LUNG OPACITIES DETECTION USING AUXILIARY DEEP GENERATIVE MODELS

Søren Møller Rasmussen (s173141), Christoffer Riis (153147), Malte E. K. Jensen (s153962)

Technical University of Denmark (DTU)

ABSTRACT

Access to trained medical doctors that can diagnose x-rays are scarce in the development countries, which could be solved by machine learning. Further, well labeled x-ray images are expensive and cannot be found in great amounts. This paper investigates the use of variational autoencoders to detect opacities in x-ray images of the chest with semi-supervised learning. All proposed models performed similary, with the best having an accuracy of 89.5 % with 50 % labeled. The performance of the different amount of labeled data ranged from 89.5 % at 50 % labeled to 78.5 % at 0.6 % labeled.

1. INTRODUCTION

Access to a diagnosis of medical X-rays by trained medical doctors is not available to many patients in development countries around the World. Further, well-labeled chest X-rays are a scarce resource and existing X-rays are expensive to get annotated by trained medical doctors. This paper investigates the use of auxiliary deep generative models in a semi-supervised setting to classify whether opacities are present or not. Opacities on X-ray images are of interest, as they represents infiltrations in the lung tissue which is most often a sign of a pathological condition and will form the basis of the diagnosis and further investigation.

2. THEORY

Using the method described in *Auxiliary Deep Generative Models* [1] a semi-supervised probabilistic model was constructed by using an inference neural network and a generative neural network consisting of the input data x (X-ray images) with the class information y, the latent variable z, and the auxiliary variable a.

Initially, the inference neural network $q(z|x)$ and the generative neural network $p(x|z)$ were constructed using only z without y and a. Together the two neural networks acts as a variational autoencoder, where the inference model is the encoder and the generative model is the decoder. Both the inference model and the generative model were parameterized by deep neural networks consisting of both linear and convolutional layers denoted respectively by $q_\phi(x|z)$ with parameter ϕ and $p_\theta(z|x)$ with parameter θ.

By adding the class information as an extra latent variable y, the model is changed from unsupervised to semi-supervised. In this paper the data consist of only two categorical classes: "Lung Opacity" and "Normal".

Lastly, the auxiliary variable a was added in order to let the marginal distribution $q(x|z)$ fit more complex posteriors $p(z|x)$. The auxiliary variable was also inferred using x such that $q(a,z|x) = q(z|a,x)q(a|x)$. This gives the two following models. The inference model Q was defined as $q_\phi(a|x)q_\phi(y|a,x)q_\phi(z|a,y,x)$, where each term was defined as

$$\begin{aligned} q_\phi(a|x) &= \mathcal{N}(a|\mu_\phi(x), \operatorname{diag}\left(\sigma^2_\phi(x)\right)), \\ q_\phi(y|a,x) &= \operatorname{Bernoulli}(y|\pi_\phi(a,x)), \\ q_\phi(z|a,y,x) &= \mathcal{N}(z|\mu_\phi(a,y,x), \operatorname{diag}\left(\sigma^2_\phi(a,y,x)\right)). \end{aligned}$$

The generative model P was defined as $p(z)p(y)p_\theta(a|z,y,x)p_\theta(x|z,y)$, where each term is defined as

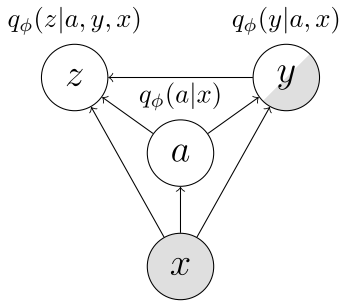
$$\begin{aligned} p(z) &= \mathcal{N}(z|0,\mathrm{I}), \\ p(y) &= \operatorname{Bernoulli}(y|\pi), \\ p_\theta(a|z,y,x) &= f(a;z,y,x,\theta), \\ p_\theta(x|z,y) &= f(x;z,y,\theta). \end{aligned}$$

The two models are illustrated in Figure 1 to clearly show the dependence of the different variables.

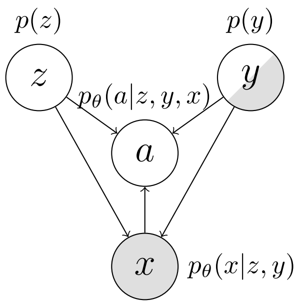
In order to relate this to the architecture of the deep neural network the following list shows the connection between the above equations and the deep convolutional and deep linear neural networks.

