

BACHELOR THESIS
ARTIFICIAL INTELLIGENCE

Radboud University



**Object classification under the
predictive processing approach**

Author:
Thomas de Haer
s1008589

Supervisor:
dr. P.L. Lanillos Pradas
Donders Institute for
Brain, Cognition and
Behaviour
p.lanillos@donders.ru.nl

Second reader:
dr T.C. Kietzmann
Donders Institute for
Brain, Cognition and
Behaviour
t.kietzmann@donders.ru.nl



June 18, 2021

Abstract

Current state-of-the-art algorithms for visual object classification are based on training large convolutional artificial neural networks on labeled data sets. This means that first, the features are directly learned for object classification and the computation is performed using the forward pass of the network. Conversely, evidence in computational neuroscience reveals recurrency, top-down modulation and unsupervised learning in the visual cortex. Here, I study the potential of the predictive processing approach in classification tasks. First, I compared the classification performance of unsupervised learnt features (using a variational autoencoder) against standard supervised approaches (VGG). Second, I studied the advantages of modeling the classification task as an inference process, with bottom-up and top-down modulation. While using predictive processing for classification on the CIFAR-10 data set, was shown to be inferior to that of the VGG model that is trained directly on classification error in a supervised manner with 56.71% against 84.23% classification accuracy, the classification accuracy of 98.63% on the MNIST data set showed there is definitely potential. By modeling the classification task as an inference process, with bottom-up and top-down modulation, I showed that this method can help with the reduction of uncertainty that is present in the real world.

Contents

1	Introduction	2
2	Related Work	4
3	Methods	5
3.1	The Variational AutoEncoder	5
3.2	Predictive Processing	6
3.3	Experiments	7
3.4	Architecture	7
4	Results	9
4.1	CIFAR-10	9
4.1.1	Classification	9
4.1.2	Predictive processing	12
4.2	MNIST	14
4.2.1	Classification	14
4.2.2	Predictive processing	16
5	Discussion	18
5.1	Classification	18
5.2	Predictive processing	19
A	Appendix	22

Chapter 1

Introduction

Convolutional neural networks (CNNs) in a supervised learning scheme (trained directly on classification error), currently achieve the best performance in classification and object detection tasks [4]. However, even though a CNN has state-of-the-art performance, there is still room for improvement. For instance generalisation can still pose as a problem. Mainly, because of the gap between theory and practise [18] [7]. I will look into the question whether using prediction feature encoding is a good representation for classification. This method might come closer to how the human brain infers its environment, since it does not only use bottom-up processing, but also top-down processing [15]. This approach may improve the generalisation of a given problem and thus possibly result in a higher classification accuracy and better properties for interpreting uncertain information.

Since data is of extreme importance in training a neural network, but can be limited, maximizing its generalisation towards the data can be greatly beneficial. For instance where the desired object to classify only occurs rarely in the available data set, not generalizing can be the difference between failure or success. An example could be the classification of malignant lesions in certain tissues [3]. The vast majority of the available data is most likely normal or benign. Making it of great importance to generalize to the malignant cases that are available in the data set.

By using feature encoding coming from a generative model, the hypothesis is that using this top-down processing, the model will gain a great understanding of the classification problem and thus result in a better classification accuracy as opposed to CNNs that are trained purely on classification error. In order to reconstruct the visual input from the latent space that is formed, the information has to be stored in an optimal way, for it to have the reconstructed image as similar as possible to the input image. To achieve this, I will use a Variational Autoencoder (VAE) [10].

For training a VAE you will make use of the Evidence Lower Bound (ELBO). Since the model will be trained in an unsupervised manner, based on the

ELBO and not straight on classification error, the hypothesis is that different information is represented in the latent space of the VAE than at the end of a typical classifier such as the VGG framework [14] that is trained only on classification error in a supervised manner.

In summary this work addresses two main objectives:

- 1) Comparing the classification performance of unsupervised learnt features by making use of the VAE against the well established supervised approach with the VGG framework [14].
- 2) Modeling the classification task as an inference process, with bottom-up and top-down modulation by making use of predictive processing [11].

The thesis is organized as follows: In related work, I will give some background of similar work that has been done in this domain. Then I will elaborate on my experimental setup in the methods chapter. After that I will present the results that I obtained. Followed by the discussion chapter where I will recap on the obtained results in the previous chapter.

Chapter 2

Related Work

The approach of making use of a VAE for classification problems has been used before, such as [12] where a separate VAE is trained for every class. They tested their model with the MNIST and MNIST Fashion data set. Both relatively easy data sets in terms of dimensions. The authors mention that for future work their approach is desirable to be tested on the CIFAR-10 data set.

Bayesian Neural Networks have also been used for classification tasks [5]. Using this approach more attention can be put into the distributions of data and their uncertainties.

The loss function of a VAE has also been used purely for classification of a binary problem [2]. Here a VAE was trained to reconstruct images of football matches. The goal was to let the model predict whether it was watching the actual football match or whether it was watching commercials. A threshold value was set with regards to the loss function and when the loss surpassed said threshold the input would be classified it as a commercial. When the loss value did not surpass the threshold, the input would be classified as an image of a football match. While this approach worked reasonably well, it had a hard time differentiating visually similar looking images from different classes, such as American football versus football.

In 2014, Kingma et al. constructed a model that used semi-supervised classification that was likewise based on a VAE [8]. It achieved rather good performance, however it lacked interpretability. In [16] a VAE was trained in an unsupervised way and the encoder was used as feature extractor for classification. They made use of disentanglement in the VAE which entails that the latent space formed contains variables that have as little overlap as possible, forcing the model to capture as much information as possible in the latent space.

