

Хиймэл Нейроны Сүлжээг Ашиглан Нэг Хэмжээт Брату Бодлогын Шийдийг Тооцоолох нь

Бадарчийн Төмөрхуяг*, Балтын Батгэрэл

Шинжлэх Ухааны Академи, Математик, тоон технологийн хүрээлэн, Улаанбаатар 13330, Монгол улс

*Холбоо барих зохиогч: tumurkhuuyag.b@mas.ac.mn; ORCID:0000-0003-4710-5435

Өгүүллийн мэдээлэл: Хүлээн авсан: 2022.09.09; Зөвшөөрөгдсөн: 2022.10.01;
Нийтлэгдсэн: 2022.12.26

Хураангуй: Брату бодлогыг дулаан дамжуулалт, шаталтын процесс гэх мэт үзэгдлүүдийг загварчлахад өргөн ашигладаг. Тэгшитгэл параметрийн тодорхой утгуудад хоёр шийдтэй бөгөөд доод шийдийг олох нь төвөггүй, тооцон бодох математикийн стандарт аргуудыг хэрэглэх бүрэн боломжтой. Харин дээд шийдийг олох нь бэрхшээлтэй бөгөөд өндөр эрэмбийн нийлэлттэй алгоритмуудыг ашиглах шаардлага тулгардаг. Энэхүү судалгаанд Брату бодлогын тоон шийдийг хиймэл нейроны сүлжээ ашиглан тооцоолох аргыг авч үзлээ. Нейроны сүлжээг байгуулахдаа идэвхжилтийн функцээр синусоидийг, оновчлолын аргаар RMSprop (Root Mean Squared Propagation) аргыг ашиглав. Ингэснээр Брату бодлогын хоёр шийдийг тооцоолох боломжтой болов.

Түлхүүр үгс: Машин сургалт, Дифференциал тэгшитгэл, Захын нөхцөлт бодлого, Стандарт бус төгсгөлөг ялгаврын арга

1. Оршил

Сүүлийн үед дифференциал тэгшитгэлийн тоон шийдийг тооцоолоход хиймэл оюун ухааны аргуудыг эрчимтэй ашиглах болж байна. Энэхүү судалгаанд нэг хэмжээт Брату бодлогын шийдийг хиймэл оюун ухааны нейроны сүлжээ ашиглан тооцоолох талаар авч үзэв. Брату бодлого нь түлшний шаталтын загвар, цацрагийн дулаан дамжуулалт, дулааны урвал, химийн реакторын онол, нанотехнологи гэх мэт хими, физик болон инженерийн ухааны салбарт элбэг тохиолдох практик ач холбогдол ихтэй бодлого юм. Нэг хэмжээт Брату бодлого нь дараах хэлбэртэй байна.

$$u''(x) + \lambda e^{u(x)} = 0, \quad x \in [0, 1], \quad (1.1)$$

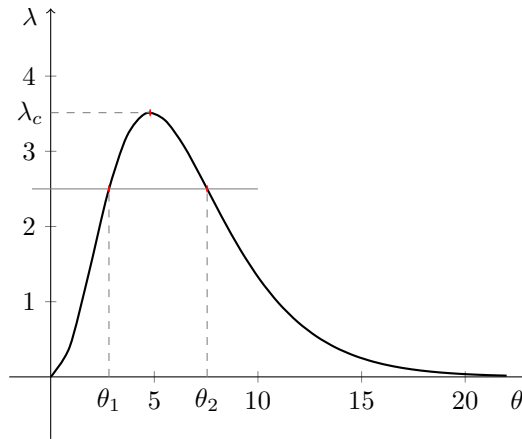
$$u(0) = u(1) = 0. \quad (1.2)$$

Энд: $u(x)$ нь үл мэдэгдэх функц, $\lambda > 0$ нь өгөгдсөн бодит тоо болно. Аналитик шийд

$$u(x) = -2 \ln \left[\frac{\cosh\left(\left(x - \frac{1}{2}\right)\frac{\theta}{2}\right)}{\cosh\left(\frac{\theta}{4}\right)} \right] \quad (1.3)$$

хэлбэртэй бөгөөд θ дараах тэгшитгэлийг хангана.

$$\theta = \sqrt{2\lambda} \cosh\left(\frac{\theta}{4}\right). \quad (1.4)$$

Зураг 1: λ ба θ -ийн хамаарал.