Encoder	$q_\phi(z|a,y,x)$ (Conv + Linear)
Decoder	$p_\theta(x|z,y)$ (Linear + Deconv)
Aux. Encoder	$q_\phi(a|x)$ (Linear)
Aux. Decoder	$p_\theta(a|z,y,x)$ (Linear)
Classifier	$q_\phi(y|a,x)$ (Linear)

The four Gaussian distributions $q_\phi(z|a,y,x)$, $p_\theta(x|z,y)$, $q_\phi(a|x)$, and $p_\theta(a|z,y,x)$ were modelled by splitting up the output from the last layers in the deterministic layers in each deep neural network into a mean $\mu_{\phi\vee\theta}$ and a log variance



(a) Inference model Q.



(b) Generative model P.

Fig. 1: Illustration of the inference model and generative model of the ADGM for semi-supervised learning. The grey nodes denote the known data. The node for the label y is half grey showing the semi-supervised aspect.

$\log \sigma^2_{\phi \vee \theta}$. The mean and variance outputs for each of these Gaussian distributions are also illustrated in the sketch of the architecture (Figure 2) in the outputs of these these four neural networks.

All the parameters were optimized by maximizing the lower bound on the likelihood as well as the classification loss. There are both a likelihood from the labeled and unlabeled case. For the unlabeled case the lower bound on the likelihood is

$$\begin{aligned}\log p(x) &= \log \int_a \int_z \sum_y p(x,y,a,z)dzda \\ &\geq \mathbb{E}_{q_\phi(a,y,z|x)}\left[\log \frac{p_\theta(x,y,a,z)}{q_\phi(a,z,y|x)}\right] \\ &\equiv -\mathcal{U}(x),\end{aligned}$$

where nominator and the denominator can respectively be written as

$$\begin{aligned}p_\theta(x,y,a,z) &= p_\theta(x|y,z)p_\theta(a|x,y,z)p(y)p(z), \\ q_\phi(a,z,y|x) &= q_\phi(z|a,y,x)q_\phi(y|a,x)q_\phi(a|x).\end{aligned}$$

For the labeled case the lower bound on the likelihood is

$$\begin{aligned}\log p(x,y) &= \log \int_a \int_z p(x,y,a,z)dzda \\ &\geq \mathbb{E}_{q_\phi(a,z|x,y)}\left[\log \frac{p_\theta(x,y,a,z)}{q_\phi(a,z|x,y)}\right] \\ &\equiv -\mathcal{L}(x,y),\end{aligned}$$

where the nominator was the same as for the unlabeled case and the denominator can be written as

$$q_\phi(a,z|x,y) = q_\phi(z|a,x,y)q_\phi(a|x).$$

The integrals over a and z are approximated by sampling from the latent space of a and z with a gaussian distribution.

In the labeled case it was also possible to explicitly calculate a classification loss. This loss was calculated using the binary cross entropy, where the class variable $y \in \{0,1\}$. Let $p_y = \{y, 1-y\}$ and $q_{\hat{y}} = \{\hat{y}, 1-\hat{y}\}$, then the binary cross entropy is defined as

$$\begin{aligned}\mathbb{E}_{q_\phi(a|x)}[-\log q_\phi(y|a,x)] &= -\sum_i p_i \log q_i \\ &= -y\log\hat{y} - (1-y)\log(1-\hat{y}).\end{aligned}$$

The weighting between the labeled likelihood and the classification loss was defined as

$$\alpha = \beta \frac{N_u + N_l}{N_l},$$

where β is a scaling constant, N_u is the number of unlabeled training samples and N_l is the number of labeled training samples. Thus, the labeled loss was defined as

$$\mathcal{L}_l(x,y) = \mathcal{L}(x,y) + \alpha \mathbb{E}_{q_\phi(a|x)}[-\log q_\phi(y|a,x)]$$

Finally, the overall loss function was

$$\mathcal{M} = \sum_{x,y} \mathcal{L}_l(x,y) + \sum_x \mathcal{U}(x).$$

2.1. Balanced Accuracy

In order to take the class imbalance into account the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) are used to calculate the balanced accuracy. Let P be the sum of TP and FP, and let N be the sum of TN and FN, when the balanced accuracy [2] is defined as

$$acc_{bal} = \frac{1}{2}\left(\frac{\text{TP}}{\text{P}} + \frac{\text{TN}}{\text{N}}\right).$$

By using the balanced accuracy it was ensured that the two classes were weighted equally. An alternative would be to simple stratify the data so the two classes would be equally represented. However, this would result in data from the large class to be discarded, and therefore this method was not chosen.

