An Investigation Into Tranformation-Invariant Variational Autoencoders

Henry Williams

November 5, 2024

1 Introduction

In this paper we investigate the transformation invariance properties of Variational Autoencoders, how spatial transformations impact the classification accuracy of linear models on the latent space of images, and how we can improve the resiliance of VAEs on spatialy transformed data.

Variational Autoencoders (VAEs) are an architecuture for probablistic generative models, which learn a distribution over their training data [8]. They were first introduced by Kingma and Welling in their 2022 publication [6]. This architecuture is widely used in research for a number of applications, such as unsupervised anomaly detection in medical imaging [1], image generation [12], and classification [2].

One of the most desirable properties of any machine learning task is an invariance to certain transformations. We consider a model to be invariant to transformations if the output for some data $\mathbf{x}$ is equivalent to the output of $t[\mathbf{x}]$ where t applies some transformation to its input. This enables the model to better generalise to real world inputs.

2 Background

2.1 Autoencoding Models

The Variational Autoencoder architecuture acts as the probablistic variant of Autoencoding Models, first introduced by Rumelhart, Hinton, and Williams in their 1986 publication [9]. These models use unsupervised learning to determine efficient encodings of high dimensional data, which can be then used to produce a reconstruction of the original input data. An autoencoding model is comprised of two learnable functions, the encoder $E(\mathbf{x},\phi_1)$, and the decoder $D(\mathbf{z},\phi_2)$. The encoder takes our high dimensional input $\mathbf{x}$, and produces a lower-dimensional latent representation $\mathbf{z}$. The decoder uses this latent representation to reproduce the original input. The full architecuture can be expressed formally as seen in Equation 1. For a higher level, visual representation of this architecture, see Figure 1a

$$\mathbf{f}[\mathbf{x},\phi] = D(E(\mathbf{x},\phi_1),\phi_2) \approx \mathbf{x} \tag{1}$$

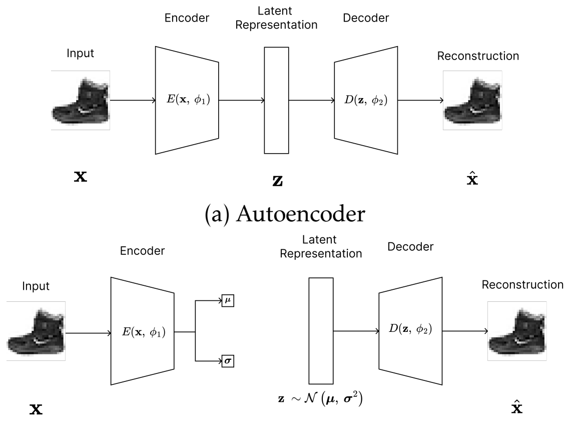
Training such a model requires reducing the reconstruction error between the output from our decoder $\hat{\mathbf{x}}$, and the original input to our model $\mathbf{x}$. To quantify this error, we use the loss function described in Equation 2, known as the Mean-Squared Error.

$$\mathcal{L}_{\phi}(\mathbf{x}) = |\mathbf{x} - \hat{\mathbf{x}}|^2 = |\mathbf{x} - \mathbf{f}[\mathbf{x},\phi]|^2 \tag{2}$$

Thus, the model can be trained by minimising the loss with respect to its parameters ϕ over a training set $X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_N\}$, Equation 3.

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \left[\frac{1}{N} \sum_{i=0}^{N} \mathcal{L}_{\phi}(\mathbf{x}_i) \right] \tag{3}$$

Where $\hat{\phi}$ are the optimal parameters for the model $\mathbf{f}$.



(a) Autoencoder

(b) Variational Autoencoder

Figure 1: A high level overview of both the Autoencoder and Variational Autoencoder architectures

2.2 Variational Autoencoders

As stated earlier in this section, Variational Autoencoders act as the probabalistic variant of an Autoencoder. Instead of predicting the latent representation $\mathbf{z}$, the encoder instead predicts a set of parameters $\boldsymbol{\mu},\boldsymbol{\sigma}$ of a multivariate normal distribution from which the latent representation is sampled from (Figure 1b).

To train such a model, we make use of the Evidence Lower Bound (ELBO) as our loss function, which takes the form of the reconstruction loss, explored above, and the Kullback Leibler Divergence [7], which measures the distance between two probability distributions. The loss function for such a model is as follows in Equation 4.

$$\mathrm{ELBO}[\phi] \approx \log[P(\mathbf{x}|\mathbf{z},\,\phi_2)] - D_{\mathrm{KL}}[q(\mathbf{z}|\mathbf{x},\phi_1) \| P(\mathbf{z})] \tag{4}$$

2.3 Spatial Transformer Networks

Spatial Transformer Networks (STNs) are a kind of artificial neural network which seek to learn spatial transformations of data. They are typically used to transform data before it is fed into another model, in order to increase the robustness of the secondary model to said transformations.

