SEMI-SUPERVISED CHEST X-RAY PNEUMONIA CLASSIFICATION USING AUXILIARY DEEP GENERATIVE MODELS

Stefano Cerri *Marco Fraccaro*

DTU Compute Unumed

ABSTRACT

Pneumonia accounts for over 15% of all deaths of children under 5 years old internationally. In 2015, 920,000 children under the age of 5 died from the disease. Chest X-Rays (CXRs) are the most commonly performed diagnostic imaging study. The availability of radiologists is however low: as a result radiologists are overburdened, and unqualified generalist practitioners are often left with the task of image analysis. The availability of ground truth data, in some cases, is scarse. With these premises, it is clear that there is a strong need of an automatic classification model that can learn with just few examples. We propose a generative model that can classify CXRs images from few labelled samples while obtaining competitive results.

Index Terms— chest X-Rays, pneumonia, generative models, semi-supervised, classification.

1. INTRODUCTION

Generative models are a powerful way of learning the data distribution using unsupervised learning and they achieved great success in the last few years [1, 2, 3]. All types of generative models aim at learning the true data distribution of the training set so as to generate new data points with some variations. Usually, due to intractable integrals, it is not possible to learn the exact distribution of the data, so variational inference (VI) is used [4]. The idea behind VI, is to model a distribution which is as similar as possible to the true data distribution, while having an analytical approximation.

In [5], Kingma et al. revisited the approach to semi-supervised learning with generative models and developed new models that allow for effective generalisation from small labelled data sets to large unlabelled ones.

2. AUXILIARY DEEP GENERATIVE MODEL

2.1. Model

The Auxiliary Deep Generative Model (ADGM) [6] is a generative model that include, from [5], an auxiliary approach [7] in order to learn a classifier from labeled and unlabeled data. The auxiliary variables leave the generative model unchanged while making the variational distribution more expressive. The probabilistic graphical model consists of a generative model P and a inference model Q (see Figure 1). The generative model P is defined as $p(y)p(z)p_\theta(a|z, y, x)p_\theta(x|y, z)$:

$$p(z) = \mathcal{N}(z|0, I), \tag{1}$$

$$p(y) = \text{Cat}(y|\pi), \tag{2}$$

$$p_\theta(a|z, y, x) = f(a; z, y, x, \theta), \tag{3}$$

$$p_\theta(x|z, y) = f(x; z, y, \theta), \tag{4}$$

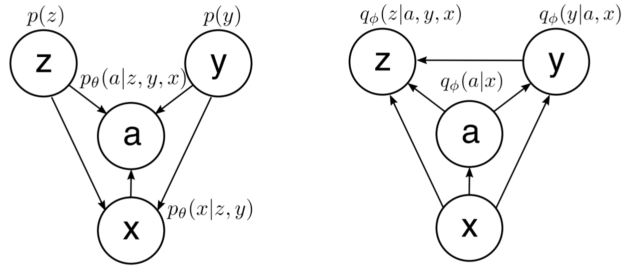
where a, y, z are the auxiliary variables, class labels, and latent features, respectively. $\text{Cat}(\cdot)$ is a multinomial distribution, where y is treated as a latent variable for the unlabeled data points. $f(x; z, y, \theta)$, in these experiments, is a Gaussian distribution for the continuos observation x. p_θ are deep neural networks with parameters θ. The inference model Q is defined as $q_\phi(a|x)q_\phi(z|a, y, x)q_\phi(y|a, x)$:

$$q_\phi(a|x) = \mathcal{N}(a|\mu_\phi(x), \text{diag}(\sigma^2_\phi))), \tag{5}$$

$$q_\phi(y|a, x) = \text{Cat}(y|\pi_\phi(a, x)), \tag{6}$$

$$q_\phi(z|a, y, x) = \mathcal{N}(z|\mu_\phi(a, y, x), \text{diag}(\sigma^2_\phi(a, y, x))), \tag{7}$$

where q_ϕ are deep neural networks with parameters ϕ. In order to model Gaussian distributions $p_\theta(a|z, y, x)$, $p_\theta(x|z, y)$, $q_\phi(a|x)$, $q_\phi(z|a, y, x)$ we define two separate outputs from the top deterministic layer in each neural network, $\mu_{\theta\vee\phi}(\cdot)$ and $\log \sigma^2_{\theta\vee\phi}(\cdot)$. From these outputs we are able to approximate the expectations $\mathbb{E}$ by applying the reparametrization trick [1, 2].



