

Универзитет у Београду
Математички факултет



Михајло Србакоски

Примена Гаусових процеса у статистичком учењу

мастер рад

Београд, 2022.

Ментор:

др Бојана МИЛОШЕВИЋ, ванредни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

др Младен НИКОЛИЋ, ванредни професор
Универзитет у Београду, Математички факултет

др Милан ЈОВАНОВИЋ, доцент
Универзитет у Београду, Математички факултет

Датум одбране: 26. септембар 2022. године

Захваљујем се професорки Бојани Милошевић на помоћи, подршци и стрпљењу током писања овог рада.

Захвалио бих се и осталим члановима комисије, професору Младену Николићу и професору Милану Јовановићу, на стрпљењу приликом читања рада и свим корисним сугестијама.

Захвалност дугујем и свом дугогодишњем другу Вуку на помоћи око избора теме и подршци коју ми даје.

Посебну захвалност дугујем својој најбољој другарици Аји за подршку током читавих студија и безброј заједно спремљених испита.

Садржај

1	Увод	1
1.1	Гаусов процес	2
1.2	Својства нормалне расподеле	3
2	Коваријациони кернели	4
2.1	Комбинација кернела	7
2.2	Репрезентација кернела	10
2.3	Дубоки кернели	11
3	Гаусова регресија	12
3.1	Оптимизација	14
3.2	Предвиђање	14
3.3	Предности Гаусових процеса	15
3.4	Ограничења Гаусових процеса	16
4	Проређени Гаусови процеси	18
5	Потпуно повезане неуронске мреже	23
5.1	Архитектура мреже	24
5.2	Оптимизација	25
6	Дубоки Гаусови процеси	27
6.1	Мотивација	27
6.2	Дефиниција дубоког Гаусовог процеса	28
6.3	Оптимизација	30
7	Конволутивне неуронске мреже	31
7.1	Конволутивни слој	32
7.2	Агрегациони слој	33

8	Конволутивни Гаусови процеси	35
8.1	Конволутивни кернел	35
8.2	Дубоки Гаусови процеси са конволутивним кернелима	36
9	Преглед библиотека <i>GPflow</i> и <i>GPflux</i>	37
9.1	<i>GPflow</i>	37
9.2	<i>GPflux</i>	41
10	Примена Гаусових процеса на реалне податке	44
10.1	Предвиђање цена некретнина	44
10.2	Предвиђање цена авионских карата	46
11	Закључак	48
	Библиографија	50

Глава 1

Увод

Гаусови процеси су један од најдоминантнијих приступа у непараметарском бајесовском учењу. Имају широку примену у регресионим задацима, а у последње време проналазе све већу примену и у задацима класификације. Међутим, постоји неколико проблема који у великој мери ограничавају њихову употребу, а највећи проблем је рачунска захтевност. У овом раду ће бити приказано на који начин се Гаусови процеси користе у задацима надгледаног учења, при чему ћемо их упоредити са другим методама машинског учења у регресионим задацима. Биће приказани и дубоки Гаусови процеси који су у последњој деценији постали врло значајна област истраживања.

Предност Гаусових процеса, а самим тим и дубоких Гаусових процеса, је та што је приступ бајесовски, па приликом предвиђања добијамо расподелу циљне променљиве, а не само вредност. Већина метода дубоког учења су фреквентистичког типа, односно за вредност циљне променљиве добијамо само број, без информације о поузданости предикције, али је и код њих на неки начин могуће конструисати интервале предвиђања. Познавање поузданости предикције може бити од великог значаја у бројним ситуацијама, поготово у трговини акцијама на берзи. На пример, можемо предвидети да ће сутрашња цена неке акције на берзи имати нормалну расподелу са математичким очекивањем 100\$ и стандардном девијацијом 20\$. Стандардна девијација нам говори колико је наше предвиђање поуздано.

Још једна предност Гаусових процеса у односу на већину осталих приступа је могућност инкорпорирања априорних знања. Наиме, када правимо неки

модел, можемо консултовати доменске експерте и њихово знање искористити за добијање бољег модела.

Сада ћемо формално увести основне појмове које ћемо користити у овом раду.

1.1 Гаусов процес

Дефиниција 1.1.1 *Фамилија случајних величина $\{X_t | t \in T\}$ дефинисаних на неком простору вероватноће $(\Omega, \mathcal{A}, P)$, где је T неки бесконачан скуп, зове се случајни процес. Ако је $(\mathcal{R}, \mathcal{B})$ простор вредности за све $X_t, t \in T$, онда се $\{X_t | t \in T\}$ зове реалан случајни процес.*

Дефиниција 1.1.2 *Средња вредност (математичко очекивање) случајног процеса $\{X_t | t \in T\}$ је функција $m : T \rightarrow \mathcal{R}$ дефинисана са:*

$$m(t) = EX_t.$$

Дефиниција 1.1.3 *Коваријациона функција реалног случајног процеса $\{X_t | t \in T\}$ је функција $K : T \times T \rightarrow \mathcal{R}$, дефинисана са:*

$$\begin{aligned} K(t, s) &= \text{cov}(X_t, X_s) = E(X_t - EX_t)(X_s - EX_s) = \\ &= EX_t X_s - EX_t EX_s. \end{aligned}$$

Дефиниција 1.1.4 *Реалан случајни процес $\{X_t | t \in T\}$ назива се Гаусов процес ако сваки његов коначно-димензиони засек има вишедимензиону нормалну расподелу.*

Нека је $\{X_t | t \in T\}$ Гаусов процес. Посматрајмо произвољну коначну колекцију случајних величина $X_{i_1}, X_{i_2}, \dots, X_{i_d}$ која припада том Гаусовом процесу. Тада важи:

$$(X_{i_1}, X_{i_2}, \dots, X_{i_d}) \sim \mathcal{N}(\mu, \Sigma),$$

где је μ вектор математичких очекивања, а Σ коваријациона матрица.

Густина случајног вектора $(X_{i_1}, X_{i_2}, \dots, X_{i_d})$ дата је са:

$$p(\mathbf{x}) = \frac{1}{\sqrt{|\Sigma|(2\pi)^d}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right),$$

где је $\mathbf{x} = (x_{i_1}, x_{i_2}, \dots, x_{i_d})^T \in \mathcal{R}^d$, а $|\cdot|$ је ознака за детерминанту.

1.2 Својства нормалне расподеле

Вишедимензиона нормална расподела је затворена у односу на условљавање и маргинализацију. То значи да је резултујућа расподела, при тим операцијама, такође нормална, што многе проблеме у статистици и статистичком учењу чини израчунљивим. То је формално записано у следећој теореми.

Теорема 1.2.1 *Нека су X и Y случајни вектори и нека (X, Y) има вишедимензиону нормалну расподелу, односно*

$$(X, Y) \sim \mathcal{N}(\mu, \Sigma) = \mathcal{N}\left(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix}\right).$$

Тада за маргиналне расподеле вектора X и Y важи:

$$X \sim \mathcal{N}(\mu_X, \Sigma_{XX}),$$

$$Y \sim \mathcal{N}(\mu_Y, \Sigma_{YY}).$$

За условне расподеле ће важити:

$$X|Y = y \sim \mathcal{N}(\mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(y - \mu_Y), \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}),$$

$$Y|X = x \sim \mathcal{N}(\mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \mu_X), \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}).$$

Можемо приметити да вектор очекивања зависи од условне случајне величине, а да коваријациона матрица не зависи. Навешћемо још две теореме које ће нам бити од користи.

Теорема 1.2.2 *Нека су $X_1, X_2, \dots, X_n$ независне случајне величине, n произвољан природан број и нека важи $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, за свако $i, 1 \leq i \leq n$.*

Тада важи:

$$X_1 + X_2 + \dots + X_n \sim \mathcal{N}(\mu_1 + \mu_2 + \dots + \mu_n, \sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2).$$

Теорема 1.2.3 *Ако случајни вектор X има нормалну расподелу $\mathcal{N}(\mu, \Sigma)$, онда важи $AX + b \sim \mathcal{N}(A\mu + b, A\Sigma A^T)$, где је A одговарајућа квадратна матрица, а b одговарајући вектор.*

Глава 2

Коваријациони кернели

Вишедимензиона нормална расподела је у потпуности одређена вектором очекивања и коваријационом матрицом, па је Гаусов процес у потпуности одређен функцијом средње вредности и коваријационом функцијом. Поставља се питање како на основу матрице података израчунати коваријациону функцију. Одговор нам дају коваријациони кернели, односно оцену коваријационе функције добијамо израчунавањем функције сличности која се назива коваријациони кернел. Дакле, у идеалном случају, коваријациони кернел је функција k таква да важи:

$$k(\mathbf{x}_1, \mathbf{x}_2) = Cov(f(\mathbf{x}_1), f(\mathbf{x}_2)),$$

где су $\mathbf{x}_1$ и $\mathbf{x}_2$ вектори атрибута у неким временским тренуцима, а f је Гаусов процес.

Што је вредност коваријационог кернела за неке две инстанце већа, то се оне могу сматрати сличнијим. Што је мања, то се те две инстанце могу сматрати различитијим. Коваријациони кернели најчешће имају одређене параметре, којима се финије подешава њихово понашање.

Дефиниција 2.0.1 Коваријациони (или Мерцеров¹) кернел (у даљем тексту само кернел) је пресликавање $k : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}$ такво да важи симетричност и позитивна дефинитност, односно:

- $k(\mathbf{x}_1, \mathbf{x}_2) = k(\mathbf{x}_2, \mathbf{x}_1), \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{R}^n$,
- $\forall m \in \mathbf{N}, \forall \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m \in \mathcal{R}^n$ матрица димензије $m \times m$, са елементима $k(\mathbf{x}_i, \mathbf{x}_j)$ на позицији (i, j) , је позитивно дефинитна.

¹James Mercer(1883-1932) - енглески математичар

Један битан тип кернела који дефинишемо је стационарни кернел.

Дефиниција 2.0.2 Кернел $k : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}$ је стационаран ако је,
 $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{R}^n$, $k(\mathbf{x}_1, \mathbf{x}_2)$ функција која зависи само од $\mathbf{x}_1 - \mathbf{x}_2$.

Најчешће коришћени кернели су:

- константни (енг. *constant*):

$$k_C(\mathbf{x}_1, \mathbf{x}_2) = c,$$

- квадратно-експоненцијални (енг. *squared-exponential*):

$$k_{SE}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2l^2}\right),$$

- периодични (енг. *periodic*):

$$k_{PER}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \exp\left(-\frac{2}{l^2} \sin^2\left(\pi \frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{p}\right)\right),$$

- линеарни (енг. *linear*):

$$k_{LIN}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 (\mathbf{x}_1 - c) \cdot (\mathbf{x}_2 - c),$$

- различите варијанте Матерн² кернела:

$$k_{M12}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{l}\right),$$

$$k_{M32}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \left(1 + \frac{\sqrt{3}\|\mathbf{x}_1 - \mathbf{x}_2\|}{l}\right) \exp\left(-\frac{\sqrt{3}\|\mathbf{x}_1 - \mathbf{x}_2\|}{l}\right),$$

$$k_{M52}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \left(1 + \frac{\sqrt{5}\|\mathbf{x}_1 - \mathbf{x}_2\|}{l} + \frac{5\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}\|\mathbf{x}_1 - \mathbf{x}_2\|}{l}\right),$$

- рационално-квадратни (енг. *rational-quadratic*):

$$k_{RQ}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \left(1 + \frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\alpha l^2}\right)^{-\alpha},$$

- косинусни (енг. *cosine*):

$$k_{COS}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \cos\left(\frac{2\pi\|\mathbf{x}_1 - \mathbf{x}_2\|}{p}\right),$$

²Bertil Matérn(1917-2007) - шведски статистичар

- бели шум (енг. *white noise*):

$$k_{WN}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \delta(\mathbf{x}_1, \mathbf{x}_2).$$

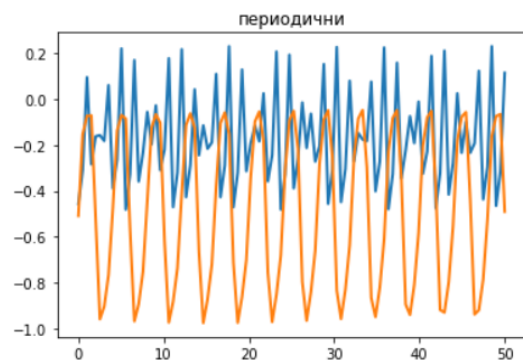
где су $\mathbf{x}_1$ и $\mathbf{x}_2$ вектори, $\|\cdot\|$ је ознака за норму, $\cdot$ је ознака за скаларни производ, а $\delta(\cdot, \cdot)$ је Кронекедова делта функција.