Chapter 3

Methods

3.1 The Variational AutoEncoder

A Variational AutoEncoder is an artificial network architecture suitable for unsupervised learning that can act as a generative model. As depicted in Figure 3.1, the main components of a VAE are the encoder, the latent space and the decoder. The VAE provides a probabilistic way of creating the latent space, which is the space where the input data is transformed and compressed to. This is done by the encoder part of the VAE. To get the latent space to capture the features from the input as well as possible, the network will learn the the mean and log variance and use these parameters to draw a sample z from a distribution. From this sample z , the latent space, the compressed data enters the decoder in order to be decompressed and be scaled back up to the the same dimensions as the input data, where ideally the input and output are identical.

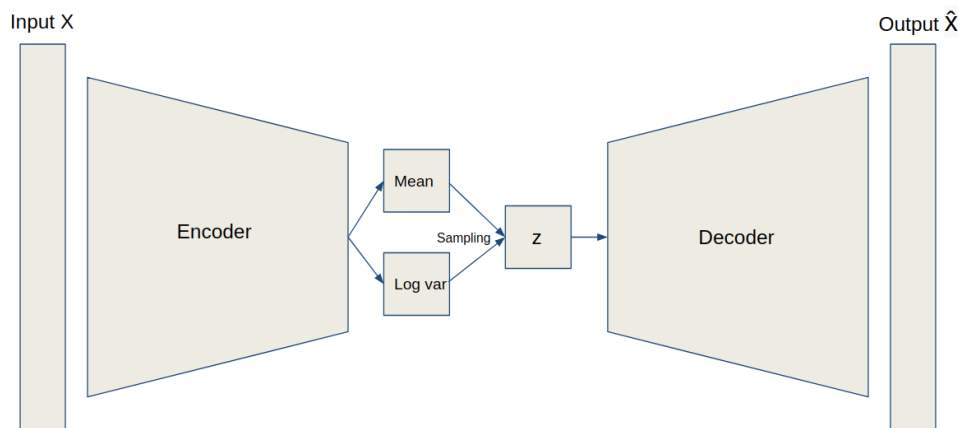


Figure 3.1: Visualisation of a Variational Auto Encoder

The network learns by making use of the ELBO. Implemented as a loss

function for a neural network it will look as follows:

$$\min \mathbb{E}_q[\log q(z|x) - \log p(z)] - \mathbb{E}_q[\log p(x|z)]$$

The first part is the Kullback–Leibler divergence. This part will act as a regularizer, to help form the latent space. We draw a sample z from the q distribution. Very often this distribution is taken as a normal distribution which I have done as well. The chance of z coming from distribution q given x is calculated as well as the probability that z came from the p distribution. Where q is the variational posterior and p is the actual posterior. The goal is to minimize the difference between these distributions, resulting in the variational posterior approximating the actual posterior as well as possible. The second term is the reconstruction term. In this equation z is again sampled from q . Now the probability of seeing the input x given the sample z is calculated. To do that we use the decoder part of the VAE. The sample z will be the input for the decoder and be scaled back up to the dimensions of the input x giving us \hat{x} . I will use the Mean Squared Error (MSE) as a standard reconstruction loss [17].

3.2 Predictive Processing

The advantages of modeling the classification task as an inference process, with bottom-up and top-down modulation by making use of predictive processing are tested, by taking inspiration from the work of Cansu Sancaktar et al. [13] [9]. The latent space is formed by making use of the ELBO, which is in term the same as the negative Free Energy. The Free Energy Principle (FEP) is proposed by Karl Friston [6]. It is an information theory measure that bounds or limits the data samples, given a generative model. To minimize this free energy, predictive processing is used. In this thesis it will be set up as follows: The latent space of the model will be set with a certain image, setting the believe of the internal model to see a certain image. However, by then showing a completely different image to the model, I updated the internal representation of the model using the following equation:

$$\delta z = \delta_z e_v^T \Sigma^{-1} e_v$$

δz will be the partial derivative with respect to the latent variable z . e_v will be the visual prediction error. This is computed by subtracting the decoded image $g(v)$ from the actual input image. Σ^{-1} is the variance of the generative process with respect to the input. Here that will mean the strength of every update.

$\delta_z e_v^T$ will result in the Jacobian $\frac{\delta g}{\delta z}$. When multiplying the transpose of this Jacobian with the error $\frac{\delta F_g}{\delta g}$ you will get the gradients to update the latent variable z :

$$\frac{\partial F_g}{\partial \mathbf{z}} = \underbrace{\begin{bmatrix} \frac{\partial g_{1,1}}{\partial z_1} & \frac{\partial g_{1,2}}{\partial z_1} & \cdots & \frac{\partial g_{w,h}}{\partial z_1} \\ \frac{\partial g_{1,1}}{\partial z_2} & \frac{\partial g_{1,2}}{\partial z_2} & \cdots & \frac{\partial g_{w,h}}{\partial z_2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_{1,1}}{\partial z_4} & \frac{\partial g_{1,2}}{\partial z_4} & \cdots & \frac{\partial g_{w,h}}{\partial z_4} \end{bmatrix}}_{\left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}}\right)^T} \underbrace{\begin{bmatrix} \frac{\partial F_g}{\partial g_{1,1}} \\ \frac{\partial F_g}{\partial g_{1,2}} \\ \vdots \\ \frac{\partial F_g}{\partial g_{w,h}} \end{bmatrix}}_{\frac{\partial F_g}{\partial \mathbf{g}}}$$

3.3 Experiments

For the experiments two data sets were used, namely the CIFAR-10 data set and the MNIST data set. The CIFAR-10 data set consists of RGB images of 32 by 32 pixels adding up to 3096 dimensions. The MNIST data set will be lower in dimensions, it consists of 10 digits and will only be 28 by 28 pixels in gray scale adding up to 784 dimensions in total. Both the CIFAR-10 as well as the MNIST data itself did not undergo any preprocessing other than normalization.

