

# Semantic Artificial Neural Networks

Sotirios Batsakis<sup>1,2</sup>(✉), Ilias Tachmazidis<sup>1</sup>, George Baryannis<sup>1</sup>, and Grigoris Antoniou<sup>1</sup>

<sup>1</sup> University of Huddersfield, UK

<sup>2</sup> Technical University of Crete, Greece

{s.batsakis,i.tachmazidis,g.bargiannis,g.antoniou}@hud.ac.uk

**Abstract.** Neural networks have achieved in recent years human level performance in various application domains, including critical applications where accountability is a very important issue, closely related to the interpretability of neural networks and artificial intelligence in general. In this work, an approach for defining the structure of neural networks based on the conceptualisation and semantics of the application domain is proposed. The proposed approach, called Semantic Artificial Neural Networks, allows dealing with the problem of interpretability and also the definition of the structure of neural networks. In addition, the resulting neural networks are sparser and have fewer parameters than typical neural networks, while achieving high performance.

## 1 Introduction

Neural networks have been an important machine learning method for many decades but impressive results in recent years have brought them to the spotlight of artificial intelligence (AI) research and the wider discussion around AI’s impact in society. Since AI is applied on critical domains, interpretability, accountability and legal compliance become significant requirements. Although neural networks achieve impressive performance, they are problematic with respect to these requirements and are commonly characterised as a “black box” approach [2]. Instead, interpretable machine learning approaches can be employed, such as linear and logistic regression, Bayes classifiers and decision trees [3]. These approaches are often efficient but not always as performing as non-interpretable ones, such as Support Vector Machines and neural networks. Extracting interpretation rules from noisy data when employing machine learning is also an alternative approach [4], applied after the creation of a neural network.

In this work, a knowledge graph based approach is proposed for achieving interpretability as in regression methods, while still employing neural networks and thus taking advantage of related work and advances (e.g. in deep learning) in this very prolific research area. The main idea, presented in detail in Section 2, is to construct the network in such a way that dataset features correspond to inputs and outputs, and the nodes of hidden layers correspond to concepts associated with inputs and outputs in a domain conceptualisation represented in graph form.

---

**Algorithm 1** Semantic Artificial Neural Networks Construction

---

**Require:** Dataset  $D$ ,  
**Require:** Ontology (Domain Conceptualization)  $O$

- 1: Create empty Neural Network Graph  $G$
- 2: **for all** output features  $o_j \in D$  **do**
- 3:   Map  $o_j \in D$  to concepts or attributes  $c_j^o \in O$
- 4:   Add corresponding  $c_j^o \in O$  into  $G$
- 5: **end for**
- 6: **for all** input features  $i_j \in D$  **do**
- 7:   Map  $i_j \in D$  to concepts or attributes  $c_j^i \in O$
- 8:   Add corresponding  $c_j^i \in O$  into  $G$
- 9: **end for**
- 10: **while**  $\exists c_j^o \in G$  not connected to  $c_j^i \in G$  **do**
- 11:   **for all** nodes  $c_j \in G$  **do**
- 12:     Find concept(s)  $c_k \in O$  connected to node(s)  $c_j \in G$
- 13:     Add node(s)  $c_k$  in  $G$
- 14:     Add arc(s) connecting  $c_j, c_k$
- 15:   **end for**
- 16: **end while**
- 17: **return** Graph  $G$

---

## 2 Semantic Artificial Neural Networks

The proposed method is based on the construction of a neural network by mapping its structure to an existing or purposely created conceptualisation or ontology in graph form, containing definitions of inputs and outputs of the neural network. Dataset features that correspond to the nodes of the input and output layers of the neural network are mapped to ontology concepts and attributes. Additional concepts within the ontology form the hidden layers of the neural network. This is based on the observation that both ontologies and neural networks are represented using a graph structure. This allows neural network nodes to be mapped to concepts and their properties in the ontology. If properly constructed, the semantics of nodes in the resulting Semantic Artificial Neural Network (SANN) can be determined directly.