They consist of three components, a localisation network, tasked with learning the transformation parameters of the input, such as rotation, scale, or translation, a grid generator, which uses the parameters predicted by the localisation network to produce a sampling grid which is then used by the final component, the sampler, to transform the original input based on the predicted transformation parameters.

This model was first introduced by Jaderberg et al. in their 2016 publication [4], and they are widely used throughout machine learning research to improve invariance to spatial transformations.

2.4 Linear Classification

Linear classification models predict linear decision boundaries separating different classes based on the features of the data. These boundaries are then used to map unseen data to a class label. Common linear classification algorithms include Logistic Regression, which make use of the sigmoid function applied to a linear combination of the input features and predict the probability of the data belonging to each of the classes, and Support Vector Machines, which find a hyperplane which maximises the margin between different classes in the data space.

While deep learning methods may find a more accurate solution for classification problems, they suffer from a number of problems, such as high complexity, and the lack of interpretability in their classifications. Linear classification models however, are far less complex, offer interpretability, and are less computationally complex. However, they are unable to capture more complex interactions, and are very susceptable to outliers in the data.

3 Methods

3.1 Data and Preprocessing

We chose to select the FashionMNIST [13] dataset for our experiments. This dataset consists of 70,000 greyscale images with a 28×28 resolution, each containing one of 10 different types of common articles of clothing, such as shoes, coats, bags, etc. This dataset was selected as it provides a challenging task for our models, while still being relatively quick to train on, due to its low resolution.

The images in this dataset were normalised such that the mean of the dataset take the value 0, and a standard deviation of 1. This is a common practice in image classification problems, as it ensures each of the features contribute equally to the model, thereby increasing both model performance and stability [10]. We refer to this dataset throughout this paper as the "base dataset".

In order to test the performance of the model on spatially transformed images, we create a secondary dataset, which we refer to as our augmented dataset, consisting of the base dataset with small geometric transformations applied to each of the images. We chose to focus on small rotations of ±10°, translations of ±4 pixels in both the x and y axes, and scaling operations by ±20%. Each of these transforms have a 50% chance of being applied to an image in the dataset, which is independent of any other transformations that may have already been applied to the image. The two datasets we use for our experiments are shown in Figure 2.

3.2 Models

In our experiments, we test the performance on two variations of a VAE architecture, namely the original architecture outlined in [6], and the categorical VAE, outlined in [5]. We chose to experiment with the Categorical VAE (CVAE) as they better understand categorical distributions of data, which we hope will lead to better classification accuracy. Both of these models encode their input to a 10-dimensional latent representation. However, the encodings generated by the CVAE require the inclusion of a categorical distribution of each of the classes in each latent dimension, thus the latent representation takes the form of a 10-dimensional square matrix.

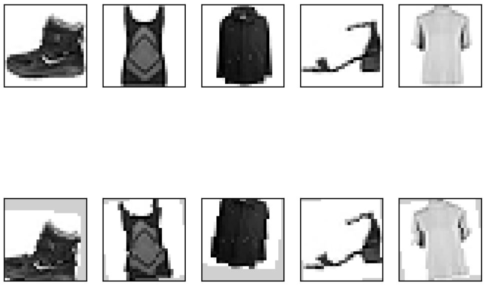


Figure 2: A sample of images in our base and augmented datasets

The implementations for both of these models are based on the excellent work of Subramanian in their *PyTorch-VAE* github repository [11]. However, some changes were made to the architecture in order for it to work correctly on the FashionMNIST dataset. We also reduced the number of hidden layers in both the encoder and decoder from 5 to 3 to decrease the time spent training our models.

We also chose to investigate the effect of Spatial Transformer Networks, as they seek to normalise the spatial representation of our data, reducing the effect of any transformations applied to the datasets. Our implementation for this model was based on the work done by Hamrouni in their *Spatial Transformer Networks Tutorial* [3], and was trained with the augmented dataset as its inputs, and the base dataset as the ground truth data for 20 epochs using a learning rate of 0.001.

We test the performance of both of these models on the base, and the augmented datasets, both with and without the spatial transformer network pre-processing its inputs. The models were each trained for 20 epochs, with a learning rate of 0.001, using pytorch's AdamW implementation of gradient descent.