2.2. Deterministic Warm-Up

The lower bounds on the likelihoods contain the Kullback-Leibler (KL) divergence $KL(p||q)$ as an regularization term. By rewriting the expectation of the unsupervised case without auxiliary variables (without loss of generality this can be expanded to fit the case with labeled data and auxiliary variables) the KL divergence appear:

$$\begin{aligned}
&\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \\
&\quad = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z) + \log p_\theta(z) - \log q_\phi(z|x)\right] \\
&\quad = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(z)}{q_\phi(z|x)}\right] \\
&\quad = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - KL(q_\phi(z|x)||p_\theta(z)).
\end{aligned}$$

The KL divergence will be zero if $q_\phi(z|x) = p_\theta(z)$, which implies that the inference model is independent of the input images. In other words, this regularization term might causes some of the latent units to be inactive during training and the network do might not manage to activate such units again. In order to overcome this problem a linear deterministic warm-up [4] of the KL divergence is used such that the above rewriting will be written as

$$\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \gamma \cdot KL(q_\phi(z|x)||p_\theta(z)),$$

where γ is a constant dependent on the number of epochs N_e and an increment factor γ_{inc} such that $\gamma = N_e\gamma_{inc}^{-1}$.

3. METHOD

The data set used for both training and test was the Pneumonia Data set from the 2018 Kaggle competition: ”RSNA Pneumonia Detection Challenge” [5]. The data set consists of X-ray images of the thorax of 26,684 patients, that has been diagnosed by a specialized radiologist. The data set was labeled into three classes: ”Lung Opacity”, ”No Lung Opacity / Not Normal”, and ”Normal”. Only the two classes, ”Lung Opacity” and ”Normal”, were used for this paper to keep the classification problem more simple. This resulted in 12,288 images.

All images were preprocessed, using downscaling to a size of 112×112 pixels and histogram equalization was then applied, to enhance the differences in pixel intensity (see Figure 3 for the resizing).

An exploratory experiment were run with 32, 64 or 128 latent variables, combined with either 0 or 64 auxiliary variables and with or without KL warm-up for 75 epochs. The warm-up was defined as a linear increase from 0% to 100% of the KL term over 75 epochs. The models were trained on 10,240 training images, where 50% were unlabeled. The test data set had 2048 images and the batch size was 64. Optimization were done using the Adam optimizer with a learning rate of 10^{-3}. The best performing model were used afterwards to train on varying proportions of labeled and unlabeled data.

A model were then chosen to train with different proportions of labeled and unlabeled data: VAE with warm-up, 32 latent features and no auxiliary variables where chosen, due being the most simple among the performing models (see Table 1). In the different setup for the proportions of labeled and unlabeled data all consisted of 10,240 images such that $N_l + N_u = 10,240$ for all setups. The results can be seen in Table 3.

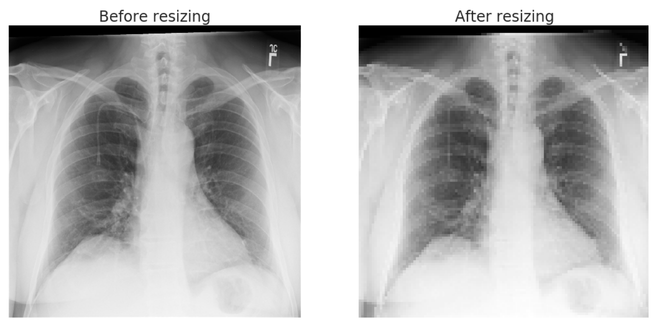


Fig. 3: Example of an X-ray image before and after the resizing to 112x112 pixels.

