Siamese Neural Networks for IUGR Identification in Cardiotocographic Recordings

Giulio Steyde[1], Luca Subitoni[1], Edoardo Spairani[2], Giovanni Magenes[2], Maria Gabriella Signorini[1]

[1]Department of Electronics, Information and Bioengineering , Politecnico di Milano, Milan, Italy
[2]Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Pavia, Italy

Abstract

The Fetal Heart Rate (FHR) is a valuable source of information which is routinely monitored to assess fetal well-being via cardiotocography (CTG). Recently, Siamese Neural Networks (SNN) have been introduced as an effective strategy for metric learning, due to their capability to learn complex embeddings useful for classification with little training data. In this paper, we present preliminary results on the use of SNNs for intra-uterine growth restriction (IUGR) identification in CTG recordings. The developed model comprises three identical residual branches and computes the Euclidean distance among the embeddings of the inputs (i.e., the triplet). The dataset employed is extracted from a wider CTG database and is balanced with respect to the class and gestational age. In particular, 466 recordings are used for training and validation and 92 as a hold-out test set. The trained SNN is then used to project the data into a lower dimensional space where they are classified using a k-Nearest Neighbors classifier. The proposed approach allows to discriminate between IUGR and physiological pregnancies with an accuracy score on the test set ranging from 63% to 72.6% when varying the number of neighbors. The results, which are still far from being satisfactory for clinical applications, align with previous studies, showing the validity of the proposed approach.

1. Introduction

Intrauterine growth restriction (IUGR), defined as “a fetus that fails to reach his potential growth” [1], is one of the most common pregnancy complications, with an incidence of around 10% [2]. IUGR affected fetuses have increased perinatal mortality and morbidity [3]. Early identification of IUGR can help clinicians choose the appropriate clinical intervention and planning delivery, thus reducing adverse outcomes by better balancing the risk of prematurity against that of stillbirth [2].

Cardiotocography (CTG) is a very common exam which is performed both during delivery and in the antepartum period, and it consists in the synchronous acquisition of the fetal heart rate (FHR) and uterine activity. It is well known that the heart rate is regulated by the autonomic nervous system. Therefore, CTG provides an indirect measure of the development and integrity of the fetal nervous system. It has been shown in previous research works that IUGR affects the modulation of the FHR, and therefore CTG can be used to discriminate between healthy and IUGR fetuses [4, 5]. Moreover, it offers complementary information compared to the echographic exam, and it can help in distinguishing actual IUGR from fetuses which are small for their gestational age compared to the population (SGA) but healthy.

In recent years, deep learning models have been used in a variety of medical applications, often showing state of the art performances. Siamese Neural Networks (SNNs) show interesting properties which make them particularly suitable in the analysis of medical data [6]. SNNs project input data in a multidimensional space and learn a similarity function, with the aim of minimizing distance between samples of the same class and maximizing distance between samples of different classes [7]. During training, the network is fed with combinations of data rather than single samples, thus dramatically increasing the number of possible inputs and allowing for effective training even when the number of samples is relatively low. Moreover, this framework is inherently very robust to class imbalance and can be applied in multiclass problems.

1.1. Aim of the work

In this paper, we present a preliminary investigation concerning the use of SNNs in CTG signal classification. In particular, we test the ability of this framework in distinguishing IUGR-affected fetuses from healthy ones. The approach can be easily extended to include more classes.

The code employed is available at:
https://github.com/steyde/SiameseNN_CINC2024

 ISSN: 2325-887X DOI: 10.22489/CinC.2024.280

2. Materials and Methods

2.1. Data Selection and Preparation

The data used in this study come from a comprehensive dataset of CTG records collected between 2013 and 2021 at the University Hospital Federico II in Naples, Italy [8]. From this dataset, we extracted records of healthy and IUGR-affected singleton pregnancies from the 27th to the 40th week of gestation. We considered only IUGR-affected pregnancies that were clinically confirmed at birth, hence excluding SGA fetuses, balancing the number of healthy and IUGR-affected pregnancies for each week of gestation by randomly under-sampling the majority class (i.e., the healthy pregnancies). The decision to balance the classes, while not strictly required for SNN training, was made for simplicity. Additionally, decreasing the number of samples helps reduce the computational training time, which is particularly high for Siamese networks.