Већина горенаведених кернела су стационарни. Такви кернели су инваријантни у односу на транслацију, а та претпоставка врло често није испуњена, па из тог разлога желимо да имамо и нестационарне кернеле. Од наведених кернела само линеарни кернел и бели шум нису стационарни. Такође, примећујемо да кернели имају параметре. Они се не дефинишу директно већ представљају параметре модела.

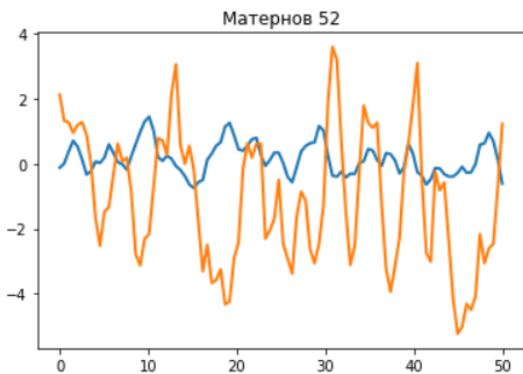
На сликама 2.1, 2.2, 2.3 и 2.4 су приказане по две реализације Гаусових процеса са квадратно-експоненцијалним, периодичним, Матерн 52 и линеарним кернелом.



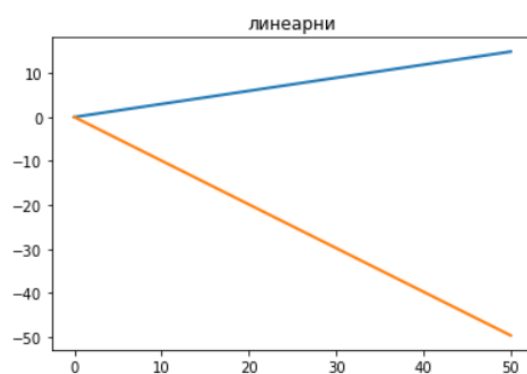
Слика 2.1



Слика 2.2



Слика 2.3



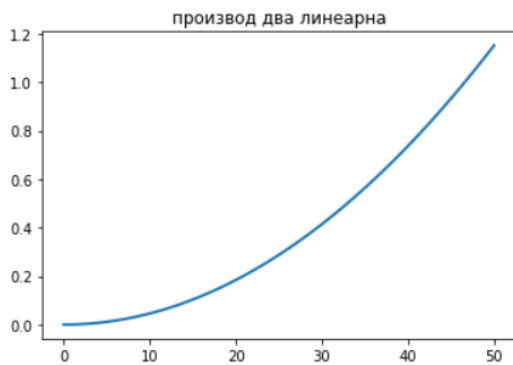
Слика 2.4

2.1 Комбинација кернела

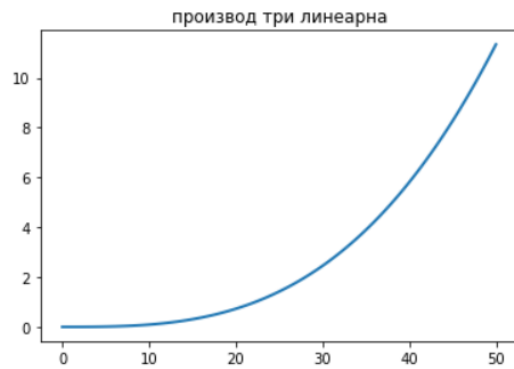
Нестационарне кернеле можемо добити сабирањем и множењем основних кернела. Сабирање и множење је дозвољено јер се при тим операцијама добија кернел који је такође позитивно семидефинитан.

Примери множења кернела:

- Множењем t линеарних кернела добијамо кернел степена t . На слици 2.5 је приказана реализација Гаусовог процеса са квадратним кернелом, а на слици 2.6 реализација Гаусовог процеса са кернелом степена 3.



Слика 2.5



Слика 2.6

- Множењем неког кернела квадратно-експоненцијалним кернелом се глобална структура квадратно-експоненцијалног кернела преноси на локални ниво. На слици 2.7 је приказана реализација Гаусовог процеса са кернелом који је једнак производу квадратно-експоненцијалног и периодичног кернела.
- Множењем неког кернела линеарним кернелом добија се Гаусов процес чија стандардна девијација маргиналних расподела расте линеарно почевши од тачке s дате у линеарном кернелу. На слици 2.8 је приказана реализација Гаусовог процеса са кернелом који је једнак производу линеарног и квадратно-експоненцијалног кернела.



Слика 2.7

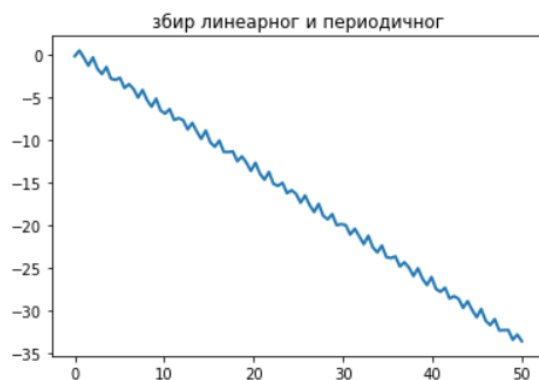


Слика 2.8

Сабирање кернела користимо кад хоћемо да моделујемо функцију за коју сматрамо да се може представити као збир независних Гаусових процеса са различитим кернелима. Заиста, то можемо радити јер је збир два независна Гаусова процеса такође Гаусов процес (то следи из својства нормалне расподеле које је дефинисано у теорему 1.2.2).

Примери сабирања кернела:

- Сабирањем линеарног и периодичног кернела добијамо периодични Гаусов процес са линеарним трендом. То је приказано на слици 2.9.
- Сабирањем квадратно-експоненцијалног и периодичног кернела добија се периодичан Гаусов процес, са присутним шумом. То је приказано на слици 2.10.



Слика 2.9



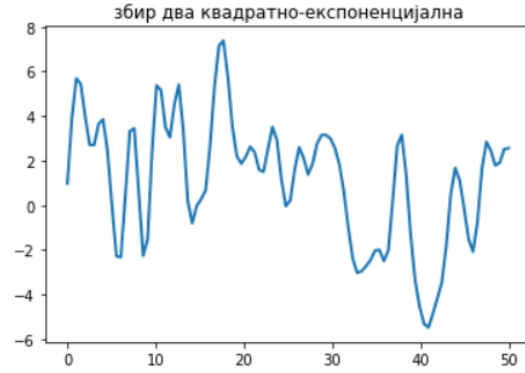
Слика 2.10

- Сабирањем квадратно-експоненцијалног и линеарног кернела добијамо Гаусов процес који има линеарни тренд, са присутним шумом, као што је приказано на слици 2.11.

- Сабирањем два квадратно-експоненцијална кернела различитих ширина добијамо Гаусов процес са брзим и спорим варијацијама. То је приказано на слици 2.12.



Слика 2.11



Слика 2.12

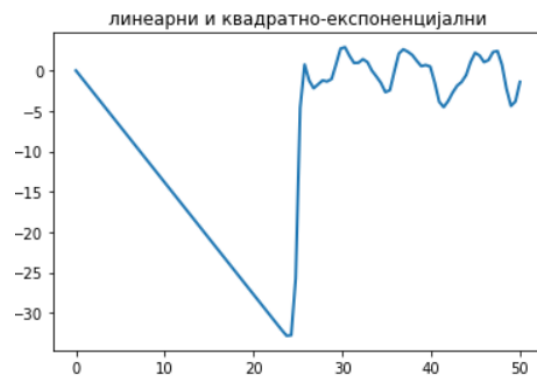
Навешћемо још један начин за добијање нових кернела. Ако желимо да добијемо Гаусов процес који има другачију структуру у различитим деловима, можемо користити сигмоидну функцију $\sigma(x) = \frac{1}{1+e^{-\frac{(x-c)}{d}}}$, где је c параметар локације, а d параметар ширине. Нека су k_1 и k_2 произвољни кернели. Нови кернел (енг. *changepoint kernel*), у ознаци k_{CP} , дефинишемо са:

$$k_{CP}(k_1, k_2)(\mathbf{x}_1, \mathbf{x}_2) = \sigma(\mathbf{x}_1)k_1(\mathbf{x}_1, \mathbf{x}_2)\sigma(\mathbf{x}_2) + (1 - \sigma(\mathbf{x}_1))k_2(\mathbf{x}_1, \mathbf{x}_2)(1 - \sigma(\mathbf{x}_2)).$$

Овако дефинисан кернел ће у првом делу имати структуру кернела k_1 , а у другом делу структуру кернела k_2 . Тачка прелаза са једног кернела на други одређена је параметром локације сигмоидне функције. На слици 2.13 је приказана реализација Гаусовог процеса са квадратно-експоненцијалним кернелом, при чему се на средини интервала мењају параметри кернела, док је на слици 2.14 приказана реализација Гаусовог процеса чији кернел на средини интервала прелази са линеарног на квадратно-експоненцијални.

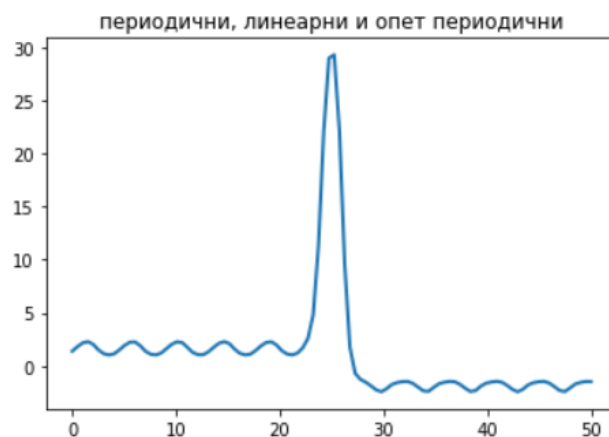


Слика 2.13



Слика 2.14

Индуктивно можемо добити Гаусов процес који мења структуру у произвољно много тачака. На слици 2.15 је приказана реализација Гаусовог процеса који на почетку има периодични кернел, затим линеарни, а на крају опет периодични кернел.



Слика 2.15

2.2 Репрезентација кернела

Као што смо рекли, желимо да од основних кернела добијемо компликованије кернеле. Још један од начина да то урадимо је компоновање кернела. Пре него што дефинишемо композицију кернела, наводимо једну врло значајну теорему из функционалне анализе, коју је доказао Џејмс Мерцер 1909. године.

Теорема 2.2.1 Нека је $k : (a, b)^n \times (a, b)^n \rightarrow \mathcal{R}$ Мерцеров кернел. Нека је оператор $T : L^2(a, b)^n \rightarrow L^2(a, b)^n$ дефинисан са:

$$Tf(\mathbf{x}_1) = \int_{[a,b]^n} k(\mathbf{x}_1, \mathbf{x}_2) f(\mathbf{x}_2) d\mathbf{x}_2,$$

где је $\mathbf{x}_2 \in (a, b)^n$.

Тада за јединичне сојсџвене функције e_j , које одговарају сојсџвеним вредносџима λ_j оператора T , важи

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{\infty} \lambda_j e_j(\mathbf{x}_1) e_j(\mathbf{x}_2),$$

при чему овај ред ајсолуџно и равномерно конверџира.

На основу теореме 2.2.1, сваки Мерцеров кернел се може представити као скаларни производ, односно за кернел k ће постојати пресликавање h из $\mathbf{R}^n$ у неки векторски простор са скаларним производом, тако да важи:

$$k(\mathbf{x}_1, \mathbf{x}_2) = h(\mathbf{x}_1)^T \cdot h(\mathbf{x}_2).$$

Ова репрезентација кернела важи и у много општијем облику. Наиме, скуп $\mathbf{R}^n$ можемо заменити произвољним непразним скупом X . На тај начин, за произвољне скупове, над којима можемо дефинисати кернеле, можемо имати значајан део техничких погодности које пружа скаларни производ.

2.3 Дубоки кернели

Компоновањем кернела добијамо нови кернел који ћемо звати дубоки кернел (енг. *deep kernel*). Посматрајмо кернеле k_a и k_b . На основу теореме 2.2.1. имају следеће репрезентације:

$$k_a(\mathbf{x}_1, \mathbf{x}_2) = h_a(\mathbf{x}_1)^T \cdot h_a(\mathbf{x}_2)$$

и

$$k_b(\mathbf{x}_1, \mathbf{x}_2) = h_b(\mathbf{x}_1)^T \cdot h_b(\mathbf{x}_2).$$

Њихову композицију дефинишемо са:

$$(k_b \circ k_a)(\mathbf{x}_1, \mathbf{x}_2) = k_b(h_a(\mathbf{x}_1), h_a(\mathbf{x}_2)) = [h_b(h_a(\mathbf{x}_1))]^T \cdot h_b(h_a(\mathbf{x}_2)).$$

Глава 3

Гаусова регресија

Једна од главних примена Гаусових процеса у статистичком учењу је управо решавање регресионих проблема. На пример, можемо се бавити предвиђањем количине тешких метала у води на основу удаљености од загађивача, врсте воденог тока, врсте земљишног загађивача и слично.