The dataset to analyse and the corresponding domain conceptualisation are necessary in order to construct an SANN. Specifically, the dataset consists of samples and a set of features  $D$ . The conceptualisation or ontology  $O$  consists of a set of nodes  $V$  corresponding to features/attributes and a set of edges  $E$  connecting related concepts/attributes in  $V$ . All features in  $D$  correspond to nodes in  $V$ . Notice that, instead of using existing conceptualisations and ontologies, users can develop task-specific ontologies when defining the structure of the neural network. The SANN construction process is shown in Algorithm 1 and is defined as follows: given a conceptualisation or an ontology graph  $O$ , the properties corresponding to the output features in  $D$  are mapped to an output layer node in the neural network  $G$  (lines 2-5). Input properties in  $D$  correspond to input layer nodes in  $G$  (lines 6-9) and intermediate nodes appearing in the path between input and output in  $G$  form the hidden layers (lines 10-16).

The proposed SANN approach can be applied even when a formal domain ontology is not available. In this case the structure can be defined using methods as in knowledge engineering, based on the stakeholders' understanding and conceptualization of the domain and an analysis of related concepts. This can be achieved by slightly modifying Algorithm 1, specifically lines 3, 7 and 12: instead of mapping dataset features to ontology concepts or attributes, such concepts or attributes have to be defined. We expect that defining the network structure is a simpler task compared to a typical knowledge engineering task, because network inputs and outputs are given in advance in the form of dataset features. Thus, it is only necessary to define concepts connecting or associating inputs with outputs, instead of creating a complete domain conceptualisation.

### 3 Evaluation

In this section the performance of SANNs is compared with that of typical dense neural networks. Several diverse datasets are used for comparison for both classification and regression tasks. Specifically, ten datasets covering diverse domains are used (8 of which are from the UCI machine learning collection [1]), with five datasets used for classification and five for regression. The classification datasets are: UCI diabetes [6]<sup>3</sup>, UCI heart disease<sup>4</sup>, UCI Iris<sup>5</sup>, UCI credit card default<sup>6</sup> and the prostate cancer dataset<sup>7</sup>. For regression the datasets are: UCI Auto-mpg<sup>8</sup>, UCI wine quality<sup>9</sup>, UCI real estate valuation<sup>10</sup>, UCI Istanbul stock exchange<sup>11</sup> and the graduate admissions dataset<sup>12</sup>. Min-max normalization was performed on all input features in the datasets.

We present indicatively the SANN created for the UCI diabetes dataset [6] by combining lexical description of inputs and DBpedia concepts. Note that similar SANNs have been created for all 10 datasets but are not shown here due to space limitations. The UCI diabetes dataset contains the following 8 attributes: number of times pregnant (*preg*), plasma glucose concentration after 2 hours in an oral glucose tolerance test (*plas*), diastolic blood pressure in mm Hg (*pres*), triceps skin fold thickness in mm (*skin*), 2-hour serum insulin in  $\mu\text{U/ml}$  (*insu*), body mass index measured as weight in  $\text{kg}/(\text{height in m})^2$  (*mass*), diabetes pedigree function (*pedi*) and age in years (*age*). The diabetes pedigree function (*pedi*) provides data on diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient. Of these features *preg*, *pedi* and *age* are grouped together as attributes directly associated with the *person* concept in DBpedia. Skin thickness (*skin*) and BMI mass indicators (*mass*) are associated with *anatomy/physiology* of the person's body while *plas*, blood pressure (*pres*) and *insu* are

<sup>3</sup> <https://www.kaggle.com/uciml/pima-indians-diabetes-database>

<sup>4</sup> <https://www.kaggle.com/ronitf/heart-disease-uci>

<sup>5</sup> <https://www.kaggle.com/uciml/iris>

<sup>6</sup> <https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset>

<sup>7</sup> <https://www.kaggle.com/multi8ball/prostate-cancer>

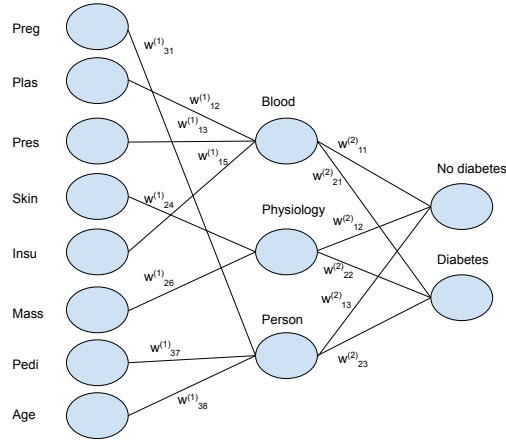
<sup>8</sup> <https://www.kaggle.com/uciml/automp-g-dataset/>