The experiments consisted out of two main parts. The first is were the VAE was trained on one of the two data sets. After training on reconstruction the data was converted to the latent space using the encoder part of the VAE. Then a classifier was trained on these latent space representations. To get a good impression of the performance, I compared it to the VGG's performance, that is trained in a supervised manner on classification error, from which is known to work great on classification tasks. To make comparison between these two models fair, I used the same classifier for both. Meaning two fully connected layers of size 512 to a fully connected layer of size 10 ending with a soft-max activation function.

In the second part, predictive processing is used to update the model's beliefs and to reduce uncertainty from its input.

3.4 Architecture

The architecture of the VAE is based on the following code by keras [1] and is for the CIFAR-10 data modified to take the 32 by 32 by 3 images as input. The following describes the architecture for the CIFAR-10 VAE. The encoder part consists of two convolutional layers followed by two fully connected layers that represent the parameters of the distribution that the sample \mathbf{z} , will be sampled from. One of these fully connected layers will represent the mean of the distribution, while the other will represent the log variance. Both these fully connected layers get their input from the last convolution layer. The final layer will be a custom sampling layer. This sampling layer uses the mean and the log variance as parameters to draw

a vector from a normal distribution. The sample z will then represent the latent space of the VAE. The convolutional layers all have a kernel size of 3 stride of 2, using the relu activation function. The first convolutional layer has 32 filters and the second one has 64 filters. The latent space dimension is chosen to be 128.

The decoder is designed to be roughly "the opposite" of the encoder. The decoder starts off with a fully connected layer of size $8 * 8 * 64$. Then it will go through three convolutional transpose layers. The first two as well as the fully connected layer have a relu as activation function while the last one has it set to a sigmoid activation function. All convolutional transpose layers have a kernel size of 3 and the first two have the stride set to 2. As for the filters it goes from 64 to 32 to 3. A full visualisation can be found in the appendix.

Chapter 4

Results

4.1 CIFAR-10

4.1.1 Classification

After training the VAE for 300 epochs training was stopped and the reconstruction results can be seen in Figure 4.1. Upon visual inspection, although slightly more blurry, for most reconstructed images it is clear what they represent.

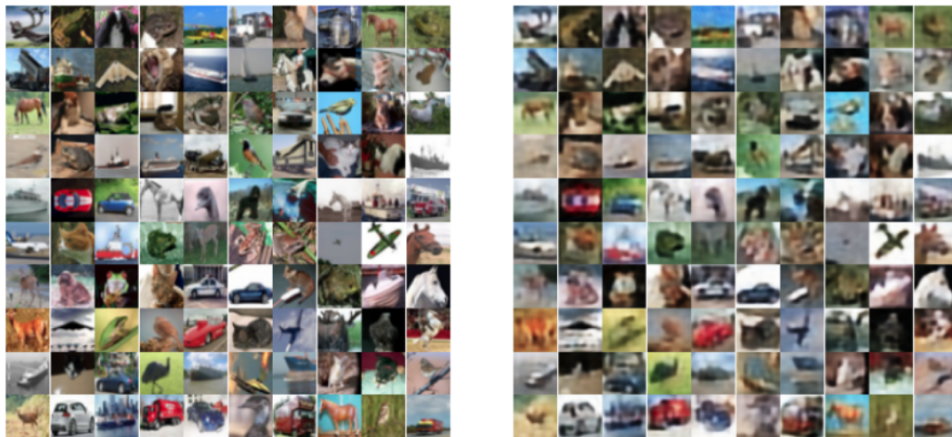


Figure 4.1: Original Images (left) and reconstructed images (right)

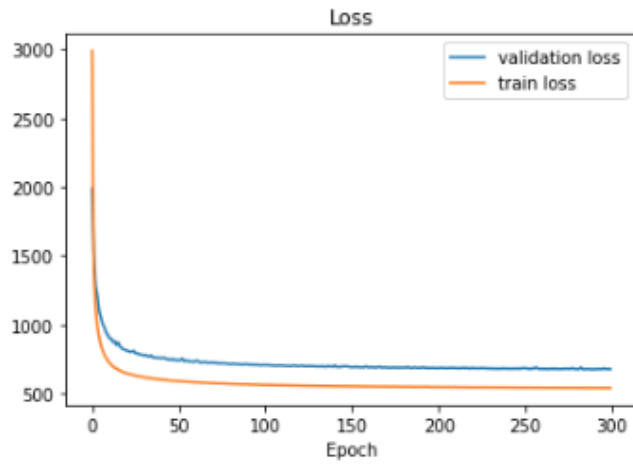


Figure 4.2: Train and validation loss

After converting all the data to its latent space and training the classifier I ended up with an accuracy of 56.71% on the test data. Even with dropout layers added to the classifier, it still seems to over-fit massively. The accuracy graph can be seen in figure 4.3 and the confusion matrix in 4.4. This confusion matrix clearly shows that some classes get confused more than others. Take for instance the car and the truck class.