The selected portion of the dataset comprised 558 unique CTG recordings. Among these, 309 were used for training, 157 for validation, and 92 as a hold-out test set. Since a single pregnancy can be associated with multiple recordings, to ensure the validity of the results and prevent data leakage, the train, validation and test sets did not share recordings from the same individuals. The signals used to train the SNN included the FHR trace (Sampling Frequency = 2 Hz) and the Fetal Movement Profile (FMP), which consists in a binary signal indicating the time instants in which the fetus is moving. The duration of the recordings ranges from 20 minutes to over 1 hour. Signals were partitioned into 20-minute intervals (2400 samples), with a maximum overlap of 33%.

All data were scaled to facilitate training. The FHR signals were standardized by subtracting the population mean (i.e., the average of the means of all signals in the training set) and then by dividing with the average standard deviation (i.e., the average of the standard deviations of all signals in the training set). This was done to not discard the information regarding the baseline and average variability from the single recordings. The gestational age was scaled to a range between 0 and 1.

2.2. Model Architecture

The developed Siamese network is composed of three identical branches (the embedding models) that are linked together by a layer which computes the Euclidean distances between their outputs.

The embedding model is a Residual Network [9], which was chosen since it has shown good performance in time series classification [10]. It consists of three sequentially connected residual blocks, which process the FHR and FMP signals. The output from these blocks is then concatenated with the gestational age and passed to a multilayer perceptron (MLP). Finally, the MLP's output is normalized using an L2 normalization layer. An overview of the network architecture is depicted in Figure 1.

The input to the first residual block is a 2400 points (20 minutes) long multivariate time series consisting of two channels, one for the FHR signal and one for the FMP. Each residual block consists of three sequential 1D convolutional layers, each one followed by batch normalization and ReLU activation function, and a parallel 1D convolution. The three sequential convolutions within each residual block have a kernel size of 8, 5 and 3, respectively, while the 1D convolution of the parallel branch has kernel size 1. For the selection of the activation function and the kernel size we employed the same values as [10]. A dropout layer is added after the last sequential convolutional layer in each residual block. The value of the dropout was optimized as explained in *section 2.3.3. Hyperparameters Selection*. The MLP has a single hidden layer with size 42, and an output of size 10.

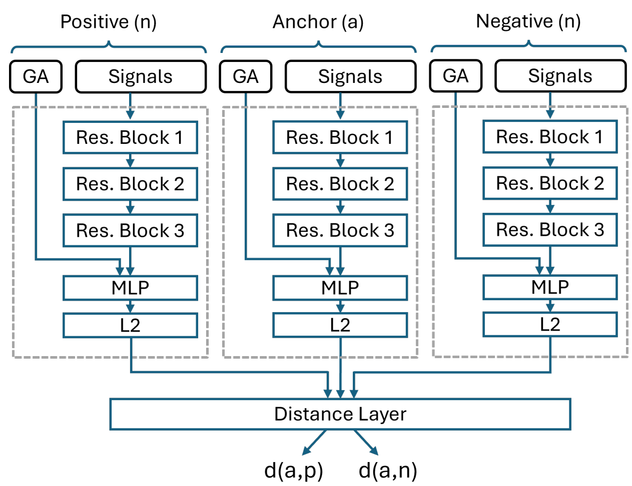

Figure 1. Implemented siamese residual neural network.

2.3. Training Procedure

The training objective is to map observations from the same class close to each other, while mapping those from different classes farther apart. During training, observations are fed to the network as triplets, consisting of an anchor sample, a positive sample from the same class, and a negative sample from a different class. The network's objective function to be minimized, called triplet loss [11], is defined as: $J = \max\{0, d_{AP} - d_{AN} + \mu\}$, where d_{AP} is the difference between the distance from the anchor to the positive samples, and d_{AN} is the distance from the anchor to the negative samples. μ is a margin which is added to promote better separation between positive and negative pairs and was set to 0.2.

