**Министерство образования Российской Федерации**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**им. Н.Э. БАУМАНА**

Факультет: Информатика, искусственный интеллект и системы управления

Кафедра: Информационная безопасность (ИУ8)

ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Лабораторная работа №3 на тему:

## «исследование нейронных сетей с радиальными базисными функциями (RBF) на примере моделирования булевых выражений»

Вариант 15

**Преподаватель:**

Строганов И. С.

**Студент**:

Рыбаков Д. В.

**Группа:**

ИУ8-63

Москва 2022 г.

### Цель работы

Исследовать функционирование нейронной сети (НС) с радиальными базисными функциями (RBF) и ее обучение по правилу Видроу-Хоффа.

**Постановка задачи**

Постановка задачи. Получить модель булевой функции (БФ) на основе RBF-НС с двоичными входами , единичным входом смещения , синаптическими весами , двоичным выходом c пороговой ФА выходного нейрона, скрытыми RBF-нейронами с гауссовой ФА и координатами центров

Требуется найти минимальный набор векторов, используемых для обучения X.

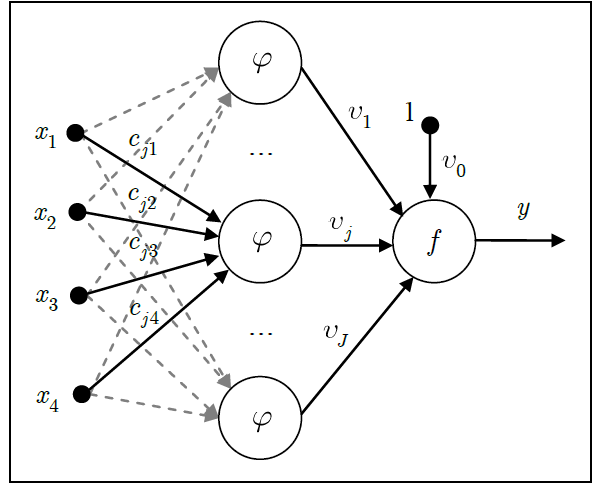
****

Рисунок 1 Нейронная сеть RBF

**Ход работы**

Получим таблицу истинности для моделируемой БФ:



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| x2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| x3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| x4 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 𝐹 | **1** | **1** | **1** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | **0** | **0** |

Норма обучения для всех случаев выбирается η = 0.3.

**Веса при обучении c пороговой ФА**

**V** = [

0.3,

0.465396791851418,

-0.2552658184207676,

-0.6313674243728802,

0.4889449218219064,

0.1374902594440197,

-0.9527564532004641,

-0.5482201349914202

]