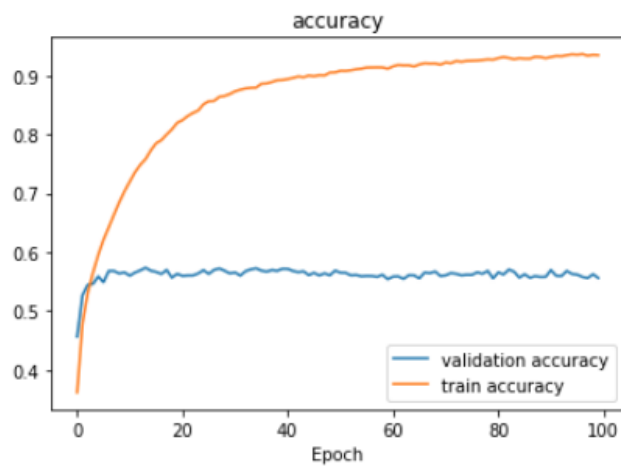


Figure 4.3: Accuracy of the classifier trained on the latent space formed

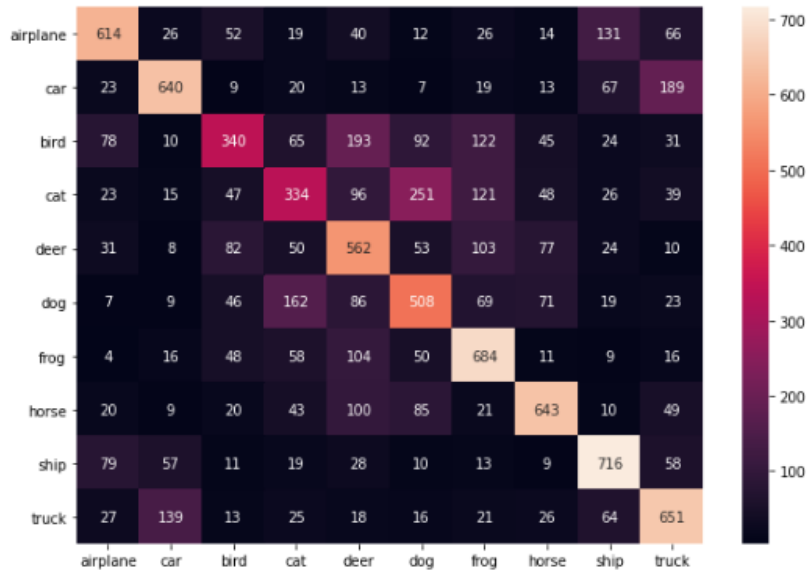


Figure 4.4: Confusion matrix of model with 56.71% classification accuracy

To get a good comparison of how well training and classifying this way was, I now combined the encoder and classifier directly and trained on classification error instead. So ending up with the same model for classification, but with a totally different way of training, namely supervised instead of unsupervised. The accuracy ended up being 68.81% as can be seen in figure 4.5.

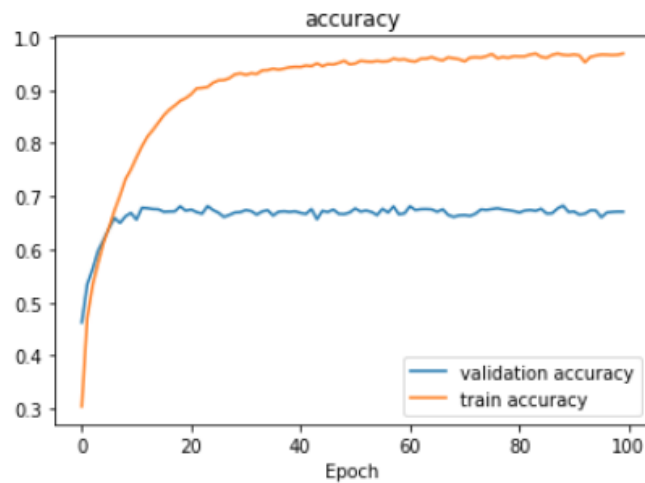


Figure 4.5: Accuracy of training on classification error

To get a proper comparison with a CNN trained on classification error

I implemented and trained the VGG16 model [14]. Since computing power was an issue, it is trained for only 50 epochs. The accuracy after these 50 epochs, ended up being 84.23% as can be seen in figure 4.6.

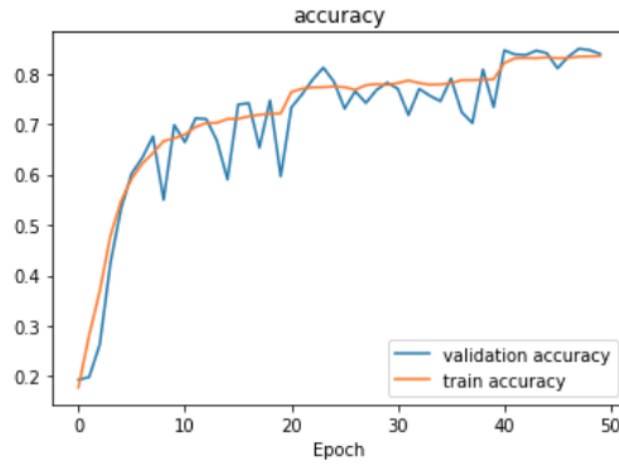


Figure 4.6: Accuracy on VGG16

4.1.2 Predictive processing

Setting the initial latent space representation to the image of a car and use the formula mentioned in methods to update its internal believes to that of a plane I obtained the following results. In 4.5 you can see a decoded image of its latent space for every 10 iterations. In 4.6 you can see the internal believes over those iterations.