Гаусова регресија је заправо кернелизована Бајесова линеарна регресија. Разлика између кернелизованог и обичног приступа је у томе што кернелизовани приступ води локалности модела. Наиме, код линеарне регресије се зависност циљне променљиве од атрибута изражава помоћу коефицијената који контролишу дејство промене тог атрибута на промену циљне променљиве. Велике вредности атрибута воде већим вредностима циљне променљиве, а мале мањим, или супротно. Насупрот томе, приступ заснован на коваријационим кернелима се заснива на блискости вредности атрибута. Уколико су вредности атрибута нове инстанце блиске вредностима атрибута неке инстанце из скупа за обучавање, тада ће и вредност циљне променљиве нове инстанце бити блиска вредности циљне променљиве те инстанце из скупа за обучавање, без обзира да ли су вредности атрибута нове инстанце саме за себе високе или ниске. Дакле, код Гаусове регресије зависности атрибута могу бити различите у различитим деловима простора предиктора, а вредност циљне променљиве нове инстанце ће бити одређена на основу законитости које важе у њеној околини.

Сада ћемо мало детаљније објаснити Гаусову регресију. Претпоставимо да имамо на располагању d предиктора, тренинг скуп величине n и тест скуп величине n' . Задатак је да помоћу вредности на тренинг скупу пронађемо Гаусов процес који најбоље описује вредности на тест скупу. Тај Гаусов процес

ћемо означити са f .

Уведимо следеће ознаке:

- $m(\cdot)$ - функција средње вредности Гаусовог процеса,
- $k(\cdot, \cdot)$ - коваријациони кернел, односно оцена коваријационе функције Гаусовог процеса,
- X - матрица предиктора на тренинг скупу (матрица димензије $n \times d$),
- X' - матрица предиктора на тест скупу (матрица димензије $n' \times d$),
- y - вектор зависне променљиве на тренинг скупу,
- y' - вектор зависне променљиве на тест скупу,
- $f(X)$ - вектор реализације Гаусовог процеса на тренинг скупу,
- $f(X')$ - вектор реализације Гаусовог процеса на тест скупу.

Циљ је пронаћи Гаусов процес f који најбоље описује податке. Наш модел је облика:

$$y = f(X) + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2 I),$$

где је σ^2 дисперзија белог шума и то је такође параметар нашег модела. Расподела вектора $f(X)|X$ је нормална $\mathcal{N}(m(X), k(X, X))$, па ће, због независности вектора $f(X)$ и ε , расподела вектора $y|X$ бити нормална $\mathcal{N}(m(X), k(X, X) + \sigma^2 I)$. Како и $f(X')|X'$ има нормалну расподелу

$$f(X')|X' \sim \mathcal{N}(m(X'), k(X', X')),$$

то ће важити:

$$\begin{bmatrix} y|X \\ f(X')|X' \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(X) \\ m(X') \end{bmatrix}, \begin{bmatrix} k(X, X) + \sigma^2 I & k(X, X') \\ k(X', X) & k(X', X') \end{bmatrix}\right).$$

На основу теореме 1.2.1 важи:

$$f(X')|X', X, y \sim \mathcal{N}(E(f(X')|X', X, y), Cov(f(X')|X', X, y)),$$

где је

$$E(f(X')|X', X, y) = k(X', X)(k(X, X) + \sigma^2 I)^{-1}y$$

и

$$\text{Cov}(f(X')|X', X, y) = k(X', X') - k(X', X)(k(X, X) + \sigma^2 I)^{-1}k(X, X').$$

Једини параметри Гаусове регресије су параметри коваријационог кернела и дисперзија белог шума. Након фиксирања кернела, који представља хиперпараметар модела, треба наћи оне вредности параметара који максимизују функцију веродостојности, односно логаритам те функције. То се назива оптимизација модела.

3.1 Оптимизација

Како је расподела циљне променљиве $y|X$ нормална $\mathcal{N}(m(X), k(X, X) + \sigma^2 I)$, функција веродостојности, у ознаци L , је дефинисана са:

$$L(y|X, \Theta) = \frac{1}{\sqrt{|k(X, X) + \sigma^2 I|(2\pi)^n}} \exp\left(-\frac{y^T(k(X, X) + \sigma^2 I)^{-1}y}{2}\right),$$

где је са Θ означен скуп свих параметара које треба оценити.

Сада је логаритам функције веродостојности, у ознаци l , дефинисан са:

$$l(y|X, \Theta) = -\frac{1}{2}y^T(k(X, X) + \sigma^2 I)^{-1}y - \frac{1}{2}\ln(|k(X, X) + \sigma^2 I|) - \frac{n}{2}\ln(2\pi).$$

Ако је ова функција диференцијабилна, можемо је максимизовати минимизовањем њене негативне вредности на пример методом градијентног спуста (енг. *gradient descent*), која је описана у глави 5. Због овог корака је пожељно изабрати диференцијабилни кернел.

Кључни проблем у раду са Гаусовим процесима је управо оптимизација и решавање проблема на које се наилази ћемо детаљно описати у глави 4. Сада ћемо објаснити корак који долази након оптимизације, а то је предвиђање помоћу нашег модела.

3.2 Предвиђање

Предикције нису само вредности, већ имају расподелу. Та расподела је условна расподела случајног вектора $f(X')|X', X, y$. На основу теореме 1.2.1 важи:

$$f(X')|X', X, y \sim \mathcal{N}(E(f(X')|X', X, y), \text{Cov}(f(X')|X', X, y))$$

где је

$$E(f(X')|X', X, y) = k(X', X)(k(X, X) + \sigma^2 I)^{-1}y$$

и

$$Cov(f(X')|X', X, y) = k(X', X') - k(X', X)(k(X, X) + \sigma^2 I)^{-1}k(X, X').$$

Вектор очекивања и коваријациону матрицу можемо израчунати јер смо претходно оценили параметре кернела и дисперзију белог шума.

3.3 Предности Гаусових процеса

Наводимо неколико разлога зашто су Гаусови процеси погодни за решавање проблема надгледаног учења.

- **Аналитичко закључивање.** Користећи функцију кернела и инстанце из тренинг скупа, предиктивна апостериорна расподела се може израчунати у затвореном облику. Ово је ретко својство за непараметарске моделе.
- **Изражајност.** Избором различитих кернела можемо добити широку класу модела.
- **Мали број параметара.** За разлику од неуронских мрежа, које могу имати и неколико милиона параметара, методи засновани на Гаусовим процесима их имају само неколико. Због тога врло тешко долази до преприлагођавања, па регуларизација, која врло често уме да буде компликована, није потребна.
- **Погодни су за статистичку анализу.** Може изгледати лоше да се ограничимо на неку ограничену класу функција, уместо да вршимо закључивање у скупу свих израчунљивих функција. Међутим, једноставни модели имају своје предности. Попут линеарних модела, који чине изузетно ограничену класу модела, једноставни су, лако се анализирају и лако се инкорпорирају у друге моделе. Гаусови процеси се могу посматрати као проширење линеарних модела који задржавају ова корисна својства.

- **Лако се компонују.** Као што смо већ рекли, Гаусови процеси се лако инкорпорирају у друге моделе, а самим тим могу и међусобно да се компонују, по узору на неуронске мреже. Међусобним компоновањем Гаусових процеса добијамо дубоке Гаусове процесе са којима ћемо се упознати у глави 6.

3.4 Ограничења Гаусових процеса

Наводимо и неколико проблема који ограничавају употребу Гаусових процеса.

- **Временска и просторна сложеност.** Ово је највеће ограничење Гаусових процеса и разлог зашто и даље немају довољно широку примену. Наиме, да бисмо добили апостериорну расподелу циљне променљиве, а самим тим и предикције нашег модела, неопходно је израчунати инверз матрице $k(X, X) + \sigma^2 I$. Временска сложеност те операције је $O(n^3)$, где је n број инстанци у тренинг скупу, а то може бити број реда величине неколико десетина хиљада или чак милиона. О решавању овог проблема ће бити више речи у наредној глави. Након што једном израчунамо инверз те матрице, временска сложеност рачунања предикције је $O(n^2)$. Такође, како је за прављење предикција потребна матрица димензије $n \times n$, просторна сложеност је $O(n^2)$. Због тога је употреба Гаусове регресије ограничена само на скупове од неколико хиљада инстанци. Решење за овај проблем су проређени Гаусови процеси. Међутим, тај проблем је врло сложен и и даље не постоји довољно добро решење. Наиме, апроксимација која се врши некада не даје очекиване резултате, па се добијају лоши модели.
- **Избор кернела.** Неопходно је изабрати одговарајући кернел, након чега се његови параметри добијају максимизовањем функције веродостојности. С обзиром да постоји велики број кернела, некад може бити тешко наћи најбољи. Такође, за неке податке и не постоји одговарајући кернел који може довољно добро да уочи законитости између атрибута и циљне променљиве. Овај проблем се решава дубоким Гаусовим процесима, који нам дају ту врсту флексибилности.

Међутим, код дубоких Гаусових процеса се додатно продубљује проблем временске и просторне сложености.

Глава 4

Проређени Гаусови процеси

Као што смо видели, модели засновани на Гаусовим процесима имају две велике предности. Прва је што су непараметарски, а друга је што је приступ бајесовски, уместо тачкастих оцена тражених вредности, добијамо њихову расподелу, а можемо и инкорпорирати априорна знања у модел. Један од највећих проблема овог приступа је рачунска сложеност и у овом делу ћемо се детаљније упознати са тим проблемом и изложити неке начине за његово превазилажење.

Проблем на који се наилази при раду са Гаусовим процесима је рачунање инверза коваријационе матрице. Временска сложеност те операције је $O(n^3)$, где је n број инстанци у скупу података. Како скуп података може бити изузетно велики, можемо доћи у ситуацију да рачунање инверза траје превише дуго, па је циљ извршити неку апроксимацију и на тај начин смањити рачунску сложеност.

Најједноставнији приступ је користити подскуп X_1 скупа података X . Постоји више начина за бирање подскупа X_1 . Циљ је да X_1 садржи што различитије опсервације из X , јер хоћемо да у малом скупу података имамо што више информација из почетног скупа. На овај начин смањујемо рачунску сложеност на $O(m^3)$, где је m број опсервација у скупу X_1 . Постоје разне технике за избор скупа X_1 , али овај приступ није превише ефикасан, па их нећемо описивати.

Други приступ се назива проређени кернел и огледа се у томе да поставимо на нулу оне вредности коваријационе матрице које су мање од неке унапред постављене границе. На тај начин ћемо добити коваријациону матрицу која ће имати велики број нула. Рачунска сложеност израчунавања инверза те

матрице биће $O(\alpha n^3)$, где је $0 < \alpha < 1$.

Трећи приступ који ћемо описати је уједно и тренутно најкоришћенији. Циљ је апроксимирати инверз коваријационе матрице. Овим приступом смањујемо рачунску сложеност на $O(nm^2)$, где је m много мање од n . Процес добијен овим поступком назива се проређени Гаусов процес (енг. *Sparse Gaussian Process*) и сада ћемо детаљно објаснити цео поступак. Овај приступ подразумева проширивање скупа података са додатних m тачака.

Означимо са X_u матрицу вредности атрибута тих додатних m тачака и са u вектор вредности њихових зависних променљивих. Претпоставићемо да вредности вектора u такође припадају Гаусовом процесу f . Циљ је пронаћи што боље X_u .

Да се подсетимо, наш модел је облика:

$$y = f(X) + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2 I).$$

У овом делу ћемо векторе $f(X)$ и $f(X')$ означавати само са f и f' , а густину расподеле ћемо означавати са p . Посматрамо векторе f , y , f' , u и матрице X , X' и X_u .

Прво ћемо навести једну дефиницију коју ћемо користити.

Дефиниција 4.0.1 Нека су X и Y апсолутно непрекидне случајне величине са густинама p и q . Кулбак³-Лајблерова⁴ дивергенција, у ознаци $KL(p||q)$, дефинисана је са:

$$KL(p||q) = \int p(x) \ln \frac{p(x)}{q(x)} dx.$$

Кулбак-Лајблерова дивергенција је мера једнакости две расподеле. Што је мања, то су две расподеле сличније. Може се показати да је ненегативна. Приметимо да важи:

$$KL(p||p) = \int p(x) \ln \frac{p(x)}{p(x)} dx = \int p(x) \ln 1 dx = 0.$$

У глави 3 смо навели да важи:

$$f'|X', X, y \sim \mathcal{N}(k(X', X)(k(X, X) + \sigma^2 I)^{-1} y, \\ k(X', X') - k(X', X)(k(X, X) + \sigma^2 I)^{-1} k(X, X')).$$

³Solomon Kullback(1907-1994) - амерички математичар

⁴Richard Leibler(1914-2003) - амерички математичар

Да бисмо олакшали запис, нећемо писати X , X' и X_u у условним случајним векторима, него ћемо то имплицитно подразумевати.

