Simple neuro-fuzzy system with combined learning for pattern recognition under conditions of short training set in medical diagnostics tasks

Yevgeniy Bodyanskiy[a], Olha Chala[b], Ivan Izonin[c] and Sergiy Popov[d]

[a] Institute of Theological Studies of Our Lady of Immaculate Conception, Gorodok, Khemlnytska obl., Ukraine
[b] Artificial intelligence department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
[c] Department of Artificial Intelligence of Lviv Polytechnic National University, Lviv, Ukraine
[d] Control systems research laboratory, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

Abstract

There is a widespread problem in the medical diagnostic tasks of the training datasets deficit. The medical data is hard to clinically collect and process into the ready-to-use dataset for supervised learning leading to difficulties in achieving computer-aided detection and diagnosis. The traditional approaches can works with big enough training datasets, however, thay cannot show their efficiency under conditions of limited number of samples.

We propose a system that combines various learning paradigms and a neuro-fuzzy approach to solve medical classification problems under conditions of a limited number of training observations-images. The distinctive feature of the proposed system is the usage of the "scatter partitioning" of input space, which provides better system performance both in learning and classification. The results of the computer experiment proved the effectiveness of the proposed system in solving image recognition in the medical diagnostic task. The computational experiment showed that the proposed model works better with limited training datasets than the advanced systems, however, the proposed one yields with bigger amount of training observations.

Keywords 1

Neuro-fuzzy system, combined learning, medical diagnostics, short training set, overlapping classes

1. Introduction

The task of image classification-recognition is one of the primary ones in the Data Mining domain. Many approaches to its solution have been developed, and Deep Neural Networks (DNN) [1-3] are quite prominent among them, which demonstrate really impressive results in terms of accuracy, but not without significant drawbacks. First of all, it is the requirements of large volumes of training data, which are not always available in practical situations. At the same time, the use of transfer learning does not always conquer this problem. Secondly, deep neural networks are rather "slow" systems that require a lot of time for their training, hence online training is not possible in this case. Additionally, there are some numerical implementation problems, the "vanishing gradient" problem in the first place. It can be overcome either by employing piecewise linear activation functions, which will lead to the increase of the number of adjustable synaptic weights, or by techniques like "dropout", "shortcut" or "skeeping" that complicate, i.e. increase in time, the processes of synaptic weights adjustment, whose number in modern DNNs is in the order of billions or more.

IDDM'2022: 4rd International Conference on Informatics & Data-Driven Medicine, November 18 - 20, 2022, Lyon, France
EMAIL: yevgeniy.bodyanskiy@nure.ua (A. 1); olha.chala@nure.ua (A. 2), ivan.v.izonin@lpnu.ua (A. 3), serhii.popov@nure.ua (A. 4)
ORCID: 0000-0001-5418-2143 (A. 1); 0000-0002-7603-1247 (A. 2), 0000-0002-9761-0096 (A. 3), 0000-0002-1274-5830 (A. 4)

Neuro-fuzzy systems [4-6], which belong to the class of so-called hybrid systems of computational intelligence, are free of most of the abovementioned drawbacks [5]. These systems have their drawbacks too. NFS with grid partitioning, which from a formal point of view are analogs of radial-basis function neural networks (RBFN) [7, 8], also require significant volumes of training data. In addition, the absolute majority of known NFS solve the problems of approximation-extrapolation, but not classification directly, in contrast to the widely used convolutional neural networks.

It is possible to improve the quality of NFS in pattern recognition tasks by using the combined learning/self-learning of both synaptic weights and membership functions based on objective functions directly related to the classification task, which requires both the modification of the algorithms for system parameters adjustment and the NFS architecture.

2. Architecture of a neuro-fuzzy system with combined learning

Figure 1 shows the architecture of the proposed NFS, which contains five layers of information processing: the first – fuzzification layer, the second – aggregation layer, the third layer of adjustable synaptic weights, the fourth – accumulation layer, and finally, the output layer of softmax activation functions.

