To appear in *Optimization Methods & Software*
Vol. 00, No. 00, Month 20XX, 1–9

An Analysis of Robust Cost Functions for CNN in Computer-Aided Diagnosis

Adrian Barbu[a*], Le Lu[b], Holger Roth[b], Ari Seff[b] and Ronald M. Summers[b]

[a]*Department of Statistics, Florida State University* [b]*Imaging Biomarkers and Computer-Aided Diagnosis Laboratory, Radiology and Imaging Sciences, National Institutes of Health Clinical Center, Bethesda, MD*

(*November 2015*)

Deep Convolutional Neural Networks (CNN) have proven to be powerful and flexible tools that advance the state-of-the-art in many fields, e.g., speech recognition, computer vision and medical imaging. Usually deep CNN models employ the logistic (soft-max) loss function in the training process of classification tasks. Recent evidence on a computer vision benchmark dataset indicates that the hinge (SVM) loss might give smaller misclassification errors on the test set compared to the logistic loss (i.e. offer better generality). In this paper, we study and compare four different loss functions for deep CNNs in the context of computer-aided abdominal and mediastinal lymph node detection and diagnosis (CAD) using CT images. Besides the logistic loss, we compare three other CNN losses that have not been previously studied for CAD problems. The experiments confirm that the logistic loss performs the worst among the four losses, and an additional 3% increase in detection rate at 3 false positives/volume can be obtained by just replacing it with Lorenz loss. The FROC curves of two of the three loss functions consistently outperform the logistic loss in testing.

1. Introduction

Highly accurate detection of enlarged lymph nodes (LNs) is an important medical imaging task since the size or volume of "swollen" LNs are biomarkers used to stage cancer and measure the progress of cancer treatments, for lung cancer, lymphoma and inflammation. LN size is measured by following the RECIST guideline [14] and a LN is considered enlarged if its smallest diameter is $\geq$ 10mm on an axial CT slice. There are many previous works on building automated computer-aided detection (CADe) systems for LNs using various image features and supervised machine learning techniques. One of the top performing LN CADe systems is described in [13], which employs a five layer convolutional neural network (CNN) with the logistic (soft-max) loss function, trained for binary classification of LN vs. non-LN from a collection of 3D LN candidate subvolumes. For another medical imaging application, a deep CNN model with the soft-max loss also produced the state-of-the-art results for magnetic resonance imaging (MRI) knee cartilage segmentation [12].

There are a number of reasons why one should study different loss functions for convolutional neural networks. First, different losses offer different ways to penalize the easy examples (examples that are correctly classified and far from the decision boundary). Three of the losses studied in this work give zero cost to the easy examples. The ex-

[*] Corresponding author. Email: abarbu@stat.ucla.edu

amples that have zero cost could in principle be removed from training and ways of speeding-up training could be obtained this way, as well as ways to handle large amounts of training data. Second, different losses have different degrees of robustness to labeling noise, i.e. robustness to training examples that have the wrong label, due to human error during labeling. These factors can make different loss functions important for different applications, depending on the data size, quality of the manual annotation, problem difficulty, etc.

1.1 *Related Work*

The hinge loss function used in the support vector machines (SVM), implemented in the Caffe architecture [5], has been used recently in the deeply supervised CNNs [8] and obtained smaller test errors on the MNIST dataset [7] compared to the logistic loss on the same CNN configuration. The handwritten digit MNIST dataset is much easier than the lymph node data, because the digits are clearly distinguishable from each other, with a small number of exceptions. This is why it is very easy to obtain a 0% training error with a deep CNN on the MNIST dataset after less than 100 epochs. In contrast, the lymph nodes are not so easily distinguishable from the hard negatives and even after 1500 epochs the training error hovers around 5-10%.

Furthermore, we observed that it was hard to train our deep CNNs described in Section 3.1 with the hinge loss because it is not differentiable, so it required a small learning rate and many training epochs. We chose to study a differentiable version of the hinge loss from [2] instead, which has the following desirable properties:

1) It is differentiable everywhere, which makes it easy to use for training deep CNNs by back-propagation.
2) It is monotonically decreasing on $(-\infty, 1+h)$ (which is $(-\infty, 1.1)$ in our application), to correctly classify most training examples with high confidence.
3) It is zero on $(1+h, \infty)$ so that easy examples don't contribute to the loss.

