

$$weight_i = \frac{digit_i}{digit_1 + digit_2 + digit_3 + digit_4 + digit_5 + digit_6} \quad (2)$$

where $digit_i$ is the number in position $(9+i)$, $digit_1$ is the number in position $(9+1)$, ..., $digit_6$ is the number in position $(9+6)$. In the case when the last six symbols are zeros, i.e. "000000", the weight will be : $weight_i=1/6$.

All the six rules are connected with conjunction [3] for the selection mechanism:

$$\text{if } \left[\frac{value_1}{1 + weight_1} \geq level_1 \right] \text{ and ... and } \left[\frac{value_6}{1 + weight_6} \geq level_6 \right] \text{ then select} \quad (3)$$

where $level_1$ is the value of a solution considered in the previous cycle, i.e. $level_1=(value_1/(1+weight_1))_{iteration-1}$, $value_1$ is the value that expresses the quality of a solution with regard to *Rule 1*.

From the expressions it follows that the greater the meaning of $weight_i$ is, the higher requirements are imposed on the objects by the i -th rule which is to participate in forming the next generation, whereas if there is generated a small weight, the meaning of $value_i$ might also be small. The experiments have proved that coefficients converge to certain values.

REFERENCES

1. Goldberg D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading/MA, USA: Addison-Wesley.
2. Oliver J. Finding Decision Rules with Genetic Algorithms. (1994). *AI Expert*, March, 33-39.
3. Hwang C-L. and Yoon, K. (1981). *Multiple Attribute Decision Making. Methods and Applications*. Lecture Notes in Economics and Mathematical Systems, 186.

NEIRONU TĪKLA UN LĒMUMU KOKA SALĪDZINĀJUMS IZPLŪDUŠAS INFORMĀCIJAS APSTRĀDĒ

ĒRIKS TIPĀNS

Intelektuālo Datortehnoloģiju Profilinstitūts

Rīgas Tehniskā Universitāte

Kaļķu ielā 1, Rīga LV-1658, Latvija

Tālr.: +371 7089 530

Fakss: +371 782 0094

E-mail: aborisov@egle.cs.rtu.lv

Darba mērķis bija salīdzināt neironu tīkla un lēmumu koka ģenerēšanas algoritmu izplūdušas informācijas apstrādes uzdevumā. Izplūdušas informācijas apstrādei ir izstrādātas vairākas pieejas – gan dažādu arhitektūru neironu tīkli, gan arī dažādi lēmumu koka indukcijas algoritmi.

Savā darbā es izmantoju daudzslāņu tiešās izplatīšanās neironu tīklu ar kļūdas atpakaļizplatīšanās algoritmu, pielāgojot tīkla ieejas un izejas izplūdušu vērtību attēlošanai saskaņā ar [2]-[H. Ishibuchi, H. Okada, H. Tanaka, Interpolation of Fuzzy If-Then Rules by Neural Networks].

Par lēmumu koka indukcijas algoritmu, ar kuru salīdzināt neironu tīkla rezultātus, es izvēlējos [1]-[Yufei Yuan, Michael J. Shaw, Induction of fuzzy decision trees] aprakstīto pieeju.

Risināmais uzdevums

Par risināmā uzdevuma pamatu es ņēmu [1] darbā aprakstīto klasifikācijas problēmu. Tajā runa iet par laika apstākļiem sestdien un par to, kādu sporta veidu lai izvēlas atkarībā no laika apstākļiem.

Laiks var būt aprakstīts ar četriem atribūtiem:

$$A = \{Outlook, Temperature, Humidity, Wind\}$$

un katram atribūtam iespējamās vērtības:

$$Outlook = \{Sunny, Cloudy, Rain\},$$

$$Temperature = \{Cool, Mild, Hot\},$$

$$Humidity = \{Humid, Normal\},$$

$$Wind = \{Windy, Not_windy\}.$$

Savukārt, klases ir sporta veidi, ko varētu izvēlēties nedēļas nogalei (*Plan*):