На основу формуле потпуне вероватноће важи:

$$p_{f'|y}(x) = \int p_{f'|u=v}(x)p_{u|y}(v)dv.$$

Да бисмо израчунали густину $p_{f'|y}$, а након тога и вредности f' , неопходно је да знамо расподелу случајног вектора $u|y$. Случајни вектор $u|y$ има нормалну расподелу. Међутим, матрица X_u је непозната, па не можемо да израчунамо параметре те расподеле. Зато ћемо је апроксимирати расподелом чију ћемо густину означити са ϕ . Дакле ϕ је густина нормалне $\mathcal{N}(\mu_u, A_u)$ расподеле, где су μ_u и A_u оцене стварних параметара расподеле вектора $u|y$. Циљ је наћи што боље оцене параметара расподеле.

Нека је:

$$q_{f'}(x) = \int p_{f'|u=v}(x)\phi(v)dv.$$

Дакле, $q_{f'}$ је апроксимација густине $p_{f'|y}$.

Да бисмо нашли $p_{f'|y}$ неопходно је да знамо и $p_{f'|u}$. Заједничка расподела вектора u и f' је дата са:

$$\begin{bmatrix} u \\ f' \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(X_u) \\ m(X') \end{bmatrix}, \begin{bmatrix} k(X_u, X_u) & k(X_u, X') \\ k(X', X_u) & k(X', X') \end{bmatrix}\right),$$

па је условна расподела $f'|u$ дата са:

$$f'|u \sim \mathcal{N}(k(X', X_u)k(X_u, X_u)^{-1}u, k(X', X') - k(X', X_u)k(X_u, X_u)).$$

Сада, користећи да је расподела вектора $u|y$ нормална $\mathcal{N}(\mu_u, A_u)$, добијамо да је $q_{f'}$ густина нормалне расподеле са очекивањем:

$$k(X', X_u)k(X_u, X_u)^{-1}\mu_u$$

и дисперзијом:

$$\begin{aligned} & k(X', X') - k(X', X_u)k(X_u, X_u)^{-1}k(X_u, X') + \\ & + k(X', X_u)k(X_u, X_u)^{-1}A_u k(X_u, X_u)^{-1}k(X_u, X'). \end{aligned}$$

На овај начин се сложеност рачунања расподеле вектора $f'|y$ смањује са $O(n^3)$ на $O(nm^2)$. Међутим, неопходно је да прво нађемо оптималне

вредности за μ_u , A_u и X_u . Кулбак-Лајблерова дивергенција је мера сличности две расподеле, па ћемо користећи тренинг скуп наћи q_f , односно вредности μ_u , A_u и X_u , које минимизују Кулбак-Лајблерову дивергенцију између q_f и $p_{f|y}$. Може се показати да је тај проблем еквивалентан максимизацији доње границе (у ознаци $\mathcal{L}(q)$) функције веродостојности, односно логаритма те функције, вектора y . Заиста, важи следећи низ једнакости:

$$\begin{aligned} KL(q_f||p_{f|y=y}) &= \int q_f(x) \ln \frac{q_f(x)}{p_{f|y=y}(x)} dx = E_{q_f} \left(\ln \frac{q_f}{p_{f|y=y}} \right) = \\ &= E_{q_f} \left(\ln \frac{q_f p_y(\mathbf{y})}{p(f,y)} \right) = E_{q_f} \left(\ln p_y(\mathbf{y}) - \ln \frac{p(f,y)}{q_f} \right) = \ln p_y(\mathbf{y}) - E_{q_f} \left(\ln \frac{p(f,y)}{q_f} \right) = \\ &= \ln p_y(\mathbf{y}) - \mathcal{L}(q). \end{aligned}$$

Дакле, из једнакости

$$KL(q_f||p_{f|y}) = \ln p_y(\mathbf{y}) - \mathcal{L}(q)$$

можемо закључити да је минимизација $KL(q_f||p_{f|y=y})$ еквивалентна максимизацији $\mathcal{L}(q)$.

Дакле, треба максимизовати функцију $\mathcal{L}(\mu_u, A_u, X_u)$. Тај проблем се може решити аналитички по μ_u и A_u , односно добија се да је $\mathcal{L}$ максимално за:

$$\mu_u = \frac{1}{\sigma^2} k(X_u, X_u) Sk(X_u, X) y$$

и

$$A_u = k(X_u, X_u) Sk(X_u, X_u),$$

где је S матрица димензије $m \times m$ дата следећим изразом:

$$S = \left(k(X_u, X_u) + \frac{1}{\sigma^2} k(X_u, X) k(X, X_u) \right)^{-1}.$$

Након што μ_u и A_u уврстимо у $\mathcal{L}$, $\mathcal{L}$ ће зависити само од X_u :

$$\mathcal{L}(X_u) = \ln \left(\frac{1}{\sqrt{|Q + \sigma^2 I|} (2\pi)^n} \exp \left(-\frac{y^T (Q + \sigma^2 I)^{-1} y}{2} \right) \right) - \frac{1}{2\sigma^2} Tr(k(X, X) - Q),$$

где је $Tr(\cdot)$ ознака за траг матрице, а Q је матрица димензије $n \times n$ дата следећим изразом:

$$Q = k(X, X_u) k(X_u, X_u)^{-1} k(X_u, X).$$

Горњи израз можемо записати у следећем облику:

$$\mathcal{L}(X_u) = -\frac{y^T(Q + \sigma^2 I)^{-1}y}{2} - \frac{1}{2} \ln |Q + \sigma^2 I| - \frac{n}{2} \ln(2\pi) - \frac{1}{2\sigma^2} \text{Tr}(k(X, X)) - \frac{1}{2\sigma^2} \text{Tr}(Q).$$

Рачунска сложеност рачунања вредности $\mathcal{L}(X_u)$ је $O(nm^2)$.

Матрица X_u представља параметре модела и рачуна се неком оптимизационом методом, заједно са параметрима кернела. Након што израчунамо матрицу X_u , можемо израчунати μ_u и A_u . Коначно, након што израчунамо X_u , μ_u и A_u , можемо израчунати расподелу вектора $f'|y$.

Глава 5

Потпуно повезане неуронске мреже

У овој глави ће бити описане потпуно повезане неуронске мреже (енг. *fully connected neural networks*), које су послужиле као инспирација за настанак дубоких Гаусових процеса, који ће бити представљени у наредној глави. Такође, биће описана и најкоришћенија метода за оптимизацију параметара - градијентни спуст (енг. *gradient descent*). Делови текста у овој глави су приказани по узору на скрипту [8].

Потпуно повезане неуронске мреже се састоје из више неурона организованих у слојеве, који примају информације од свих неурона из претходног слоја и након одређених трансформација их преносе неуронима наредног слоја. Овај тип мреже састоји се из улазног слоја, скривених слојева и излазног слоја. Величина различитих слојева, односно број неурона у њима, не мора бити иста. Мрежа која се састоји од барем два скривена слоја назива се дубока неуронска мрежа (енг. *deep neural network*). Неурон представља линеарну комбинацију неурона претходног слоја и на њу се додаје слободни члан. Затим се над том вредношћу примењује активациона функција. Резултат активационе функције је улаз неурона наредног слоја. Неуронска мрежа конструише нове атрибуте у својим скривеним слојевима. Сваки слој се надограђује над претходним и тако гради све сложеније атрибуте.

5.1 Архитектура мреже

Модел се дефинише на следећи начин:

$$h_0 = \mathbf{x},$$

$$h_i = g(W_i h_{i-1} + w_i), 1 \leq i \leq L$$

где је $\mathbf{x}$ вектор улазних атрибута, L је број слојева, W_i је матрица чија j -та врста представља вектор вредности параметара j -тог неурона у i -том слоју, w_i представља вектор слободних чланова i -тог слоја, h_i вектор атрибута у i -том слоју, а g је активациона функција.

Активациона функција је нека нелинеарна функција. У пракси се најчешће користи нека од наредних функција:

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

$$\text{relu}(x) = \max(0, x),$$

$$\text{lrelu}(x) = \max(\alpha x, x), \alpha \in (0, 1).$$

Потпуно повезана неуронска мрежа се може користити и за регресију и за класификацију. У случају регресије, на линеарну комбинацију вредности неурона последњег скривеног слоја се не примењује активациона функција и излазни слој има тачно један неурон, што је приказано на слици⁵ 5.1. У случају класификације излазни слој има онолико неурона колико има могућих класа, што је приказано на слици⁶ 5.2. Вероватноће припадања класама се добијају тако што се на вредности излазног слоја $\mathbf{x} = (x_1, x_2, \dots, x_m)$ примењује функција меког максимума (енг. *softmax function*), која је дефинисана са:

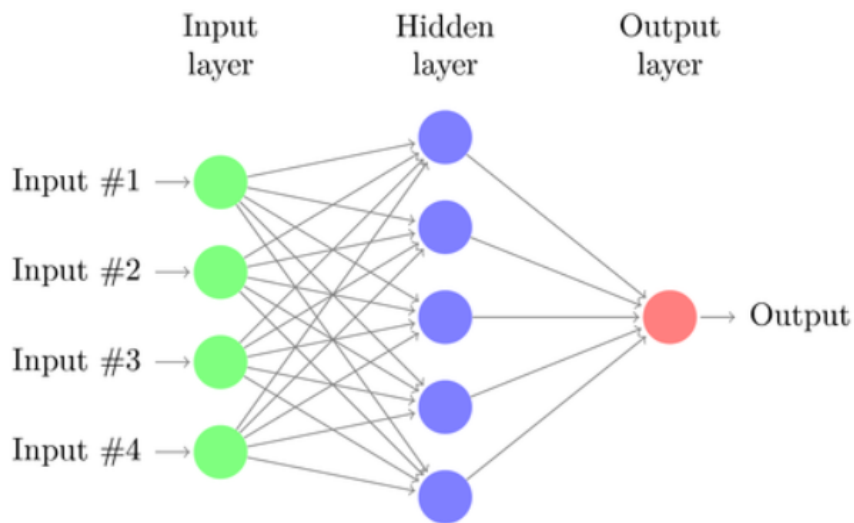
$$\text{softmax}(\mathbf{x}) = \left(\frac{e^{x_1}}{\sum_{k=1}^m e^{x_k}}, \frac{e^{x_2}}{\sum_{k=1}^m e^{x_k}}, \dots, \frac{e^{x_m}}{\sum_{k=1}^m e^{x_k}} \right),$$

где је m је број могућих класа.

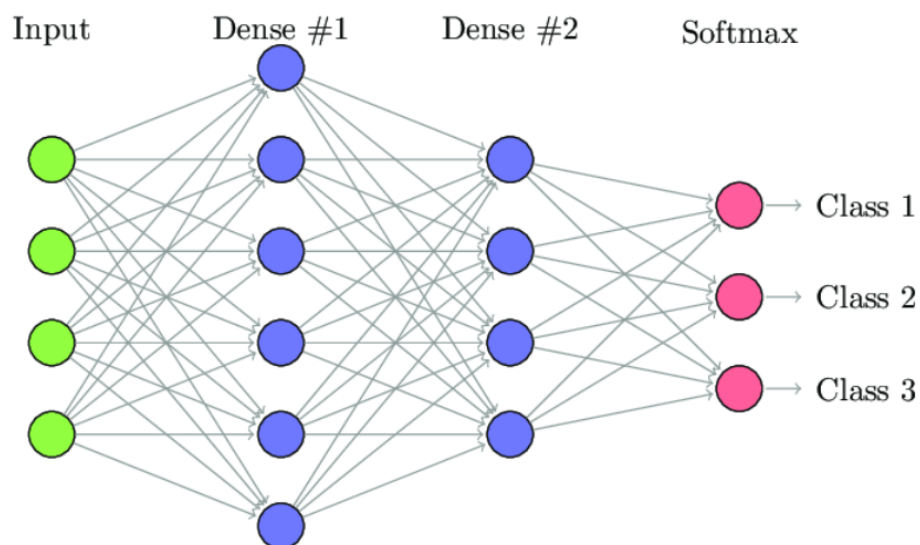
Главни задатак је оптимизовати модел, односно пронаћи што боље матрице параметара $W_i, 1 \leq i \leq L$.

⁵Слика је преузета са: <https://mineetha.com/2020/09/03/artificial-neural-network/>

⁶Слика је преузета са: https://www.researchgate.net/figure/Example-of-fully-connected-neural-network_fig2_331525817



Слика 5.1: Архитектура мреже (са једним скривеним слојем) у случају регресије.



Слика 5.2: Архитектура мреже (са два скривена слоја) у случају класификације са 3 класе.