We are looking for loss functions that obey the above three properties, plus another property that makes them robust to labeling noise:

4) The loss function should grow slowly towards $-\infty$ so that mislabeled examples have a limited influence in the total cost.

We will study two losses obtained from the binary classification literature, but which have not been used for training CNNs before. In [11] were presented two novel loss functions, named Savage loss and Tangent loss, together with boosting algorithms for each, named SavageBoost and TangentBoost. The tangent loss had the property that it is positive everywhere and it reaches value zero at a single point, located between 0 and 1, after which it becomes positive again. While the tangent loss proposed in [11] did not obey the above four properties, we present a version that obeys all four properties for our CNN architecture. One major modification was to set it to 0 after the point where it first becomes zero (point which was moved to 1 in our version).

The Lorenz loss was introduced in [1] and was used for for binary classification with a training algorithm named Feature Selection with Annealing. It was experimentally observed in [1] through simulations with artificial noise that the Lorenz loss was more robust to labeling noise than the smoothed SVM and logistics losses. It also obtained better prediction results on a number of real datasets. Since it obeys all four desired properties described above, it was used unmodified in our CNN training.

In [13], a CNN architecture similar to the one in this paper was used for lymph node detection. The approach detected a number of LN candidates with close to 100% accuracy and extracted 100 random 2D views at each candidate LN location, in random directions.

Each random view has three 2D channels that are orthogonal cross-sections extracted from the 3D CT image at or near the LN candidate location. The final candidate score was obtained as the average of the 100 CNN scores of the random views. This paper follows the same general architecture, but uses only 36 views at each candidate location, and trains the CNN using four different loss functions in order to evaluate the differences in training with these loss function and in the final detection result.

2. Neural Network Cost Functions

For classification, given a training set $D = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^D \times \mathbb{Z}, i = \overline{1,N}\}$, a neural network is a function $\boldsymbol{\varphi}_W : \mathbb{R}^D \to \mathbb{R}^L$ that given a D-dimensional input $\boldsymbol{x} \in \mathbb{R}^D$ outputs a L-dimensional response vector $\boldsymbol{\varphi}_W(\boldsymbol{x})$ representing the confidence for the L possible outcomes (classes).

Even though LN detection is a binary classification problem ($L = 2$), we first present for completeness the more general context of multi-class loss functions existent in the literature, which could be used in other applications.

2.1 *Multi-Class Loss Functions*

The neural network has a set of parameters W that are learned through the optimization of a cost function

$$L_D(W) = \sum_{i=1}^{N} L(\boldsymbol{\varphi}_W(\boldsymbol{x}_i), y_i), \tag{1}$$

which depends on a loss function $L(\boldsymbol{u}, y)$ that specifies how to compare the output vector with the ground truth label y.

In general the logistic loss is used,

$$L(\boldsymbol{u}, y) = -\ln \frac{e^{u_y}}{\sum_{k=1}^{L} e^{u_k}}, \tag{2}$$

which is defined in terms of the normalized *soft-max* probability $\dfrac{e^{u_y}}{\sum_{k=1}^{L} e^{u_k}}$ of the ground truth label y.

It is not easy to derive good multi-class loss functions $L(\boldsymbol{u}, y)$ when the number of labels is $|L| \geq 3$. However, there are notable examples that have been derived for the multi-class SVM, the first one due to Vapnik [15]

$$L(\boldsymbol{u}, y) = \sum_{k \neq y} \ell(u_y - u_k), \tag{3}$$

another one from Crammer [4]

$$L(\boldsymbol{u}, y) = \ell(u_y - \max_{k \neq y} u_k) \tag{4}$$

and a more recent one from [9], which requires $\sum_{k=1}^{L} u_k = 0$:

$$L(\boldsymbol{u}, y) = \sum_{k \neq y} \ell(-u_k). \tag{5}$$

All are formulated in terms of a one-dimensional loss function $\ell(x)$ that penalizes the *margin* x by a large amount if $x < 0$ and by a small (or zero) amount if $x > 0$. Examples of loss functions $\ell(x)$ are shown in Figure 1 and described below.

