Stochastic Simulation with Generative Adversarial Networks

Lukas Mosser, Olivier Dubrule, Martin J. Blunt

lukas.mosser15@imperial.ac.uk, o.dubrule@imperial.ac.uk, m.blunt@imperial.ac.uk
(Deep) Generative Methods

- Task: Draw (new) samples from unknown density given a set of samples

  **Main Problem: How to find the generative model?**

- Generative Adversarial Networks (GAN)
  - Two competing Neural Networks

- Variational Autoencoders (VAE)
  - Bayesian Graphical Model of data distribution

- Autoregression (Pixel-CNN)
  - Conditional Distribution on every sample

  - Many More ...
Generative Adversarial Networks – Toy Example

- Noise prior
  - Latent space $z$

- Generator ($z$)

- Gradient-based Feedback

- Discriminator ($x$)

- Training Data $x$

- $p_{\text{data}}(x)$

(Goodfellow et al. 2014)
Generative Adversarial Networks – Training

- Requirements:
  - Training Set of data
  - Generator – creates samples $G(z)$

- Discriminator – evaluates samples

- Cost function:
  $$
  \min_\theta \max_\omega \{ \mathbb{E}_{x \sim p_{data}} [\log D_\omega(x)] + \mathbb{E}_{x \sim p_z} [\log 1 - D_\omega(G_\theta(z))] \}
  $$

- GAN training – two step procedure in supervised way
  - Discriminator training step – Generator fixed
    - Train on real data samples
    - Train on fake samples
  - Generator training step – Discriminator fixed
    - Push generator towards “real” images
Ketton Limestone Dataset and Preprocessing

- Oolitic Limestone
- Intergranular pores
- Intragranular Micro-Porosity
- Ellipsoidal grains
- 99% Calcite
- Image Size:
  - $900^3$ voxels @ 26.7 $\mu m$

Extract Non-Overlapping Training Images ($64^3$ voxels)
Network Architecture - 3D Convolutional Network

Represent $G(z)$ and $D(x)$ as deep neural networks:

Discriminator: Binary Classification Network -> Real / Fake
Reconstruction Quality – Unconditional Simulation

Ketton Training Image

GAN generated sample

Intergranular Porosity
Moldic Features
Micro-Porosity

Training Time: 8 hours
Generation: 5 sec.

High visual quality
Needs quantitative measures
Reconstruction Quality Criteria

Statistical Properties

- Two-Point Probability Function $S_2(r)$
  - Radial Average / Directional

$$S_2(r) = P(x \in P, x + r \in P) \text{ for } x, r \in \mathbb{R}^3$$

Minkowski Functionals

- Porosity $\phi$
- Specific Surface Area $S_v$
- Integral of Mean Curvature
- Specific Euler Characteristic $\chi_v$
- Compute as function of image gray-level
  $\Rightarrow$ Characteristic Curves

Flow Properties: Solve Stokes flow in pore domain

- Permeability + Velocity Distributions

\[ \nabla \cdot \mathbf{v} = 0 \]
\[ \mu \nabla^2 \mathbf{v} = \nabla p \]
Ketton Comparison – Directional $S_2(r)$

- **Isotropic Covariance**
- **Pronounced Oscillations** -> “Hole-Effect”
  - Captured by GAN model

**Smaller Variance of GAN model**
Isotropic Permeability
Range of effective (flowing) porosity: Data (0.29 - 0.37) GAN (0.31 - 0.33)
Same order of magnitude and $\bar{k} - \phi$ relationship
Opening the GAN black box

What does the Generator learn?

Multi-scale Representation of pore space

Smaller Variance in GAN generated samples: Why?

Generator can miss modes of the data distribution -> Mode-Collapse
Latent Space Interpolation

\[ z^* = \beta z_{\text{start}} + (1 - \beta) z_{\text{end}}, \beta \in [0, 1] \]

Interpolation in latent space:
Shows that generator has learned a meaningful representation in a lower dimensional space!
### Computational Effort

Main Computation cost training:
Amortizes with number of samples due to low per sample cost / runtime

<table>
<thead>
<tr>
<th>Authors</th>
<th>Method</th>
<th>Size [voxels$^3$]</th>
<th>Run time ($\times$ 1) (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational run time comparison</td>
<td>Simulated annealing</td>
<td>300$^3$</td>
<td>22–47</td>
</tr>
<tr>
<td>Pant (2016)</td>
<td>Patch-based</td>
<td>$1000^2 \times 300$</td>
<td>0.1</td>
</tr>
<tr>
<td>Tahmasebi et al. (2017)</td>
<td>MPS</td>
<td>150$^3$</td>
<td>12</td>
</tr>
<tr>
<td>Okabe and Blunt (2004)</td>
<td>GAN</td>
<td>450$^3$</td>
<td>8</td>
</tr>
</tbody>
</table>

![Graph showing computational runtime vs. number of realizations.](image)

- **Proportional Cost**
- **Training-based**

Number of Realizations vs. Computational Runtime graph.

- **Proportional Cost**
- **Training-based**

13
Image Inpainting (Yeh et al. 2016)

Task: Restore missing details given a corrupted / masked image $M \cdot \tilde{x}$

Use a generative model $G(z)$ to find missing details, conditional to given information.

Contextual Loss: $L_{content} = \lambda \left| M \cdot G(z) - M \cdot \tilde{x} \right|_2$

Perceptual Loss: $L_{perc} = \log(1 - D(G(z)))$

Optimize loss by gradient descent on latent vector $z$

Corresponds to likelihood

Regularization for prior

Stay close to “real” images

Credit: Kyle Kastner
Conditioning – Pore Scale Example

Two-dimensional data at pore-scale more abundant e.g. thin-sections
Combine 3D generative model G(z) with 2D conditioning data

Generative Model: Ketton Limestone GAN (Part 1)
Mask: Three orthogonal cross-sections, honor 2D data in a 3D image

Contextual Loss:

\[ L_{content} = \lambda \left| |M \cdot G(z) - M \cdot \bar{x}|_2 \right| \] on orthogonal cross-sections

Perceptual Loss:

\[ L_{perc} = \log(1 - D(G(z))) \] on whole volumetric generated image G(z)

\[ L_{Total} = \lambda L_{content} + L_{perceptual} \]

Optimize Total Loss, by modifying latent vector (GAN parameters fixed)
- Many local minima at error threshold -> stochastic volumes that honor 2D data
Conditioning – Pore Scale Example

Same 2D conditioning data leads to varied realizations in 3D
Conditioning – Reservoir Scale Example

Maules Creek Training Image (Credit G. Mariethoz)

Pre-trained 3D-Generative Adversarial Network

Condition to single well (1D conditioning) from ground truth data:

![Images showing single realization, mean, and standard deviation](image_url)
Conclusions:

Generative Adversarial Networks are:
- Parametric – Latent Vector
- Differentiable – Allow for optimization
- Learned from training examples

That allow continuous reparameterizations of geological models.
- Can be conditioned to existing grid-block scale data.

Possibly very useful for solving stochastic inverse problems

Main Idea: Represent prior with a (deep) generative model

Thank you!

References