5.2 Оптимизација

Циљ оптимизације је пронаћи оптималне вредности параметара модела. То се ради минимизовањем функције губитака l на тренинг скупу градијентним спустом. Основна идеја градијентног спуста је да се, полазећи од насумице изабраних вредности параметара модела, низом корака у правцу градијента

функције губитака, дође врло близу оптималних вредности параметара. Тај поступак је дат следећом формулом:

$$W_i^{k+1} = W_i^k - \gamma_k \nabla l(W_i^k)$$

где је W_i матрица параметара у i -том слоју, k је редни број итерације алгоритма, а γ_k је константа којом контролишемо дужину вектора градијента.

Рачунање градијента се састоји од пропагације унапред и пропагације уназад, које се наизменично смењују.

Алгоритмом пропагације унапред се рачуна вредност функције губитака на тренинг скупу, помоћу тренутних вредности параметара модела. Вредност функције губитака ће бити потребна за рачунање парцијалних извода функције губитака по вредностима параметара, а то нам је потребно за пропагацију уназад.

Схема по којој се врши пропагација унапред:

$$X \rightarrow XW_1 \rightarrow H_1 = g(XW_1) \rightarrow H_1W_2 \rightarrow \dots \rightarrow H_L \rightarrow H_LW_{L+1} \rightarrow \hat{y} \rightarrow l,$$

где је X матрица улазних атрибута, W_i матрица параметара у i -том слоју, H_i матрица атрибута у i -том слоју, $\hat{y}$ вредности предикција зависне променљиве на тренинг скупу, а l функција губитака на тренинг скупу.

Циљ пропагације уназад је израчунати градијент функције губитака, а затим ажурирати вредности параметара. Алгоритам пропагације уназад заснива се на правилу израчунавања парцијалног извода сложене функције. У свакој итерацији се, крећући се по слојевима неуронске мреже од последњег ка првом, обављају следећи задаци:

- проширивање до тада израчунатог парцијалног извода изводом активационе функције у складу са правилом за рачунање извода сложене функције,
- израчунавање вредности градијента функције губитака по вредностима параметара на текућем слоју, зарад чега се до тада израчунати парцијални извод множи улазима јединица које ти параметри множе,
- проширивање до тада израчунатог парцијалног извода изводом линеарне комбинације по улазима.

Глава 6

Дубоки Гаусови процеси

Дубоки Гаусови процеси су инспирисани методама дубоког учења. Главна разлика је у томе што у овом случају имамо непараметарски приступ. Слојеви су организовани тако да је веза између њих функција која зависи само од података и неколико параметара.

6.1 Мотивација

Наведимо прво једну врло значајну теорему, доказану у раду [25], која важи за потпуно повезане неуронске мреже са једним скривеним слојем.

Теорема 6.1.1 (*Теорема о универзалној апроксимацији*) Нека је g ограничена и строго растућа функција. Тада за сваку функцију $f \in C[0, 1]^d$ и свако $\varepsilon > 0$, постоји број $m \in \mathbf{N}$, матрица $W \in \mathbf{R}^{m \times d}$, вектор $w_0 \in \mathbf{R}^m$ и вектор $v \in \mathbf{R}^m$, тако да за свако $x \in [0, 1]^d$ важи:

$$|v^T g(Wx + w_0) - f(x)| < \varepsilon.$$

Односно, скуп свих потпуно повезаних неуронских мрежа са једним скривеним слојем је свуда густ на скупу функција непрекидних на интервалу $[0, 1]^d$. Међутим, иако таква неуронска мрежа постоји, тешко ју је наћи, односно главни задатак је наћи оптималну ширину m скривеног слоја и затим истренирати ту мрежу на одговарајући начин.

Показано је да постоји врло занимљива веза између Гаусовог процеса и потпуно повезане неуронске мреже са једним скривеним слојем, и тај резултат представља основну мотивацију да се код Гаусових процеса

примени исти приступ као код неуронских мрежа. Наиме, у раду [14] је наведено да је, под одређеним претпоставкама, потпуно повезана неуронска мрежа, са једним скривеним слојем у којем има бесконачно много чворова, еквивалентна Гаусовом процесу.

6.2 Дефиниција дубоког Гаусовог процеса

Неопходно је да прво дефинишемо матричну нормалну расподелу и вишедимензиони Гаусов процес.

Дефиниција 6.2.1 *Случајна матрица X , димензије $n \times p$, има матричну нормалну расподелу, у ознаци $X \sim \mathcal{MN}(M, U, V)$, ако је њена заједничка густина даћа са:*

$$p_X(\mathbf{X}) = \frac{\exp\left(-\frac{1}{2}\text{Tr}(V^{-1}(\mathbf{X} - M)^T U^{-1}(\mathbf{X} - M))\right)}{\sqrt{(2\pi)^{np} |V|^n |U|^p}},$$

где је M матрица димензије $n \times p$, U позитивно дефинићна матрица димензије $n \times n$, а V позитивно дефинићна матрица димензије $p \times p$.

Дефиниција 6.2.2 *Реалан случајни процес $\{X_t | t \in T\}$, где су X_t случајни вектори, назива се вишедимензиони Гаусов процес ако сваки његов коначно-димензиони засек има матричну нормалну расподелу.*

Дефиниција 6.2.3 *Дубоки Гаусов процес дубине L , у ознаци $f^{(1:L)}$, је дефинисан са:*

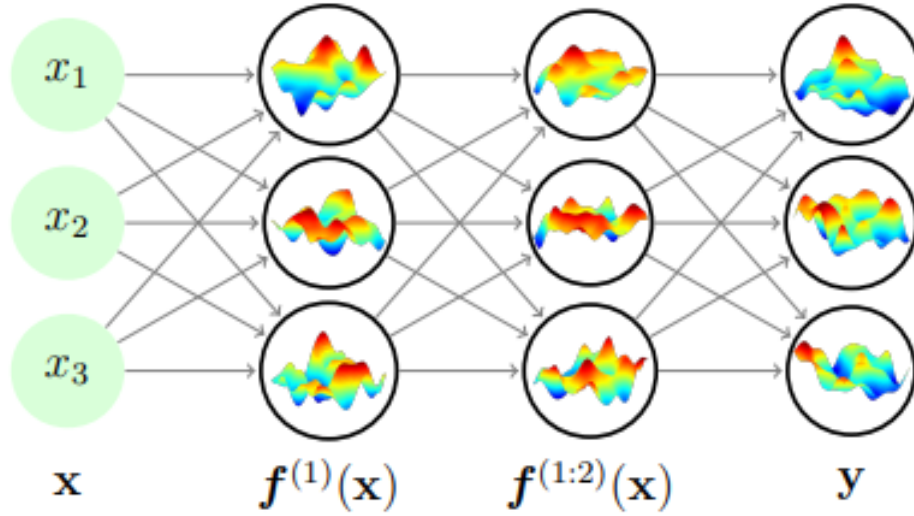
$$f^{(1:L)}(X) = f^{(L)}(f^{(L-1)}(\dots f^{(2)}(f^{(1)}(X)) \dots)),$$

где је X матрица априорија, а $f^{(1)}, f^{(2)}, \dots, f^{(L)}$ су независни вишедимензиони Гаусови процеси.

Дакле, дубоким Гаусовим процесом вршимо трансформисање предиктора, што је илустровано сликом⁷ 6.1.

Улазна вредност првог слоја је матрица димензије $n \times d$, где је n величина тренинг скупа, а d број предиктора. Излазна вредност првог слоја је такође матрица, чије су димензије $n \times d_1$, где је d_1 број нових предиктора. Та матрица

⁷Слика је преузета из [2]



Слика 6.1: Дубоки Гаусов процес дубине 2.

је улазна вредност наредног слоја. И тако даље, улазна вредност последњег (L -тог) слоја је матрица димензије $n \times d_{L-1}$, а излазна вредност је матрица димензије $n \times C$, где је C број могућих класа, ако вршимо класификацију, односно $C = 1$, ако је у питању регресија.

Посматрајмо један слој (на пример l -ти слој, за неко фиксирано $l, 1 \leq l \leq L$). Улазна вредност овог слоја је матрица F^{l-1} (напоменимо да је $F^0 = X$) димензије $n \times d_{l-1}$, а излазна вредност је матрица F^l димензије $n \times d_l$. Посматрани слој састоји се од d_l Гаусових процеса $f_1^{(l)}, f_2^{(l)}, \dots, f_{d_l}^{(l)}$, који пресликавају улазну матрицу у излазну матрицу на следећи начин: j -ти Гаусов процес, у ознаци $f_j^{(l)}$, прима улазну матрицу F^{l-1} , а као излаз даје j -ту колону излазне матрице, у ознаци F_j^l . Дакле, $f_j^{(l)}$ је Гаусов процес са функцијом средње вредности $m_j^l(\cdot)$ и коваријационим кернелом $k_j^l(\cdot, \cdot)$ и важи $F_j^l = f_j^{(l)}(F^{l-1})$. Последњи слој садржи један Гаусов процес у случају регресије, односно C Гаусових процеса у случају класификације са C класа, након којих применимо *softmax* функцију да бисмо добили вероватноће припадања свакој од класа.

Главни задатак је пронаћи оптималне параметре модела и томе је посвећено наредно поглавље.

6.3 Оптимизација

Постоји више приступа овом проблему. Најзначајнији резултати описани су у радовима [6] и [11]. Библиотека *GPflux*, која ће бити описани у глави 9, је изграђена на резултату добијеном у раду [11], који ћемо сада представити.

Настављамо да користимо ознаку F^l за матрицу излазних вредности у l -том слоју. Врсте матрице F^l ћемо означити са $\mathbf{f}_{i,l}$. Као у глави 4, са X_u ћемо означити матрицу апроксимације матрице предиктора X , коју користимо ради смањења рачунске сложености. Имамо и матрице $U^l, 0 \leq l \leq L, U^0 = X_u$, које представљају излазне вредности l -тог слоја ако кренемо од X_u . У [11] је показано да је логаритам функције веродостојности ограничен одоздо на следећи начин:

$$l(y|X, \Theta) \geq \sum_{i=1}^n E_{q_{\mathbf{f}_{i,L}}} \left(\ln p_{y_i|\mathbf{f}_{i,L}} \right) - \sum_{l=1}^L KL(q_{U^l} || p_{U^l}),$$

где је са p означена стварна густина, а са q оцењена густина.

Дакле, проблем максимизовања функције веродостојности по скупу параметара Θ , се у пракси замењује максимизовањем израза са десне стране.

Глава 7

Конволутивне неуронске мреже

Конволутивне неуронске мреже (енг. *convolutional neural networks*) представљају унапређење потпуно повезаних неуронских мрежа, специјализовано за обраду различитих врста сигнала. Послужиле су као инспирација за конволутивне Гаусове процесе који ће бити описани у наредној глави. Делови текста у овој глави су приказани по узору на скрипти [8].

Називају се конволутивним зато што уче филтере којима применом конволуције трансформишу матрицу атрибута. У обради сигнала су раније коришћени филтери дизајнирани од стране стручњака и значај конволутивних неуронских мрежа је управо у томе што не захтевају људски ангажман, већ саме установљавају која су својства битна, кроз учење адекватних филтера. Највећа предност у односу на потпуно повезане неуронске мреже јесте то што узимају у обзир топологију сигнала. На пример, при обради слика, конволутивне мреже су конструисане имајући у виду да то што су пиксели једни у околини других има посебан значај.

Конволутивне мреже највећу примену имају у обради слика, па ћемо њихову структуру описати на таквом примеру. Структура конволутивне мреже подразумева два типа слојева - конволутивни слој (енг. *convolutional layer*) и слој агрегације (енг. *pooling layer*). Након ових слојева имамо надовезану потпуно повезану неуронску мрежу.

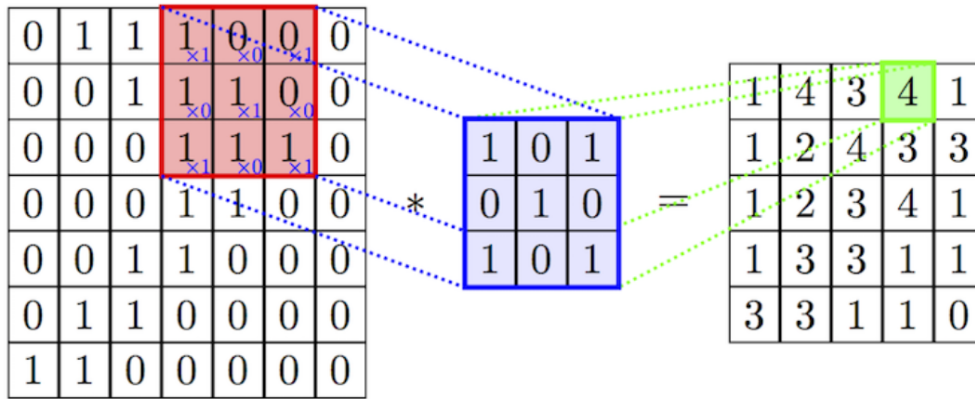
7.1 Конволутивни слој

Конволутивни слој садржи више филтрираних слика које су настале процесом конволуције почетне слике и филтера, како би се издвојили упечатљиви облици на сликама попут ивица објеката. Конволуција подразумева померање филтера дуж полазне слике (матрице), при чему се у сваком померају рачуна конволуција одговарајућег блока полазне матрице и филтера. На излазе конволутивног слоја се примењује активациона функција. У нижим конволутивним слојевима мреже се препознају једноставнији облици, док се у вишим слојевима могу разазнати и неки компликованији облици који се налазе на полазној слици.

