Application of variational autoencoders for aircraft turbomachinery design

Jonathan Zalger
SUID: 06193533
jzalger@stanford.edu
SCPD Program
Final Report
December 15, 2017

1 Introduction

1.1 Motivation

Machine learning and optimization have been used extensively in engineering to determine optimal component designs while meeting various performance and manufacturing constraints.

As the complexity of the design problem increases and additional analysis disciplines are included, more and more features must be added to capture nuances in the underlying physics. In many scenarios, it can be difficult or impractical to explicitly define these features or targets in advance. In addition, experienced engineers often rely on intuition and the evaluation of analysis plots to make design decisions.

By teaching an algorithm to identify images of superior vs. inferior plots, we may be able to capture more abstract engineering intuition into the model which is only attained through extensive experience.

1.2 Approach

In this project, the goal is to train a variational autoencoder[1] to model supersonic airflow characteristics of a NASA rotor 37 compressor blade [2] in response to changing mass flow conditions. The input vector includes a flattened colour image representing the relative mach number contours as well as the associated mass flow boundary condition. Once the trained model is built, novel mach contour plots and boundary conditions can then be sampled from the posterior latent distribution given constraints on the latent manifold. This latent space sampling approach provides an alternative to parametric design space exploration where a cost function is optimized using the engineering parameters directly and is the rational for selecting a VAE.

1.3 Related Work

At the time of writing, a survey of the main aerospace and turbomachinery journals revealed no similar work applying generative VAE models in an aerospace design context. This is likely due to both the recent publication of the variational autoencoding algorithm as well as the dominance of parametric optimization methods. Recent state-of-the art modeling in the aerospace field involves the application of surrogate models [3] to accelerate the time to prediction and reduce the overall number of simulation points required to perform a design optimization. To some extent, the presented application of a VAE can be seen as a surrogate for the underlying simulations, although the generative/sampling approach differs from typical polynomial, kriging, or gaussian process based models [4].

Outside of the aerospace field, there have been recent applications of variational autoencoders in pharmacutical drug optimization [5][6] and material design [7][8]. Although presented with different compositions of input vectors, these works provided useful baseline references for setting

the initial autoencoder paremeters and network architecture design. In addition, the applications of latent space arithmetic and convolutional encoders [6] are inspiring for future work.

2 Dataset & Features

The dataset for this project is generated via a numerical simulation technique known as computational fluid dynamics (CFD). These simulations calculate the flow field around the airfoil shape at various boundary conditions (one set per simulation). Two example training images are shown in figure 1. In the left image, a shock wave can be seen near the leading edge of the airfloil, and in the right image the shock wave is driven deeper (swallowed) into the flow passage between blades. These states constitute the two main states in the dataset and the main states to be predicted by the model. Along with the flow field image showing contours of relative mach number, the corrected mass flow boundary condition is also saved for use in training.

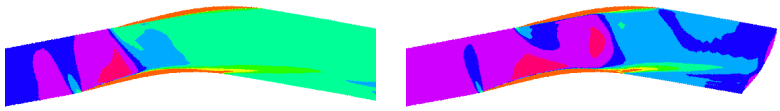

Figure 1: Example training set images showing relative mach contours. A leading edge shock wave (left) and deep (swallowed) shock wave (right) can be seen.

To generate the thousands of CFD simulations required for the dataset, an automated workflow was used which builds a Design Of Experiments (DOE) matrix using latin hypercube sampling (LHS). This results in a random sampling of cases which covers the full mass flow domain [9]. A CFD simulation was then run for each mass flow condition generated in the DOE matrix.

As simulations are completed, a script is run which collects and processes each design point. This involves extracting the relative mach number data from a specified plane in the 3D flow field which is then scaled, rotated, and saved as an image. Prior to training, the dataset is standardized to zero mean and unit variance.

In total, 4962 simulations were completed and split into training (4219), development (498), and final test (245).