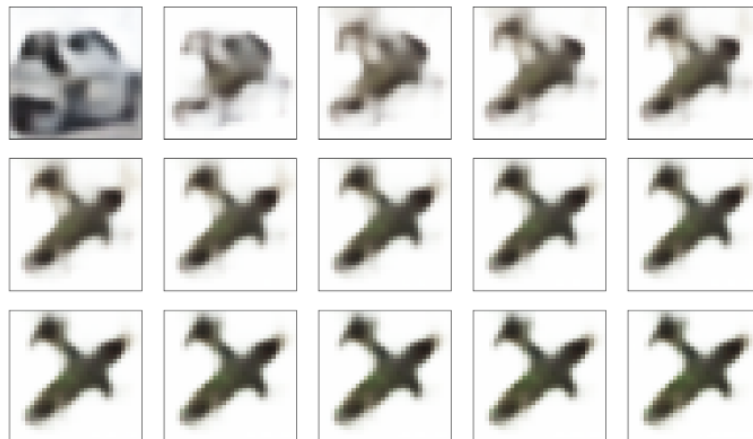


Figure 4.7: From left to right and top to bottom the changes of the internal representation when updating the internal believe from a "Car" to a "Plane"

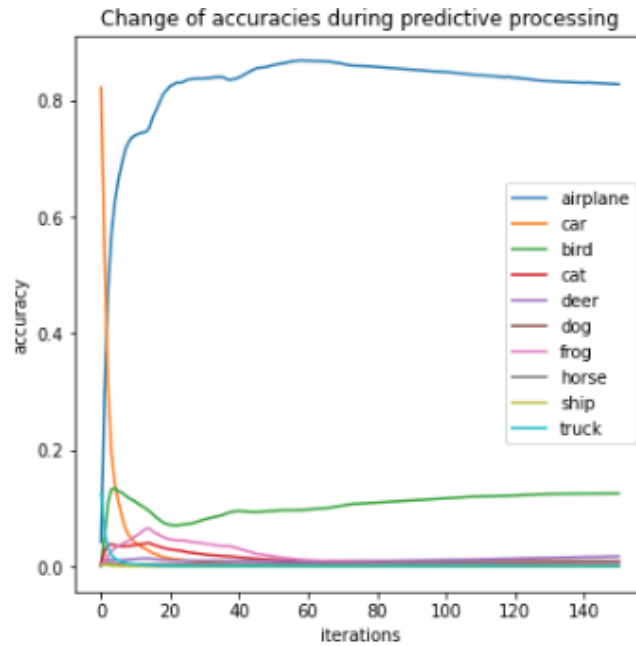


Figure 4.8: The classifiers predictions based on the internal representation after every iteration

These problems might not necessarily correspond to real world problems. However, in real life it would be definitely possible to see only part of a certain object. Since you already have an internal representation of that object you know that the part you are not seeing is still there and can even imagine how that would look like. I tried to model this as well by using the same techniques as before. This time by showing the same plane image, but now I removed one of its wings. The model already has a internal representation of a plane and will try to minimize the free energy again with the same calculation resulting in the following:



Figure 4.9: From left to right and top to bottom the changes of the internal representation when updating the internal believe from a plane with a missing wing to the same plane with both wings

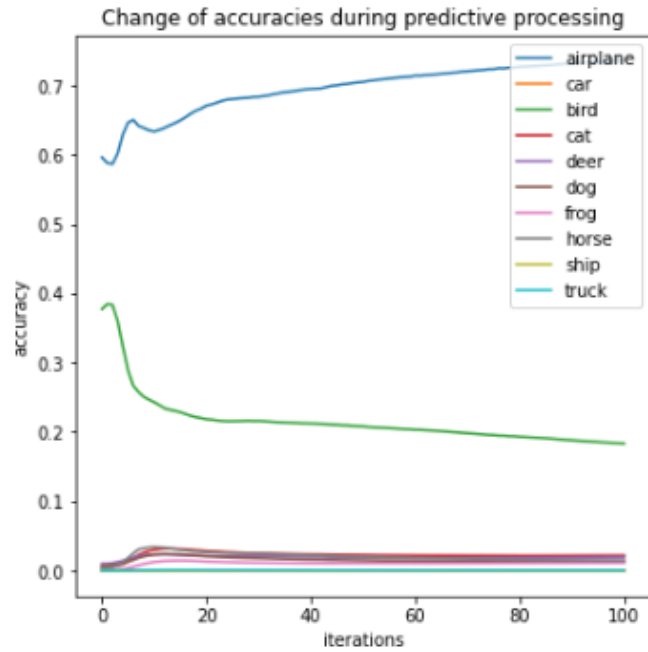


Figure 4.10: The classifiers predictions based on the internal representation after every iteration

4.2 MNIST

4.2.1 Classification

When testing on the MNIST data set, keeping the latent space dimensions the same, namely 128, I got a classification accuracy of 98.63% as can be seen in figure 4.10. Since this data is way smaller in dimensions as compared to the CIFAR-10 data set, the accuracy was also tested with a smaller latent space. With a latent space as small as 8, I managed to get an accuracy of 95.88% as shown in figure 4.10 as well. The original versus the reconstructed images can be seen in Figure 4.9.