(b) Generative Model P

(c) Inference Model Q

Fig. 1. Probabilistic graphical model of the ADGM for semisupervised learning. The incoming joint connections to each variable are deep neural networks with parameters θ and ϕ.

2.2. Variational Lower Bound

We optimize the model by maximizing the lower bound on the likelihood. The variational lower bound on the marginal likelihood for a single labeled data point is

$$\begin{aligned}\log p(x,y) &= \log \int_a \int_z p(x,y,a,z) dz da \\ &\geq \mathbb{E}_{q_\phi(a,z|x,y)}\left[\log \frac{p_\theta(x,y,a,z)}{q_\phi(a,z|x,y)}\right] \equiv -\mathcal{L}(x,y),\end{aligned} \quad (8)$$

with $q_\phi(a,z|x,y) = q_\phi(a|x)q_\phi(z|a,y,x)$. For unlabeled data we introduce the variational distribution for y, $q_\phi(y|a,x)$:

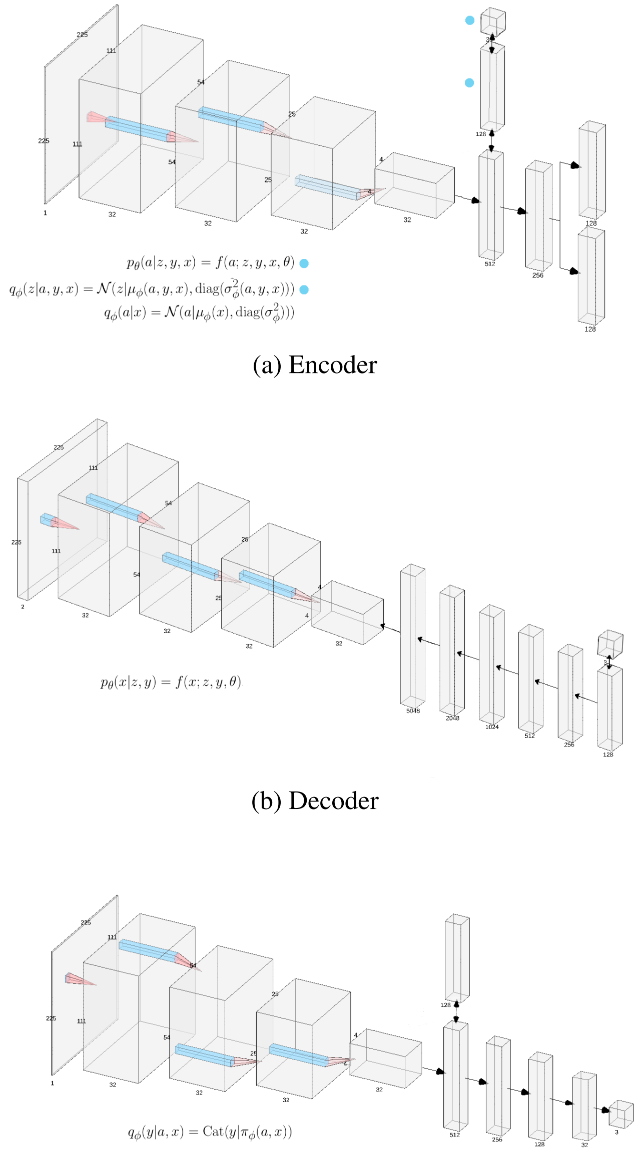
$$\begin{aligned}\log p(x) &= \log \int_a \int_y \int_z p(x,y,a,z) dz dy da \\ &\geq \mathbb{E}_{q_\phi(a,y,z|x)}\left[\log \frac{p_\theta(x,y,a,z)}{q_\phi(a,y,z|x)}\right] \equiv -\mathcal{U}(x),\end{aligned} \quad (9)$$