To calculate the loss of these models, we take both the reconstruction error between the reconstruction of the input from the decoder, and the original input to the encoder using the mean-squared-error loss function, and the KL divergence [7] between the predicted probability distribution, and the true probability distribution weighted by 0.001.

To classify the images based on their latent space representation, we use the Logistic Regression model, due to its excellent performance, and ease of use. This model is trained on the encodings of the training set images generated by the encoder of our models, and their associated labels.

3.3 Metrics

At each episode of training, we collect the average loss of our models on both the base test set, and the augmented test set, in order to see how the models performance changes on both tasks throughout the training process.

The classification accuracy for each model is given by the number of correctly classified data samples over the total number of test samples. We collect the accuracy for both the base dataset, and the augmented dataset.

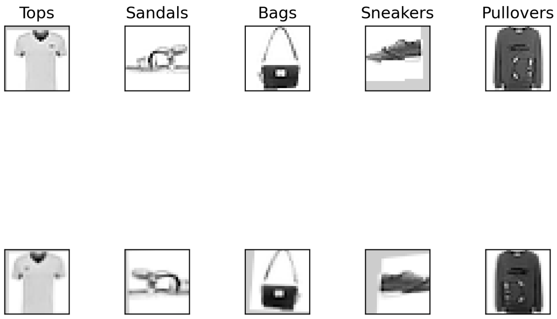


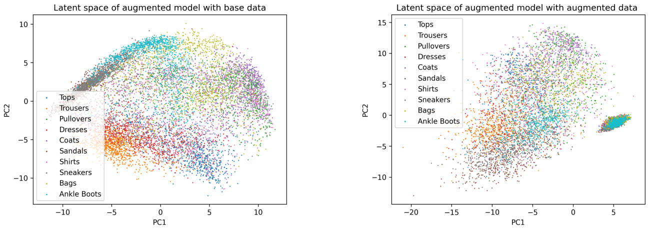
Figure 3: Inputs to the spatial transformer network, and their transformed outputs

4 Results

The results from our experiments are displayed in Table 1. We found a Categorical VAE achieved the best classification accuracy on the spatially transformed dataset. Interestingly, we also found that the use of a Spatial Transformation Network does not have as pronounced effect on the classification accuracy as expected, at times even causing a net-negative effect. We believe that this could be due to the fact that the Spatial Transformer Network was trained prior to the training of each of the models. This was in an attempt to standardise the model across our experiments, in addition to the fact that the backward pass operation for the 2D Grid Sampler pytorch module is not implemented for the apple MPS backend, which was used for the training of the models in this investigation. To bypass this restriction, the STN was trained on the CPU, and doing this for each of the tested models in this investigation would significantly increase the time spent training. Future work could investigate the impact of training the spatial transformer at the same time as the various VAE models. The outputs from our trained Spatial Transformer Network are shown in Figure 3.

We also see that despite having significantly higher test loss values, Conditional VAEs have a significantly higher classification accuracy across both the base test dataset, and the augmented test dataset. However, the conditional VAE encodes the images to a 100 dimensional latent representation (10 latent dimensions × 10 class labels), whereas the VAE encodes images to a 10 dimensional latent representation. This suggests that the CVAE is only able to achieve higher classification accuracy because it has a larger latent representation to work with. Future work in this field should consider how the dimensionality of the latent representation effects the classification accuracy of the tested models and architectures.

Furthermore, we also observe the classification accuracy between the base and augmented datasets are largely consistent between each of our experiments. On average, this difference is approximately 10%. The consistency of this difference indicates that each of the models we tested had little to no effect on their invariance to spatial transformations, and that other architectures and training procedures should be tested. However, we do see that across our experiments, models trained on an augmented dataset exhibit better classification accuracy on the base dataset despite having worse classification accuracy on the augmented test set. This is shown in the latent space plots of our test datasets (Figure 4), which is used to classify data by a linear classification model. This diagram shows a well separated set of class clusters in 4a, which would likely be easier to classify. Figure 4b however demonstrates little clustering between classes, instead showing a single cluster with multiple classes present, and a sparse distribution of samples outside this primary cluster. The distribution of points resembles a gaussian distribution, indicating the model struggles to learn properties of its training data, and to meaningfully encode this information into its latent representation, thus reducing the classification accuracy.