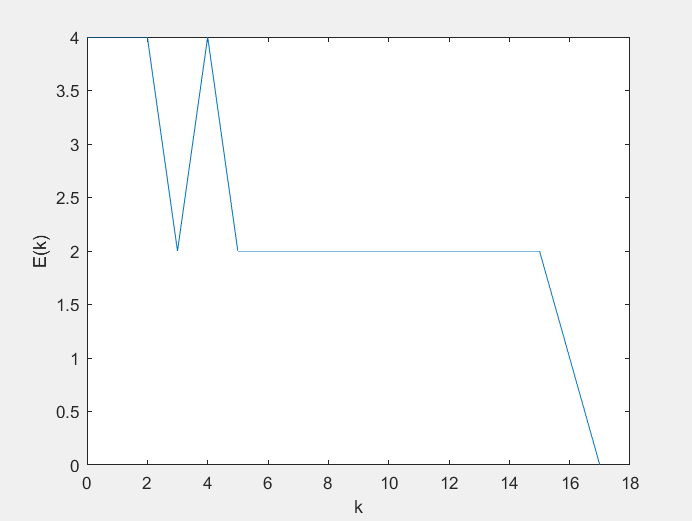


Рисунок 2 График суммарной ошибки НС по эпохам обучения для пороговой ФА

{

coefficients: [

0,

0.17463238806526132,

-0.21530063210919193,

-0.044098782919824255,

0.04409878291982427,

0.05132892892124925,

-0.17463238806526132,

-0.05132892892124925

],

Error: 4,

y: [

1, 1, 1, 1, 0, 1,

1, 0, 0, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 0

{

coefficients: [

0,

0.2795015287500738,

-0.4049367997577592,

-0.11386203030027314,

0.11386203030027314,

0.07699339338187387,

-0.3398233472867837,

-0.11209928668623742

],

Error: 4,

y: [

0, 1, 1, 1, 0, 1,

1, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 1

{

coefficients: [

0,

0.38437066943488624,

-0.3351735523773103,

-0.183625277680722,

0.2187311709850856,

0.10265785784249849,

-0.40958659466723263,

-0.17286964445122563

],

Error: 4,

y: [

0, 1, 1, 1, 0, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 2

{

coefficients: [

0,

0.3938120982786252,

-0.26541030499686136,

-0.15796081322009736,

0.19306670652446098,

0.09321642899875957,

-0.47934984204768155,

-0.19853410891185028

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 3

{

coefficients: [

0,

0.4032535271223641,

-0.45504647264542863,

-0.22772406060054623,

0.4268016570928525,

0.17920271199609417,

-0.5491130894281304,

-0.22419857337247492

],

Error: 4,

y: [

1, 1, 0, 1, 1, 0,

1, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 4

{

coefficients: [

0,

0.41269495596610306,

-0.3852832252649797,

-0.20205959613992158,

0.4011371926322279,

0.16976128315235522,

-0.6188763368085792,

-0.24986303783309957

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 5

{

coefficients: [

0,

0.422136384809842,

-0.3155199778845308,

-0.17639513167929693,

0.3754727281716033,

0.16031985430861628,

-0.688639584189028,

-0.2755275022937242

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 6

{

coefficients: [

0,

0.43157781365358094,

-0.2457567305040819,

-0.1507306672186723,

0.34980826371097873,

0.15087842546487734,

-0.7584028315694769,

-0.3011919667543488

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 7

{

coefficients: [

0,

0.4410192424973199,

-0.17599348312363303,

-0.12506620275804764,

0.32414379925035414,

0.1414369966211384,

-0.8281660789499257,

-0.3268564312149734

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 8

{

coefficients: [

0,

0.4504606713410588,

-0.10623023574318413,

-0.099401738297423,

0.29847933478972954,

0.13199556777739946,

-0.8979293263303745,

-0.35252089567559797

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 9

{

coefficients: [

0,

0.42479620688043424,

-0.2958664033917514,

-0.1691649856778719,

0.27281487032910495,

0.12255413893366053,

-0.9676925737108233,

-0.37818536013622256

],

Error: 2,

y: [

1, 1, 0, 0, 1, 1,

1, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 10

{

coefficients: [

0,

0.42479620688043424,

-0.2958664033917514,

-0.26459269751894543,

0.3079207636334685,

0.12255413893366053,

-0.9676925737108233,

-0.4132912534405861

],

Error: 2,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 11

{

coefficients: [

0,

0.42479620688043424,

-0.2958664033917514,

-0.36002040936001894,

0.343026656937832,

0.12255413893366053,

-0.9676925737108233,

-0.4483971467449496

],

Error: 2,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 12

{

coefficients: [

0,

0.42479620688043424,

-0.2958664033917514,

-0.45544812120109246,

0.37813255024219555,

0.12255413893366053,

-0.9676925737108233,

-0.48350304004931316

],