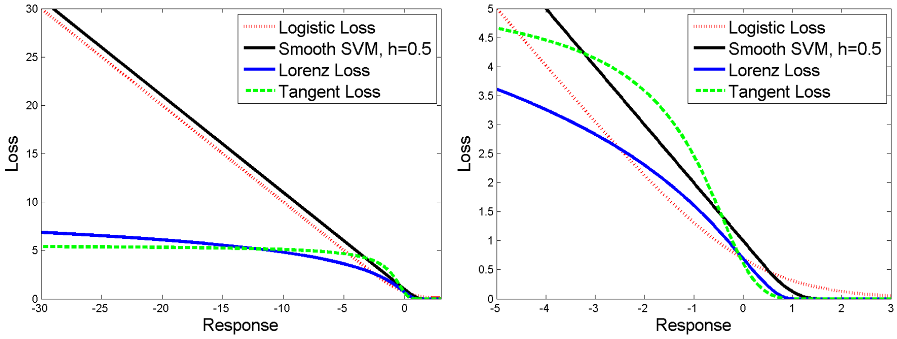

Figure 1. Left: The four loss functions $\ell(x)$ that are used in this work. Right: zoom-in on the interval $[-5, 3]$.

2.2 *Loss Functions for Binary Classification*

For binary classification, things are much simpler, and all these losses are equivalent to a loss of the form $L(\boldsymbol{u}, y) = \ell(yu_1)$, where y is assumed to take values $y \in \{-1, 1\}$ and u_1 is the response for class $y = 1$. Beside the logistic loss defined in Eq. (2), we investigate three other loss functions $\ell(x)$, illustrated in Figure 1. All these losses can also be used for training multi-class problems.

(1) The differentiable approximation of the SVM loss [2]

$$\ell(x) = \begin{cases} 0 & \text{if } x > 1 + h \\ \dfrac{(1 + h - x)^2}{4h} & \text{if } |1 - x| \le h \\ 1 - x & \text{if } x < 1 - h \end{cases} \tag{6}$$

with a parameter h defining the degree of smoothness. A small h brings the loss closer to the hinge SVM loss, but makes it harder to work with, requiring small learning rates and more training epochs. The value of h was fixed to $h = 0.1$ in our experiments.

(2) The Lorenz loss [1]

$$\ell(x) = \begin{cases} 0 & \text{if } x > 1 \\ \ln(1 + (x - 1)^2) & \text{else.} \end{cases} \tag{7}$$

which was observed experimentally in [1] to be robust to labeling noise.

(3) A modified version of the Tangent loss [11],

$$\ell(x) = \begin{cases} 0 & \text{if } x > 1 \\ (\text{atan}(x) - \pi/4)^2 & \text{else.} \end{cases} \tag{8}$$

also supposed to be robust to labeling noise since it gives a bounded cost to a misclassified example no matter how far from the decision boundary.

Each of these losses has its own advantages and disadvantages for binary classification. The two-class logistic loss is convex, which was important for logistic regression, but it is not important in the context of deep CNNs, since the overall cost function (1) is not convex anyway. The other three losses except the logistic loss take value zero for ”easy” examples that are classified correctly with a large margin. This means they have no contribution to the total cost (1) or the gradient with respect to the weights W. Therefore, they don’t need to be involved in back-propagation, and the training could be made faster.

The Lorenz and tangent losses are robust to labeling noise because they penalize a gross error by the log of the margin $y\boldsymbol{\varphi}_1(\boldsymbol{x})$ in the case of the Lorenz loss and by at most a constant value in the case of the Tangent loss. This way examples that have a wrong label will not have a strong influence in the loss and in the training result.

3. System Architecture

In this paper we focus on lymph node detection using CNNs, following the setup from [13], described in more detail below. The same CNN loss functions could be used in other medical imaging CAD problems where conclusions could be different depending on the problem difficulty, label quality and other factors.

Preprocessing. Lymph node candidate locations were obtained using a Random Forest approach [3] for the abdominal volumes and a multi-label fusion [10] for the mediastinal lymph nodes. These candidate generators have a detection rate close to 100% with about 40 false positives per volume. At each of the candidate locations a number of $N = 36$ triplets of orthogonal cross-sections (views) [12] were extracted at random directions and scales and were used as input to the CNN. The average of the CNN scores of the N views at each candidate location was used as the final candidate score, and the results are reported based on these final scores. We experimented with other aggregation schemes such as median and percentiles but the average offered best performance. All reported results are obtained from 6-fold cross-validation.

