Reduced Order Models for Fluid Flow With Generative Adversarial Networks (GANs)

by

Mirko Kemna

Interim Thesis and Literature Report

Student Number: 5606896

Faculty Electrical Engineering, Mathematics, and Computer Science

Delft University of Technology

Chapter 1

Introduction

Chapter 2

Fluid Modeling

2.1 Governing Equations

2.2 Numerical Solution

Chapter 3

Machine Learning

3.1 Artificial Neural Networks

3.2 Generative Adversarial Networks

3.2.1 Motivation

Before a neural network can be trained, one has to lay out the "grading system" that is the cost function. For some types of problems, such as classification, the choice is typically straightforward. Take for instance the task of detecting whether an image contains a cat. Given a set of labeled training data, we penalize incorrect guesses by the network, taking into account its level of confidence with the cross-entropy cost function. A much more difficult task, however, is to *generate* realistic images of cats. The key difference is that for the classification problem, we always know exactly in which direction the output should be moved in order to improve the result. For the generative problem on the other hand, this is all but clear.

A naive approach would be to compare generated images to images from the training set using a simple pixel-based loss. However this approach will likely only produce results akin to a superposition of training data, instead of realistic images. One may try to improve upon it by adding additional terms to the cost function such as penalties on blurred edges etc., but the network will still be essentially unaware of any meaningful conditional distributions underlying cat images and truly convincing results are likely to remain elusive.

Taking a step back and seeing the problem from a higher vantage point, we may come to the conclusion that ideally, we would like to abandon simple cost functions in favor of having an observer that can judge the generated images for their realism. One may at first consider relying on human subjects for this task, similar to how they are usually employed to generate the labels for training data. However, apart from the fact that this would make the training process prohibitively slow and expensive, there is an even more fundamental limitation to this idea. In order to be useful for training, it is not sufficient to obtain a value representing the credibility of the image, but we also need to know how every pixel affected that judgment (and therefore what the local gradients are). This requirement is clearly infeasible with human subjects.

A much more practical approach would therefore be to first train a separate neural network on the aforementioned classification task of detecting images of cats, and afterwards use it to train the generative network (generator). This can be done by using the generated output as input for the discriminative network (discriminator), and back-propagate through both networks to obtain information on how to change the generators weights in order to increase the output of the discriminator (i.e. its confidence that the generated output is real). The issue with this approach is that the generator, since it has in a sense perfect knowledge of the inner workings of the discriminator, will typically find "shortcuts" to overpowering the discriminator by exploiting its imperfections and biases. In order to circumvent this, we train both generator and discriminator together in an adversarial zero sum game. This way both networks can be kept in balance, and improve each other until a satisfactory solution is reached. This framework is known as generative adversarial networks (GANs). It was first proposed in [1].

3.2.2 Fundamentals and Variants

The goal of training a GAN is typically to find a useful mapping from a latent space $\mathbf{z}$ to the probability distribution $p(\mathbf{x})$ underlying the training data $\mathbf{x} \in X$. A classical example is generating images of human faces. However, GANs can also easily be extended to work with conditional distributions $p(\mathbf{x}|\mathbf{c})$. For the example of facial generation, $\mathbf{c}$ could be as simple as a value $\in [0, 1]$ representing the lightness of the skin, or as complex as a sketch acting as basis for the generation.

Generative problems fall on a spectrum; on the one side there are setting where we are actively interested in sampling the data distribution through the latent space, such as when building a universal face generator. On the other side, we have more translative problems, which may or may not be strictly deterministic, but for which we are typically only interested in getting a single, high-quality result (e.g. image upscaling). In this case, we may choose to omit the latent space altogether.

...

3.3 Scientific Machine Learning

As a general purpose tool for pattern recognition and automated data analysis, machine learning continues to find many applications in scientific research. An emerging field is the use of machine learning, and neural networks in particular, as surrogate models for complex problems from fields such as fluid mechanics. The hope is that the networks can learn useful relations from data, just as a human can build up a physical intuition. If successful, this could produce surrogate models that can be evaluated at much lower computational effort compared to a full numerical simulation, if perhaps at the cost of some loss of generality.

...

Chapter 4

Initial Experiments

4.1 Methodology

4.2 Results

Chapter 5

Project Scope

Bibliography

[1] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).