Figure 4.11: Original Images (left) and reconstructed images (right) with latent space of 8 dimensions

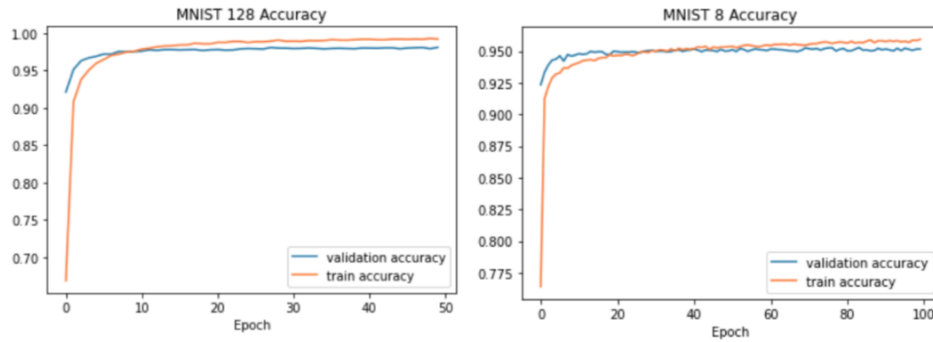


Figure 4.12: Accuracy on MNIST with latent space of 128 on the left and with a latent space of 8 on the right

When compared to the VGG framework that is trained in a supervised manner on classification error, it shows that the VGG model achieves an accuracy as high as 99.43% accuracy. The values of the validation accuracy for the first epochs can be explained by the use of dropout layers.

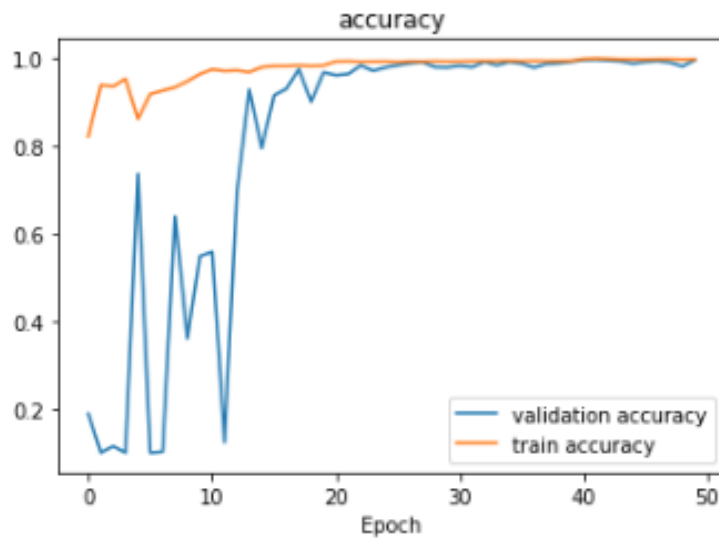


Figure 4.13: Accuracy with the VGG framework over training iterations

4.2.2 Predictive processing

Using the same methods as for the CIFAR-10 data set, now setting the initial latent space to the digit 9 and letting it update to the digit 6, I got the following results. From Figure 4.15 it is clear to see the change of the models internal believes ending after full conversion with the believe of seeing the digit 6.

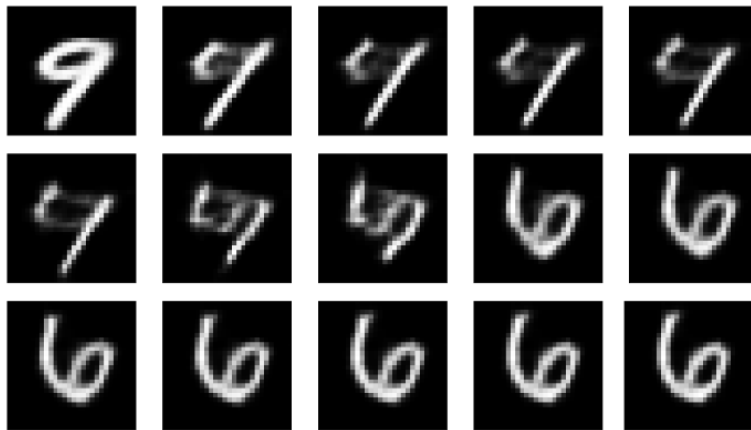


Figure 4.14: From left to right and top to bottom the changes of the internal representation when updating the internal believe from a "9" to a "6"

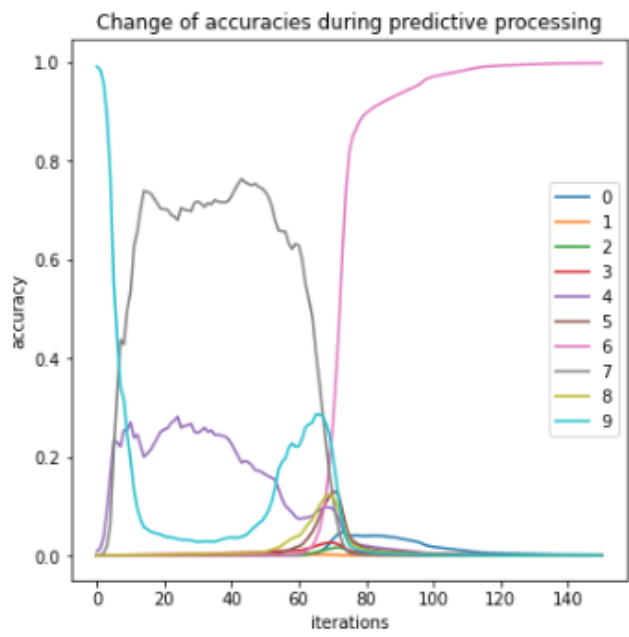


Figure 4.15: The classifiers predictions based on the internal representation after every iteration

Chapter 5

Discussion

5.1 Classification

While the VAE showed good reconstruction performance, the classification accuracy on the CIFAR-10 data set was lacking when compared to the VGG classifier. When testing the performance of the VAE classifier trained in a supervised way directly on classification error, it showed immediate improvements on the classification error. However, still not nearing the VGG implementation and showing massive signs of overfitting. From these facts the conclusions can be drawn that the model's architecture has to be improved. When training both the VAE's encoder part connected to a classifier in a supervised manner on classification error, ideally this would show comparable performance to that of the VGG architecture, which it in this case did not.