3 Algorithm

The initial encoder-decoder network architecture was chosen after a thorough review of recent literature on similar applications in the materials and pharmacutical fields [5][8]. Linear activations were chosen for the output layer of the encoder as the outputs are intended to represent the mean and variance of the latent posterior. Sigmoid activations were chosen for the decoder output after which the values are inverse transformed through a standardization scaler. Exponential linear units (ELU) were chosen for all hidden layer activation functions based on recent VAE research demonstrating an improvement in learning behaviour over ReLu and Leaky ReLu implementations [10][5]. Both the encoder and decoder have two hidden layers with 200 neurons each based on similar models [1] and for training efficiency. A hyperparameter study also confirmed this to be the optimal value allowing a reduction in overfitting while not significantly impacting the quality of the results. Over the course of the project, dropout was also added to the second hidden layer of the encoder network to improve regularization of the latent space. Dropout was not used in the decoder as it resulted in obvious artifacting of the generated images. The dimension of the latent space was restricted to two. This enables the latent space to be directly interpreted which is valuable in the context of this project. Gaussians were used for all approximating distributions.

The objective function of the model, shown in equation 1, is based on maximizing the variational lower bound representing the reconstructive error of the autoencoder and the KL divergence between the estimated latent distribution $q_\phi(z|x)$ and a normal gaussian prior, $p_\theta(z)$ [1]. The KL divergence term acts to regularize the latent space and prevent the model from simply memorizing the training dataset. The encoder and decoder networks are used to approximate the posterior distributions of the latent space $q_\phi(z|x)$ and output $p_\theta(x|z)$.

$$L(\theta, \phi, x^{(i)}) = \frac{1}{L}\sum_{l=1}^{L}(\log p_\theta(x^{(i)}|z^{(i,l)})) - D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) \quad (1)$$

The model was implemented in Python using the TensorFlow framework [11] (v1.4) for efficient distributed calculation. The Scikit-Learn package [12] was used for preprocessing and standardization. Model calculations were primarily conducted on a quad-core PC and 16 core Haswell-based HPC cluster.

4 Results & Discussion

4.1 Optimization and Hyperparameter Selection

Several optimization algorithms were tested including SGD, Adagrad, and ADAM. The Adagrad method consistently provided the most robust training. This is likely due to the efficiency of the adaptive learning rate and its favourable performance on sparse datasets [13]. The ADAM method was also explored although the performance was not as robust as Adagrad. Given its similarity to Adagrad and theoretical advantages, it is likely that further hyperparameter tuning of the ADAM implementation may produce improved results.

The initial learning rate was chosen at 0.001 after a parameter sweep from 0.0001 to 0.1. Also, a constant parameter γ was added to the KL divergence term in 1 to control the strength of the regularization. The γ value was set to 2.5 after a sweep in the range of 0 to 3. This parameter is discussed further in section 4.4. Batch size was tested from 32 to 256 and set to 64. This decision was made to optimize the model performance on the primary hardware used for computation (Dell M series notebook). Variations in batch size in the range tested had minimal impact on the performance metrics. The selection criteria for all hyperparameters included likelihood convergence performance and similarity of SSIM indicies on the training and development sets.

4.2 Latent Space Exploration

To determine how well the latent space was generalizing, three visualizations were created assessing the performance of the encoder, the visual reconstructions, and mass flow distribution.

To assess the regularization performance of the KL divergence term in 1, the development set points were passed through the encoder and the resultant latent values were plotted in 2D space. As the KL divergence is being computed with respect to a unit gaussian, the model should be driving towards a representation that is similar, however we do expect clusters or features to emerge which represent the intrinsic characteristics of the flow field being modeled. In figure 2, we can see two distinct clusters emerging, which represent the two dominant flow states found in the dataset (leading edge and swallowed shocks). The distributions above and to the side of the main plot illustrate the regularization of the individual latent variables and are discussed further in section 4.4.

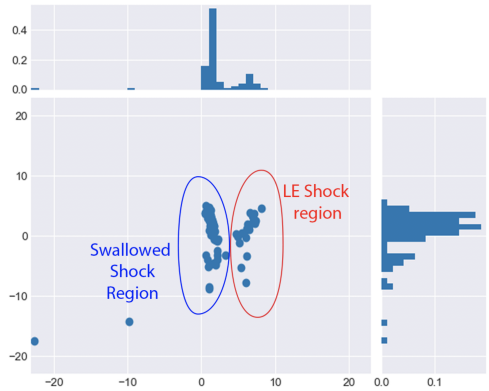

Figure 2: Visualization of test points encoded into the 2D latent space

To confirm the clusters depicted in figure 2 indeed correspond to meaningful regions of the flow field, a 5x5 linear spaced sampling of the latent distribution was conducted from -10 to 10 in both dimensions. These results are presented in figure 3.

