Notes on Intuitionistic Fuzzy Sets
Print ISSN 1310–4926, Online ISSN 2367–8283
Vol. 26, 2020, No. 4, 80–89
DOI: 10.7546/nifs.2020.26.4.80-89

A generalized net model of the stochastic gradient descent and dropout algorithm with intuitionistic fuzzy evaluations

Plamena Yovcheva[1] and Sotir Sotirov[2]

[1] "Prof. Dr. Assen Zlatarov" University
1 "Prof. Yakimov" Blvd., Burgas 8010, Bulgaria
e-mail: plamena.iovcheva@abv.bg

[2] "Prof. Dr. Assen Zlatarov" University
1 "Prof. Yakimov" Blvd., Burgas 8010, Bulgaria
e-mail: ssotirov@btu.bg

Received: 7 August 2020 **Revised:** 1 December 2020 **Accepted:** 10 December 2020

Abstract: In the paper, we consider a stochastic gradient descent algorithm in combination with a dropout method. We used the theory of intuitionistic fuzzy sets for the assessment of the equivalence of the respective assessment units. We also consider a degree of uncertainty when the information is not enough.

Keywords: Neural networks, Dropout algorithm, Generalized net, Stochastic gradient descent algorithm, Intuitionistic fuzzy sets.

2010 Mathematics Subject Classification: 68Q85, 03E72.

1 Introduction

Stochastic gradient descent is a very popular and common algorithm used in various machine learning algorithms, the most important being the basis of Neural Networks (NN). Gradient descent is a method of finding a local extremum (minimum or maximum) of a function by moving along the gradient. Dropout [25] works by switching off neurons in a network during training to force the remaining neurons to take on the load of the missing neurons. This is typically done randomly with a certain percentage of neurons per layer being switched off. To find the average weight of each neuron, we use avg_k and avg_k is the average weight input of a neuron on the k-th layer and $W_{jk}^{(i)}$ is the matrix of the weight for the current iteration i before beginning the training and n is the number of neurons in the k-th layer.

$$avg_k = \frac{1}{n}\sum_{j=1}^{n}(|\, W_{jk}^{(i)}|)$$

In the present work, we use the apparatus of intuitionistic fuzzy sets, defined by Atanassov [1, 2] in 1983 as an extension of the theory of fuzzy sets created by L. Zadeh [28].

Let E be a fixed set. The set A^* is called intuitionistic fuzzy set if there is:

$$A=\{\langle x\, \mu_A(x), \nu_A(x)\rangle \mid x \in E\}, \qquad (1)$$

where functions $\mu_A : E \rightarrow [0; 1]$ and $\nu_A : E \rightarrow [0; 1]$, set respectively the degree of membership and non-membership of the elements $x \in E$ to the set A, which is a subset of E and for each $x \in E$:

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1.$$

The function π_A that sets the degree of uncertainty of the membership of the elements $x \in E$ to the set A is determined by the formula:

$$\pi_A(x) = 1 - \mu_A(x) + \nu_A(x)$$

In the case of a fuzzy set $\pi_A(x) = 0$, for each $x \in E$.

The comparison between the elements of any two Intuitionistic fuzzy sets, say A and B, involves a double comparison between the degree of membership and non-membership of the respective elements to the two networks.

In intuitionistic fuzzy logic (IFL) [4, 6], the degree of membership and non-membership can be noted as:

$$\mu_A(x) = \frac{m}{u}, \qquad \nu_A(x) = \frac{n}{u},$$

where m is the lower boundary of the “narrow” range; u – the upper boundary of the “broad” range; n – the upper boundary of the “narrow” range.

1.1 Generalized nets

Generalized nets (GNs) [3, 5, 7] are defined in a way that is principally different from the ways of defining the other types of Petri nets. During the time GN have become a tool for modelling parallel operating systems. Models for neural networks [8, 9] and data mining methods [11–14] have been developed.

The first basic difference between GNs and ordinary Petri nets is the “place – transition” relation. Here the transitions are objects of a more complex nature. A transition may contain m input places and n output places where $m, n \geq 1$.