3.1 *Architecture and training of the CNN*

CNN architecture. The following CNN architecture is used in all experiments, using a modified version of the open source GPU implementation of [6]:

- The input data is 32×32 with three channels for the three orthogonal cross-sections at the LN candidate location.
- Two convolutional layers with 64 filters of size 5×5, each followed by a max-pooling layer over a 3×3 range, with a stride of 2.
- Two local fully connected layers with DropConnect [16] and filters of size 3×3 with stride 1. The first layer has 64 filters and the second layer has 32 filters.
- A fully connected layer with 512 neurons, DropConnect [16], and rectified linear unit (ReLU) activation.
- The output layer with 2 neurons.

A final cost layer is used for training the CNN and evaluating the performance on the training and test sets. Four versions of the cost layer were used, to reflect the loss functions from equations (2), (6), (7) and (8). We also experimented with the standard hinge loss $\ell(x) = \max(1 - x, 0)$, but it was very difficult to work with because it required a very small learning rate and mini-batch size, so training was at least 10 times slower than with the other losses.

Training details. Since the experiments are with six-fold cross-validation, the data is divided in six batches, and at each fold five batches are used for training and the sixth one for testing. The number of training epochs was tuned on one of the six cross-validation folds using the logistic loss. Then the same training was applied to all four losses. The training consists of 420 epochs in four stages:

- The first 100 epochs of training are run on four out of the five training batches, with mini-batch size of 64.
- Another 200 epochs are trained on all five training batches, with mini-batch size of 64.
- The learning rate is reduced 10 times and 100 more epochs are trained, with mini-batch size of 32.
- The learning rate is again reduced 10 times and 20 more epochs are trained, with mini-batch size of 16.

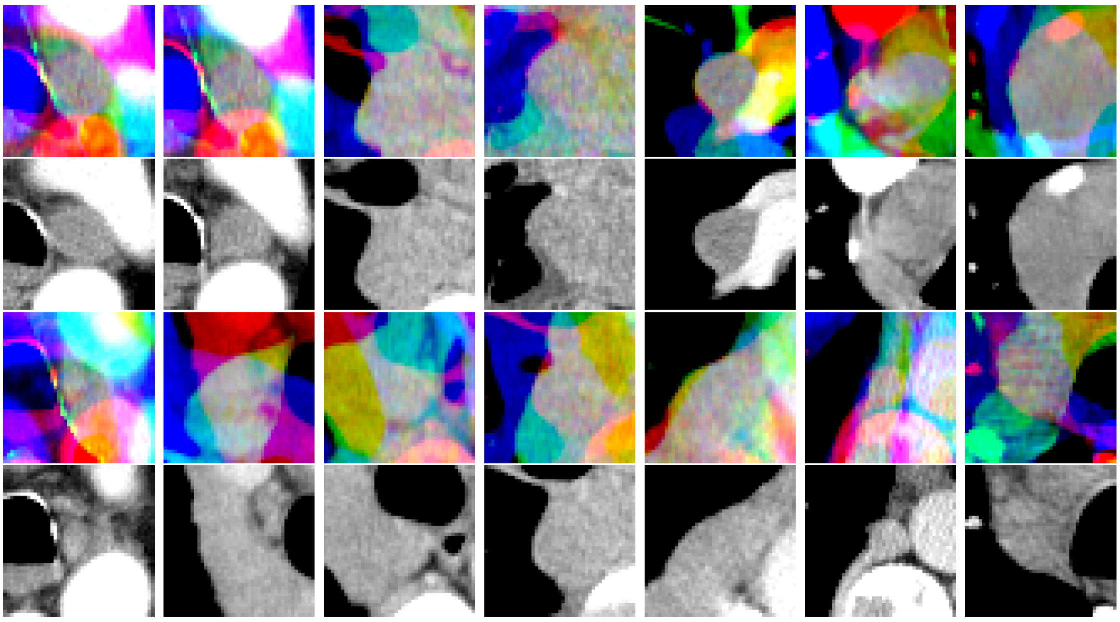

Figure 2. Examples of LN positive candidates (top) and negatives, i.e. non-LN candidates (bottom). The three orthogonal cross-sections are shown as three color channels. The axial cross-section is also shown below each color image, for clarity.