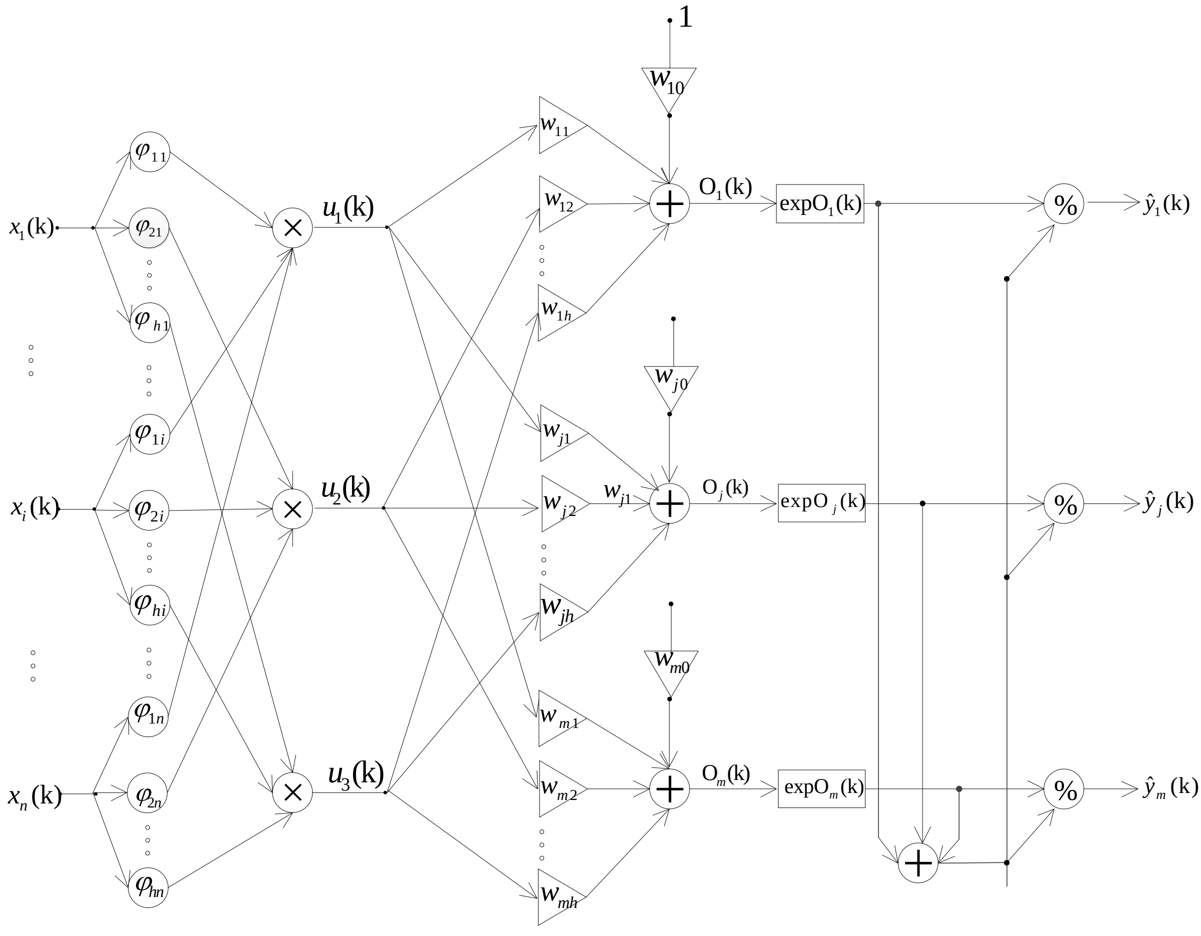


Figure 1: neuro-fuzzy system with combined learning for pattern recognition

The input information for the considered NFS training is a classified set of observations $X = \{x(1), x(2),..., x(k),..., x(N)\}, x(k) = (x_1(k),..., x_i(k),..., x_n(k))^T \in R^n$ (here $k = 1, 2,..., N$ is either the current observation index in the training set, or the current discrete time if training is implemented in online mode). It is assumed that all components $x(k)$ are pre-coded on some interval, usually in fuzzy systems $0 \le x_i(k) \le 1 \forall i = 1, 2,...n; k = 1, 2,..., N$. The input information enters the first layer of the system which is formed by one-dimensional membership functions $\varphi_{li}(x_i, c_{li}, \sigma_{li})$ (here c_{li} – the center of the corresponding function, σ_{li} – parameter determining its width). Usually one-dimensional Gaussian is used for this purpose

$$\varphi_{li}(x_i, c_{li}, \sigma_{li}) = exp\left(-\frac{(x - c_{li})^2}{2\sigma^2}\right), l = 1,2, \dots, h. \tag{1}$$