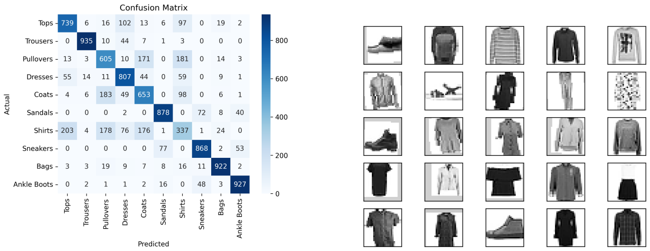


(a)

(b)

Figure 4: Latent space plots of both the base (4a), and augmented test sets (4b)

As shown in Figure 5a, we see the classification model has the most difficulty when attempting to classify the shirt and pullover classes. This suggests that the latent representation of images with these classes are clustered together, leading to the classification model struggling to correctly predict the class of the image. This is somewhat expected, as these images have a similar shape (Figure 5b). We also see from this figure that incorrectly classified images include some un-augmented images, suggesting the errors in classification are not necessarily caused by spatial transformations, and likely a cause of the model struggling to encode images with similar shapes, but distinct classes.



(a) Confusion matrix of the linear classification model

(b) A sample of incorrectly classified images

5 Conclusion

In this investigation, we have investigated how spatial transformations impact the classification accuracy on the Fashion-MNIST dataset, and potential mitigations such as making use of Spatial Transformer Networks, and training our models on a dataset with random small geometric transformations. We tested the effects of spatial transformations on the linear classification of data using the latent representation of data generated by both the original Variational Autoencoder ar-

Experiment	Train Dataset	Model	STN	Accuracy		Test Loss	
				Base	Aug	Base	Aug
1	Base	VAE	False	0.779	0.6463	0.2389	0.4427
			True	0.7478	0.6093	0.1826	0.2244
2	Augmented	VAE	False	0.7713	0.7133	0.2574	0.2808
			True	0.757	0.6384	0.1889	0.2045
3	Base	CVAE	False	0.8564	0.7452	7.107	13.0395
			True	0.8515	0.7492	5.2893	6.5253
4	Augmented	CVAE	False	0.851	0.7635	7.6776	8.6028
			True	0.851	0.7504	5.4501	5.9202

Table 1: Results obtained from our experiments

chitecture, and the Categorical Variational Autoencoder. We found that the best classification accuracy was achieved by a CVAE, trained on an augmented dataset, without the use of a spatial transformer network to normalise spatial properties of images before training. Future work could investigate the impact of increasing the dimensionality of the latent representations of data in the standard VAE architecture, and the effects of training the spatial transformer network at the same time as the Variational Autoencoding models.

References

[1] Christoph Baur et al. "Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study". In: *Medical Image Analysis* 69 (2021), p. 101952. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2020.101952. URL: https://www.sciencedirect.com/science/article/pii/S1361841520303169.

[2] Lahiru D. Chamain, Siyu Qi, and Zhi Ding. "End-to-End Image Classification and Compression With Variational Autoencoders". In: *IEEE Internet of Things Journal* 9.21 (2022), pp. 21916–21931. DOI: 10.1109/JIOT.2022.3182313.

[3] Ghassen Hamrouni. *Spatial Transformer Networks Tutorial*. 2017. URL: https://pytorch.org/tutorials/intermediate/spatial_transformer_tutorial.html.

[4] Max Jaderberg et al. *Spatial Transformer Networks*. 2016. arXiv: 1506.02025 [cs.CV]. URL: https://arxiv.org/abs/1506.02025.

[5] Eric Jang, Shixiang Gu, and Ben Poole. *Categorical Reparameterization with Gumbel-Softmax*. 2017. arXiv: 1611.01144 [stat.ML]. URL: https://arxiv.org/abs/1611.01144.

[6] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML]. URL: https://arxiv.org/abs/1312.6114.

[7] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79 –86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

[8] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023. URL: http://udlbook.com.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning internal representations by error propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, 318–362. ISBN: 026268053X.

[10] Dalwinder Singh and Birmohan Singh. "Investigating the impact of data normalization on classification performance". In: *Applied Soft Computing* 97 (2020), p. 105524. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2019.105524. URL: https://www.sciencedirect.com/science/article/pii/S1568494619302947.

[11] A.K Subramanian. *PyTorch-VAE*. https://github.com/AntixK/PyTorch-VAE. 2020.

[12] Arash Vahdat and Jan Kautz. *NVAE: A Deep Hierarchical Variational Autoencoder*. 2021. arXiv: 2007.03898 [stat.ML]. URL: https://arxiv.org/abs/2007.03898.

[13] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: cs.LG/1708.07747 [cs.LG].