  $\text{constraint}$  به مسئله، اضافه کردن  $\text{regularization term}$  یا همان  $\text{penalty}$  به سیگما است. اگر بتوان حدس زد که توزیع داده  $\text{prior}$  چیست، توزیعی را که از همه بهتر است در اینجا می گذاریم.
- در یادگیری ماشین مسئله ای به نام  $\text{expressibility}$  وجود دارد.  $\text{Expressibility}$  این است که یک سری داده داریم و فیچرهای کمی مثلاً ۲ تا داریم. سپس مسئله را به مدل می دهیم تا حل شود تا پارامترها بدست بیاید. مدل یادگیری ماشین بدست آمده  $\text{expressibility}$  پایینی دارد و باید تعداد فیچرها را افزایش دهیم. برای این کار می توانیم فیچرها را در هم ضرب کنیم. یک روش دیگر هم  $\text{projection}$  نام دارد که در آن خود را از بعد کم به بعد بالاتر تصویر میکنیم و بعد از تصویر، تابع را در بعد بالاتر می نویسیم.
- گوگل کولب مرتب شود و دارای ساختار باشد. هر جلسه با یک نام شروع شود و اطلاعات آن جلسه مانند شماره سوال، کد و توضیحات کامل آورده شود.