(1.4) тэгшитгэлийн θ хувьсагч нь λ параметрийн утгаас хамаарч Зураг 1-т үзүүлснээр нэг шийдтэй, хоёр шийдтэй эсвэл шийдгүй байна. Иймд (1.1)-(1.2) бодлого нь $\lambda = \lambda_c = 3.513830719$ критик утгад нэг шийдтэй, $\lambda > \lambda_c$ тохиолдолд шийдгүй, харин $\lambda < \lambda_c$ үед дээд ба доод гэсэн хоёр шийдтэй байна [1]. Хоёр шийдтэй үед дээд шийдийг тооцоолоход стандарт аргууд тохиромжгүй бөгөөд өндөр эрэмбийн нийлэлттэй аргууд ашиглах шаардлага тулгардаг. Сүүлийн жилүүдэд Брату бодлогын шийдийг судлах, тооцоолох ажлууд нэлээдгүй хийгдсэн [1-11]. Үүнээс [1] ажилд Брату бодлогын шийдийг төгсгөлөг ялгаврын арга, Коллокацийн арга, Адамын олон гишүүнтэд задлах арга, Онилох арга зэрэг аргуудаар тооцоолж, үр дүнг харьцуулан судалжээ. [2] ажилд дурдсанаар дээд шийдийн хүндрэлийг шийдийн асимптотик чанарыг багасгах замаар арилгах боломжтой гэж онцолжээ. Мөн сплайн функц ашиглан тооцоолох ажлууд [10], [11] хийгджээ. Нэг ба хоёр хэмжээст үед хиймэл нейроны сүлжээ ашиглан машин сургалтын аргаар тооцоолсон ажлууд [3], [4], [5], [6], [7] хийгдсэн байна. Эдгээрээс [3] ажилд хиймэл нейроны сүлжээ ашиглан захын нөхцөлт бодлогын шийдийг тооцоолох талаар авч үзсэн бол [4] ажилд Брату бодлогын шийдийг нэг хэмжээст тохиолдолд тооцоолсон бөгөөд дээд шийдийг хараахан гаргаагүй байна. [5] өгүүлэлд хоёр хэмжээст Брату бодлогын шийдийг нейроны сүлжээ ашиглан тооцоолох талаар судалжээ. Харин [6] өгүүлэлд хиймэл нейроны сүлжээ ашиглан анхны ба захын нөхцөлт дифференциал тэгшитгэлүүдийн шийдийг тооцоолох ерөнхий алгоритмыг санал болгосон бөгөөд Брату бодлогыг жишээ байдлаар оруулан зөвхөн нэг шийдийг тооцоолсон бол [7] өгүүлэлд машин сургалтын аргыг Ньютоны аргатай хослуулсан алгоритмыг санал болгосон бөгөөд Брату бодлогын зөвхөн дээд шийдийг жишээ байдлаар тооцоолсон байна.

Машин сургалт ба Ньютоны арга болон бусад тооцон бодох математикийн аргуудыг хослуулан ашиглах нь тооцооллын тоог нэмэх, тэр дундаа Якобиан матрицын урвууг тооцоолоход параметрийн тоо ихсэх тутамд гүйцэтгэх үйлдлийн тоо үлэмж хэмжээгээр нэмэгдэх болон Якобиан матрицын урвуу оршихгүй байх магадлалтай. Иймд бид энэхүү судалгаанд нэг хэмжээст Брату бодлогын шийдийг хиймэл нейроны сүлжээг ашиглан тооцоолох алгоритмыг санал болгож байна. Энэхүү ажил нь Брату бодлогын дээд ба доод шийдийг хоёуланг нь олох боломжтойн дээр, дээд шийд рүү нийлэх нийлэлтийг удирдах удирдлагын коэффициентуудыг тодорхойлсон. Мөн дээд шийдийг тооцоолох хүндрэлийн шалтгааныг олж, уламжлалыг тооцоолохдоо стандарт бус алхамтай төгсгөлөг ялгаврын аргыг хэрэглэсэн зэргээрээ онцлог болсон.