<sup>9</sup> <https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009>

<sup>10</sup> <https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set>

<sup>11</sup> <https://www.kaggle.com/uciml/istanbul-stock-exchange/>

<sup>12</sup> <https://www.kaggle.com/mohansacharya/graduate-admissions>



**Fig. 1.** Semantic Artificial Neural Network for UCI Diabetes dataset - Classification.

attributes of a person’s blood, hence they are grouped together under the label *blood*. DBpedia and Linked Open Data were used for selecting proper concepts and relations when building the Semantic Artificial Neural Network. The resulting network is presented in Figure 1.

After constructing SANNs for all datasets, the Weka machine learning software [5] was used for performance evaluation, based on default hyperparameters. The multilayer perceptron implementation was used for implementing SANNs by modifying the network construction phase. Specifically, instead of automatically selecting a predefined structure (typically fully connected networks) provided by Weka, the structure is defined manually for each dataset using the semantics of the related input/output features. Apart from the definition of the network structure, all other components of the SANN implementation are identical to those of the multilayer perceptron implementation in Weka. In addition, for the experimental evaluation all hyperparameters of SANNs and multilayer perceptrons were identical and set to the default hyperparameters of multilayer perceptrons of Weka. This ensures that the evaluation focuses on the effects of the network topology of SANNs: all other things being equal, determine the effects in performance of the network topology defined using Algorithm 1 and the one produced by Weka.

The average performance over five classification and five regression experiments for SANNs and dense multilayer perceptrons using Weka is reported in Table 1. For classification, the performance metric is accuracy, while for regression, the performance metric is correlation coefficient. The best performing algorithm for a given metric is highlighted in bold.

The experiments indicated that no algorithm was dominant in terms of performance, with dense neural networks slightly outperforming SANNs on average in case of classification and SANNs achieving better average performance in case of regression. Overall, the performance of SANNs compared with fully connected Neural Networks was

**Table 1.** Comparison between SANNs and dense neural networks - average performance for classification (5 datasets) and regression (5 datasets)

Dataset/Metric	Multilayer perceptron	Semantic Artificial Neural Network
Classification/Accuracy	<b>81.94</b>	81.76
Regression/Correlation coefficient	0.659	<b>0.697</b>

comparable if not superior on average. This is a quite positive result considering the additional advantages afforded by SANNs: explainability, reduced number of weights and reduced overall complexity.

## 4 Conclusion

In this work a novel approach for constructing neural networks, called Semantic Artificial Neural Networks, is proposed. The structure of the network reflects the conceptualisation of the application domain by means of a knowledge graph with the objective to create networks that are easy to interpret. Since lack of interpretability is a major issue of neural networks, the proposed approach can be used to create networks where hidden layer nodes correspond to specific concepts and have a specific meaning. In addition, the resulting networks are typically sparse and have fewer parameters, which is typically an advantage during training. Compared with fully connected neural networks with the same number of layers, Semantic Artificial Neural Networks are interpretable, have fewer parameters to train and achieve comparable and, in many cases, better performance.

In future work, we intend to further explore the applicability of SANNs, by also considering the effect of selecting different conceptualisations, which, in its essence, is a knowledge representation problem. For instance, we will compare the different SANNs created based on generic or domain specific ontologies and complex versus minimalist taxonomies.

## References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
2. Castelvechi, D.: Can we open the black box of AI? *Nature News* **538**(7623), 20 (2016)
3. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: A survey. In: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO). pp. 0210–0215. IEEE (2018)
4. Evans, R., Grefenstette, E.: Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research* **61**, 1–64 (2018)
5. Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H., Trigg, L.: Weka-a machine learning workbench for data mining. In: *Data mining and knowledge discovery handbook*, pp. 1269–1277. Springer (2009)
6. Smith, J.W., Everhart, J., Dickson, W., Knowler, W., Johannes, R.: Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In: *Proceedings of the Annual Symposium on Computer Application in Medical Care*. p. 261. American Medical Informatics Association (1988)