2.3.1. Triplet Mining

The number of all possible combinations scales very rapidly with the amount of data. Using all possible combinations during training is computationally very demanding and is not recommended for effective learning [7]. In this work we opted for online mining of informative triplets. During each epoch, the data are divided in batches and for each, all possible triplets inside the batch are generated. However, only the triplets that satisfy a “difficulty” criterion are used in the training step to update the weights. Indeed, triplets that are “too easy” either have no impact on the training process or may even hinder learning. Triplets are selected only if $d_{AP} - d_{AN}$ is bigger than a threshold, which is initialized at the margin value $-\mu$ and is progressively increased up to 0 during training.

We utilized the Adam optimizer and trained the model for 500 epochs. The learning rate was optimized as explained in *section 2.3.3*. At the end of training, the model’s weights are restored to the values they had at the epoch with the best class distance, as will be detailed in the next chapter.

2.3.2. Model Evaluation

To monitor model performance, at the end of each epoch the validation data are projected using the embedding model and their separability in the embedding space is evaluated with the following metric:

$$D = \log_{10}\left(1 + \frac{\text{AvgDiff}}{\text{AvgSame}}\right)$$

where AvgDiff is the average of all the pairwise distances between validation samples belonging to different classes, and AvgSame of samples belonging to the same class. Higher values of D indicate better separability.

2.3.3. Hyperparameters Selection

Performances of machine learning models strongly depend on the choice of appropriate hyperparameters. To address this, various methods and frameworks for automated hyperparameter optimization, such as “Optuna”, have been developed [12]. The high computational time required to train SNNs however prevented us from fully exploiting these frameworks, which include advanced search algorithms. In this work, 20 models have been trained by simple random sampling from the search space reported in Table 1. The network which is selected is the one with the highest D value in validation.

Table 1. Search space of the model hyperparameters.

Hyperparameter	**Minimum**	**Maximum**
Learning Rate	10^{-7}	10^{-4}
N. Feature Maps	32	80
Dropout	0.1	0.7

2.4. Data Embedding: visualization and classification

The embedding model learns to map high-dimensional data into a lower-dimensional Euclidean space where the different classes should be well-separated. In this space the classification task can then be solved by a simple model. Specifically, we trained a k-Nearest Neighbors (k-NN) classifier using the training and validation data projected by the embedding model and applied it to the unseen test data for classification.

During inference, if the time series were longer than the input dimension of the embedding model, multiple patches were extracted with a maximum overlap of 33%, were separately projected and classified by the k-NN classifier. The final label was then obtained by employing a majority voting approach.

The low-dimensional space in which data are projected by the embedding model can then be further reduced to allow to visually inspect the model performance. We employed Principal Component Analysis (PCA) to reduce the dimensionality of the embedding space from 10 to 2, thus allowing to generate scatterplots of the input data. PCA was trained using train and validation data and then was used to transform all data.

3. Results

The highest validation performance was achieved by the model with dropout=0.6, a learning rate of 10^{-5}, and 38 feature maps, resulting in a D value of 0.879. This model was selected for testing. The projections produced by the embedding model for the train and validation sets are shown in panel a) of Figure 2, while the projection of the test set obtained using the same transformation are shown in panel b).

Even though there is a considerable overlap between the two, the samples of the train and validation set belonging to the “healthy” and “IUGR” classes are separated in the embedding space by the trained embedding model. Moreover, it is noticeable that the test samples, which have never been seen by the network, tend to occupy the same portions of the space when projected. Indeed, it becomes possible to separate the two classes using a simple k-NN classifier.

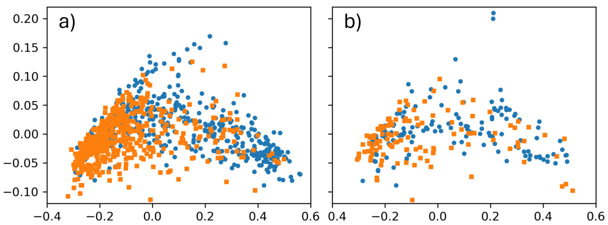

Figure 2. Projections of the embedding space obtained via PCA for a) the train and validation sets and b) the test set. IUGR class in orange; healthy class in blue.