2. Арга зүй

2.1. Нейроны сүлжээг ашиглан дифференциал тэгшитгэлийн шийдийг тооцоолох нь

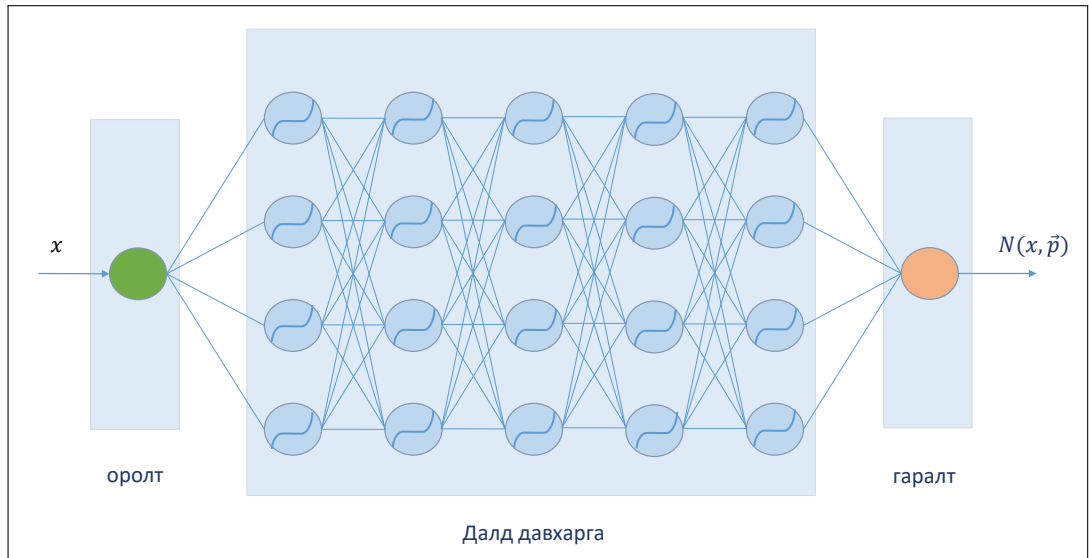
Хиймэл нейроны сүлжээг ашиглаж ердийн ба тухайн уламжлалт дифференциал тэгшитгэлүүдийн шийдийг тооцоолох аргыг I. E. Lagaris, A. Likas, D. I. Fotiadis нар [12] ажилд дэвшүүлсэн байна.

$$G(\vec{x}, \Psi(\vec{x}), \nabla\Psi(\vec{x}), \nabla^2\Psi(\vec{x})) = 0, \vec{x} \in D \quad (2.1)$$

бодлого авч үзье. Энд тодорхой захын нөхцөл (Дирихлей, Нейман гэх мэт) өгөгдөх бөгөөд $\Psi(\vec{x})$ нь үл мэдэгдэх функц байна. (2.1) дифференциал тэгшитгэлийн D тасралтгүй мужийг төгсгөлөг тооны зангилааны цэгүүд бүхий \hat{D} мужаар солих бөгөөд энэ үед бодлого дараах хэлбэрийн тэнцлүүдэд шилжинэ:

$$G(\vec{x}_i, \Psi(\vec{x}_i), \nabla\Psi(\vec{x}_i), \nabla^2\Psi(\vec{x}_i)) = 0, \forall \vec{x}_i \in \hat{D}. \quad (2.2)$$

(2.2) нь харгалзах захын нөхцөлийг хангана. Дифференциал тэгшитгэлийн шийдийг тооцоолохдоо эхлээд нейроны сүлжээг байгуулна (Зураг 2).



Зураг 2: Нейроны сүлжээ.

Энд: оролт нь D мужийн цэгүүд, гаралт нь сүлжээний \vec{p} параметрүүдээс хамаарах $N(x, \vec{p})$ функц байна. Дундын давхаргын тоо ба түүний нейроны тоог чөлөөтэй сонгох боломжтой. Сүлжээний нейрон хоорондын холбоос бүрт харгалзах w_j жин, b_j хазайлтын параметрүүдийг нийтэд нь нейроны сүлжээний \vec{p} параметрүүд гэнэ. \vec{p} параметрүүдийн анхны утгыг санамсаргүй байдлаар үүсгэнэ. Нейроны сүлжээг томьёо байдлаар илэрхийлбэл:

$$N(x_i, \vec{p}) = \begin{cases} MultilayerPerceptron(x_i) : \\ layer1 = \sigma(x_i \cdot w_1 + b_1) \\ layer2 = \sigma(layer1 \cdot w_2 + b_2) \\ layer3 = \sigma(layer2 \cdot w_3 + b_3) \\ \dots \\ output = layer(j-1) \cdot w_j + b_j \end{cases}$$

Энд: σ - идэвхжилтийн функц. (2.1) тэгшитгэлийн шийдийг сонгон авсан сүлжээнээс үл хамаарч, өгөгдсөн захын нөхцөлийг ямагт биелүүлдэг байлгахын тулд дараах хэлбэрт тавина:

$$\Psi_t(\vec{x}, \vec{p}) = A(\vec{x}) + F(\vec{x}, N(\vec{x}, \vec{p})). \quad (2.3)$$