Нека су f и g две матрице димензија $m \times n$ и $p \times q$. Конволуција ових матрица, у ознаци $f * g$, је матрица чији је елемент са индексом (i, j) дефинисан са:

$$(f * g)_{i,j} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} f_{i-k,j-l} g_{k,l}.$$

Ова операција је приказана на слици³ 7.1.



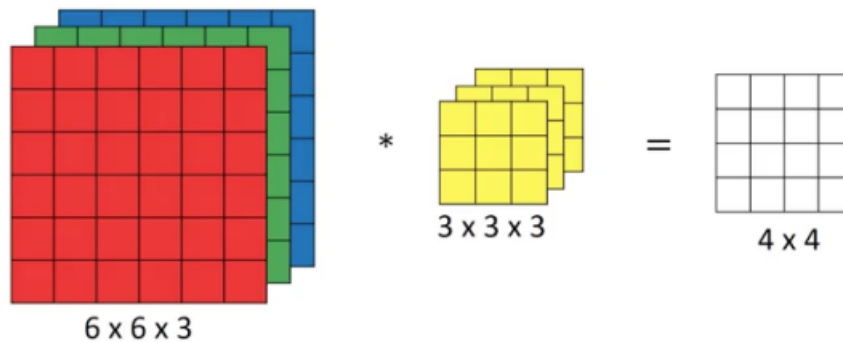
Слика 7.1: Конволуција матрице података и филтера.

У нашем случају f је матрица података, а g је филтер.

Конволутивним слојем се смањује димензија матрице. Да бисмо сачували димензије можемо на неки начин проширити матрицу, на пример нулама или вредностима које се већ налазе на ободу. Напоменимо да се приликом рачунања конволуција филтер дуж слике не мора померати за један пиксел, већ може и за више пиксела.

³Слика је преузета са: <https://towardsdatascience.com/demystifying-convolutional-neural-networks-384785791596>

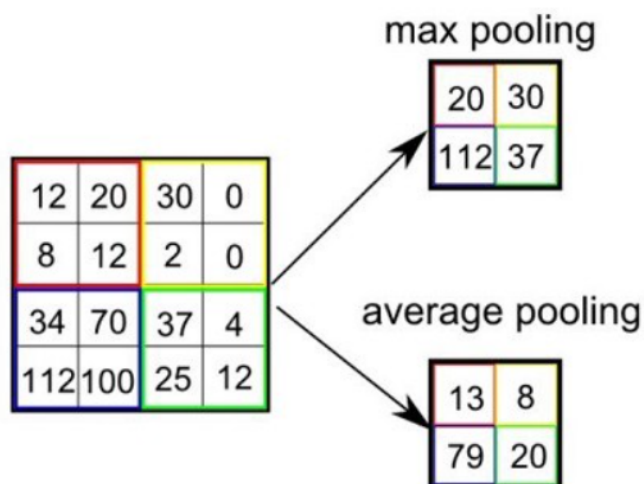
Важно је напоменути и да филтери делују одједном над више канала из претходног слоја, што је илустровано сликом⁸ 7.2.



Слика 7.2: Филтер који делује на више канала одједном.

7.2 Агрегациони слој

Агрегациони слој укрупњује информације које добија из претходног слоја, тако што рачуна најчешће максимум или просек суседних вредности претходног слоја, као што је приказано на слици⁹ 7.3.



Слика 7.3: Агрегација максимумом и просеком.

⁸Слика је преузета са: <https://medium.com/swlh/convolutional-neural-networks-part-3-convolutions-over-volume-and-the-convnet-layer-91fb7c08e28b>

⁹Слика је преузета са: <https://poojamahajan5131.medium.com/max-pooling-210fc94c4f11>

Уколико агрегира, на пример, 3×3 пиксела, онда је број излаза овог слоја 9 пута мањи од броја излаза претходног слоја. Рачунањем функције агрегације губи се информација где је тачно пронађено неко својство, али и даље имамо информацију да је својство пронађено, што је много битније. На пример, ако су на слици пронађени око, кљун и крила, информација о тачној позицији вероватно није битна за одлучивање да ли се на слици налази птица. Главна улога агрегационог слоја је смањење броја параметара у потпуно повезаној неуронској мрежи која је надовезана након конволутивних и агрегационих слојева.

Глава 8

Конволутивни Гаусови процеси

Конволутивне неуронске мреже су направиле пробој у домену компјутерског вида (енг. *computer vision*). Са друге стране, Гаусови процеси су неефикасни у задацима компјутерског вида. Зато се дошло на идеју да се комбинују конволутивне мреже и Гаусови процеси. Постоји више приступа. Један приступ је да се потпуно повезани део конволутивне мреже замени дубоким Гаусовим процесом. Ми ћемо разматрати други приступ, односно разматраћемо Гаусове процесе са конволутивним кернелом, чиме добијамо неосетљивост на транслацију, што је врло пожељна особина када на пример вршимо класификацију слика.

8.1 Конволутивни кернел

Претпоставимо да пред собом имамо скуп слика, односно матрица, димензије $D = W \times H$, којима је додељена нека од могућих C класа. Посматрајмо неку слику из тог скупа, означимо је са X . Хоћемо на неки начин да трансформишемо њене атрибуте, односно пикселе. Посматрамо делове те слике, односно подматрице димензије $E = w \times h$. Таквих подматрица има укупно $P = (W - w + 1) \cdot (H - h + 1)$. Нека је $g : \mathcal{R}^E \rightarrow \mathcal{R}$ функција која подматрицама додељује реалне бројеве. Означимо са f реалну функцију којом вршимо класификацију, $f : \mathcal{R}^D \rightarrow \mathcal{R}$. Оваква конструкција нас за сада подсећа на конволутивне неуронске мреже. Међутим, функције g и f ће бити дефинисане као Гаусови процеси, са функцијом средње вредности која је једнака нули и кернелима k_g и k_f , редом. Гаусов процес f

дефинисан је са:

$$f(X) \sim \mathcal{GP}(0, \sum_{p=1}^P \sum_{s=1}^P k_g(x^{[p]}, x^{[s]})), X \in \mathcal{R}^D.$$

Можемо приметити да све подматрице деле кернел k_g , односно кернел k_g је оно што је филтер код конволутивних мрежа.

8.2 Дубоки Гаусови процеси са конволутивним кернелима

Код конволутивних мрежа је флексибилност једног слоја контролисана бројем филтера, док је дубина мреже важна да би се омогућила довољна нелинеарност. По узору на то, можемо конструисати више слојева, и на сваки слој ћемо применити више Гаусових процеса (филтера), са идејом да ће различити филтери ухватити различите законитости међу пикселима.

Означимо са g_m^1 m -ти Гаусов процес (филтер) у првом слоју. Тада је:

$$g_m^1(x^{[p]}) \sim \mathcal{GP}(0, k_{g_m^1}(x^{[p]}, x^{[s]})).$$

Излаз првог слоја ће бити тензор који се састоји од жељеног броја матрица, при чему су те матрице мањих димензија од почетне.

Означимо са g_m^l m -ти Гаусов процес у l -том слоју и означимо са F^{l-1} улазну матрицу тог слоја и са $f_l^{[p]}$ њену p -ту подматрицу. Тада је:

$$g_m^l(f_l^{[p]}) \sim \mathcal{GP}(0, k_{g_m^l}(f_l^{[p]}, f_l^{[s]})).$$

На излаз последњег конволутивног слоја се примењује Гаусов процес f који врши класификацију.

Глава 9

Преглед библиотека *GPflow* и *GPflux*

*GPflow*¹⁰ је библиотека у програмском језику *Python* за изградњу модела базираних на Гаусовим процесима, а користи *Tensorflow*.

*GPflux*¹¹ је библиотека за изградњу дубоких Гаусових процеса, користећи градивне блокове из библиотеке *GPflow*.

У овој глави ћемо се кроз једноставне примере упознати са функцијама из ове две библиотеке.

9.1 *GPflow*

GPflow садржи функцију *gpflow.models.GPR* за прављење регресионог модела без апроксимације, као и функцију *gpflow.models.SGPR* која користи апроксимацију проређеним Гаусовим процесима.

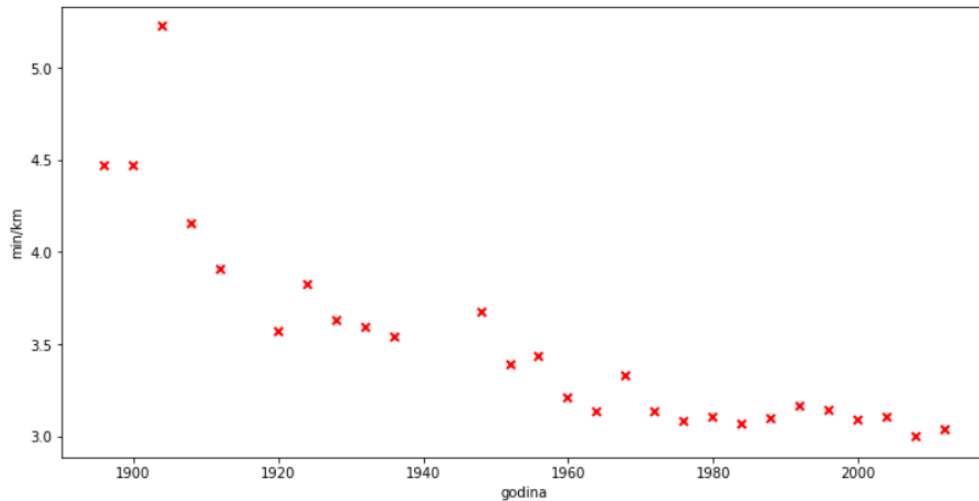
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import tensorflow as tf
4 import gpflow
```

Користићемо базу података из библиотеке *pods*. База има две колоне, што је врло погодно за визуализацију. У првој колони се налази година одржавања Олимпијских игара (од 1896. до 2012. године), а у другој колони је темпо трчања победника мушког маратона. Темпо трчања је информација колико

¹⁰<https://www.gpflow.org>

¹¹<https://secondmind-labs.github.io/GPflux/>

је просечно минута било потребно победнику да пређе један километар. На слици 9.1 видимо да је у последње време темпо трчања много бржи него у почетним годинама. Такође, можемо приметити да податак из 1904. године одудара од осталих података.



Слика 9.1: Зависност темпа трчања од године одржавања олимпијских игара

```

1 import pods
2 data = pods.datasets.olympic_marathon_men()
3 X = data['X']
4 Y = data['Y']
5 plt.plot(X,Y, "kx", color="r")

```

У библиотеци *GPflow* имплементирано је неколико основних кернела, а могуће је и њихово комбиновање у циљу добијања нових кернела, а такође је могуће и креирање потпуно нових основних кернела. У нашем примеру ћемо прво користити квадратно-експоненцијални кернел:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2l^2}\right).$$

Правимо модел без апроксимације проређеним Гаусовим процесима.

```

1 kernel = gpflow.kernels.RBF()
2 model = gpflow.models.GPR(data=(X,Y), kernel=kernel, mean_function=None)

```

Постоји више могућности за оптимизацију параметара модела, а ми ћемо користити *gpflow.optimizers.Scipy()*.