There was however, although slight, improvement in classification accuracy when trained in a supervised manner. This can possibly be explained by the fact that purely training on reconstruction might not be ideal for classification tasks. This way, less important parts of the image still are stored in the latent space, causing noise for the classifier. Upon inspecting the confusion matrix, it is evident that classes such as dogs and cats, or cars and trucks get confused a lot. These classes obviously share similar features, however they also share comparable environments, potentially causing the wrong classification. For future work, it might be interesting to go for the approach of training on reconstruction, however, when then training on classification error still update the weights of the VAE to form the latent space more ideally for classification.

While these results were disappointing, the results on the MNIST data set looked promising. With a 98.63% almost nearing the VGG's accuracy of 99.43%. This massive improvement on classification accuracy when compared to the accuracy on the CIFAR-10 data can be explained by the lower dimension data, however the fact that the MNIST digits do not have a

background can possibly play a big role as well. This way the model can fully focus on the reconstruction of the object itself, something that was not possible for the CIFAR-10 data set.

5.2 Predictive processing

When looking at the advantages of modeling the classification task as an inference processes, using bottom-up and top-down modulation, some interesting results were found. For both the MNIST example as for the CIFAR-10 example when changing the internal believes from one object to another, it is clearly visible that the latent space has a clear representation of the object itself. Taking Figure 4.9 as an example, you can see the classification believes changes over the amount of iterations. For a short duration, the digit 7 seems to be the digit the model would classify the internal representation as, when looking at the corresponding images above it is understandable on how these believes are inferred.

The same patterns are also visible for the CIFAR-10 example. Furthermore, when examining the experiment with the plane with one wing cut off, it is interesting to see that even though the model initially classifies the object as a plane, that when the model adjusts for the wing that it is not seeing, the internal believes for seeing a plane goes up and the internal believes for seeing a bird, which obviously shares similar features, goes down. Showing great potential in reducing uncertainty that is always present in the real world.

Combining these results: creating a classifier this way should not be immediately excluded as a method to get a good classification accuracy. However, work on the architecture of the VAE is definitely needed in order to approach the VGG's classification accuracy. Even though classification accuracy was not state-of-the-art as of now, accuracy on the MNIST data set showed that great accuracy is achievable. Furthermore, it already showed great potential of using top-down modulation for reducing uncertainty in classification for both the MNIST as for the CIFAR-10 data set.

Bibliography

- [1] *Variational AutoEncoder*, <https://keras.io/examples/generative/vae/>, (2020).
- [2] Samson Afolabi, *Image classification using the variation autoencoder*, (2020).
- [3] Jan-Jurre Mordang Kathy Schilling Sylvia H. Heywang-Köbrunner Ioannis Sechopoulos Ritse M. Mann Alejandro Rodríguez-Ruiz, Elizabeth Krupinski, *Detection of Breast Cancer with Mammography: Effect of an Artificial Intelligence Support System*, Pubs.Rsna.Org **294** (2020), no. 1, 19–28.
- [4] Dan C Cires, Ueli Meier, Jonathan Masci, and Luca M Gambardella, *IJCAI11-210.pdf*, Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence Flexible (2003), 1237–1242.
- [5] Giacomo Deodato, Christopher Ball, and Xian Zhang, *Bayesian neural networks for cellular image classification and uncertainty analysis*, bioRxiv (2019).
- [6] Karl Friston, *The free-energy principle: A unified brain theory?*, Nature Reviews Neuroscience **11** (2010), no. 2, 127–138.
- [7] Hossein Hosseini, Baicen Xiao, Mayoore Jaiswal, and Radha Pooven-dran, *On the limitation of convolutional neural networks in recognizing negative images*, Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017 **2017-December** (2017), 352–358.
- [8] Diederik P. Kingma and Max Welling, *Auto-encoding variational bayes*, 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings (2014), no. ML, 1–14.
- [9] Lanillos Pradas P.L., *pl-robotdecision*, <https://github.com/pl-robotdecision>, GitHub repository (2020).

- [10] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin, *Variational autoencoder for deep learning of images, labels and captions*, Advances in Neural Information Processing Systems (2016), no. Nips, 2360–2368.
- [11] Rajesh P.N. Rao and Dana H. Ballard, *Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects*, Nature Neuroscience **2** (1999), no. 1, 79–87.
- [12] Shideh Rezaeifar, Olga Taran, and Slava Voloshynovskiy, *Classification by re-generation: Towards classification based on variational inference*, European Signal Processing Conference **2018-September** (2018), 2005–2009.
- [13] Cansu Sancaktar, Marcel A.J. Van Gerven, and Pablo Lanillos, *End-to-End Pixel-Based Deep Active Inference for Body Perception and Action*, ICDL-EpiRob 2020 - 10th IEEE International Conference on Development and Learning and Epigenetic Robotics (2020), no. 741941.
- [14] Karen Simonyan and Andrew Zisserman, *Very deep convolutional networks for large-scale image recognition*, 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2015), 1–14.
- [15] Jan Theeuwes, *Top-down and bottom-up control of visual selection*, Acta Psychologica **135** (2010), no. 2, 77–99.
- [16] Chris Varano and Lytton Ave, *Disentangling Variational Autoencoders for Image Classification*, **4** (2017), 98.
- [17] Zhou Wang and Alan C Bovik, *Mean Squared Error : Love It or Leave It ?*, (2009), no. January, 98–117.
- [18] Pan Zhou and Jiashi Feng, *Understanding generalization and optimization performance of deep CNNs*, arXiv (2018).