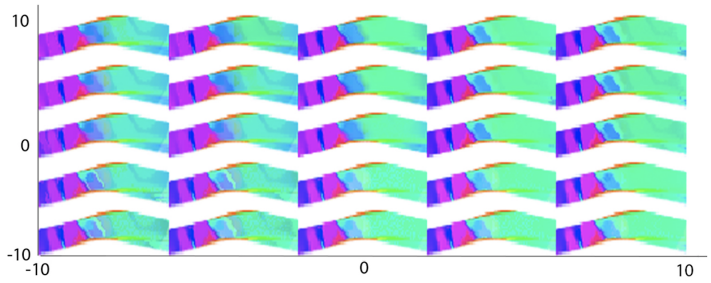

Figure 3: Reconstructed images as sampled from the 2D latent space

Although there is still uncertainty due to the training set size, the model is able to represent the distinct leading edge shock and uniform downstream flow (bottom right) as well as the more uniform upstream flow of the swallowed shock condition (upper left). Note that the model had more difficulty correctly determining the mach contours in the swallowed shock regime, which is further discussed in section 4.4.

4.3 Reconstructive Similarity

One popular way of measuring the performance of VAE algorithms is assessing the log-likelihood of the reconstructive posterior $p_\theta(x|z)$ [1]. Although this is convenient for hyperparameter tuning and comparing architecture, a more intuitive approach is to use a similarity metric between the input and reconstructed image. Initially, the Mean Squared Error (MSE) metric was explored, but this was determined to be too sensitive for practical comparisons. Some softness in the reconstructed model is expected given the regularization affect of the KL divergence term. The image Structural Similarity metric was chosen instead which includes perceptual differences such as texture and exposure[14][15].

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2}$$

A perfect (100%) SSIM score would indicate poor generalization by the algorithm, whereas a low score would indicate the model is struggling to reproduce a coherent image. The target for this project was for high (but not perfect) SSIM scores, and more importantly, similar scores across the training, development, and test sets.

Training Set (4129 samples)	90.5 %
Development Set (498 samples)	92.2 %
Test/Validation Set (245 samples)	90.3 %

Table 1: Reconstructive Similarity (SSIM)

Table 1 presents the calculated SSIM scores across each set. The data shows similar values across all three with the development set slightly more accurate. It is possible this is correlated to the test procedure, although more likely due to the random composition of the datasets which were generated via LHS sampling.

4.4 Error Analysis and Mitigation

In early error analysis, the largest issue identified was overfitting. This is evident in early latent space models where at least one of the latent variables would not conform to a gaussian distribution efficiently (figure 4 left). In addition, the joint latent space exhibited highly correlated and widespread encoding, not characteristic of the input data. Further evidence of overfitting can be seen plotting

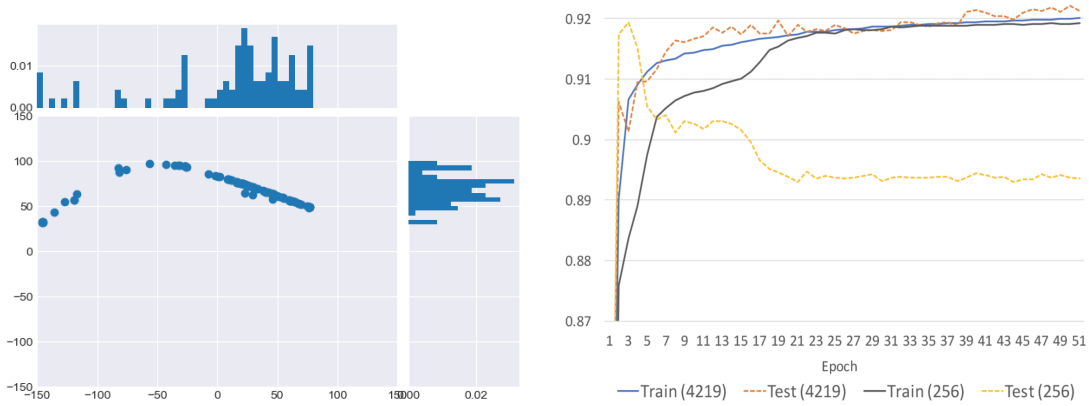

Figure 4: Evidence of overfitting: Poor latent space regularization (left) and large variance between train-development SSIM across dataset sizes (right).