with $q_\phi(a,y,z|x) = q_\phi(z|a,y,x)q_\phi(y|a,x)q_\phi(a|x)$. The classifier (6) appears in $-\mathcal{U}(x_u)$, but not in $-\mathcal{L}(x_l,y_l)$. The classification accuracy can be improved by introducing an explicit classification loss for labeled data:

$$\mathcal{L}_l(x_l,y_l) = \mathcal{L}(x_l,y_l) + \alpha \cdot \mathbb{E}_{q_\phi(a|x_l}\left[\log q_\phi(y_l|a,x_l)\right], \quad (10)$$

where α is a weight between generative and discriminative learning. The α parameter is set to $\beta \cdot \frac{N_l+N_u}{N_l}$, where β is a scaling constant, N_l is the number of labeled data points and N_u is the number of unlabeled data points. The objective function for labeled and unlabeled data is

$$\mathcal{J} = \sum_{(x_l,y_l)} \mathcal{L}_l(x_l,y_l) + \sum_{(x_u)} \mathcal{U}(x_u) \quad (11)$$


$p_\theta(a|z,y,x) = f(a;z,y,x,\theta)$
$q_\phi(z|a,y,x) = \mathcal{N}(z|\mu_\phi(a,y,x), \mathrm{diag}(\sigma^2_\phi(a,y,x)))$
$q_\phi(a|x) = \mathcal{N}(a|\mu_\phi(x), \mathrm{diag}(\sigma^2_\phi))$
$p_\theta(x|z,y) = f(x;z,y,\theta)$
$q_\phi(y|a,x) = \mathrm{Cat}(y|\pi_\phi(a,x))$

(a) Encoder

(b) Decoder

(c) Classifier

Fig. 2. Architecture of the 5 neural networks of the model. Light blue dots indicate module inclusion.

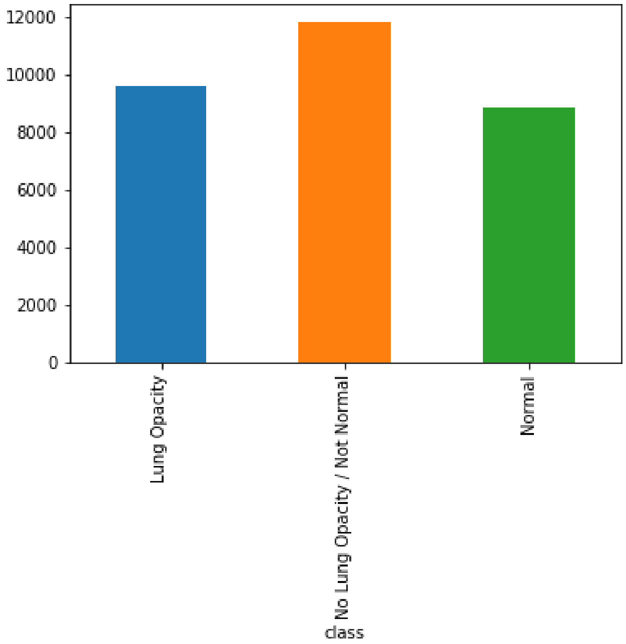
3. IMPLEMENTATION

The model is parametrized by 5 convolutional neural networks (CNN) (see Figure 2): (1) auxiliary inference model $q_\phi(a|x)$, (2) latent inference model $q_\phi(z|a,y,x)$, (3) classification model $q_\phi(y|a,x)$, (4) generative model $p_\theta(a,\cdot)$, and (5) the generative model $p_\theta(x,\cdot)$. We apply ReLU, batch normalization and dropout (0.2) between each convolutional/deconvolutional layer. We trained the model for

100 epochs for the two classeds classification and for 200 epochs for the threes classes classification. We used Adam optimizer with learning rate of 1e-4 and first and second momentum at 0.9 and 0.999, respectively. The β constant was set to 1 and weight decay to 1e-5. We set the temperature on the KL-divergence going from 0 to 1 within the first 100 epochs of training as done in [8, 9]. All the code is publicly available at `https://github.com/ste93ste/ChestXRayClassification`.