$$C = \{Swimming, Volleyball, Weight_lifting\}.$$

Tā kā šeit visi atribūti un klases attēlo cilvēka uztveri un vēlēšanos, tie ir neskaidri (izplūduši). Piemēram, cilvēka sajūtas par aukstu, remdenu vai karstu ir neskaidra un starp tām nav noteiktu stingru robežu. Lai gan no temperatūras neskaidrības var izvairīties ar skaitlisku mērīšanu, likums, kas ieviests lēmumu kokā, var saturēt mākslīgu stingru robežu, piemēram: "IF *temperature* ≥ 20 °C THEN *swimming*". Bet kā rīkoties gadījumā, ja temperatūra ir 19 °C? Vai personai pilnīgi noteikti nav jāiet peldēties? Acīmredzams, ka mākslīga stingra robeža šeit nav vēlama. Lai gan neskaidrība starp *swimming* vai *volleyball* varētu arī nebūt, tomēr klasifikācija, ja tā tiek interpretēta kā cilvēka vēlme, var joprojām būt neskaidra. Piemēram, laiks var būt perfekts vai arī vienkārši samērā labs volejbolam. Klasifikācijas nenoteiktība (divdomība) arī var parādīties. Piemēram, laiks varētu būt ļoti labs gan peldēšanai, gan arī volejbolam, un cilvēkam varētu būt grūti izvēlēties.

Zemāk redzamajā 1.tabulā ir neironu tīkla apmācībā un lēmumu koka ģenerēšanā izmantotie dati. Tās ir atribūtu $\{Outlook, Temperature, Humidity, Wind\}$ piederības funkciju vērtības, kā arī klases (*Plan*) piederības funkciju vērtības.

Izplūduša lēmumu koka indukcijas algoritms

Izplūduša lēmumu koka indukcijas process sastāv no sekojošiem soļiem:

1. Apmācības datu fuzifikācija;
2. Izplūdušā lēmumu koka inducēšana;
3. Lēmumu koka pārveidošana likumu kopā;
4. Izplūdušo likumu pielietošana klasifikācijā.

1. Apmācības datu fuzifikācija

Apmācības datu fuzifikācija ir konceptualizācijas process, ko cilvēki lieto, lai samazinātu lēmuma veikšanas procesā izmantotās informācijas pārāk lielo daudzumu. Piemēram, skaitliskās vērtības tiek aizvietotas ar lingvistiskiem terminiem, kā *daudz*, *vidēji*, *maz* utt. To piederības funkcijas var tikt aptuveni noteiktas balstoties uz ekspertu pieredzi, vai cilvēka vispārējo uztveri. Alternatīvi, piederības funkcijas var tikt iegūtas

no statistikas datiem. Arī izplūdušī klasterizācija, kas balstās uz pašapmācību, var tikt izmantota piederības funkciju noteikšanai.

1. tabula

Neironu tīkla apmācībā un lēmumu koka ģenerēšanā izmantotie dati

Case	Outlook			Temp.			Humidity		Wind		Plan		
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not windy	Volleyb.	Swimm.	W_lift.
1	0.9	0.1	0.0	1.0	0.0	0.0	0.8	0.2	0.4	0.6	0.0	0.8	0.2
2	0.8	0.2	0.0	0.6	0.4	0.0	0.0	1.0	0.0	1.0	1.0	0.7	0.0
3	0.0	0.7	0.3	0.8	0.2	0.0	0.1	0.9	0.2	0.8	0.3	0.6	0.1
4	0.2	0.7	0.1	0.3	0.7	0.0	0.2	0.8	0.3	0.7	0.9	0.1	0.0
5	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5	0.0	0.0	1.0
6	0.0	0.7	0.3	0.0	0.3	0.7	0.7	0.3	0.4	0.6	0.2	0.0	0.8
7	0.0	0.3	0.7	0.0	0.0	1.0	0.0	1.0	0.1	0.9	0.0	0.0	1.0
8	0.0	1.0	0.0	0.0	0.2	0.8	0.2	0.8	0.0	1.0	0.7	0.0	0.3
9	1.0	0.0	0.0	1.0	0.0	0.0	0.6	0.4	0.7	0.3	0.2	0.8	0.0
10	0.9	0.1	0.0	0.0	0.3	0.7	0.0	1.0	0.9	0.1	0.0	0.3	0.7
11	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.2	0.8	0.4	0.7	0.0
12	0.2	0.6	0.2	0.0	1.0	0.0	0.3	0.7	0.3	0.7	0.7	0.2	0.1
13	0.9	0.1	0.0	0.2	0.8	0.0	0.1	0.9	1.0	0.0	0.0	0.0	1.0
14	0.0	0.9	0.1	0.0	0.9	0.1	0.1	0.9	0.7	0.3	0.0	0.0	1.0
15	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.8	0.2	0.0	0.0	1.0
16	1.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0	0.0	1.0	0.8	0.6	0.0