Error: 2,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 13

{

coefficients: [

0,

0.42479620688043424,

-0.2958664033917514,

-0.5508758330421659,

0.4132384435465591,

0.12255413893366053,

-0.9676925737108233,

-0.5186089333536767

],

Error: 2,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 14

{

coefficients: [

0,

0.42479620688043424,

-0.2958664033917514,

-0.6463035448832394,

0.4483443368509226,

0.12255413893366053,

-0.9676925737108233,

-0.5537148266580404

],

Error: 2,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 15

{

coefficients: [

0.3,

0.465396791851418,

-0.2552658184207676,

-0.6313674243728802,

0.4889449218219064,

0.1374902594440197,

-0.9527564532004641,

-0.5482201349914202

],

Error: 1,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 16

{

coefficients: [

0.3,

0.465396791851418,

-0.2552658184207676,

-0.6313674243728802,

0.4889449218219064,

0.1374902594440197,

-0.9527564532004641,

-0.5482201349914202

],

Error: 0,

y: [

1, 1, 1, 0, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 17

**Веса при обучении c логистической ФА**

**V** = [

0.13735073723246125,

0.19614184764226983,

-0.05851966498417733,

-0.32978779583128737,

0.25645262879604036,

0.11427058940174964,

-0.47336096850904114,

-0.272366197882416

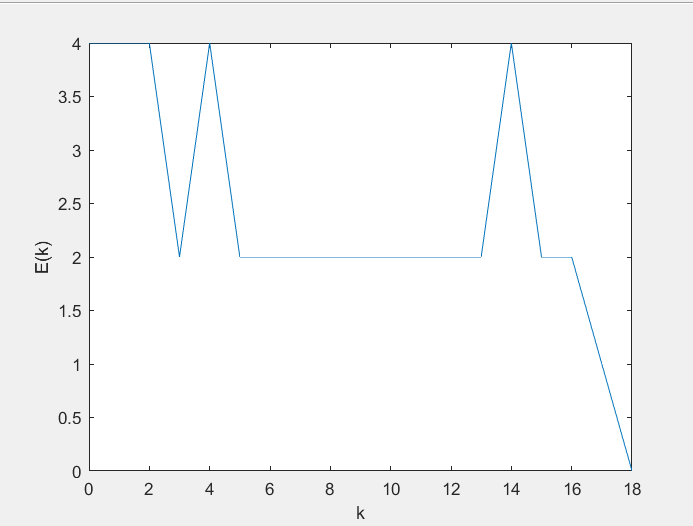


Рисунок 3 График суммарной ошибки НС по эпохам обучения для логистической ФА

{

coefficients: [

-0.008069433405992366,

0.08431569735444168,

-0.1093768505381028,

-0.02310955248924875,

0.020204176759502496,

0.024560643419415693,

-0.08795135057358637,

-0.02605447590764872

],

Error: 4,

y: [

1, 1, 1, 1, 0, 1,

1, 0, 0, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 0

{

coefficients: [

-0.005674219168544714,

0.13686462682458958,

-0.2031583916138629,

-0.05760985341239605,

0.05539334922820247,

0.03750601652248655,

-0.16964036350023426,

-0.056106184332950663

],

Error: 4,

y: [

0, 1, 1, 1, 0, 1,

1, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 1

{

coefficients: [

-0.002962533993280808,

0.18953986460005226,

-0.16796097876527671,

-0.09163209150034571,

0.10787029584904474,

0.05045442109190498,

-0.20420161283053276,

-0.08610009119502386

],

Error: 4,

y: [

0, 1, 1, 1, 0, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 2

{

coefficients: [

-0.0035222955328112204,

0.19420943227179135,

-0.13345728128104353,

-0.07893889055149751,

0.09507347052766091,

0.04574673214390575,

-0.238986990563615,

-0.0988969165164077

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 3

{

coefficients: [

-0.004060103929578368,

0.19888957132610277,

-0.2285553755482016,

-0.11392352432696876,

0.21194133722493672,

0.08874001763539818,

-0.27381371953816175,

-0.11170895410939323

],

Error: 4,

y: [

1, 1, 0, 1, 1, 0,

1, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 4

{

coefficients: [

-0.00603092779908701,

0.20350222863097242,

-0.19447219195737733,

-0.10138502179423264,

0.19903799114287893,

0.08399314188948906,

-0.30888865071933763,

-0.124612300191451

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 5

{

coefficients: [

-0.007928583272539369,

0.20812336396145925,

-0.16032636375965972,

-0.08882347359852023,

0.1861251445033859,

0.07924277108381982,

-0.34398940709314996,

-0.13752514683094402

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 6

{

coefficients: [

-0.00971042039454556,

0.21275349403015212,

-0.12611407293674248,

-0.07623747516939057,

0.1732092881960543,

0.07449129308322802,

-0.3790983445923576,

-0.15044100313827563

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 7