```

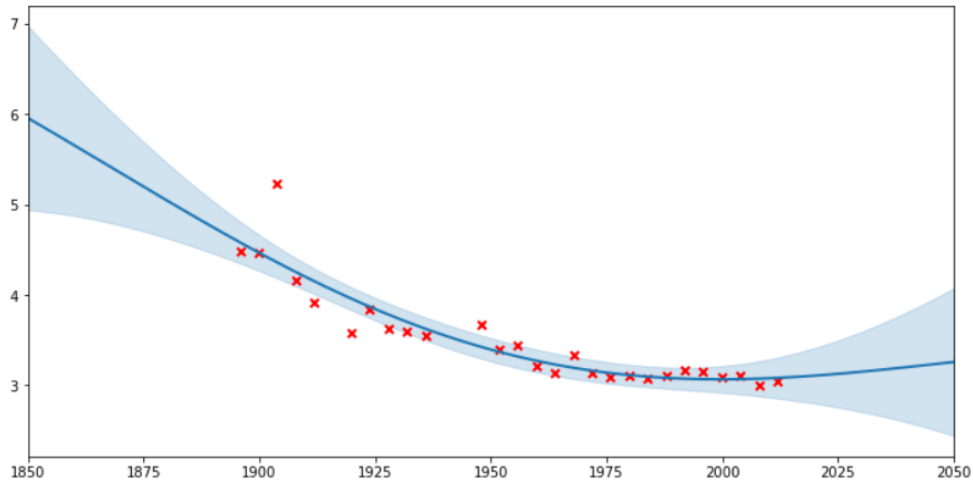
1 optimizer = gpflow.optimizers.Scipy()

```

```
2 optimizer_logs = optimizer.minimize(model.training_loss,model.  
    trainable_variables)
```

Предикција циљне променљиве је случајна величина са нормалном расподелом. Функција *predict_f* нам даје математичко очекивање и дисперзију расподеле, и то очекивање користимо као вредност наше предикције. Корисна је и функција *predict_f_samples*, јер је њена повратна вредност узорак из расподеле циљне променљиве у прослеђеној тачки. Помоћу тог узорка можемо да добијемо интервал поверења за нашу предикцију. На слици 9.2 се налазе добијен резултати.

```
1 test_data = np.linspace(1850,2050,200).reshape(200,1)  
2 mean,var = model.predict_f(test_data)  
3 samples = model.predict_f_samples(test_data,5)  
4 plt.figure(figsize=(12,6))  
5 plt.plot(X,Y,"kx",color='r')  
6 plt.plot(test_data,mean,color='C0',lw=2)  
7 plt.fill_between(test_data[:,0],mean[:,0]-1.96*  
8 np.sqrt(var[:,0]),mean[:,0]+1.96*np.sqrt(var[:,0]),color='C0',alpha  
    =0.2)  
9 plt.xlim(1850,2050)
```

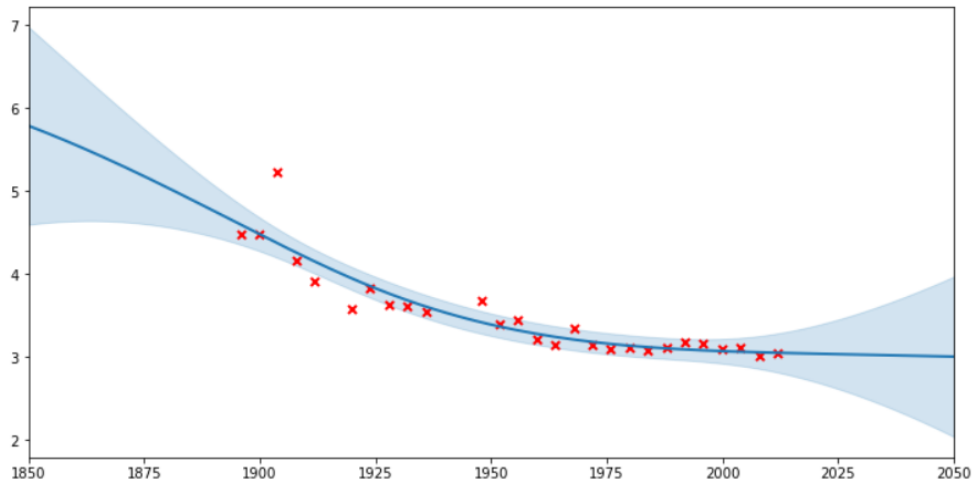


Слика 9.2: Модел са квадратно-експоненцијалним кернелом

Испробаћемо још неке кернеле.

На слици 9.3 је приказан модел са једним од Матерн кернела, који је дефинисан са:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \left(1 + \frac{\sqrt{5} \|\mathbf{x}_1 - \mathbf{x}_2\|}{l} + \frac{5 \|\mathbf{x}_1 - \mathbf{x}_2\|^2}{3l^2} \right) \exp \left(-\frac{\sqrt{5} \|\mathbf{x}_1 - \mathbf{x}_2\|}{l} \right).$$

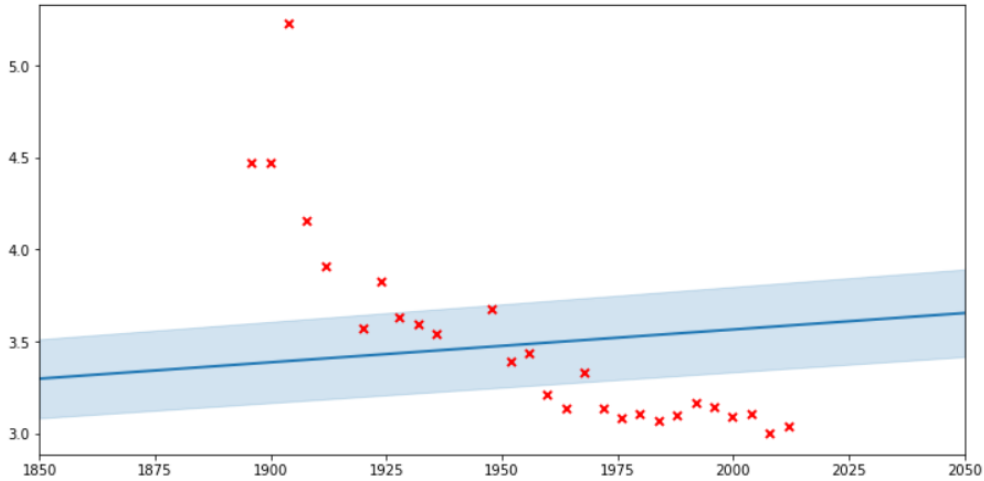


Слика 9.3: Модел са Матерн кернелом

Добили смо сличан модел као у првом случају.

На слици 9.4 је приказан модел са линеарним кернелом, који је дефинисан са:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 (\mathbf{x}_1 - c) \cdot (\mathbf{x}_2 - c).$$



Слика 9.4: Модел са линеарним кернелом

Линеарни кернел се у овом случају показао као врло лоша опција.

9.2 *GPflux*

Ова библиотека садржи функције за рад са дубоким Гаусовим процесима. Циљ овог мастер рада је упоређивање дубоких Гаусових процеса са дубоким неуронским мрежама и у те сврхе ћемо користити ову библиотеку. У овом делу ћемо описати како се користе функције из ове библиотеке, а у наредној глави ћемо правити моделе помоћу њих.

- за конструисање појединачних слојева користимо функцију `GPLayer` из пакета `gpflux.layers`,
- за спајање слојева у дубоки Гаусов процес користимо функцију `DeepGP` из пакета `gpflux.models`.

Прво учитавамо неопходне библиотеке и податке.

На истом примеру као у поглављу 9.1 ћемо показати како се користе наведене функције.

```

1 import pods
2 import numpy as np
3 from matplotlib import pyplot as plt
4 import tensorflow as tf
5 import gpflow
6 import gpflux

1 data = pods.datasets.olympic_marathon_men()
2 X = data['X']
3 Y = data['Y']
4 X = np.reshape(X, (-1, 1))
5 n = len(Y)

```

Правимо модел помоћу дубоког Гаусовог процеса са три слоја, при чему у сваком слоју користимо квадратно-експоненцијални кернел. Прво конструишемо појединачне слојеве. Подразумевано се користе проређени Гаусови процеси, а број тачака, помоћу којих се врши апроксимација, се подешава помоћу параметра *inducing_variable*. Даље, помоћу параметра *kernel* подешавамо који кернел користимо, а параметар *num_data* представља број инстанци у почетној бази података. Параметром *num_latent_gps* подешавамо број Гаусових процеса у слоју и тај број је једнак броју излаза, а то је у нашем случају 1. Подразумевана вредност параметра *mean_function* је идентичка функција, па је неопходно да тај параметар у излазном слоју променимо на нула функцију.

```

1 kernel1 = gpflow.kernels.SquaredExponential()
2 inducing_points1 = gpflow.inducing_variables.InducingPoints(X.copy())
3 layer1 = gpflux.layers.GPLayer(kernel=kernel1, inducing_variable=
    inducing_points1, num_data=n, num_latent_gps=1)
4 kernel2 = gpflow.kernels.SquaredExponential()
5 inducing_points2 = gpflow.inducing_variables.InducingPoints(X.copy())
6 layer2 = gpflux.layers.GPLayer(kernel=kernel2, inducing_variable=
    inducing_points2, num_data=n, num_latent_gps=1)
7 kernel3 = gpflow.kernels.SquaredExponential()
8 inducing_points3 = gpflow.inducing_variables.InducingPoints(X.copy())
9 layer3 = gpflux.layers.GPLayer(kernel=kernel3, inducing_variable=
    inducing_points3, num_data=n, num_latent_gps=1, mean_function=gpflow.
    mean_functions.Zero())

```

Сада ћемо спојити слојеве у дубоки Гаусов процес. Неопходно је да проследимо и параметар *likelihood*, којим подешавамо функцију веродостојности коју ћемо користити.

```

1 likelihood=gpflux.layers.LikelihoodLayer(gpflow.likelihoods.Gaussian()
    )
2 deep_gp = gpflux.models.DeepGP([layer1,layer2,layer3],likelihood=
    likelihood)
3 model = deep_gp.as_training_model()
4 model.compile(tf.optimizers.Adam(0.01))
5 model.fit({"inputs":X,"targets":Y},epochs=3000,verbose=0)

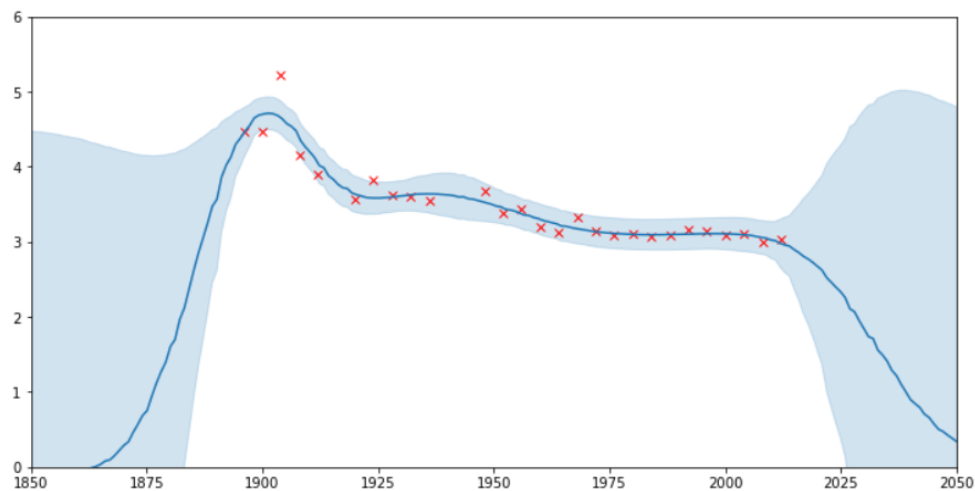
```

Након што смо истренирали модел, правимо предикције помоћу добијених вредности параметара нормалне расподеле. Добијени Гаусов процес је приказан на слици 9.5.