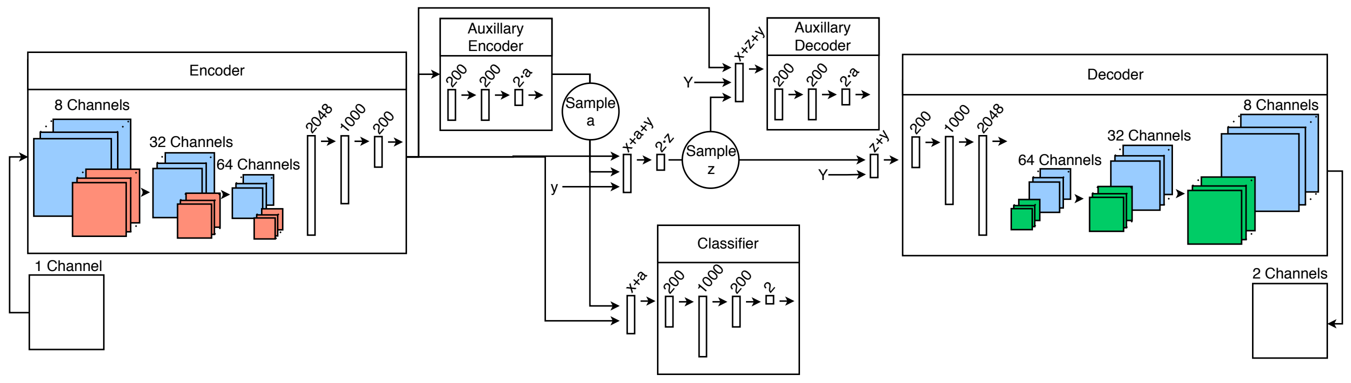


Fig. 2: Architecture for Auxiliary Deep Generative Model implemented by M.E.K. Jensen, S.M. Rasmussen. and C. Riis with inspiration from J. Wohlert’s notebook [3].

3.1. Network architecture

The network was build as an variational autoencoder (VAE). The network architecture is shown in Figure 2. The Encoder and Decoder was based on resp. convolutional and transpose-convolutional layers with Max-pooling in the encoder to downsize the images, and Interpolation in the decoder to upsample to the correct size. The convolutional and transpose-convolutional layers did not change the size of the input, but was instead done by resp. max-pooling and interpolation. This was done to avoid upsampling checkerboard artifacts as is common when using transpose convolutional layers to perform upsampling. Furthermore all layers were followed by batch-normalization and dropout.

4. RESULTS

Two different parameters were tested by this network. Firstly the inclusion of KL warm-up and auxillary variables was tested, to find the best model. Secondly this model was used to test the performance at different fractions of labeled data. [1]

4.1. Model selection

In Table 1 the balanced accuracy for each combination of variational autoencoder (VAE), deterministic warm-up (WU), and auxiliary variables (aux.). It is seen that all constellations gave similar results. For VAE and VAE + WU + aux. the accuracy increased as the number of latent features increased. However, this was not the case for VAE + WU. It was seen that the model with the highest balanced accuracy was VAE + WU with 32 latent variables.

No. latent features	**VAE**	**VAE + WU**	**VAE + WU + aux.**
32	88.1 %	**89.5 %**	88.0 %
64	88.4 %	89.1 %	88.9 %
128	88.7 %	89.3 %	89.2 %

Table 1: Balanced accuracy for the combination of variational autoencoder (VAE), deterministic warm-up (WU), and auxiliary variables (aux.). The number of latent features is for the latent variable z.

An other interesting aspect was the latent space for the different models. In Table 2 the T-SNE representations of the latent spaces for these models are shown. It could be seen that the results of the different latent spaces were similar, but it seemed that the lowest separation in the two classes were seen for the model consisting of VAE and highest separation in the two classes for the model consisting of VAE + WU.

[1] The implemented code can be found at `https://github.com/Soren941/Opacity-Detection-in-thorax-x-ray`

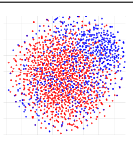
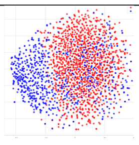
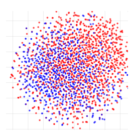
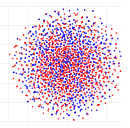
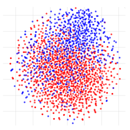
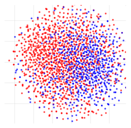
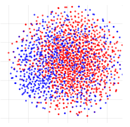
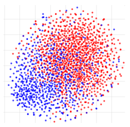
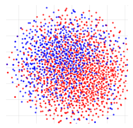
No. latent features	**VAE**	**VAE + WU**	**VAE + WU + aux.**
32			
64			
128			

Table 2: The latent spaces generated with test data for the combination of variational autoencoder (VAE), deterministic warm-up (WU), and auxiliary variables (aux.). The number of latent features is for the latent variable z.

4.2. Varying number of labeled data

In Table 3 the balanced accuracy for different amount of labeled training data is shown. As described earlier in the section "Model", the amount of data for each setup was 10,240 images. It appears from the table that the balanced accuracy decreased as the percentile of labeled data decreased.

No. labeled images		**VAE + WU**
5120	(50.0 %)	**89.5 %**
2560	(25.0 %)	86.8 %
1280	(12.5 %)	86.0 %
640	(6.3 %)	83.1 %
256	(2.5 %)	80.0 %
128	(1.3 %)	78.4 %
64	(0.6 %)	78.5 %

Table 3: Balanced accuracy for varying amounts labeled and unlabeled images with the combination: variational autoencoder (VAE) and deterministic warm-up (WU). The first number designates the amount of labelled images and the second is the percentage this amounts to.