Formally, every transition is described by a seven-tuple (Fig. 1):

$$Z = \langle L', L'', t_1, t_2, r, M, \square\rangle,$$

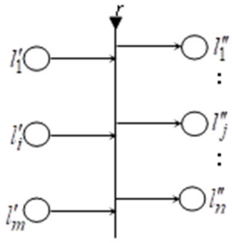

Figure 1. A GN-transition

where:

(a) L' and L'' are finite, non-empty sets of places (the transition's input and output places, respectively). For the transition in Fig. 1 these are

$$L' = \{ l'_1, l'_2, \ldots, l'_m \}, \quad L'' = \{ l''_1, l''_2, \ldots, l''_n \};$$

(b) t_1 is the current time-moment of the transition's firing;

(c) t_2 is the current value of the duration of its active state;

(d) r is the condition of the transition to determine which tokens will pass (or transfer) from the inputs to the outputs of the transition; it has the form of an Index Matrix:

$$r = \begin{array}{c|ccccc} & l''_1 & \ldots & l''_j & \ldots & l''_n \\ \hline l'_1 & & & & & \\ \vdots & & & r_{i,j} & & \\ l'_i & & & & & \\ \vdots & & & (r_{i,j} - \text{predicate}) & & \\ l'_m & & & (1 \leq i \leq m,\ 1 \leq j \leq n) & & \end{array},$$

$r_{i,j}$ is the predicate that corresponds to the i-th input and j-th output place. When its truth value is "true", a token from the i-th input place transfers to the j-th output place; otherwise, this is not possible;

(e) M is an IM of capacities of transition's arcs:

$$M = \begin{array}{c|ccccc} & l''_1 & \ldots & l''_j & \ldots & l''_n \\ \hline l'_1 & & & & & \\ \vdots & & & m_{i,j} & & \\ l'_i & & & (m_{i,j} \geq 0 - \text{natural number}) & & \\ \vdots & & & (1 \leq i \leq m,\ 1 \leq j \leq n) & & \\ l'_m & & & & & \end{array};$$

(f) □ is an object of a form similar to a Boolean expression. It may contain as variables the symbols that serve as labels for a transition's input places, and □ is an expression built up from variables and the Boolean connectives $\wedge$ and $\vee$ and the semantics of which is defined as follows:

$\wedge\ (l_{i_1}, l_{i_2}, ..., l_{i_u})$ – every place $(l_{i_1}, l_{i_2}, ..., l_{i_u})$ must contain at least one token,

$\vee\ (l_{i_1}, l_{i_2}, ..., l_{i_u})$ – there must be at least one token in all places $(l_{i_1}, l_{i_2}, ..., l_{i_u})$,

where $\{l_{i_1}, l_{i_2}, ..., l_{i_u}\} \subset L'$. When the value of a type (calculated as a Boolean expression) is "true", the transition can become active, otherwise it cannot.

2 Generalized net model

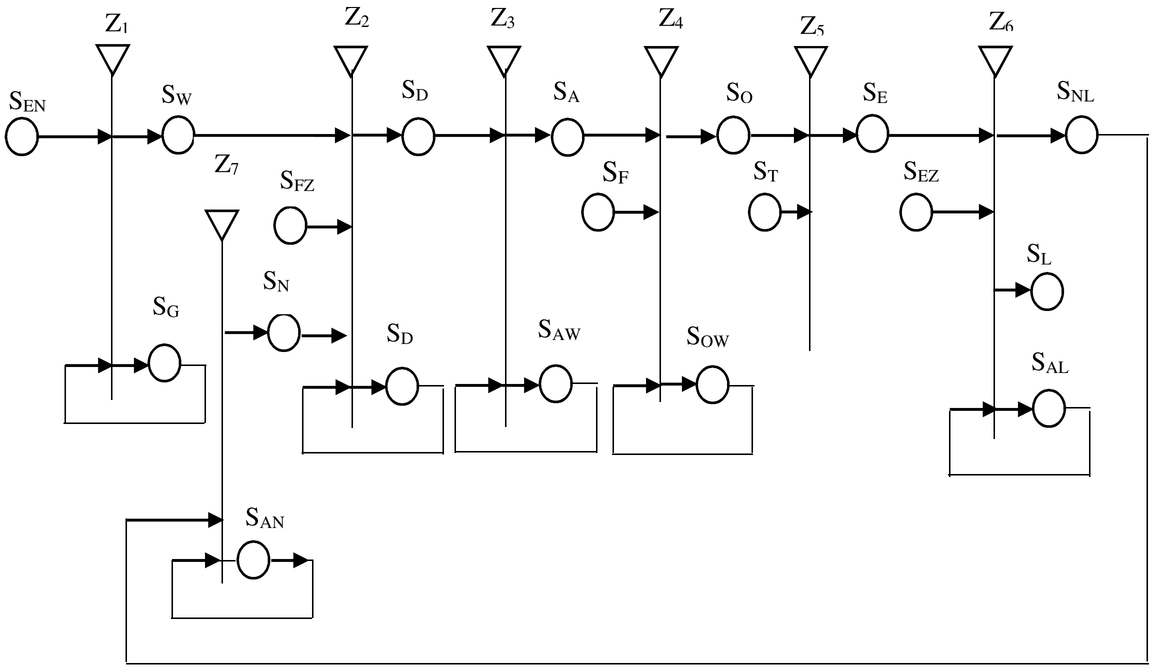

Figure 2. A Generalized net model of the Stochastic Gradient Descent and Dropout Algorithm with intuitionistic fuzzy evaluations

The following tokens stay in the generalized net.