```

1 model = deep_gp.as_prediction_model()
2 test_data = np.linspace(1850,2050,200).reshape(200,1)
3 out = model(test_data)
4 mu = out.f_mean.numpy().squeeze()
5 var = out.f_var.numpy().squeeze()
6 test_data = test_data.squeeze()
7 lower = mu-1.96*np.sqrt(var)
8 upper = mu+1.96*np.sqrt(var)
9 plt.figure(figsize=(12,6))
10 plt.plot(X,Y,"kx",color='r')
11 plt.plot(test_data,mu,"C0")
12 plt.fill_between(test_data,lower,upper,color="C0",alpha=0.2)
13 plt.xlim(1850,2050)
14 plt.ylim(0,6)
15 plt.show()

```



Слика 9.5: Дубоки Гаусов процес

Глава 10

Примена Гаусових процеса на реалне податке

Модели засновани на Гаусовим процесима упорђени су са линеарном регресијом (LR), методом потпорних вектора (SVM), потпуно повезаним неуронским мрежама ($FCNN$) и методом градијентног појачавања $XGBoost$. Мере квалитета које смо користили су квадратни корен средњеквадратне грешке ($RMSE$) и средња апсолутна грешка (MAE). Кодови су писани у програмском језику *Python* и могу се пронаћи на следећем *GitHub* репозиторијуму [32].

Одабрали смо податке *Real estate* [33] и *Clean_Dataset* [34].

1. Скуп података *Real estate* садржи информације о ценама некретнина. Скуп садржи 414 опсервације које су описане помоћу 6 променљивих.
2. Скуп података *Clean_Dataset* садржи информације о ценама авионских карата. Скуп садржи 300261 опсервацију које су описане помоћу 10 променљивих. Издвојили смо подскуп од 10000 опсервација и 6 променљивих и на њему смо правили моделе.

10.1 Предвиђање цена некретнина

Подаци су подељени тако да тренинг скуп садржи 290 опсервација, а валидациони и тест скуп по 62 опсервације. Подела није постојала приликом преузимања података, већ је настала случајним одабиром.

У табели 10.1 су приказани резултати добијени на валидационом скупу, а у табели 10.2 резултати добијени на тест скупу.

	LR	SVM	GP	DGP1	DGP2	FCNN	XGB
RMSE	8.001	7.175	7.677	7.229	7.090	7.326	7.163
MAE	6.183	5.046	5.631	5.228	5.074	5.152	5.449

Табела 10.1: Резултати на валидационом скупу

	LR	SVM	GP	DGP1	DGP2	FCNN	XGB
RMSE	8.203	6.798	7.353	8.458	9.591	6.844	7.471
MAE	6.209	4.805	5.635	6.844	6.685	5.323	5.220

Табела 10.2: Резултати на тест скупу

GP представља проређени Гаусов процес проређен помоћу 64 тачке. Користили смо следећих 7 кернела: k_{M12} , k_{M32} , k_{M52} , k_{SE} , k_{RQ} , k_{PER} , k_{COS} . Модели су тренирани у 50 епоха, а за оптимизацију је коришћен оптимизатор *Adam* са коефицијентом учења 0.01. И у свим наредним моделима заснованим на Гаусовим процесима је коришћен исти оптимизатор, па то више нећемо посебно наводити. Најбољи резултат на валидационом скупу постигао је модел са рационално-квадратним кернелом и тај резултат је приказан у табели 10.1.

$DGP1$ представља проређени дубоки Гаусов процес дубине 2. У првом слоју смо користили 64, а у другом слоју 32 тачке. Наведених 7 кернела смо комбиновали на следећи начин: бирали смо по 2 кернела и њихов збир смо користили и у првом и у другом слоју. На тај начин смо направили 49 различитих модела. Модели су тренирани у 50 епоха. Најбољи резултат на валидационом скупу постигао је модел са збиром Матерн32 и Матерн52 кернела и тај резултат је приказан у табели 10.1.

$DGP2$ такође представља проређени дубоки Гаусов процес дубине 2. Једина разлика у односу на $DGP1$ је у томе што смо уместо збира користили производ два кернела. Најбољи резултат на валидационом скупу постигао је модел са производом квадратно-експоненцијалног и Матерн52 кернела и тај резултат је приказан у табели 10.1.

$FCNN$ је потпуно повезана мрежа са два скривена слоја. У првом слоју има 32, а у другом 16 неурона. Мрежа је тренирана у 50 епоха, а за

оптимизацију је коришћен оптимизатор *Adam* са коефицијентом учења 0.001.

SVM модел је најбољи резултат на валидационом скупу постигао са хиперпараметрима $\gamma=0.1$ и $C=100$ и тај резултат је приказан у табели 10.1.

XGB модел је најбољи резултат на валидационом скупу постигао са 50 стабала, максималне дубине 5 и тај резултат је приказан у табели 10.1.

10.2 Предвиђање цена авионских карата

Подаци су подељени тако да тренинг скуп садржи 7000 опсервација, а валидациони и тест скуп по 1500 опсервација. Подела није постојала приликом преузимања података, већ је настала случајним одабиром. Мере квалитета које смо користили су средњеквадратна грешка и средња апсолутна грешка.

У табели 10.3 су приказани резултати добијени на валидационом скупу, а у табели 10.4 резултати добијени на тест скупу.

	LR	SVM	GP	DGP1	DGP2	FCNN	XGB
RMSE	7048.68	10813.55	7245.02	5583.66	5424.38	6508.82	5257.03
MAE	4557.82	6305.84	5194.50	3356.85	3310.77	4431.11	3119.84

Табела 10.3: Резултати на валидационом скупу

	LR	SVM	GP	DGP1	DGP2	FCNN	XGB
RMSE	7099.03	11286.95	7506.98	5931.45	5889.97	6789.63	5522.19
MAE	4756.33	6767.94	5194.50	3582.69	3603.98	4667.18	3348.02

Табела 10.4: Резултати на тест скупу

GP представља проређени Гаусов процес проређен помоћу 256 тачака. Користили смо следећих 7 кернела: k_{M12} , k_{M32} , k_{M52} , k_{SE} , k_{RQ} , k_{PER} , k_{COS} . Модел су тренирани у 150 епоха. Најбољи резултат на валидационом скупу постигао је модел са периодичним кернелом и тај резултат је приказан у табели 10.3.

DGP1 представља проређени дубоки Гаусов процес дубине 2. У првом слоју смо користили 256, а у другом слоју 128 тачака. У оба слоја је коришћен збир Матерн32 и Матерн52 кернела. Модел је трениран у 150 епоха.

DGP2 представља проређени дубоки Гаусов процес дубине 2. У првом слоју смо користили 256, а у другом слоју 128 тачака. У оба слоја је коришћен производ квадратно-експоненцијалног и Матерн⁵² кернела. Модел је трениран у 150 епоха.

FCNN је потпуно повезана мрежа са два скривена слоја. У првом слоју има 32, а у другом 16 неурона. Мрежа је тренирана у 150 епоха, а за оптимизацију је коришћен оптимизатор *Adam* са коефицијентом учења 0.001.

SVM модел је најбољи резултат на валидационом скупу постигао са хиперпараметрима $\gamma=0.1$ и $C=100$ и тај резултат је приказан у табели 10.3.

XGB модел је најбољи резултат на валидационом скупу постигао са 50 стабала, максималне дубине 5 и тај резултат је приказан у табели 10.3.

Можемо приметити да су Гаусови процеси на првом скупу података дали добре резултате на валидационом скупу, али врло лоше резултате на тест скупу. На другом скупу података је ситуација значајно боља. Дубоки Гаусови процеси су дали значајно боље резултате од свих осталих модела, осим *XGB* модела.

Глава 11

Закључак

У овом раду представљени су Гаусови процеси и њихова примена у статистичком учењу. Гаусови процеси су погодни за моделовање зависне променљиве због повољних својстава вишедимензионе нормалне расподеле, као што су лако налажење маргиналне и условне расподеле.

У првом делу рада су дати примери основних кернела и начини за генерисање нових, описани су и дубоки кернели, а затим је описано на који начин се Гаусови процеси користе у регресионим задацима.

Наведене су предности и мане Гаусових процеса. Главне предности су мали број параметара и могућност рачунања поузданости предикције, док је највећа мана рачунска сложеност. Затим су описани проређени Гаусови процеси, који представљају апроксимацију Гаусових процеса која је настала у циљу смањивања временске и просторне сложености.

Фокус рада је у највећој мери био на дубоким Гаусовим процесима, који су инспирисани идејама из дубоког учења. У шестој глави су дефинисани и описано је на који начин се врши трансформација предиктора, а у осмој глави је описан конволутивни слој који је користан нпр. при класификацији слика.

Затим су описане библиотеке за рад са дубоким Гаусовим процесима у програмском језику *Python* и дати су примери примене функција из тих библиотека.

Коначно, у десетој глави, дубоки Гаусови процеси су упоређени са методама машинског учења у регресионим задацима.

Правци за даљи рад:

- Већина модела Гаусових процеса захтева пажљиву иницијализацију

модела и подешавање хиперпараметара, јер у супротном неће дати жељене резултате. Избор најбољег кернела је изузетно тежак посао због могућности комбиновања кернела, као и конструисања дубоких кернела. Аутоматизација овог процеса и дефинисање правила за иницијализацију модела и избор хиперпараметара би знатно олакшали рад са Гаусовим процесима.

- Показало се да дубоки Гаусови процеси раде добро на скуповима података као што је [34]. Међутим, постоје много већи скупови и на таквим скуповима дубоки Гаусови процеси нису детаљно испитани, углавном су испитани на моделима са највише десет слојева. Било би занимљиво видети како се понашају на високо структурираним скуповима података, на којима би био потребан већи број слојева.
- Неки правци истраживања се удаљавају од Гаусових процеса и уместо нормалне расподеле се разматрају неке друге расподеле попут Студентове расподеле, а разматра се и општи случајни процес параметризован помоћу неуронске мреже.

Можемо закључити да су Гаусови процеси одличан приступ моделирању података. Овај метод је прешао дугачак пут од свог настанка, нарочито у последњих десетак година, али и даље постоји доста проблема које треба решити да би добио широку примену.

Библиографија

- [1] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, Donald B. Rubin; *Bayesian Data Analysis*, Third Edition 2013
- [2] David K. Duvenaud *Automatic Model Construction with Gaussian Processes*, University of Cambridge, 2014
- [3] Kalvik Jakkala *Deep Gaussian Processes: A Survey*, University of North Carolina at Charlotte, 2021
- [4] Carl E. Rasmussen, Christopher K. I. Williams *Gaussian Processes for Machine Learning*, the MIT Press, 2006
- [5] Andreas C. Damianou *Deep Gaussian Processes and Variational Propagation of Uncertainty*, University of Sheffield, 2015
- [6] Andreas C. Damianou, Neil D. Lawrence *Deep Gaussian Processes*, 2013
- [7] Павле Младеновић *Вероватноћа и статистику*, Универзитет у Београду, 2008
- [8] Младен Николић, Анђелка Зечевић *Машинско учење*, Универзитет у Београду, 2019
- [9] Yarin Gal *Uncertainty in deep learning*, 2016
- [10] Ian Goodfellow, Yoshua Bengio, Aaron Courville *Deep Learning*, the MIT Press, 2016
- [11] Hugh Salimbeni, Marc Peter Deisenroth *Doubly Stochastic Variational Inference for Deep Gaussian Processes*, 2017
- [12] James Hensman, Nicolo Fusi, Neil D. Lawrence *Gaussian Processes for Big Data*, 2013

- [13] Radford M. Neal *Bayesian Learning for Neural Networks*, 1995
- [14] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, Jascha Sohl-Dickstein *Deep Neural Networks as Gaussian Processes*, 2018
- [15] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S.M. Ali Eslami, Yee Whye Teh *Neural Processes*, 2018
- [16] Vinayak Kumar, Vaibhav Singh, P. K. Srijith, Andreas Damianou *Deep Gaussian Processes with Convolutional Kernels*, 2018
- [17] Vincent Dutoir, Mark van der Wilk, Artem Artemev, James Hensman *Bayesian Image Classification with Deep Convolutional Gaussian Processes*, 2020
- [18] Kenneth Blomqvist, Samuel Kaski, Markus Heinonen *Deep convolutional Gaussian processes*, 2018
- [19] Liang Ding, Rui Tuo, Shahin Shahrampour *A Sparse Expansion for Deep Gaussian Processes*, 2021
- [20] Felix Leibfried, Vincent Dutoir, S. T. John, Nicolas Durrande *A Tutorial on Sparse Gaussian Processes and Variational Inference*, 2021
- [21] Michalis K. Titsias *Variational Learning of Inducing Variables in Sparse Gaussian Processes*, 2009
- [22] Michalis K. Titsias, Neil D. Lawrence *Bayesian Gaussian Process Latent Variable Model*, 2010
- [23] Jie Wang *An Intuitive Tutorial to Gaussian Processes Regression*, 2022
- [24] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, Jianfei Cai *When Gaussian Process Meets Big Data: A Review of Scalable GPs*, 2019
- [25] Kurt Hornik, Maxwell Stinchcombe, Halbert White *Multilayer Feedforward Networks are Universal Approximators*, 1989
- [26] Mark van der Wilk, Carl Edward Rasmussen, James Hensman *Convolutional Gaussian Processes*, 2017

- [27] Gia-Lac Tran, Edwin V. Bonilla, John P. Cunningham, Pietro Michiardi, Maurizio Filippone *Calibrating Deep Convolutional Gaussian Processes*, 2018
- [28] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo Leon-Villagra, Zoubin Ghahramani, James Hensman *GPflow: A Gaussian Process Library using TensorFlow*, 2017
- [29] Vincent Dutordoir, Hugh Salimbeni, Eric Hambro, John McLeod, Felix Leibfried, Artem Artemev, Mark van der Wilk, James Hensman, Marc P. Deisenroth, ST John *GPflux: A Library for Deep Gaussian Processes*, 2021
- [30] GPflow
- [31] GPflux
- [32] srbakoski/matf-master
- [33] Real estate price prediction
- [34] Flight Price Prediction

Биографија аутора

Михајло Србакоски је рођен 10. марта 1998. године у Београду. Завршио је основну школу „Бранко Ћопић” у Београду, а након тога Математичку гимназију. 2017. године уписује Математички факултет Универзитета у Београду, на смеру статистика, актуарска и финансијска математика. Основне студије завршава 2021. године са просечном оценом 9.82. Главне области интересовања су му вероватноћа и статистика, као и машинско учење.