Appendix A

Appendix

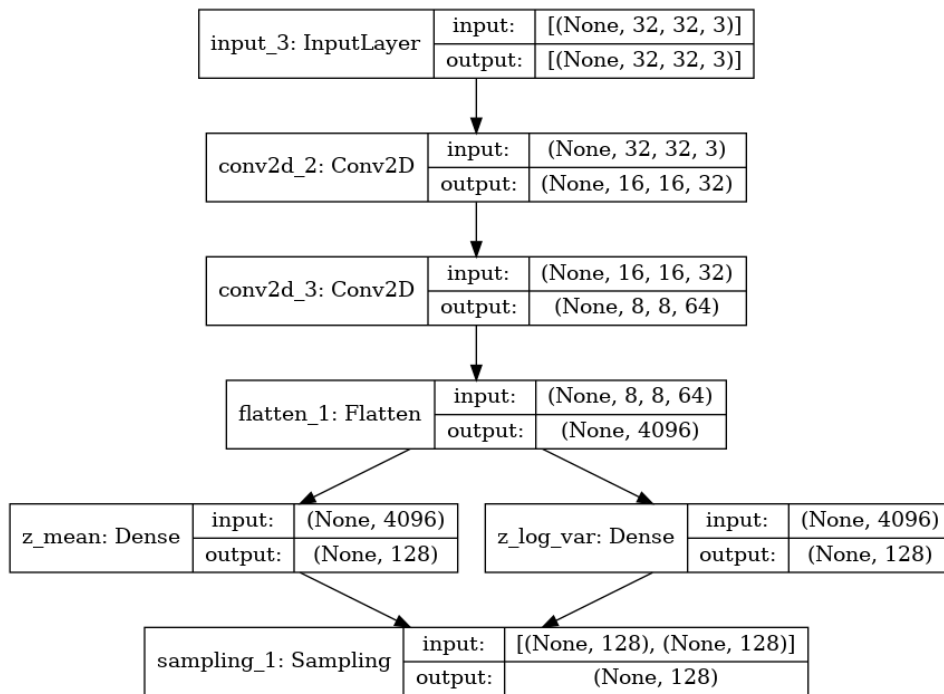


Figure A.1: Encoder

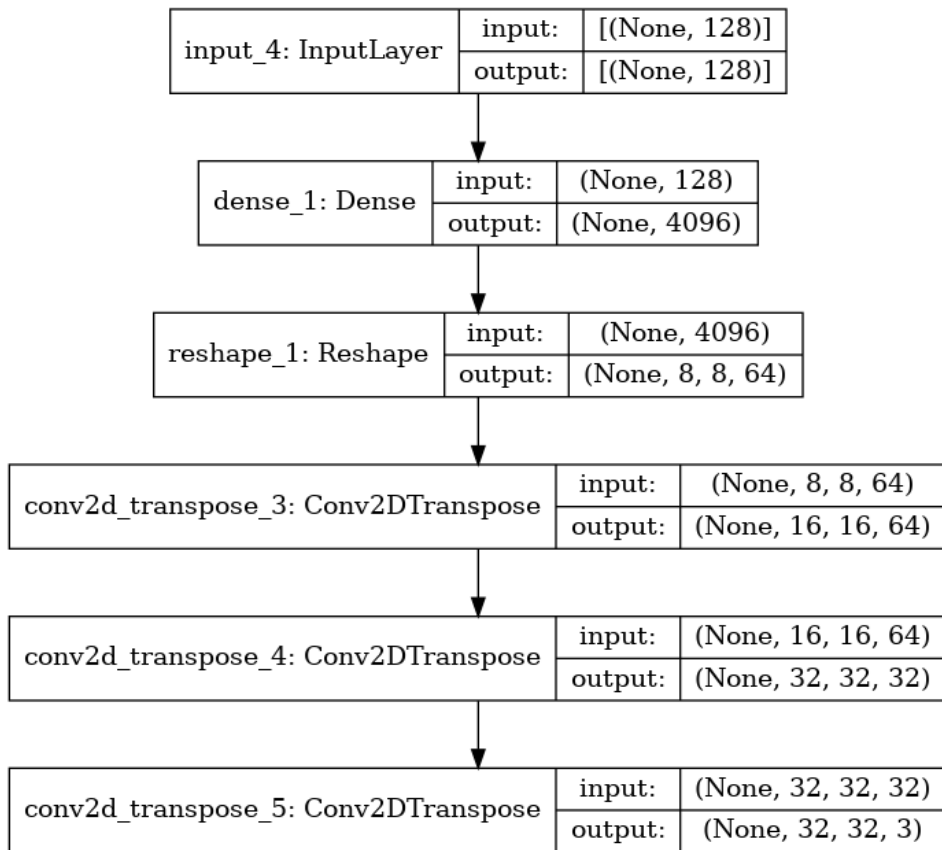


Figure A.2: Decoder