- In place S_G – one α_G token with characteristic "Random number generator" for generalizing weight coefficients.
- In each place S_F one α_i token, $1 \le i \le k$, with the characteristic "Transfer of a function from the i-th layer to the neural network".
- In place S_T – one α_t token with characteristic "Learning objective for neural network output".
- In place S_{EZ} – one α_{ez} token with characteristic "Pre-fixed error in neural network training".

The generalized net includes the following set of seven transitions:

$$A = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7\},$$

where the following events take place:

- Z_1 – generalizing random vector for values of the weight matrix W;
- Z_2 – calculating the avg_K;

- Z_3 – calculating the gradient;
- Z_4 – calculating the outputs $a_k = F_K(n_k)$ from the k-th layer;
- Z_5 – determining the difference between the received value (S_O) and the fixed learning target and the least-square error between them;
- Z_6 – determining whether the artificial neural network (ANN) has been learnt or not;
- Z_7 – calculating the new weight coefficients.

Each of the seven transitions is described below in detail.

Transition Z_1 has the following form:

$$Z_1 = \langle \{S_{EN}, S_G\}, \{S_W, S_G\}, R_1, \vee(S_{EN}, S_G)\rangle,$$

where:

$$R_1 = \begin{array}{c|cc} & S_W & S_G \\ \hline S_{EN} & false & true \\ S_G & W_{G,W} & true \end{array},$$

and $W_{G,W}$ = "Random vector is generated".
At place S_W the token obtains the characteristic "*weight coefficient W*".

Transition Z_2 has the following form:

$$Z_2 = \langle \{S_W, S_{NW}, S_{DW}\}, \{S_D, S_{FZ}, S_{DW}\}, R_2, \vee(\wedge(S_W, S_{NW}), S_{DW}\rangle,$$

where:

$$R_6 = \begin{array}{c|ccc} & S_D & S_{FZ} & S_{DW} \\ \hline S_W & false & false & true \\ S_{NW} & false & false & true \\ S_{DW} & W_{DW,D} & W_{DW,FZ} & true \end{array},$$

and

- $W_{DW,D}$ = "the calculated averages values for W are retained to obtain the outputs from the layers",
- $W_{DW,FZ}$ = "the calculated averages values for W receive an intuitionistic fuzzy estimate and are preserved".

At place S_D the token obtains the characteristic "*average value*".

Transition Z_3 has the following form:

$$Z_3 = \langle \{S_D, S_{AW}\}, \{S_A, S_{AW}\}, R_2, \vee(\wedge(S_D), S_{AW}\rangle,$$

where:

$$R_3 = \begin{array}{c|cc} & S_A & S_{AW} \\ \hline S_D & false & true \\ S_{AW} & W_{AW,A} & true \end{array},$$

and $W_{AW,A}$ = "the calculated averages for W are retained to obtain the outputs from the layers".

At place S_A the token obtains the characteristic *"Output of the NN with input DN and weight coefficient W"*.

Transition Z_4 has the following form:

$$Z_4 = \langle \{S_A, S_F, S_{OW}\}, \{S_O, S_{OW}\}, R_4, \vee(\wedge(S_A, S_F), S_{OW})\rangle,$$

where:

$$R_4 = \begin{array}{c|cc} & S_O & S_{OW} \\ \hline S_A & false & true \\ S_F & false & true \\ S_{OW} & W_{OW,O} & true \end{array},$$

and $W_{OW,O}$ = "The neural layer's output is calculated".

At place S_O the token obtains the characteristic *"Output of the NN with input AN, weight coefficient W and transfer functions F"*.

Transition Z_5 has the following form:

$$Z_5 = \langle \{S_O, S_T\}, \{S_E\}, R_5, \wedge(S_O, S_T)\rangle,$$

where:

$$R_5 = \begin{array}{c|c} & S_E \\ \hline S_O & true \\ S_T & true \end{array},$$

At place S_E the token obtains the characteristic *"The value of the least square error in the network's learning"*.