Энд: $A(\vec{x})$ нь өгөгдсөн захын нөхцөлийг хангах хэсэг бөгөөд \vec{p} параметруудээс хамаарахгүй. Харин $F(\vec{x}, N(\vec{x}, \vec{p}))$ -г захын нөхцөлд нөлөөлөхгүй байхаар сонгоно. Ингээд зангилааны цэг бүрт $\Psi(\vec{x}) \approx \Psi_t(\vec{x}, \vec{p})$ нөхцөлийг хангах \vec{p} параметруудийн оновчтой утгыг олох зорилго үүснэ. Эндээс зорилгын функц гэж нэрлэгдэх

$$L = \sum_{\vec{x}_i \in \vec{D}} G(\vec{x}_i, \Psi_t(\vec{x}_i, \vec{p}), \nabla \Psi_t(\vec{x}_i, \vec{p}), \nabla^2 \Psi_t(\vec{x}_i, \vec{p}))^2 \xrightarrow{\vec{p}} \min \quad (2.4)$$

оптимизацийн бодлогод шилжинэ. (2.4) бодлогын \vec{p} параметруудийн оновчтой утгыг оптимизацийн уламжлалт аргад суурилсан машин сургалтын алгоритмыг ашиглан тооцоолох боломжтой.

2.2. Нейроны сүлжээг ашиглан нэг хэмжээст Брату бодлогын шийдийг тооцоолох нь

Нэг хэмжээст Брату бодлогыг нейроны сүлжээ ашиглан хэрхэн бодохыг авч үзье. Өгөгдсөн $x \in [0, 1]$ муж дээр $x_i = ih, i = 0, 1, \dots, n, h = \frac{1}{n}$ цэгүүд сонгон авъя. $x = 0, x = 1$ захын цэгүүд дээр (1.2) нөхцөл биелэгддэг байхын тулд шийдийг

$$\hat{u}(x, \vec{p}) = Cx(1 - x)N(x, \vec{p}) \quad (2.5)$$

хэлбэртэй сонгон авъя. C -тогтмол коэффициент. Зорилгын функц дараах хэлбэртэй болно:

$$L = \sum_{i=1}^{n-1} \left(\hat{u}''(x_i, \vec{p}) + \lambda e^{\hat{u}(x_i, \vec{p})} \right)^2 \xrightarrow{\vec{p}} \min. \quad (2.6)$$

Брату бодлогын дээд шийдийг олоход үүсдэг хүндрэлийн шалтгаануудын нэгд зорилгын функц дахь $u''(x)$ уламжлалыг тооцоолох арга зүй ихээхэн нөлөөтэй байна. Тухайлбал Хүснэгт 1-т дээд шийдийн ойролцоох 2-р эрэмбийн уламжлалуудын утгуудыг тооцоолж харууллаа. Энд 2-р мөрөнд (1.3) шийдийн хоёрдугаар эрэмбийн уламжлалын утгуудыг, 3-р мөрөнд ойролцоо шийдийн хоёрдугаар эрэмбийн уламжлалыг *Python* хэлний стандарт команд ашиглан тооцоолсон утгууд, 4-р мөрөнд төгсгөлөг ялгаврын аргаар тооцоолсон утгууд байна.

Хүснэгт 1: Ойролцоо шийдийн 2-р эрэмбийн уламжлалууд ($\lambda = 1$).

	x_i	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Жинхэнэ шийд	$u''(x_i)$	-2.9	-8.3	-21.7	-44.9	-59.8	-44.9	-21.7	-8.3	-2.9
Ойролцоо шийд	$\hat{u}''(x_i)$	-23.4	-26.6	-29.4	-31.5	-32.8	-31.6	-29.1	-26.7	-23.6
Төгсгөлөг ялгаврын арга	$\hat{u}''(x_i)$	-3.6	-7.7	-24.1	-44.8	-56.3	-46.9	-22.3	-8.4	0.92

Хүснэгтээс харахад *Python* хэлний стандарт команд ашиглан тооцоолсон утгууд төгсгөлөг ялгаврын аргаар тооцоолсноос эрс ялгаатай байна. Үүнээс шалтгаалж зорилгын функцийг утга эрс өөрчлөгдөж, доод шийд рүү нийлж байна. Иймээс зорилгын функц дахь хоёрдугаар эрэмбийн уламжлалыг

$$\hat{u}''(x_i, \vec{p}) \approx \frac{\hat{u}(x_{i+1}, \vec{p}) - 2\hat{u}(x_i, \vec{p}) + \hat{u}(x_{i-1}, \vec{p})}{\psi(h)} \quad (2.7)$$