2. Izplūdušā lēmumu koka inducēšana

Kognitīvās nenoteiktības mēri. Literatūrā tiek ieteikti divi kognitīvās nenoteiktības mēri – nenoteiktības mērs (*vagueness*) E_v , un divdomības mērs (*ambiguity*) E_a .

E_v (*vagueness*) ir izplūdušā koka analogs ID3 algoritmā izmantotajam entropijas mēram. Pieņemsim, ka A apzīmē izplūdušu kopu universā U ar piederības funkciju $\mu_A(u)$ visiem $u \in U$. Ja U ir diskrēta kopa $U = \{u_1, u_2, \dots, u_m\}$ un $\mu_i = \mu_A(u_i)$, tad izplūdušās kopas A nenoteiktība (jeb *fuzziness*) tiek definēta kā:

$$E_v(A) = -\frac{1}{m} \sum_{i=1}^m (\mu_i \ln \mu_i + (1 - \mu_i) \ln(1 - \mu_i)).$$

Tā kā visi lingvistiskie termini un klases ir izplūdušas kopas, to nenoteiktību iespējams aprēķināt.

E_a (*ambiguity*). Pieņemsim, ka $\pi = (\pi(x) | x \in X)$ apzīmē normalizētu Y varbūtības sadalījumu uz $X = \{x_1, x_2, \dots, x_n\}$. Varbūtiskais divdomības mērs tiek definēts kā

$$E_a(Y) = g(\pi) = \sum_{i=1}^n (\pi_i^* - \pi_{i+1}^*) \ln i,$$

kur $\pi^* = \{\pi_1^*, \pi_2^*, \dots, \pi_n^*\}$ ir varbūtību sadalījuma $\pi = \{\pi(x_1), \pi(x_2), \dots, \pi(x_n)\}$ permutācija sakārtota tā, ka $\pi_i^* \geq \pi_{i+1}^*$ ja $i = 1, 2, \dots, n$, un $\pi_{n+1} = 0$.

Lai noteiktu divdomību atribūtam A starp tā lingvistiskajiem terminiem $T(A) = \{T_1, \dots, T_S\}$, mēs interpretējam piederības funkcijas

$\{\mu_{T_1}(u_i), \mu_{T_2}(u_i), \dots, \mu_{T_S}(u_i)\}$ kā varbūtības sadalījumu objektam u_i iegūt lingvistisko terminu no terminu kopas $T(A) = \{T_1, \dots, T_S\}$. Lai normalizētu varbūtības sadalījumu, pieņemsim, ka

$$\pi_{T_s}(u_i) = \mu_{T_s}(u_i) / \max_{i \leq j \leq S} \{\mu_{T_j}(u_i)\}, \text{ kur } s = 1, \dots, S.$$

Tādējādi, atribūta A divdomība objektam u_i var tikt aprēķināta kā

$$E_a(A(u_i)) = g(\pi_{T_s}(u_i)).$$

Tad atribūta A divdomība ir:

$$E_a(A) = \frac{1}{m} \sum_{i=1}^m E_a(A(u_i)).$$

Klašu divdomība tiek aprēķināta tāpat kā atribūtiem.

Izplūdušie likumi. Izplūdušajam likumam ir forma "IF A THEN B ", kas definē saistību starp nosacījumu izplūdušo kopu A un secinājumu izplūdušo kopu B . Likuma patiesuma līmenis (*degree of truth*) var tikt mērīta ar A piederību (subsethood) B . Piederība $S(A, B)$:

$$S(A, B) = \frac{M(A \cap B)}{M(A)} = \frac{\sum_{u \in U} \min(\mu_A(u), \mu_B(u))}{\sum_{u \in U} \mu_A(u)}$$

Tā ir izplūdušo kopu analogs Beijesa teorēmai.