The width parameter σ is usually chosen to be equal for all functions. The total number of membership functions in the system is hn, i.e. h functions at each input. The centers of these functions are, as a rule, distributed uniformly along the abscissa axis, hence the distance between two adjacent centers is defined as

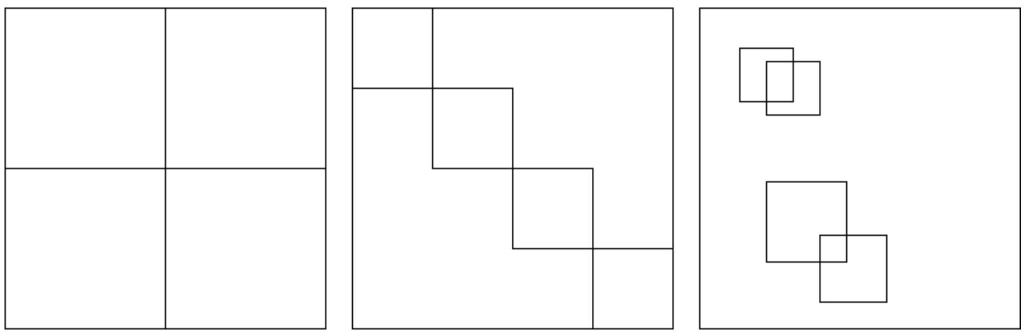
$$\Delta = \frac{1}{h-1}. \tag{2}$$

Signals from the first layer proceed the second hidden layer of aggregation, which is formed by h elementary multiplication blocks yielding the following outputs

$$u_l(k) = \prod_{i=1}^{n} exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = exp\left(-\frac{\|x(k) - c_l\|^2}{2\sigma^2}\right). \tag{3}$$

The first two layers calculate the signals formed at the outputs of many RBFN activation functions, that is, in fact, the first two layers calculate multidimensional RBFN activation functions.

It should be noted that the uniform placement of membership functions along all coordinates leads to the so-called diagonal partitioning (Figure 2(b)) of input space, i.e. the centers of multidimensional Gaussians are located along the diagonals of a unit hypercube. This, in turn, leads to the fact that observations to be classified that are located far from this diagonal will be processed with rather poor accuracy. This undesirable effect can be avoided by using grid partitioning (Figure 2(a)), however, the number of multidimensional Gaussians in the second layer will be h^n leading to the so called "curse of dimensionality". This undesirable effect can be avoided by using the so-called "scatter partitioning" (Figure 2(c)), but in this situation the issue of placing the membership functions along the feature axes remains open.



(a) (b) (c)

Figure 2: (a) Grid partitioning, (b)diagonal partitioning, (c) scatter partitioning

The third hidden layer is formed by $(h+1)m$ adjustable synaptic weights w_{jl} (here $j = 1,2, \dots, m$; m – the number of available classes in the processed dataset, $l = 0,1,2, \dots, h$; w_{j0} – bias signal), which are in the process of learning – the adopted goal function optimization.

The fourth hidden layer is formed by m elementary accumulators that calculate values

$$o_j(k) = w_{j0} + \sum_{l=1}^{h} w_{jl} u_l(k) = w_{j0} + \sum_{l=1}^{h} w_{jl} \prod_{i=1}^{n} \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2} \right) =$$
$$= w_{j0} + \sum_{l=1}^{h} w_{jl} exp\left(-\frac{\|x(k) - c_l\|^2}{2\sigma^2} \right) = w_j^T u(k), \tag{4}$$

where $w_j = (w_{j0}, w_{j1}, ..., w_{jh})^T, u(k) = (1, u_1(k), ..., u_h(k))^T$.