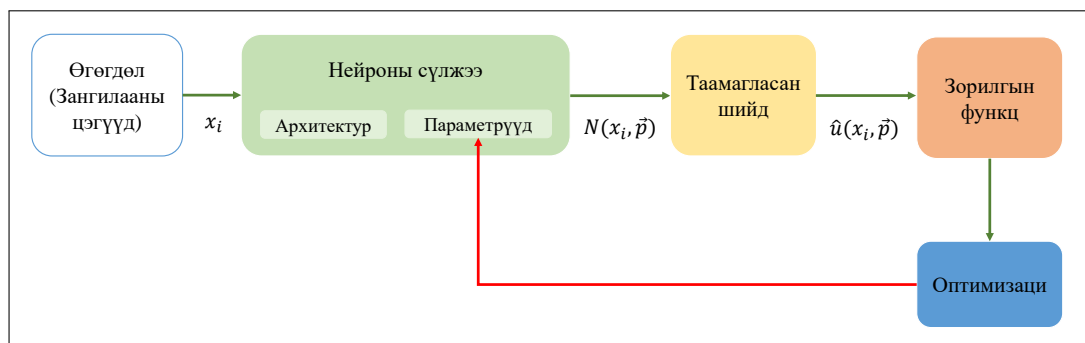
стандарт бус ялгаварт схемийн аргаар бодов. Энд: $\psi(h) = 2 \ln[\cosh(h)] = h^2 + O(h^4)$ хэлбэрийн функц байна. (2.6) бодлогын оновчтой тоон шийдийг олохын тулд дараах *RMSprop* (*Root Mean Squared Propagation*) оновчлолын алгоритмыг ашиглав.

$$\vec{p}_{t+1} = \vec{p}_t - \frac{\alpha}{E_t} \frac{\partial L(\vec{p}_t)}{\partial \vec{p}}, \quad t = 0, 1, 2, \dots \quad (2.8)$$

Энд: $\frac{\alpha}{E_t}$ сургалтын алхам бөгөөд α параметрийг бага утгатай байхаар сонгоно.

$$E_t = \sqrt{\beta(E_{t-1})^2 + (1 - \beta) \left(\frac{\partial L(\vec{p}_t)}{\partial \vec{p}} \right)^2}, \quad E_0 = \frac{\partial L(\vec{p}_0)}{\partial \vec{p}}. \quad (2.9)$$

Энд: β тогтмол коэффициент байх бөгөөд ихэнх тохиолдолд $\beta = 0.9$ утгыг сонгон авна. Зураг 3-т сургалтын загвар хэрхэн ажиллахыг харууллаа.



Зураг 3: Сургалтын загвар.

3. Үр дүн

Тоон туршилтыг λ параметрийн 0.1, 0.5, 1, 1.5, 2, 2.5, 3, 3.5 ба критик утга гэсэн тохиолдлуудад гүйцэтгэв. Сүлжээг дараах байдлаар тохируулав. Тоон мужийн хуваалтын тоо $n = 10$.

$$\text{Нейроны сүлжээ: } \begin{cases} \text{Оролтын нейроны тоо: 1} \\ \text{Далд давхаргын тоо: 5} \\ \text{Далд давхаргын нейроны тоо: 4} \\ \text{Идэвхжүүлэгч функц: Синусоид} \\ \text{Гаралтын нейроны тоо: 1} \end{cases}$$

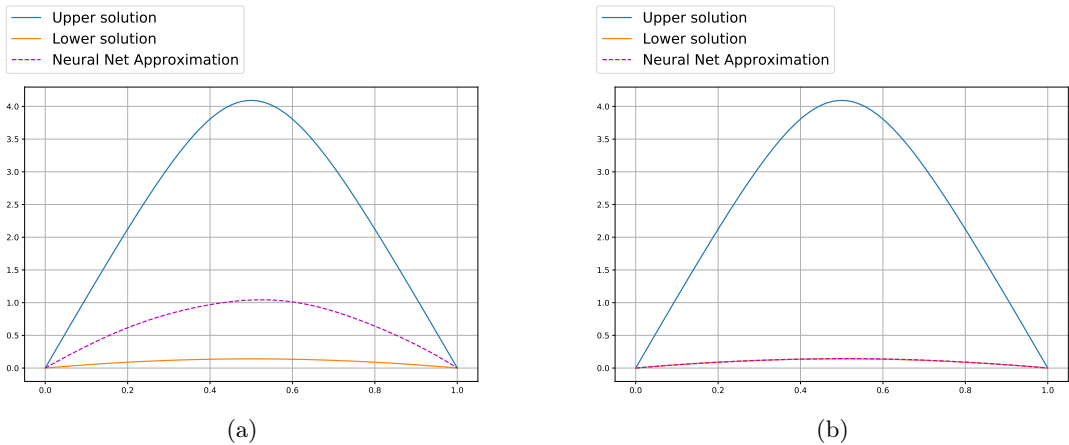
Сургалтын алхам: доод шийдэд $\alpha = 0.001$, дээд шийдэд $\alpha = 0.0005$.