Izplūdušī liecība. Klasifikācijas uzdevumā *izplūdušī liecība* (*fuzzy evidence*) ir nosacījumu izplūdušī apakškopa, definēta objektu telpā, kas attēlo lingvistiskās vērtības, ko pieņem viens vai vairāki atribūti. Piemēram, mūsu uzdevumā izplūdušī liecība E varētu būt $Hot \cap Sunny$, kas attēlo nosacījumu, kad "the Temperature is *Hot* and the Outlook is *Sunny*".

Varbūtība klasificēt objektu kā C_i , ja dota izplūdušī liecība E , tiek definēta kā:

$$\pi(C_i | E) = S(E, C_i) / \max_j S(E, C_j),$$

kur $S(E, C_i)$ attēlo klasifikācijas likuma "IF E THEN C_i " patiesuma līmenis, bet $\pi(C|E) = \{\pi(C_i|E), i = 1, \dots, L\}$ ir normalizētais varbūtības sadalījums neizplūdušā telpā $C = \{C_1, \dots, C_L\}$.

Klasifikācijas divdomība ar izplūdušo liecību E tiek definēta kā

$G(E) = g(\pi(C|E))$ un kas tiek mērīta, balstoties uz varbūtības sadalījumu $\pi(C|E)$.

Izplūdušu kopu *izplūdušī atdalīšana* (*fuzzy partition*) tiek definēta sekojoši:

ja ir dota izplūdušī liecība F un izplūdušu liecību kopa $P = \{E_1, \dots, E_k\}$ definēta objektu telpā U , tad P izplūdušī atdalīšana uz F tiek definēta kā

$P|F = \{E_1 \cap F, \dots, E_k \cap F\}$, kur katrs objekts iekš F tiek nodalīts uz E_i ar piederību $\mu_{E_i} \cap_F$. Kad $F = U$, var vienkārši rakstīt $P|U = P$.

Klasifikācijas divdomība ar izplūdušo atdalīšanu $P = \{E_1, \dots, E_k\}$ un izplūdušu liecību F tiek apzīmēta kā $G(P|F)$ un ir vidējais svērtais no klasifikācijas divdomībām katrai atdalījuma apakškopai:

$$G(P|F) = \sum_{i=1}^k w(E_i | F) G(E_i \cap F),$$

kur $G(E_i \cap F)$ ir klasifikācijas divdomība ar izplūdušo liecību $E_i \cap F$, $w(E_i|F)$ ir svars, kurš attēlo relatīvo apakškopas $E_i \cap F$ izmēru iekš F .

$$w(E_i | F) = \frac{M(E_i \cap F)}{\sum_{j=1}^k M(E_j \cap F)}$$

Pārklāšanās starp lingvistiskajiem terminiem var izraisīt lielu klasifikācijas divdomību. Lai samazinātu divdomību, objekta klasifikācijai jābalstās vairāk uz stiprajām nekā vājajām liecībām. Liecība ir stipra, ja tās piederības funkcijas vērtība

pārsniedz noteiktu **nozīmīguma līmeni**. Ja ir izplūdusi liecība E ar piederību $\mu_E(u)$, tad $E\alpha$ ir izplūdusī liecība ar nozīmīguma līmeni α un piederību:

$$\mu_{E\alpha}(u) = \begin{cases} \mu_E(u), & \text{ja } \mu_E(u) \geq \alpha, \\ 0, & \text{ja } \mu_E(u) < \alpha \end{cases}$$

Dotajai izplūdušo liecību kopai $P = \{E_1, \dots, E_k\}$ izplūdusī kopa F tiek sadalīta nozīmīguma līmenī α kā:

$P\alpha|F\alpha = \{E\alpha_1 \cap F\alpha, \dots, E\alpha_k \cap F\alpha\}$, kur $E\alpha_i$ ir liecība E_i nozīmīguma līmenī α un $F\alpha$ ir liecība F nozīmīguma līmenī α . Nozīmīguma līmenis α nodrošina filtru lai samazinātu divdomību sadalījumā. Lēmumu koka indukcijas procesā vēl tiek ieviests likuma patiesuma līmeņa sliekšnis β .