4. Experiments

Dataset. We use a dataset containing mediastinal and abdominal lymph nodes. A total of 388 mediastinal LNs and 595 abdominal LNs have been labeled by radiologists in the 176 CT volumes. The LN candidate generator produced a total of 8658 candidate locations, of which 6692 are false positives.

Results. The average free-receiver operating characteristic (FROC) curves for training (left) and testing (right) of the four loss functions with six-fold cross-validation are shown in Figure 3. We see that the smoothed SVM loss and the Lorenz loss outperform the logistic loss and that an improvement of 3% in detection rate at 3 false positives(fp)/volume can be obtained just by using the Lorenz loss instead of the logistic loss. The Area under the normalized FROC curves up to 10 fp/volume (AUC) on the test data increased from 0.698 for the logistic loss to 0.713 for the Lorenz loss, a 2.2% increase, with a p-value $p = 0.028$ based on a paired t-test on the AUCs of the six folds.

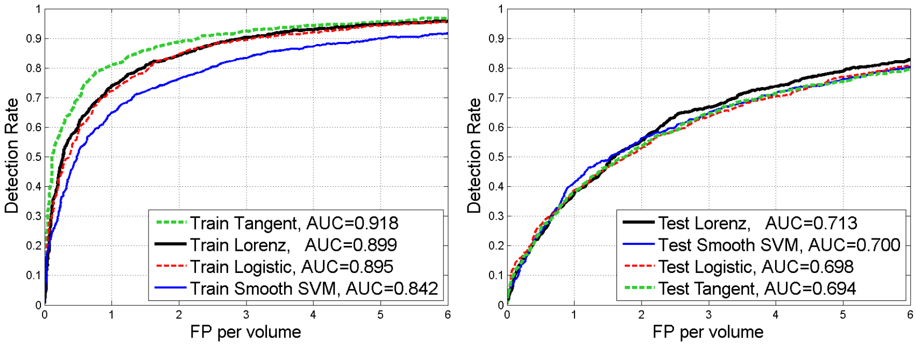

Figure 3. Detection results for identical CNNs trained with the four loss functions with 6-fold cross-validation. Left: train set, right: test set.

Comparing the training and test FROC curves in these experiments reveals that the tangent loss overfits most, and the smooth SVM loss overfits least among the four loss functions. Since the smoothed SVM loss resists better to overfitting, we increased the number of neurons in the fully connected layer from 512 to 1024 and 2048. The training and test curves are shown in Figure 4, again with 6-fold cross-validation. The Lorenz loss result from Figure 3 is also shown for comparison. One can see that the test performance improves but it cannot reach the performance of the Lorenz loss.

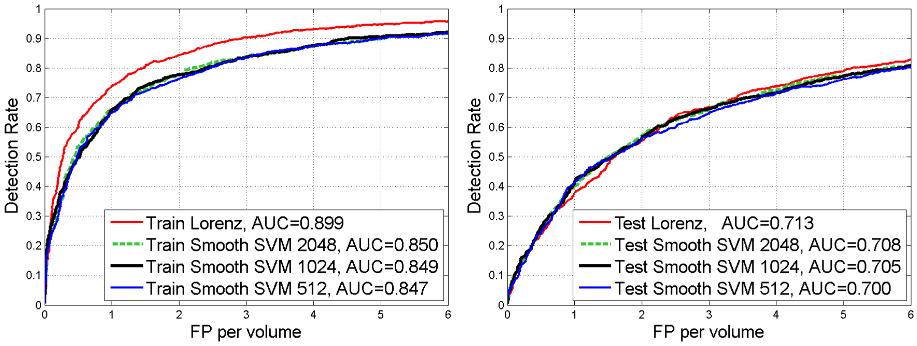

Figure 4. Detection results for the smoothed SVM loss with 512-2048 neurons in the fully connected layer. Left: train set, right: test set.