4. DATASET

The dataset used is the 2018 RSNA Pneumonia Detection Challenge [1]. It consists of 27000 images with manual annotations from radiologists. There are three classes: Opacity, No-Opacity/Not-Normal and Normal (see Figure 4). The three classes have similar size, so there are no unbalanced dataset problems. The classification between No-Opacity/Not-Normal and Opacity is a difficult task since the images labeled with No-Opacity/Not-Normal look like they contain lung opacities but they don't. We split this dataset in training set (90%) e validation set (10%). We first downsampled the images from 1024 × 1024 to 225 × 225, for computational reasons. We performed online data augmentation during training, using imgaug [2]. Every training batch is augmented with random flips and random affine transformations.

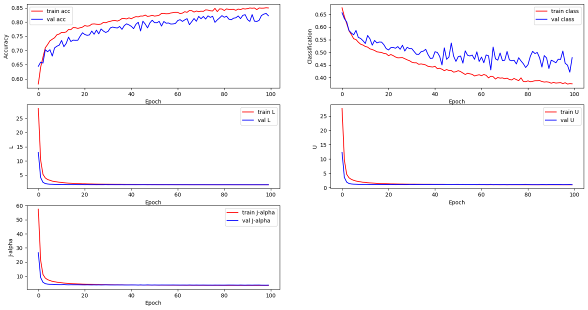


2018 RSNA Pneumonia Detection Challenge Dataset

5. EXPERIMENTS

5.1. Normal vs Abnormal

In this task we merged the No-Opacity/Not-Normal and Opacity class into one class. The task then becomes to classify normal chest X-Ray images and abnormal chest X-Ray images. Note that this is easier task than the one that we discuss in Section 5.2. Figure 3 shows an example of training with 4000 labels per class. More results can be found in Section 5.3.



Confusion Matrix

	Abnormal	Normal
Abnormal	**1384**	386
Normal	89	**810**

Abnormal accuracy:	78,19%
Normal accuracy:	90,10%
Total accuracy:	82,20%

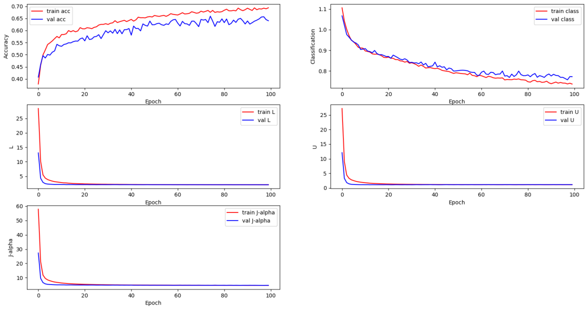
Fig. 3. 2 classes classification with 4000 labels per class: Normal vs Abnormal.

5.2. Normal vs Opacity vs No-Opacity/Not-Normal

In this task we used all the classes. As expected the No-Opacity/Not-Normal class has low accuracy. Figure 4 shows an example of training with 4000 labels per class. More results can be found in Section 5.3.

[1] `https://www.kaggle.com/c/rsna-pneumonia-detection-challenge`

[2] `https://imgaug.readthedocs.io/en/latest/`



Confusion Matrix

	Opacity	Not-Normal	Normal
Opacity	**375**	73	23
Not-Normal	436	**599**	260
Normal	20	123	**756**

Opacity accuracy:	78,95%
No-Opacity/Not-Normal accuracy:	46,25%
Normal accuracy:	84,09%
Total accuracy:	64,82%

Fig. 4. 3 classes classification with 4000 labels per class: Normal vs Opacity vs No-Opacity/Not-Normal.

5.3. Accuracy vs number of labels per class

We compared the accuracy of the model for different number of labels per class (see Figure 5 and Figure 6).

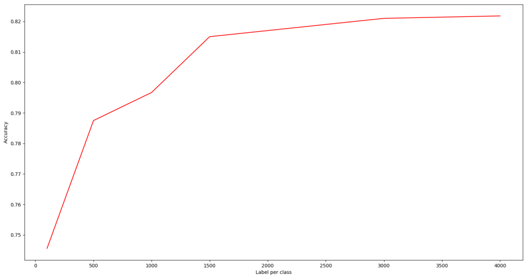


Fig. 5. Accuracy vs number of labels for the 2 classes classification.