Figure 3 shows the values of the accuracy and of the Area Under the Receiver Operating Characterisic Curve (AUC) on the test set obtained by the trained k-NN. While varying the number of neighbors from 1 to 20, the accuracy values ranged from a minimum of 63% to a maximum of 72.6%, whereas the AUC values ranged from 60.5% to 70.3%.

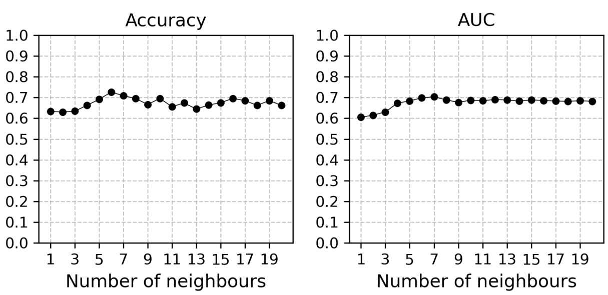

Figure 3. Accuracy and AUC values on the test set while varying the number of neighbors of the k-NN classifier.

4. Discussion and Conclusions

The described methodology showed results that align with those reported by Stroux et al. [13], who used logistic regression on handcrafted FHR features to classify IUGR fetuses in over 2000 cases. However, they are lower than those achieved by Pini et al. [14], where various machine learning algorithms were applied to a smaller dataset of 262 carefully curated cases. To summarize, the presented framework performs substantially better than a random classifier, but the results are still far from being satisfactory and ready for clinical applications.

Future developments, such as conducting a more extensive hyperparameter tuning, exploring different architectures for the embedding model, and utilizing more data in training, could potentially enhance performance to exceed state-of-the-art levels for this application. Moreover, since the presented learning framework is based on similarities among all the training samples, it is inherently very robust to class imbalance. This suggests that in future developments, it could be possible to add more classes, including those representing rare pathologies that would be challenging to address using traditional learning strategies.

References

[1] Galan H, Grobman W. ACOG Practice Bulletin No. 204: Fetal Growth Restriction. Obstet Gynecol Feb. 2019;133.

[2] Martins JG, Biggio JR, Abuhamad A. Society for Maternal-Fetal Medicine Consult Series 52: Diagnosis and management of fetal growth restriction. Am J Obstet Gynecol Oct. 2020;223.

[3] Giampaolo M, et al. Intrauterine restriction (iugr). Journal of Perinatal Medicine Jan. 2008;36.

[4] Ferrario M, Signorini MG, Magenes G. Complexity analysis of the fetal heart rate variability: early identification of severe intrauterine growth-restricted fetuses. Med Biol Eng Comput Sep. 2009;47.

[5] Rescinito R, Ratti M, Payedimarri AB, Panella M. Prediction models for intrauterine growth restriction using artificial intelligence and machine learning: A systematic review and meta-analysis. Healthcare Jun. 2023;11.

[6] Koch G, et al. Siamese neural networks for one-shot image recognition. In Proceedings of the 32nd International Conference on Machine Learning. 2015.

[7] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In 2015 IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[8] Spairani E, Daniele B, Magenes G, Signorini MG. A novel large structured cardiotocographic database. In Annu Int Conf IEEE Eng Med Biol Soc. 2022. Jul. 2022.

[9] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition. Dec. 2016.

[10] Fawaz HI, et al. Deep learning for time series classification: a review. Data Min Knowl Disc Jul. 2019;33.

[11] Dong X, Shen J. Triplet loss in siamese network for object tracking. In Proceedings of the European conference on computer vision (ECCV). Sep. 2018.

[12] Akiba T, et al. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Jul. 2019.

[13] Stroux L, Redman CW, Georgieva A, Payne SJ, Clifford GD. Doppler-based fetal heart rate analysis markers for the detection of early intrauterine growth restriction. Acta Obstet Gynecol Scand Nov. 2017;96.

[14] Pini N, Lucchini M, Esposito G, Tagliaferri S, Campanile M, Magenes G, Signorini MG. A machine learning approach to monitor the emergence of late intrauterine growth restriction. Front Artif Intell Mar. 2021;4.

Address for correspondence:

Giulio Steyde

Via Camillo Golgi, 39, 20133 Milano MI, Italy

giulio.steyde@polimi.it