The results presented in this paper are not comparable with the results from [13] because instead of training the mediastinal and abdominal separately, they were trained together, and the number of views extracted per LN candidate was smaller ($N = 36$ instead of $N = 100$), so overfitting was a more acute concern in our case.

The conclusions we can draw from these experiments are that there exist better losses than the softmax (logistic) loss such as the smoothed SVM loss or the Lorenz loss. The advantages and disadvantages of the tested losses are summarized in Table 1.

Criterion	softmax (logistic)	hinge (SVM)	smooth SVM	Lorenz	tangent
Ease of training	☺	☹	☺	☺	☺
Test Error	😐	-	☺	☺	☹
Labeling Noise Robustness	😐	😐	😐	☺	☺
Overall	✓	X	✓✓	✓✓✓	X

Table 1. Comparison between loss functions under different criteria

Even though the tangent loss fares well on two criteria, the fact that it does not generalize well is a deal breaker.

5. Conclusion

This paper presented a study of four different loss functions for training deep Convolutional Neural Networks in computer aided diagnostics, with a focus on lymph node detection. Our experiments reveal that the logistic loss based on soft-max and the tangent loss obtained the worst detection performance and the Lorenz loss obtained the best performance. The smoothed SVM loss stood out as overfitting the least and the tangent loss overfitted the most.

Our conclusion is that it is important to find the appropriate loss function for the particular classification problem that needs to be solved and that the standard softmax loss might not be the best choice. One of the factors that determine which loss function is appropriate is the amount of labeling noise in the data, since the SVM and the logistic losses are the least resistant to labeling noise. Another factor is whether the easy examples have an influence on the loss function or not, which might help with speed and generalization power.

Acknowledgments. This research was supported in part by the National Institutes of Health, Clinical Center. We thank NVIDIA Corporation for their generous donation of a Tesla K40 GPU in support of this work.

References

[1] A. Barbu, Y. She, L. Ding, and G. Gramajo, *Feature selection with annealing for big data learning*, arXiv preprint arXiv:1310.2880 (2013).

[2] O. Chapelle, *Training a support vector machine in the primal*, Neural Computation 19 (2007), pp. 1155–1178.

[3] K.M. Cherry, S. Wang, E.B. Turkbey, and R.M. Summers, *Abdominal lymphadenopathy detection using random forest*, in *SPIE Medical Imaging*, 2014.

[4] K. Crammer and Y. Singer, *On the algorithmic implementation of multiclass kernel-based vector machines*, The Journal of Machine Learning Research 2 (2002), pp. 265–292.

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, *Caffe: Convolutional architecture for fast feature embedding*, in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 675–678.

[6] A. Krizhevsky, I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks*, in *NIPS*, 2012, pp. 1097–1105.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE 86 (1998), pp. 2278–2324.

[8] C.Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, *Deeply-supervised nets*, arXiv preprint arXiv:1409.5185 (2014).

[9] Y. Lee, Y. Lin, and G. Wahba, *Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data*, Journal of the American Statistical Association 99 (2004), pp. 67–81.

[10] J. Liu, J. Zhao, J. Hoffman, J. Yao, W. Zhang, E.B. Turkbey, S. Wang, C. Kim, and R.M. Summers, *Mediastinal lymph node detection on thoracic CT scans using spatial prior from multi-atlas label fusion*, in *SPIE Medical Imaging*, 2014.

[11] H. Masnadi-Shirazi, V. Mahadevan, and N. Vasconcelos, *On the design of robust classifiers for computer vision*, in *CVPR*, 2010, pp. 779–786.

[12] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, *Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network*, in *MICCAI*, 2013, pp. 246–253.

[13] H.R. Roth, L. Lu, A. Seff, K.M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R.M. Summers, *A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations*, in *MICCAI*, 2014, pp. 520–527.

[14] P. Therasse, S. Arbuck, E. Eisenhauer, J. Wanders, R. Kaplan, L. Rubinstein, and et al., *New guidelines to evaluate the response to treatment in solid tumors*, JNCI 92 (2000), pp. 205–216.

[15] V.N. Vapnik and V. Vapnik, *Statistical learning theory*, Vol. 1, Wiley New York, 1998.

[16] L. Wan, M. Zeiler, S. Zhang, Y.L. Cun, and R. Fergus, *Regularization of neural networks using dropconnect*, in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1058–1066.