A part of the autoencoder was also to reconstruct the input images. In Figure 4 four input images and reconstructions are shown. The top row and bottom rows shows respectively images from the class *normal* and the class *lung opacity*.

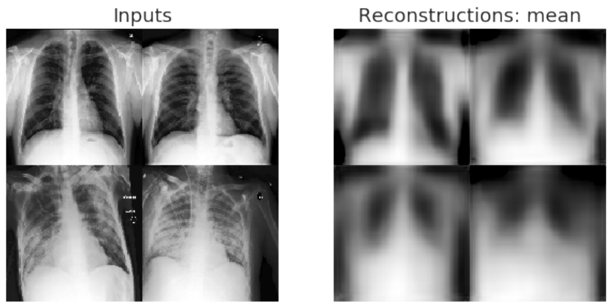


Fig. 4: Input images to the left and reconstructions to the right. The top row shows images from the class *normal* and bottom row shows images from the class *lung opacity*.

Lastly, in Figure 5 four samples from latent space are shown.

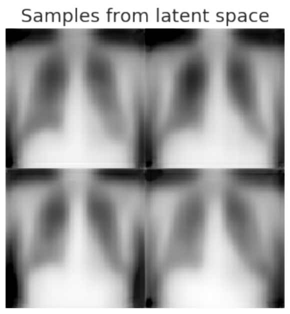


Fig. 5: Four samples from latent space.

5. DISCUSSION

From the initial results in table Table 1, no real difference in the classification performance was seen for these setups. When looking at Table 2, better separation in the latent space was seen for the VAE + WU and VAE + WU + aux. compared to only the VAE. For all cases, overfitting was observed, as the training accuracy converged to 100 % while the test error did not improve. This could indicate that the models could have more potential to generalize given a better hyper-parameter tuning.

Hence, the most simple model that still performed different in the latent space were chosen as VAE + WU with 32 latent features, to test on different amount of labeled and unlabeled data. The results of different amounts of labeled and unlabeled data can be seen in Table 3. For all these cases, poor separation was seen in the latent space (results not shown).

In Table 3 it is seen that the accuracy increases as the amount of labeled data increases. It is seen that with only 256 labeled images (124 for each class and 2.5 % of the training data) the model do still achieve a balanced accuracy at 80 %. The maximum balanced accuracy is 89.5 % for 50 % labeled data. A higher percentile of labeled data has not been tested in this paper, but would be an interesting aspect to investigate further.

From Figure 4 it could be seen that network was able to reconstruct the images to some degree, but not with good details. The overall shape of the x-ray image seemed to be encoded well and with some differences in lung shape, but no structure in the lungs could be seen. In Figure 5 samples from the latent space are presented. These were quite similar, and mainly captured the overall shape of chest on the x-ray. Some small differences in intensity of the lung area could be seen, which likely indicates that the network has tried to capture opacities on the x-ray images, but has not been able to encode detailed structures of the opacities.

In conclusion, a VAE could potentially be feasible model for diagnosing chest x-rays, but of would require more research. For the semi-supervised experiments, the balanced accuracy increased with the amount of labeled data, and had similar balanced accuracy at 12.5 % labelled data to that of 50 % labeled, and was still able to learn even at 0.6 % labeled data. With a good network structure, a semi-supervised approach could be a feasible way to exploit the vast amount of unlabeled x-ray images for a classifier.

6. REFERENCES

[1] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther, "Auxiliary deep generative models," in *Proceedings of The 33rd International Conference on Machine Learning*, Maria Florina Balcan and Kilian Q. Weinberger, Eds., New York, New York, USA, 20–22 Jun 2016, vol. 48 of *Proceedings of Machine Learning Research*, pp. 1445–1453, PMLR.

[2] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *2010 20th International Conference on Pattern Recognition*, Aug 2010, pp. 3121–3124.

[3] "GitHub: Wohlert semi-supervised pytorch," `https://github.com/wohlert/semi-supervised-pytorch`, Accessed: 03-01-2019.

[4] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther, "How to train deep variational autoencoders and probabilistic ladder networks," in *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, 2016.

[5] "Kaggle rsna pneumonia detection challenge," `https://www.kaggle.com/c/rsna-pneumonia-detection-challenge`, Accessed: 02-01-2019.