Сургалтыг зогсоох нөхцөл: $L < 0.03$ эсвэл $\varepsilon < 0.0005$ (L -зорилгын функцийн утга, ε -дараалсан хоёр дөхөлтийн ялгаврын евклид норм.)

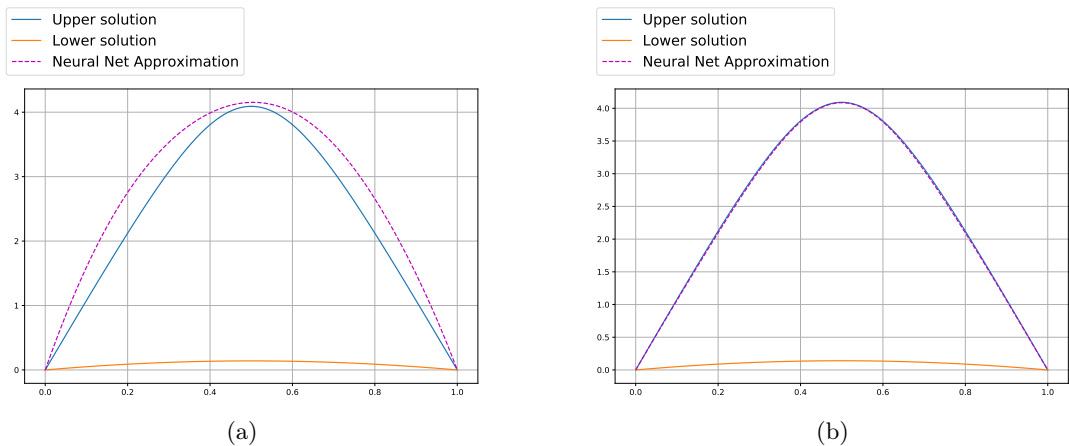
Оптимизаци: *RMSprop*.

Хүснэгт 2: C тогтмол коэффициентын утгууд.

λ	0.5	1	1.5	2	2.5	3	3.5
C	1	1	1	1	1	1	1
C_{upper}	67	30	15	8	5	3	1.25

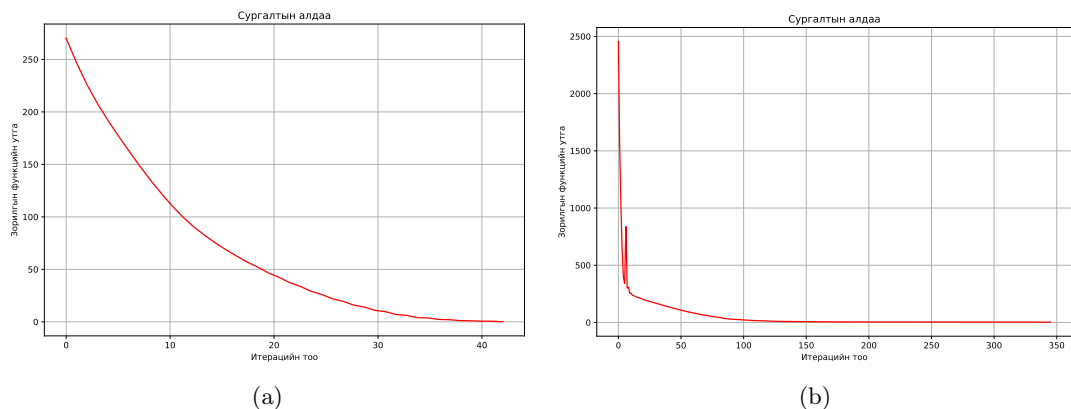


Зураг 4: доод шийдийн тооцоолол: (a)- сургалтын өмнө, (b)- сургалтын дараа ($\lambda = 1, \alpha = 0.001$).



Зураг 5: дээд шийдийн тооцоолол: (a)- сургалтын өмнө, (b)- сургалтын дараа ($\lambda = 1, \alpha = 0.0005$).