Izplūdušā lēmumu koka inducēšana notiek sekojoši

Solis 1: Noteikt klasifikācijas divdomību katram atribūtam un atribūtu ar vismazāko divdomību izvēlēties par lēmumu koka sakni.

Solis 2: Izdzēst visus lēmuma mezgla tukšos zarus. Katram lēmuma mezgla netukšam zaram izrēķināt visu objektu klasificēšanas patiesuma līmeni katrā klasē. Ja patiesuma līmenis, klasificējot vienā klasē, pārsniedz doto sliekšni β , nobeigt zaru kā lapu (klasi). Citādi, izsekot, vai papildus atribūts turpinās sadalīt zaru (t.i., vai tas ģenerēs vairāk kā vienu netukšu zaru) un samazinās klasifikācijas divdomību. Ja jā, tad izvēlēties atribūtu ar vismazāko klasifikācijas divdomību par jaunu lēmuma mezglu uz šā zara. Ja nē, nobeigt zaru kā lapu (klasi). Uz lapas visi objekti tiks ieskaitīti vienā klasē ar augstāko patiesuma līmeni.

Solis 3: Atkārtot soli 2 visiem jaunajiem ģenerētajiem lēmuma mezgliem līdz tālāka koka augšana nebūs iespējama. Tad lēmumu koks ir pabeigts.

Augstāk minētie soļi tiek veikti, ņemot vērā nozīmīguma līmeni α . Objekts pieder zaram tikai tad, ja atbilstošā piederība ir lielāka par α . Divdomības mērs arī tiek aprēķināts nozīmīguma līmenī α . Parametram α ir ļoti nozīmīga loma, filtrējot nenozīmīgas liecības un tādējādi novēršot nenozīmīgus zarus un lapas.

Patiesuma līmeņa sliekšnis β kontrolē koka augšanu. Zemāks β varētu vest pie mazāka koka, bet ar zemāku klasifikācijas precizitāti. Tomēr, kad β tiek palielināts līdz noteiktam punktam, papildus precizitātes pieaugums vairs netiek iegūts. α un β izvēle ir atkarīga no konkrētā gadījuma.

Inducējot koku pēc augstāk aprakstītā uzdevuma datiem ar $\alpha = 0.5$ un $\beta = 0.7$, tiek iegūts sekojošs izplūdis lēmumu koks:

<i>Temperature?</i> ($G(\text{Temperature}) = 0.48$)	
<i>Hot</i> ($G(\text{Hot}) = 0.45$):	<i>Outlook?</i> ($G(\text{Outlook} \text{Hot}) = 0.42$)
	<i>Sunny:</i> <i>Swimming</i> ($S = 0.85$)
	<i>Cloudy:</i> <i>Swimming</i> ($S = 0.72$)
	<i>Rain:</i> <i>Weight_lifting</i> ($S = 0.73$)
<i>Mild</i> ($G(\text{Mild}) = 0.83$):	<i>Wind?</i> ($G(\text{Wind} \text{Mild}) = 0.36$)
	<i>Windy:</i> <i>Weight_lifting</i> ($S = 0.81$)
	<i>Not_windy:</i> <i>Volleyball</i> ($S = 0.78$)
<i>Cool</i> ($G(\text{Cool}) = 0.20$):	<i>Weight_lifting</i> ($S = 0.88$),

kur G ir klasifikācijas divdomības mērs lēmuma mezglā, bet S ir klasifikācijas patiesuma līmenis uz lapas.