{

coefficients: [

-0.011333385841616694,

0.21739317023275703,

-0.09183124519482118,

-0.06362552765791575,

0.16029689350444484,

0.06974108853989364,

-0.41419787244445055,

-0.1633533978298851

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 8

{

coefficients: [

-0.009753650845899742,

0.20477478463449533,

-0.10444963079308289,

-0.11547873043211487,

0.21276018216178874,

0.08265191839547435,

-0.4012870425888699,

-0.1759717834281468

],

Error: 2,

y: [

1, 0, 1, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 9

{

coefficients: [

-0.011381545548718036,

0.20941443215168765,

-0.07016701500928452,

-0.10286686089567584,

0.19984695292979954,

0.07790140684189144,

-0.4363888389569123,

-0.188885012660136

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 10

{

coefficients: [

-0.011417871527948598,

0.19669918448969922,

-0.035621668112972404,

-0.13743048755961532,

0.20452215723651898,

0.07322372863764266,

-0.4236803160456234,

-0.2016002603221244

],

Error: 2,

y: [

1, 1, 0, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 11

{

coefficients: [

-0.011507015376065244,

0.1839564066523161,

-0.001028065170485891,

-0.17206894899906452,

0.2092038922889153,

0.06853592264785432,

-0.4109540407270347,

-0.21434303815950756

],

Error: 2,

y: [

1, 1, 0, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 12

{

coefficients: [

-0.012477491345138703,

0.18862878786590187,

0.03349642173228898,

-0.15936810005054092,

0.1963233865468212,

0.0637974493934472,

-0.4459668854271315,

-0.22722354390160168

],

Error: 2,

y: [

1, 1, 0, 1, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 13

{

coefficients: [

-0.01270974246804274,

0.17583337011938016,

-0.07882814250706702,

-0.2418588913747378,

0.20104846148119343,

0.10679951803958619,

-0.4808320398712046,

-0.24001896164812336

],

Error: 4,

y: [

1, 1, 0, 0, 1, 1,

1, 0, 1, 0, 1, 0,

1, 1, 0, 0

]

} эпоха 14

{

coefficients: [

-0.012633511534203334,

0.1758436868544027,

-0.07881782577204446,

-0.2895420680778629,

0.21860183483329715,

0.106803313354301,

-0.48082824455648976,

-0.2575606220469482

],

Error: 2,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 15

{

coefficients: [

-0.012639662181390648,

0.1758428544548235,

-0.07881865817162365,

-0.337255378101428,

0.23615363560859406,

0.10680300713160898,

-0.4808285507791818,

-0.27511336787485713

],

Error: 2,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 1, 0,

1, 1, 0, 0

]

} эпоха 16

{

coefficients: [

0.13735073723246125,

0.19614184764226983,

-0.05851966498417733,

-0.32978779583128737,

0.25645262879604036,

0.11427058940174964,

-0.47336096850904114,

-0.272366197882416

],

Error: 1,

y: [

0, 1, 1, 0, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 17

{

coefficients: [

0.13735073723246125,

0.19614184764226983,

-0.05851966498417733,

-0.32978779583128737,

0.25645262879604036,

0.11427058940174964,

-0.47336096850904114,

-0.272366197882416

],

Error: 0,

y: [

1, 1, 1, 0, 1, 1,

0, 0, 1, 1, 0, 0,

1, 1, 0, 0

]

} эпоха 18

**Поиск минимальных векторов**

Используем пороговую ФА

**Пороговая ФА имеет следующие параметры**

**V** = [ 0.3,

0.35655362471025576,

0.04060058497098381,

-0.7484855742182288,

0.3214477314058922,

0.014936120510359183,

-0.8439132860593023,

-0.5913054945075603

Минимальные вектора:

**x**(1) = [0, 0, 0, 0]

**x**(2) = [0, 0, 1, 1]

**x**(3) = [1, 0, 1, 0]

количество эпох: 20

**Логистическая ФА имеет следующие параметры**

Используем гиперболический тангенс (логистическую) как ФА:



**V** = [0.15112773578247177,

0.11703362412916155,

-0.03963032213456499,

-0.3468551806431457

,0.11708149125796327,

-0.0018967929813738888,

-0.34672506429674343,

-0.24979053204519894

Минимальные вектора:

**x**(1) = [0, 0, 0, 0]

**x**(2) = [0, 0, 1, 0]

**x**(3) = [0, 0, 1, 1]

**x**(4) = [0, 1, 1, 0]

**x**(5) = [1, 0, 1, 0]

количество эпох: 16

**Вывод**

Было проведено исследование функционирование нейронной сети (НС) с радиальными базисными функциями (RBF) и ее обучение по правилу Видроу-Хоффа. Были найдены веса на минимальных векторах используя порогувую и логистическую ФА со скрытыми RBF-нейронами с гауссовой ФА для обучения булевой функции с четырьмя переменными.

**Исходный код**

|  |
| --- |
| const learning\_rate = 0.3 |
|  | const fi0 = 1 |
|  | let tTeacher = [1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0] // значение t на наборах переменных |
|  |  |
|  | // значения всех возможных наборов аргументов |
|  | let X1 = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1] |
|  | let X2 = [0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1] |
|  | let X3 = [0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1] |
|  | let X4 = [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1] |
|  |  |
|  | let Cji = [[1, 1, 0, 0], |
|  | [0, 1, 1, 0], |
|  | [1, 1, 1, 0], |
|  | [0, 1, 0, 1], |
|  | [1, 1, 0, 1], |
|  | [0, 1, 1, 1], |
|  | [1, 1, 1, 1]] |
|  |  |
|  |  |
|  | const porogovayaFunc = net => net >= 0 ? 1 : 0 |
|  |  |
|  | const logisticFunc = out => out >= 0.5 ? 1 : 0 |
|  |  |
|  | const hyperbolicTangentOut = x => |
|  | (((Math.exp(x) - Math.exp(-x)) / |
|  | (Math.exp(x) + Math.exp(-x))) + 1) \* 0.5 |
|  |  |
|  | const fiCalculation = (X, Cji, j) => { |
|  | let sum = 0 |
|  | for (let i = 0; i < 4; i++) { |
|  | sum += ((X[i] - Cji[j][i]) \* (X[i] - Cji[j][i])) |
|  | } |
|  | return Math.exp(-sum) |
|  | } |
|  |  |
|  | const calculateNet = (Cji, V, X) => { |
|  | let net = 0 |
|  | for (let j = 1; j < V.length; j++) { |
|  | net += V[j] \* fiCalculation(X, Cji, j - 1) |
|  | } |
|  | net += V[0] |
|  | return net |
|  | } |
|  |  |
|  | const hammingDistance = (teacher, y) => { // считаем расстояние Хэмминга |
|  | let Error = 0 |
|  | for (let i = 0; i < teacher.length; i++) { |
|  | if (teacher[i] !== y[i]) { |
|  | Error++ |
|  | } |
|  | } |
|  | return Error |
|  | } |
|  |  |
|  | const stepsMinVectorNeuralNetwork = (X1, X2, X3, X4, coeff, combination) => { |
|  | let y = 0, |
|  | outputVector = [], |
|  | vectorT = [] |
|  |  |
|  | for (let i = 0; i < combination.length; i++) { |
|  | let X = [X1[combination[i]], |
|  | X2[combination[i]], |
|  | X3[combination[i]], |
|  | X4[combination[i]]] |
|  | let net = calculateNet(Cji, coeff, X) |
|  | //let out = hyperbolicTangentOut(net) |
|  | y = porogovayaFunc(net) |
|  | outputVector.push(y) |
|  | let t = tTeacher[combination[i]] |
|  | vectorT.push(t) |
|  | let beta = t - y |
|  | if (beta !== 0) { |
|  |  |
|  | coeff[0] += learning\_rate \* beta \* fi0 |
|  | for (let i = 1; i < coeff.length; i++) { |
|  | coeff[i] += learning\_rate \* beta \* fiCalculation(X, Cji, i - 1) |