The signals $o_j(k)$ are then fed to the output layer of the system formed by m softmax activation functions that calculate the NFS output signals in the form

$$\hat{y}_j(k) = softmax\, \mathrm{o}_j(k) = \frac{exp o_j(k)}{\sum_{j=1}^{m} exp o_j(k)}. \tag{5}$$

The maximum value of the signal $\hat{y}_j(k)$ determines whether the input observation belongs to a specific class $j = 1, 2, ..., m,$, as well as the level of its fuzzy membership to this class.

3. Combined learning of the neuro-fuzzy system

It is easy to see that the proposed NFS architecture is quite close to the Wang-Mendel system [9], but the main difference is that the Wang-Mendel system is focused on solving approximation problems and is tuned based on the quadratic criterion, and our NFS is focused on solving classification problems, has many outputs, softmax output activation functions, and bases its learning on crossentropy

$$E = -\sum_{k=1}^{N} \sum_{j=1}^{m} y_j(k) \ln \hat{y}_j(k), \tag{6}$$

where $y_j(k)$ – external reference signal that can take only two values 0 or 1, the so called “one hot coding”.

The combined NFS learning is performed in two stages: setting the centers of the membership functions $c_{li}, \forall l = 1, 2, ..., h; i = 1, 2, ..., n$ and tuning of synaptic weights $w_{jl} \forall\, j = 1, 2, ..., m; l = 0, 1, ..., h$.

Setting the centers of membership functions is made according to the principle of “neurons at data points” [10] based on “just in time models” [11]. In the simplest case, when $h = N$, the components of the input signals $c_{li} = x_i(k), k = 1, 2, ..., N = h$ are designated as centers. In case of $h < N$, some indistinguishability threshold δ is introduced and if $\|x(k+1) - x(k)\| < \delta$ the observation $x(k+1)$ is ignored and a new center is not formed.

In such a way, “scatter partitioning” of input space is implemented, i.e. the input space is completely covered by multidimensional activation functions (3). If the recognition quality is insufficient, the threshold δ can be reduced which automatically leads to an increase in the number of membership-activation functions.

A modified optimization procedure of criterion (6) can be used for synaptic weights tuning, which in this case has the form [12]:

$$
\begin{cases}
o_j(k) = w_j^T(k-1)u(k), \\
\hat{y}_j(k) = exp\ o_j(k)\left(\sum_{j=1}^{m} exp\, o_j(k)\right)^{-1}, \\
w_j(k) = w_j(k-1) + r^{-1}(k)(y_j(k) - \hat{y}_j(k)u(k), \\
r(k) = \alpha r(k) - 1 + \|u(k)\|^2, 0 \le \alpha \le 1.
\end{cases}
\tag{7}
$$

where α is a forgetting factor.

It is easy to see that, if $\alpha = 0$, (7) takes the form of the optimal Kaczmarz algorithm [13] and, if $\alpha = 1$, the form of adaptive Goodwin-Ramadge-Caines algorithm [14] for the stochastic objects' online identification. Thus, (7) provides online tuning of NFS synaptic weights. The smaller the value of the forgetting factor α, the higher the learning process convergence rate.

4. Experiment results

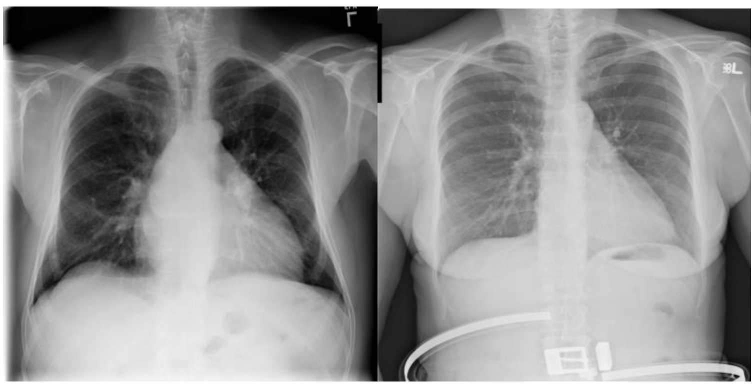
The performance of the proposed neuro-fuzzy system with one-dimensional Gaussian membership function and combined learning was compared to the convolutional neural networks such as LeNet-5 [15] and ResNet-20 [16] with transfer learning and modified classifier. Also, we experimentally tested the effectiveness of the scatter partitioning above the grid and diagonal ones.