3. Lēmumu koka pārveidošana likumu kopā

Katrs ceļš pa zariem no saknes līdz lapai var tikt pārveidots likumā, kur nosacījums reprezentē atribūtus, kas ir ceļā no saknes līdz lapai, bet secinājums reprezentē klasi uz

lapas ar augstāko klasifikācijas patiesuma līmeni. No augstāk redzamā lēmumu koka tiek iegūta sekojoša likumu kopa:

Rule 1: IF <i>Temperature</i> is <i>Hot</i> AND <i>Outlook</i> is <i>Sunny</i> THEN <i>Swimming</i> ($S = 0.85$)
Rule 2: IF <i>Temperature</i> is <i>Hot</i> AND <i>Outlook</i> is <i>Cloudy</i> THEN <i>Swimming</i> ($S = 0.72$)
Rule 3: IF <i>Temperature</i> is <i>Hot</i> AND <i>Outlook</i> is <i>Rain</i> THEN <i>Weight_lifting</i> ($S = 0.73$)
Rule 4: IF <i>Temperature</i> is <i>Mild</i> AND <i>Wind</i> is <i>Windy</i> THEN <i>Weight_lifting</i> ($S = 0.81$)
Rule 5: IF <i>Temperature</i> is <i>Mild</i> AND <i>Wind</i> is <i>Not_windy</i> THEN <i>Volleyball</i> ($S = 0.78$)
Rule 6: IF <i>Temperature</i> is <i>Cool</i> THEN <i>Weight_lifting</i> ($S = 0.88$)
Likums 3 var tikt vienkāršots uz likumu 3':
Rule 3': IF <i>Outlook</i> is <i>Rain</i> THEN <i>Weight_lifting</i> ($S = 0.89$)

Iegūto likumu kopu ir iespējams vienkāršot. Šeit tiek izmantota vienkārša metode. Katru likumu mēģinām vienkāršot, uz laiku novācot pa vienam atribūtam no tā IF daļas. Tad pilnīgai novākšanai izvēlamies atribūtu, kura novākšana dod augstāko vienkāršotā likuma patiesuma līmeni. Ja šā jaunā likuma patiesuma līmenis nav zemāks par sliekšni β , un par oriģinālā likuma patiesuma līmeni, vienkāršošana ir veiksmīga. Process turpinās līdz tālāka vienkāršošana nav iespējama nevienam likumam. Mūsu piemērā 3. likums var tikt vienkāršots uz 3.'. 3.' likuma patiesuma līmenis ir 0.89, kas ir augstāks par oriģinālā 3. likuma patiesuma līmeni 0.73. Citi likumi vairāk nevar tikt vienkāršoti.

4. Izplūdušo likumu pielietošana klasifikācijā

Kad ir iegūta attiecīgā izplūdušo likumu kopa, to var pielietot klasifikācijā. Neizplūdušajā klasifikācijā tikai viens likums tiek pielietots katram konkrētajam objektam. Turpretī izplūdušajā gadījumā, vairāki likumi var tikt pielietoti vienlaicīgi un, tādējādi, objekts var tikt klasificēts dažādās klasēs ar atšķirīgu piederības pakāpi. Klasifikācija notiek, izrēķinot katram objektam nosacījuma piederību balstoties uz tā atribūtiem. Tad secinājuma piederības pakāpe būs vienāda ar nosacījuma piederību.

Iegūtie rezultāti, to salīdzinājums

Lūk, rezultāti, kas iegūti ar izplūdušo lēmumu koku, balstoties uz 1. tabulā esošajiem datiem:

2. tabula

Izplūdušā lēmumu koka apmācības rezultāti

Case	Classification in training data			Classification with learned rules		
	Volleyb.	Swimm.	W_lift.	Volleyb.	Swimm.	W_lift.
1	0.0	0.8	0.2	0.0	0.9	0.0
2	1.0	0.7	0.0	0.4	0.6	0.0 ^a
3	0.3	0.6	0.1	0.2	0.7	0.3
4	0.9	0.1	0.0	0.7	0.3	0.3
5	0.0	0.0	1.0	0.3	0.1	0.9
6	0.2	0.0	0.8	0.3	0.0	0.7
7	0.0	0.0	1.0	0.0	0.0	1.0
8	0.7	0.0	0.3	0.2	0.0	0.8 ^a
9	0.2	0.8	0.0	0.0	1.0	0.0
10	0.0	0.3	0.7	0.1	0.0	0.7
11	0.4	0.7	0.0	0.0	0.7	0.0
12	0.7	0.2	0.1	0.7	0.0	0.3
13	0.0	0.0	1.0	0.0	0.2	0.8
14	0.0	0.0	1.0	0.3	0.0	0.7
15	0.0	0.0	1.0	0.0	0.0	1.0
16	0.8	0.6	0.0	0.5	0.5	0.0 ^b

a - nepareiza klasifikācija

b - nevar izšķirt starp divām vai vairāk klasēm

No rezultātiem ir redzams, ka no 16 apmācības piemēriem 3 gadījumos lēmumu koks deva nepareizu rezultātu, tātad tas kļūdījās 18.75% gadījumu.