SSIM indicies against dataset size (figure 4 right) where training values are markedly higher than development at lower dataset sizes.

Initially, the γ parameter was increased to strengthen the regularization. However this was not successful at improving the latent space distribution. A more dramatic improvement was seen by adding dropout to the second layer in the encoder network. In addition, the image pixel size was reduced and an additional 3600 simulations were added to the dataset.

When examining figure 3, we can also see the model does a better job predicting shock wave formation on the leading edge of the airfoil, than shock waves ingested deeper into the blade passage. Due to numerical instability in the CFD solver when simulating higher mass flow rates, a phenomena directly related to the underlying gas dynamics [16], the yield of solutions in these flow states was lower than with other mass flow values. This created a bias in the dataset which although addressed through additional simulation, was still present in the final results.

5 Conclusion

In this project we presented an alternative approach for engineering design space exploration. By training a generative variational autoencoder with engineering functional parameters and flow field visualizations, we demonstrated the ability of a model to encode a high dimensional and complex transonic flow regime into a two dimensional latent space. Distinct latent clusters emerged representing the dominant flow regimes in the dataset. We then demonstrated the ability to selectively sample from favourable regions, creating novel flow fields with their associated mass flow boundary conditions. The SSIM metric was utilized as an interpretable way to gauge the reconstructive performance of the generated flow fields and consistent results were presented across the training, development, and test datasets.

5.1 Future Work

The most advantageous future work involves addressing the overfitting and dataset bias issues. These can be improved immediately through expanding the dataset size, particularly in the swallowed shock state where CFD convergence issues reduced the overall sample yield. The use of convolutional layers may also improve the generalization capability [7].

There are also many other exciting applications within the aerospace field including the variation of geometric parameters and inclusion of other boundary conditions. Exploring the capability of latent space arithmetic may also provide new ways to explore high dimensional design spaces in a tractable and interpretable manor. Generative models have the potential of creating a new paradigm in engineering design practices and provide opportunities for significant future research.

References

[1] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv.org*, December 2013.

[2] J. D. Denton. Lessons from rotor 37. *Journal of Thermal Science*, 6(1):1–13, Mar 1997.

[3] C. Boursier Niutta, E. J. Wehrle, F. Duddeck, and G. Belingardi. *Surrogate Modeling in the Design Optimization of Structures with Discontinuous Responses with Respect to the Design Variables – A New Approach for Crashworthiness Design*, pages 242–258. Springer International Publishing, Cham, 2018.

[4] J. Sobieszczanski-Sobieski and R. T. Haftka. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization*, 14(1):1–23, Aug 1997.

[5] Ladislav Rampasek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr.VAE: Drug Response Variational Autoencoder. *arXiv.org*, June 2017.

[6] Rafael Gómez-Bombarelli, David K. Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2016.

[7] Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Timothy D Hirzel, Jorge Aguilera-Iparraguirre, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using variational autoencoders.

[8] Edward Kim, Kevin Huang, Stefanie Jegelka, and Elsa Olivetti. Virtual screening of inorganic materials synthesis parameters with deep learning. *npj Computational Materials*, 3:53, 2017.

[9] Jaroslav Menčík. Concise Reliability for Engineers. *Concise Reliability for Engineers*, (Chapter 16), April 2016.

[10] ELU-Networks: Fast and Accurate CNN Learning on ImageNet. pages 1–1, December 2015.

[11] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[14] David M Rouse and Sheila S Hemami. Understanding and simplifying the structural similarity metric. In *Image Processing, 2008. ICIP 2008. 15th lEEE International Conference on*, pages 1188–1191. IEEE, 2008.

[15] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[16] Bo Song and Wing F Ng. Performance and Flow Characteristics of an Optimized Supercritical Compressor Stator Cascade. In *ASME Turbo Expo 2005: Power for Land, Sea, and Air*, pages 199–210. ASME, 2005.