Эхлээд удирдлагын коэффициент $C = 1$ тохиолдолд сургалтыг гүйцэтгэж, доод шийдийг хялбар олно. Улмаар доод шийдийн \vec{p} параметрүүдээ ашиглаад $\hat{u}(x, \vec{p})$ функцийг Хүснэгт 2-ын дагуу $C = C_{upper}$ тохиолдолд дахин тодорхойлж, сургалтыг дахин гүйцэтгэснээр дээд шийдийг олох боломжтой болно. Сургалтыг гүйцэтгэхдээ Зураг 4-ийн (a) хэсэгт үзүүлсний дагуу $\alpha = 0.001$ сургалтын алхмаар 42 итераци сургахад зогсох нөхцөл биелсэн бөгөөд (b) хэсэгт үзүүлсний дагуу доод шийд рүү нийлсэн. Улмаар доод шийдийн параметрүүдээ ашиглан зорилгын функцээ дахин тодорхойлж, Зураг 5-ын (a) хэсэгт үзүүлсний дагуу $\alpha = 0.0005$ алхмаар 345 итераци сургахад зогсох нөхцөл биелсэн бөгөөд (b) хэсэгт үзүүлсний дагуу дээд шийд олдоно. Зураг 6-д дээд ба доод шийд рүү нийлэх нийлэлтийн алдааны графикийг харууллаа. Хэрэв дээд шийдэд дөхөхдөө α алхмыг хэт томоор сонгон авах, эсвэл сургалтыг зогсоох нөхцөлийг хэт багаар сонговол нийлэлт явагдахгүй юм уу, удаахрах магадлалтай байна. Мөн сургалтад машин сургалтын стандарт идэвхжүүлэгч функцүүд болох сигмоид, софтмакс, релу зэрэг функцүүдийг ашигласан боловч төдийлөн хурдан нийлэлтийг үзүүлэхгүй байгаа нь туршилтаар тогтоогдсон тул синусоид функцийг ашигласан. Оптимизаци хийхдээ *SGD, Adam, Adagrad, Adadelta, RMSprop, Adamax, Nadam, Ftrl* зэрэг түгээмэл ашиглагддаг оновчлолын алгоритмуудыг туршсан бөгөөд эдгээрээс *RMSprop* алгоритм нь оновчтой шийд рүү хурдан нийлж, зорилгын функцийн утгыг хамгийн сайн минимумчилж байна. Бусад оновчлолын алгоритмууд нь дээд шийд рүү огт нийлэхгүй эсвэл удаашралтай үр дүнг үзүүлснийг туршилтаар тогтоолоо.



Зураг 6: $\lambda = 1$ үед: (a)-доод шийд рүү нийлэх нийлэлтийн алдаа, ($C = 1, \alpha = 0.001$), (b)-дээд шийд рүү нийлэх нийлэлтийн алдаа, ($C = 30, \alpha = 0.0005$).

4. Дүгнэлт

Хиймэл оюун ухааны нейроны сүлжээ ашиглаж Брату бодлогын дээд ба доод шийдийг тооцоолж болохыг харуулсан. Дээд шийдийн хүндрэлийн нэг шалтгаан нь дээд шийд рүү нийлэх нийлэлтийн анхны дөхөлтийг хангалттай ойрхон өгч чадахгүй байх явдал байв. Иймээс доод шийдийг олсны дараа түүнд удирдлагын коэффициентоор үйлчлэх замаар дээд шийдийн ойролцоо орших дөхөлтийн утгуудыг гарган авсан. Дээд шийдийн хүндрэлийн дараагийн шалтгаан нь $u(x)$ функцийн хоёрдугаар эрэмбийн уламжлалын алхмыг оновчтой тохируулах явдал байв. Уламжлалыг тооцоолохдоо стандарт алхамтай төгсгөлөг ялгаврын аргыг хэрэглэж туршсан боловч зарим тохиолдолд дээд шийд рүү нийлэлт явагдахгүй байсан. Харин стандарт бус төгсгөлөг ялгаврын аргыг ашиглаж, турших үед нийлэлт тогтвортой явагдаж байгааг тоон туршилтаар олж тогтоосон. Мөн дээд шийдийн нийлэлтэд сургалтын алхмын хэмжээ, зорилгын функцийн параметруудийг оновчилж буй оптимизацийн арга чухал нөлөө үзүүлж байна. Сургалтын үед аналитик ба сургалтаар таамагласан шийдийн ялгаврын евклид нормын утгыг хэмжихэд доод шийд 6.369-өөс эхлээд 0.024 хүртэл буурсан. Харин дээд шийд 3.967-оос эхлээд 0.186 хүртэл буурсан нь дээд шийдийн нарийвчлал бага, сайжруулах шаардлагатайг харуулж байна. $\lambda \rightarrow 0$ тохиолдолд дээд шийдийн нийлэлт, $\lambda = \lambda_c$ үед критик шийдийн нийлэлтүүд удааших сул талтай байна.

Талархал

Энэхүү судалгааны ажлыг “Молекулын динамик загварчлал, хэрэглээ” сэдэвт ШуСс_2019/26 дугаартай суурь судалгааны төслөөс санхүүжүүлсэн болно. Судалгааны үед зөвлөгөө өгч, тусалж дэмжсэн төслийн баг, ажлын хамт олондоо талархсанаа илэрхийлье.