Tas pats uzdevums tika risināts, izmantojot daudzslāņu tiešās izplatīšanās neironu tīklu ar kļūdas atpakaļizplatīšanās algoritmu, pielāgojot tīkla ieejas un izejas izplūdušu vērtību attēlošanai saskaņā ar [2]. Apmācot neironu tīklu ar to pašu piemēru kopu, neironu tīkls spēja 100% gadījumos pareizi klasificēt visus apmācošajā kopā ietilpstošos piemērus, tātad kļūda bija 0% gadījumu.

Tādēļ ar neironu tīklu tika veikti papildus eksperimenti, izslēdzot no apmācošās kopas dažus piemērus un pēc tam mēģinot tos klasificēt. Jauktā secībā tika izslēgti gan 2, gan 4, gan 6 piemēri un tika novērtēta klasifikācijas kvalitāte. Jāsaka, ka tā bija stipri atkarīga no neironu tīkla parametriem – neironu skaita slēptajā slāni (tika izmēģināti no 1 līdz 6), apmācības koeficienta β , apmācības procesa iterāciju skaita un uzdotā kļūdas līmeņa. Tādēļ katrā izmēģinājumā rezultāti bija atšķirīgi un tos ir praktiski neiespējams sistematizēt tabulas vai grafika veidā. Tomēr jāsaka, ka vidējais kļūdišanās procents, klasificējot piemērus, kas nebija iekļauti apmācošajā kopā sastādīja apmēram 50%.

Secinājumi

Darba mērķis bija salīdzināt neironu tīkla un lēmumu koka ģenerēšanas algoritmu izplūdušas informācijas apstrādes uzdevumā.

Darbā iegūtie rezultāti liecina, ka, testējot apmācītu sistēmu ar apmācošajā piemēru kopā esošajiem piemēriem, neironu tīkls dod labākus rezultātus (kļūda 0%), turpretī lēmumu koks kļūdījās 18.75% gadījumu.

Testējot ar piemēriem, kas neietilpa apmācošajā kopā, neironu tīkls kļūdījās aptuveni 50% gadījumu, kas ir ļoti augsts kļūdas līmenis, kas praktiski liedz izmantot sistēmu ar šādu arhitektūru reālu uzdevumu risināšanā. Tomēr jāsaka, ka ir izstrādātas vairākas neironu tīklu arhitektūras, kas ievērojami labāk apstrādā izplūdušus datus. Dažas no tām ir apskatītas [3] - Detlef Nauck, Neuro-fuzzy systems: review and prospects, EUFIT '97, September 8 – 11, 1997., pp. 1044 – 1053.

LITERATŪRA

1. Yufei Yuan, Michael J. Shaw. Induction of fuzzy decision trees – from Fuzzy Sets and Systems 69 (1995) pp. 125 – 139
2. Ishibuchi H., Okada H., Tanaka H. Interpolation of Fuzzy If-Then Rules by Neural Networks – from Proceedings of the 2nd International Conference on Fuzzy Logic & Neural Networks (Iizuka, Japan, July 17-22, 1992) pp. 227 - 340
3. Detlef Nauck, Neuro-fuzzy systems: review and prospects, EUFIT '97, September 8 – 11, 1997 pp. 1044 – 1053
4. Alexander I. and Morton H. An introduction to Neural Computing – Chapman and Hall, 1990.
5. Горбань А. Н., Россиев Д. А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. – 276 с.
6. Kevin Knight. Connectionist Ideas and Algorithms – from Communications of the ACM, Vol. 33, No. 11. pp. 59 – 74.
7. Нильсон Н. Обучающиеся машины - М.: Мир, 1967. – 88 - 104с.