The figures show how the plot accuracy vs number of labels per class has a logarithmic trend: after 4000 labels the accuracy starts to be stationary for the 2 class classification while more labels can be used for the 3 classes classification in order to achieve better results. Note that the aim of this model is to learn from few data, if more labels are available, supervised learning model can achieve better results in terms of accuracy.

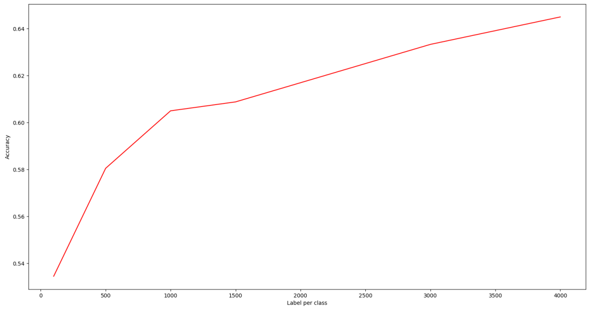


Fig. 6. Accuracy vs number of labels for the 3 classes classification.

5.4. Conditional Generation

From the model we can generate samples conditionally given some normal distributed noise z and a label y. Figure 7 shows some examples generated when y=Opacity.

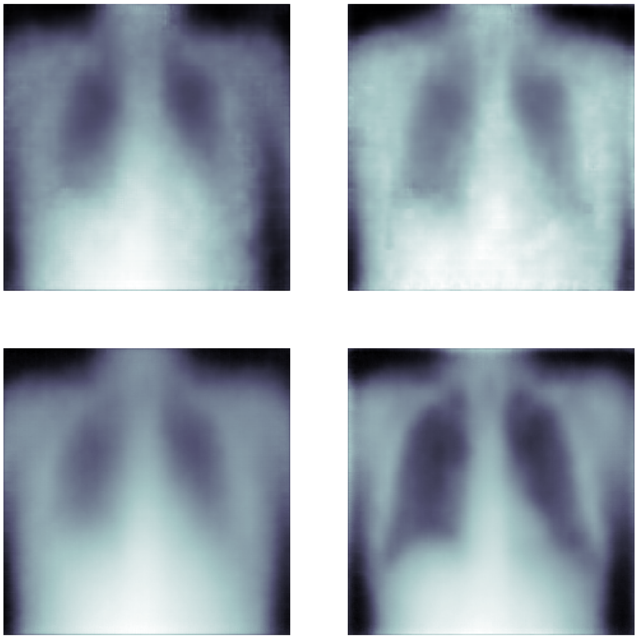


Fig. 7. 4 different samples generated conditionally from the model when y=Opacity.

6. CONCLUSION

We implemented Auxiliary Deep Generative models that can classify pneumonia on chest X-Ray images. We then show how the models can obtain competitive results with few labeled data. Finally we evaluated the accuracy of the models with different number of labels per class.

7. REFERENCES

[1] Diederik P Kingma and Max Welling, “Auto-Encoding Variational Bayes,” *ArXiv e-prints*, p. arXiv:1312.6114, Dec. 2013.

[2] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra, “Stochastic Backpropagation and Approximate Inference in Deep Generative Models,” *ArXiv e-prints*, p. arXiv:1401.4082, Jan. 2014.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative Adversarial Networks,” *ArXiv e-prints*, p. arXiv:1406.2661, June 2014.

[4] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe, “Variational Inference: A Review for Statisticians,” *ArXiv e-prints*, p. arXiv:1601.00670, Jan. 2016.

[5] Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling, “Semi-Supervised Learning with Deep Generative Models,” *ArXiv e-prints*, p. arXiv:1406.5298, June 2014.

[6] L. Maaløe, C. Kaae Sønderby, S. Kaae Sønderby, and O. Winther, “Auxiliary Deep Generative Models,” *ArXiv e-prints*, Feb. 2016.

[7] Felix Agakov and David Barber, “An auxiliary variational method,” November 2004, vol. 3316, pp. 561–566.

[8] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner, “Understanding disentangling in beta-VAE,” *arXiv e-prints*, p. arXiv:1804.03599, Apr. 2018.

[9] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther, “Ladder Variational Autoencoders,” *arXiv e-prints*, p. arXiv:1602.02282, Feb. 2016.