The LeNet-5 has three sets of convolution and max pooling, with kernel sizes 5x5 and 2x2 respectively and a ReLU activation function. Additionally, it has two fully connected layers where the first FC layer has a ReLU activation function and the second one SoftMax classifier.

The ResNet-20 was taken as a baseline for the transfer learning, so the original architecture remained, however, as a classifier was chosen the radial-basis support vector machine (RBF-SVM)[17].

4.1. Dataset

One of the most frequent diagnostic-examination image data is Chest X-ray, but X-rays in clinical diagnosis are challenging and occasionally can be much harder to read in comparison to CT scans of the chest. Such data is hard to obtain clinically thus their little amount of datasets online with annotations, so which leads to difficulties in achieving computer-aided detection and diagnosis. So, there are only a few datasets with X-ray images one of which is OpenAI with 4 143 images available and the second one NIH Chest X-ray Dataset [18].



(a) (b)

Figure 2: Observations from the NIH Chest X-ray Dataset labeled as (a) Cardiomegaly, (b) Normal

The second dataset contains 112 120 X-ray images some of which are presented in Figure 2 with various disease labels (Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Normal) from 30 805 unique patients. To label obtained observations, authors processed the associated radiological reports using Natural Language Processing, where accuracy was more than 90% and suitable for supervised learning. The dataset was split on training and validation set in ration 80% and 20%.

4.2. The results analysis

We track the accuracy of each version of the proposed neuro-fuzzy system, which is the original proposed neuro-fuzzy system, the one with the grid partitioning, diagonal partitioning, and the deep neural networks LeNet-5 and ResNet-20 with RBF-SVM classificator. As for the spread-width parameter in the proposed neuro-fuzzy system, was chosen 0,33 using the "3 sigmas" rule. Since, we consider the situation of the training dataset deficit, here to simulate this problem we randomly chose a similar amount of the observations of each class, forming a subset with 10 000 observations, but also we tested these systems on the full dataset. The results of the comparison are shown in Table 1.

Table 1
Comparison analysis on subset and full dataset

System	Accuracy, %	
	Subset	Full dataset
The proposed neuro-fuzzy system	90,26	92,12
Neuro-fuzzy system with grid partitioning	81,301	-
Neuro-fuzzy system with diagonal partitioning	63,75	72,58
LeNet-5	90,57	95,82
ResNet-20 with RBF-SVM	91,34	98,73

As we can see, the maximum accuracy we achieved in convolutional neural networks is comparable to the proposed neuro-fuzzy system. At the same time, the modifications of the neuro-fuzzy system with grid and diagonal partitioning showed poor performance.

Performing grid partitioning, we tried to improve accuracy by changing the similarity parameter, because, if we left this number as it was in the case with scatter partitioning, we ended up with the curse of dimensionality, but increasing it led to a drastic decrease in accuracy. Performing diagonal partitioning we obtained great speed which was 16,43 minutes, however, the accuracy was the worst among competitors. The reason behind that was the clusters of data were scattered through all space of features, leaving only part of it on the main diagonal, so the computed probability of the observations which were located further from the diagonal was calculated poorly.

Advanced systems such as convolutional neural networks always show outstanding accuracy, however, the processing speed is a sore spot. The time consumption of the proposed network, LeNet-5, and ResNet-20 with RBF-SVM is shown in Table 2.

Table 2
Comparison analysis on subset and full dataset

System	Time, min	
	Subset	Full dataset
The proposed neuro-fuzzy system	37,29	152,64
LeNet-5	58,1	192,7
ResNet-20 with RBF-SVM	75,58	458,32

The proposed system, in comparison to the CNNs, showed better speed on both datasets, even though the accuracy was comparable on the subset, however on the full dataset NFS yields in accuracy to the advanced systems. This is conditioned not just by the architecture of the NFS but also by combined learning. It allows, without affecting performance, narrowing down the amount of the training observations, speeding up the image processing, and avoiding the “curse of dimensionality”. Additionally, the scatter partitioning and probability nature of the Gaussian membership function helps calculate membership levers of new observations with decent accuracy.