Ном зүй

- [1] R. Buckmire, “On Exact and Numerical Solutions of the One-Dimensional Planar Bratu Problem”, *Science Direct Working Paper*, 2003.
- [2] J. P. Boyd, “One-point pseudospectral collocation for the one-dimensional Bratu equation,” *Applied Mathematics and Computation*, Vol. 217, pp. 5553-5565, 2011, doi: <https://doi.org/10.1016/j.amc.2010.12.029>.
- [3] N. Yadav, A. Yadav, and M. Kumar, “An Introduction to Neural Network Methods for Differential Equations,” *Springer Briefs in Computational Intelligence*, 2015, doi: <https://doi.org/10.1007/978-94-017-9816-7>.
- [4] A. Ahmad, M. Sulaiman, A. J. Aljohani, A. Alhindi, and H. Alrabaiah, “Design of an efficient algorithm for solution of Bratu differential equations,” *Ain Shams Engineering Journal*, Vol. 12, pp. 2211-2225, 2021, doi: <https://doi.org/10.1016/j.asej.2020.11.007>.

- [5] M. A. Z. Raja, S. Ahmad, and R. Samar, “Solution of the 2-dimensional Bratu problem using neural network, swarm intelligence and sequential quadratic programming,” *Springer Neural Comput and Applic*, Vol. 25, pp. 1723-1739, 2014, doi: <https://doi.org/10.1007/s00521-014-1664-3>.
- [6] H. C. Lucas, “Differentiable Boundary Value Problem Solver,” *University of Nevada, Reno, Master of Science in Computer Science and Engineering*, 2020.
- [7] G. Fabiani, F. Calabro, L. Russo, and C. Siettos, “Numerical solution and bifurcation analysis of nonlinear partial differential equations with extreme learning machines,” *Journal of Scientific Computing*, Vol. 44, pp. 21-25, 2021, doi: https://doi.org/10.17506/26867206_2021_21_2_25.
- [8] A. Mohsen, “A simple solution of the Bratu problem,” *Elsevier-Computers and Mathematics with Applications*, Vol. 67, pp. 26-33, 2014, doi: <https://doi.org/10.1016/j.camwa.2013.10.003>.
- [9] N. Karamollahi, G. B. Loghmani, and M. Heydari, “A computational method to find dual solutions of the one-dimensional Bratu problem,” *Journal of Computational and Applied Mathematics*, Vol. 388, pp. 113309, 2021, doi: <https://doi.org/10.1016/j.cam.2020.113309>.
- [10] H. Caglar, N. Caglar, M. Özer, A. Valaristos, and A. N. Anagnostopoulos, “B-spline method for solving Bratu’s problem,” *International Journal of Computer Mathematics*, Vol. 87, pp. 1885–1891, 2010, doi: <https://doi.org/10.1080/00207160802545882>.
- [11] F. Lang, and X. Xu, “Quintic B-spline collocation method for second order mixed boundary value problem,” *Computer Physics Communications*, Vol. 183, pp. 913-921, 2012, doi: <https://doi.org/10.1016/j.cpc.2011.12.017>.
- [12] I. E. Lagaris, A. Likas, and D. I. Fotiadis, “Artificial Neural Networks for Solving Ordinary and Partial Differential Equations,” *Computational Physics*, Vol. 1, 1997, doi: https://doi.org/10.1007/978-1-4899-2846-7_1.

Artificial Neural Network Method for Solving of Bratu's Problem

Tumurkhuyag Badarch*, Batgerel Balt

Institute of Mathematics and Digital Technology, Mongolian Academy of Sciences, Ulaanbaatar 13330, Mongolia

**Corresponding author: tumurkhuyag.b@mas.ac.mn; ORCID:0000-0003-4710-5435*

Article Info: Received: 2022.09.09; Accepted: 2022.10.01; Published: 2022.12.26

Abstract: The Bratu's problem is widely used to model phenomenas such as heat transfer and combustion theory. For certain values of the parameters, there are 2 different solutions, and finding the lower solution is not difficult, and it is quite possible to apply standard mathematical methods for it calculation. However, finding the upper solution is difficult and requires the use of high-order convergent algorithms. In this study, the method of calculating the numerical solution of the Bratu's problem using artificial neural networks is considered. When constructing the neural network, sinusoids were used as the activation function, and RMSprop (Root Mean Squared Propagation) was used as the optimization method. By doing so, its possible to calculate two solutions of the Bratu's problem.

Key words: Differential equations, Nonstandard finite difference method, Boundary value problem, Machine learning algorithms