Transition Z_6 has the following form:

$$Z_6 = \langle \{S_E, S_{EZ}, S_{AL}\}, \{S_{NL}, S_L, S_{AL}\}, R_6, \wedge(S_E, S_{EZ}, S_{AL})\rangle,$$

where:

$$R_6 = \begin{array}{c|ccc} & S_{NL} & S_L & S_{AL} \\ \hline S_E & false & false & true \\ S_{EZ} & false & false & true \\ S_{AL} & W_{AL,NL} & W_{AL,L} & true \end{array},$$

and

- $W_{AL,NL}$ = "The NN is not learnt enough",
- $W_{AL,L}$ = " The NN is learnt".

At place S_{NL} the token obtains the characteristic: *"The value of the received error for recalculating the weight coefficients"*.

Transition Z_7 has the following form:

$$Z_7 = \langle \{S_{NL}, S_{ANW}\}, \{S_{NW}, S_{ANW}\}, R_7, \wedge(S_{NL}, S_{ANW})\rangle,$$

where:

$$R_7 = \begin{array}{c|cc} & S_{NW} & S_{ANW} \\ \hline S_{NL} & false & true \\ S_{ANW} & W_{ANW,NW} & true \end{array},$$

and

$W_{ANW,NW}$ = "$W_{(n+1)}$ is calculated with the previous values of $W_{(n)}$ from the archives".

- total number of neurons – e;
- number of set values for all neurons in layers – s;
- number of total values of neurons in the layer – n;
- number of neurons whose value is greater than the average value for the layer – m;
- number of neurons whose value is less than the average value for the layer – f.

Initially, we calculate the average value for the layer,

$$S_{avg} = \frac{1}{pk} \sum_{\substack{i=1 \\ j=1}}^{\substack{i=p \\ j=k}} W_{ij}.$$

We obtain S_{avgneg}, in case when $S_{avg} > W_{ij}$, we obtain the degree of membership having the following form:

$$\mu_{layer} = \frac{S_{avgneg}}{n}.$$

We obtain S_{avgpos}, in case when $S_{avg} < W_{ij}$, we obtain the degree of non-membership having the following form:

$$\nu_{layer} = \frac{S_{avgpos}}{n}.$$

We obtain $S_{avgequal}$, in case when $S_{avg} = W_{ij}$, we obtain the uncertainty:

$$\pi_{layer} = \frac{S_{avgequal}}{n}.$$

The following new values can be obtained:

$$V_{strong\_opt} = \langle \mu_{A1}(x) + \mu_{A2}(x) + \mu_{A3}(x) + \ldots + \mu_n(x) \\ - \mu_{A1}(x)\mu_{A2}(x) - \mu_{A1}(x)\mu_{A3}(x) - \mu_{A2}(x)\mu_{A3}(x) - \ldots - \mu_{An-1}(x)\mu_{An}(x) \\ + \ldots + \mu_{A1}(x)\mu_{A2}(x)\mu_{A3}(x)\ldots\mu_n(x),\ \nu_{A1}(x)\nu_{A2}(x)\nu_{A3}(x)\ldots\nu_n(x)\rangle$$

$$V_{opt} = \langle \max(\mu_{A1}(x)\mu_{A2}(x)\mu_{A3}(x)\ldots\mu_n(x)),\ \min(\nu_{A1}(x)\nu_{A2}(x)\nu_{A3}(x)\ldots\nu_n(x))\rangle$$

$$V_{avg} = \langle (\mu_{A1}(x) + \mu_{A2}(x) + \mu_{A3}(x) + \ldots + \mu_n(x)) / n, (\nu_{A1}(x) + \nu_{A2}(x) + \nu_{A3}(x) + \ldots + \nu_n(x)) / n\rangle$$

$$V_{pes} = \langle \min(\mu_{A1}(x)\mu_{A2}(x)\mu_{A3}(x)\ldots\mu_n(x)), \max(\nu_{A1}(x)\nu_{A2}(x)\nu_{A3}(x)\ldots\nu_n(x))\rangle$$

$$V_{strong\_pes} = \langle \mu_{A1}(x)\mu_{A2}(x)\mu_{A3}(x)\ldots\mu_n(x),\ \nu_{A1}(x) + \nu_{A2}(x) + \nu_{A3}(x) + \ldots + \nu_n(x) \\ - \nu_{A1}(x)\nu_{A2}(x) - \nu_{A1}(x)\nu_{A3}(x) - \nu_{A2}(x)\nu_{A3}(x) - \ldots - \nu_{An-1}(x)\nu_{An}(x) \\ + \ldots + \nu_{A1}(x)\nu_{A2}(x)\nu_{A3}(x)\ldots\nu_n(x)\rangle.$$

3 Conclusions

A new generalized net model, simulation of the neural network learning process combining the Dropout Method and Stochastic Gradient Descent are considered. The model makes it possible to consider the different stages in the training of the neural network. An estimation with intuitionistic fuzzy sets is used. The intuitionistic fuzzy evaluations reflect the results of the system. A degree of uncertainty is also considered in case of insufficient information. A generalized net model is used to describe the whole process.