LeNet-5, in comparison to the ResNet-20 with RBF-SVM, showed drastically better speed on the full dataset, when the first took around 2,5 hours to process images, but the second one more than 7 hours. This can be explained by the fact that LeNet-5 has a “lighter” architecture than Res-Net-20: the autoencoder of LeNet-5 has fewer layers of convolution-pulling, the RBF-SVM classificatory of ResNet-20 provides better accuracy, yet is time-consuming in comparison to typical fully connected layers on LeNet.

Also, we compared these system variations, the proposed one, but with a different activation function, the Epanechnikov one, however, we did not include it in the table due to the poor performance. So, the piece-wise approximation gives relatively good speed but the highest accuracy achieved was 49%. The reason is that in the space of features the “tails” of the Gaussian function is essential to compute probabilities solving classification task under conditions of overlapping classes. Thus the usage of the triangular membership function is inappropriate for these conditions.

5. Conclusion

Architecture and algorithm of combined learning of the neuro-fuzzy system for solving classification problems under the conditions of limited training data are proposed. The system implements “scatter partitioning” of input space, provides a high rate of its parameters tuning and is characterized by simplicity of numerical implementation in comparison with known neuro-fuzzy systems. The simulation results confirm the effectiveness of the proposed system.

6. References

[1] Bengio Y, Cun YL, Hinton G (2015) Deep learning. Nature 521:436–444

[2] Schmidhuber J (2015) Deep learning” in Neural networks: An overview. Neural Networks 61:85–117

[3] Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press

[4] Souza PVC (2020) Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. Applied Soft Computing 92:

[5] Kacprzyk J, Pedrycz W (2015) Springer Handbook of Computational Intelligence. Springer, Verlag, Berlin Heidelberg

[6] S. Osowski, Sieci neuronowe do przetwarzania informacji, Warszawa: Oficijna Wydawnicza Politechniki Warszawskiej, 2006.

[7] J.Moody Cjd (1989) Fast learning in networks of locally tuned processing units. Neural Computation 1:281–294

[8] Poggio T, Girosi F (1990) Networks for approximation and learning”. Proceedings of the IEEE 78:1481–1497

[9] 10. Babuška, R., Verbruggen, H.B. (1997). Constructing Fuzzy Models by Product Space Clustering. In: Hellendoorn, H., Driankov, D. (eds) Fuzzy Model Identification. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-60767-7_2

[10] D.R. Zahirniak, R. Chapman, Rogers, Pattern recognition using radial basis function networks, in: Sixth Annual Aerospace Applications of AI Conf, Dayton, 1990, pp. 249–260.

[11] O. Nelles, Nonlinear Systems Identification. Berlin: Springer, 2001. doi:10.1007/978-3-662-04323-3.

[12] Bodyanskiy YV, Kokshenev I, Kolodyazhniy V (2003) An adaptive learning algorithm for a neo fuzzy neuron. In: Wagenknecht M, Hampel R (eds) Proceedings of the 3rd

Conference of the European Society for Fuzzy Logic and Technology, Zittau, Germany, September 10-12, 2003. University of Applied Sciences at Zittau/Görlitz, Germany, pp 375–379

[13] Kaczmarz S (1993) Approximate solution of systems of linear equations. International Journal of Control 57:1269–1271. https://doi.org/10.1080/00207179308934446

[14] G. Goodwin, P. Ramadge and P. Caines, "Discrete-time multivariable adaptive control," in *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 449-456, June 1980, doi: 10.1109/TAC.1980.1102363.

[15] Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324. https://doi.org/10.1109/5.726791

[16] He K, Zhang X, Ren S, Sun J (2015) Deep Residual Learning for Image Recognition

[17] Apostolidis-Afentoulis V (2015) SVM Classification with Linear and RBF kernels. https://doi.org/10.13140/RG.2.1.3351.4083

[18] Wang X, Peng Y, Lu L, et al (2017) ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu, HI, pp 3462–3471