|  | } |
|  |  |
|  | /\* |
|  | coeff[0] += learning\_rate \* beta \* fi0 \* |
|  | (1 / (2 \* Math.pow(((Math.exp(net) + Math.exp(-net)) / 2), 2))) |
|  | for (let i = 1; i < coeff.length; i++) { |
|  | coeff[i] += learning\_rate \* beta \* fiCalculation(X, Cji, i - 1) \* |
|  | (1 / (2 \* Math.pow(((Math.exp(net) + Math.exp(-net)) / 2), 2))) |
|  | } |
|  | \*/ |
|  |  |
|  | } else { |
|  | for (let i = 0; i < coeff.length; i++) |
|  | coeff[i] = coeff[i] |
|  | } |
|  | } |
|  |  |
|  | let countsErrorInCurrentEra = hammingDistance(vectorT, outputVector) |
|  |  |
|  | return { |
|  | coefficients: coeff, |
|  | Error: countsErrorInCurrentEra, |
|  | y: outputVector |
|  | } |
|  |  |
|  | } |
|  |  |
|  | const learningNetworkMinVector = V => { |
|  | let y = 0, |
|  | outputVector = [], |
|  | vectorT = [] |
|  | for (let i = 0; i < 16; i++) { |
|  | let x1 = X1[i], x2 = X2[i], |
|  | x3 = X3[i], x4 = X4[i] |
|  | let net = calculateNet(Cji, V, [x1, x2, x3, x4]) |
|  | //let out = hyperbolicTangentOut(net) |
|  | y = porogovayaFunc(net) |
|  | outputVector.push(y) |
|  | let t = tTeacher[i] |
|  | vectorT.push(t) |
|  | } |
|  |  |
|  | let countsErrorInCurrentEra = hammingDistance(vectorT, outputVector) |
|  |  |
|  | return { |
|  | weights: V, |
|  | Error: countsErrorInCurrentEra, |
|  | y: outputVector |
|  | } |
|  |  |
|  | } |
|  |  |
|  | const eraNeuralNetwork = (coefficient, combination) => { |
|  |  |
|  | let currentEra = {}, |
|  | k = 0; |
|  |  |
|  | while (currentEra.Error !== 0) { |
|  | currentEra = stepsMinVectorNeuralNetwork(X1, X2, X3, X4, coefficient, combination) |
|  | console.log(currentEra, 'эпоха ', k) |
|  | k++ |
|  | if (k > 150) break |
|  | coefficient = currentEra.coefficients |
|  | } |
|  |  |
|  | return { |
|  | coefficient: coefficient, |
|  | combination: combination, |
|  | Epoch\_count: k |
|  | } |
|  | } |
|  |  |
|  | function\* combinationN(array, n) { |
|  | if (n === 1) { |
|  | for (const a of array) { |
|  | yield [a]; |
|  | } |
|  | return; |
|  | } |
|  |  |
|  | for (let i = 0; i <= array.length - n; i++) { |
|  | for (const c of combinationN(array.slice(i + 1), n - 1)) { |
|  | yield [array[i], ...c]; |
|  | } |
|  | } |
|  | } |
|  |  |
|  | function\* combinations(array) { |
|  | for (let i = 1; i <= array.length; i++) { |
|  | yield\* combinationN(array, i); |
|  | } |
|  | } |
|  |  |
|  | let index = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] |
|  |  |
|  | let combinations\_generator = combinations(index) |
|  |  |
|  | let eraSearchVector = {} |
|  | for (let combinations of combinations\_generator) { |
|  | let V\_coeff = [0, 0, 0, 0, 0, 0, 0, 0] |
|  | eraSearchVector = eraNeuralNetwork(V\_coeff, combinations) |
|  | let minVector = learningNetworkMinVector(eraSearchVector.coefficient) |
|  | if (minVector.Error !== 0) continue |
|  | else break |
|  | } |
|  |  |
|  | console.log("найденные синаптические веса: " |
|  | + eraSearchVector.coefficient + "\n", |
|  | "индексы минимальных векторов: " + eraSearchVector.combination + "\n", |
|  | "количество эпох: " + eraSearchVector.Epoch\_count) |