References

[1] Atanassov, K. (1983). Intuitionistic fuzzy sets. *Proc. of VII ITKR's Session*, Sofia, June (in Bulgarian).

[2] Atanassov, K. (1986). Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, 20(1), 87–96.

[3] Atanassov, K. (1991). *Generalized nets*. World Scientific, Singapore, New Jersey, London.

[4] Atanassov, K. (1999). *Intuitionistic Fuzzy Sets*. Springer, Heidelberg.

[5] Atanassov, K. (2007). *On Generalized Nets Theory*. “Prof. Marin Drinov” Academic Publishing House, Sofia.

[6] Atanassov, K. (2012). *On Intuitionistic Fuzzy Sets Theory*. Springer, Berlin.

[7] Atanassov, K. (2016). Generalized Nets as a Tool for the Modelling of Data Mining Processes. *Innovative Issues in Intelligent Systems, Vol. 623, Studies in Computational Intelligence*, 161–215.

[8] Atanassov, K., & Sotirov, S. (2006). Optimization of a neural network of self-organizing maps type with time-limits by a generalized net. *Advanced studies in Contemporary Mathematics*, 13(2), 213–220.

[9] Atanassov, K., Sotirov, S., & Antonov, A. (2207). Generalized net model for parallel optimization of feed-forward neural network. *Advanced studies in Contemporary Mathematics,* 15(1), 109–119.

[10] Barrow, E., Eastwood, M., & Jayne, Ch. (2016). Selective Dropout for Deep Neural Networks. *Neural Information Processing*, 519–528.

[11] Bureva, V. (2014). Intuitionistic fuzzy histograms in grid-based clustering. *Notes Intuitionistic Fuzzy Sets*, 20(1), 55–62.

[12] Bureva V., Sotirova, E., & Atanassov, K. (2014). Hierarchical generalized net model of the process of clustering. *Issues in Intuitionistic Fuzzy Sets and Generalized Nets*, Vol. 1, Warsaw School of Information Technology, 73–80.

[13] Bureva V., Sotirova, E., & Atanassov, K. (2014). Hierarchical generalized net model of the process of selecting a method for clustering. *15th Int. Workshop on Generalized Nets*, Burgas, 16 October, 39–48.

[14] Bureva, V., Sotirova, E., & Chountas, P. (2015). Generalized Net of the Process of Sequential Pattern Mining by Generalized Sequential Pattern Algorithm (GSP). *Intelligent Systems'2014*, Springer, Cham, 2015, 831–838.

[15] Fukushima, K. (2005). Restoring partly occluded patterns: a neural network model, *Neural Networks*, 18(1), 33–43.

[16] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*, The MIT Press.

[17] Hagan, M., Demuth, H., & Beale, M. (2010). Neural Network Toolbox 7.

[18] Krawczak, M. (2003). Generalized Net Models of Systems. *Bulletin of Polish Academy of Science*.

[19] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *In Advances in Neural Information Processing Systems,* 25, 1106–1114.

[20] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature* 521.7553: 436.

[21] Sotirov, S. (2003). Modeling the algorithm Backpropagation for training of neural networks with generalized nets – part 1. *Proceedings of the Fourth International Workshop on Generalized Nets*, Sofia, 23 September 2003, 61–67.

[22] Sotirov, S. (2006). Generalized net model of the accelerating backpropagation algorithm. *Proceefings of the Jangjeon Mathematical Society*, 2006, 217–225.

[23] Sotirov, S. (2010). Generalized net model of the Time Delay Neural Network. *Issues in Intuitionistic Fuzzy Sets and Generalized nets*, Warsaw, 125–131.

[24] Sotirov, S., & Krawczak, M. (2007). Modeling the algorithm Backpropagation for learning of neural networks with generalized nets – Part 2. *Issues in Intuitionistic Fuzzy Sets and Generalized nets*, Warszawa, 2007, 65–70.

[25] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.

[26] Torralba, A., Fergus, R., & Weiss, Y. (2008). Small codes and large databases for recognition. *In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'08)*, 1–8.

[27] Tsuruoka, Y., Tsujii, J., & Ananiadou, S. (2009). Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. *Proceedings of the AFNLP/ACL '09.*

[28] Zadeh, L. A. (19 The following new values can be obtained:65). *Fuzzy Sets.* Information and Control, 8, 333–353.