A Deep-Learning-Based Geological Parameterization Method for History Matching

Yimin Liu    Wenyue Sun    Louis J. Durlofsky

SESAAI Meeting
March 30, 2018
Outline

- Background
- Deep-learning-based neural style transfer
- Convolutional neural network PCA (CNN-PCA) for geological Parameterization
- Parameterization results
- History matching results
- Conclusions and future work
History Matching

Actual Production From Field

Field Development Strategies

FINAL OBJECTIVE

STARTING POINT

Geology

High Resolution Geo-Cellular Model

Up-Scaled Engineering Model
History Matching Problem

\[
\arg\min_m \left\{ \frac{1}{2} (d(m) - d_{\text{obs}})^T C_D^{-1} (d(m) - d_{\text{obs}}) + \frac{1}{2} (m - \bar{m})^T C_m^{-1} (m - \bar{m}) \right\}
\]

- Decision variable: \( m \) - model parameters
- Challenges:
  - \( m \) can be high dimensional
  - \( m \) should preserve geology
- Solution: map \( m \) onto lower dimensional space

Log permeability
Reparameterization for History Matching

- Map $m$ to a new variable $\xi$
  
  $$m \approx \tilde{m} = f(\xi)$$

- Favorable properties:
  - $\dim(\xi) \ll \dim(m)$
  - $\xi$ is uncorrelated
  - $\tilde{m}$ preserves geological realism

- Optimization variable: $\xi$
Principal Component Analysis (PCA)

- PCA: Oliver (1996), Sarma et al. (2006)

  - Generate $N_r$ realizations using geostatistical algorithm
    \[ Y = \frac{1}{\sqrt{N_r} - 1} \left[ m_1 - \bar{m}, m_2 - \bar{m}, ..., m_{N_r} - \bar{m} \right] \]

  - Perform SVD and reduce dimension
    \[ Y = U \Lambda^{1/2} V^T \approx U_l \Lambda_l^{1/2} V_l^T \quad l \ll N_c \quad (N_c: \# \text{ of grid blocks}) \]

  - Generate new realization:
    \[ m_{\text{pca}} = U_l \Lambda_l^{1/2} \xi_l + \bar{m} \]
    \[ \xi_l \sim N(0, I) \]
PCA Representation for New Realizations

- Works well when $\mathbf{m}$ follows Gaussian distribution

\[ \mathbf{m}_{\text{pca}} = U_l \Lambda_l^{1/2} \xi_l + \bar{\mathbf{m}} \quad \xi_l \sim \mathcal{N}(0, I) \]

---

Gaussian

- SGEMS $N_C = 3600$
- PCA $l = 70$

---

Non-Gaussian

- SGEMS $N_C = 3600$
- PCA $l = 70$
Optimization-based PCA (O-PCA)


\[
m_{\text{opca}} = \arg\min_x \left\{ \| \mathbf{m}_{\text{pca}}(\xi_i) - x \|^2 + \gamma x^T (1 - x) \right\} \quad x_i \in [x^l, x^u]
\]

- Formulate PCA as an optimization problem with regularization
- Objective: minimize difference to \( \mathbf{m}_{\text{pca}} \) and original histogram
- Essentially post-process \( \mathbf{m}_{\text{pca}} \) with point-wise mapping

SGEMS \( N_C = 3600 \)  
O-PCA \( l = 70 \)

(figures from Vo and Durlofsky, 2014)
Limitations of O-PCA

- Underlying PCA honors only two-point correlations
- O-PCA point-wise mapping honors single point statistics
- Difficult to preserve multiple point statistics

Unconditional Realizations

SGeMS Random Real. 1  O-PCA Random Real. 1  O-PCA Random Real. 2
Neural Style Transfer

- Recent work in deep learning and computer vision
- Gatys et al. (2016), Johnson et al. (2016)
- Enables transfer of photo into artistic style

(images from Johnson: github.com/jcjohnson/fast-neural-style)
Neural Style Transfer

- Recent work in deep learning and computer vision
  - Gatys et al. (2016), Johnson et al. (2016)
  - Enables transfer of photo into artistic style

![Image of content image, PCA model, style image (training image), output image (post-processed model)]
Neural Style Transfer Algorithm

\[ O = \arg\min_O \{ L_{\text{content}}(O, C) + \gamma L_{\text{style}}(O, S) \} \]

- **O**: output image, **C**: content image, **S**: style image
- **\( L_{\text{content}}(O, C) \)**: difference between output and content image
- **\( L_{\text{style}}(O, S) \)**: difference between output and style image
- Output image preserves
  - Content (objects) in the content image
  - Style (color, texture) in the style image
  - Object and texture are characterized by multipoint statistics
Analogy to O-PCA

- Neural style transfer algorithm

\[ O = \arg\min_O \{ L_{\text{content}}(O, C) + \gamma L_{\text{style}}(O, S) \} \]

- O-PCA

\[ m_{\text{opca}} = \arg\min_x \left\{ \| m_{\text{pca}} - x \|^2 + \gamma x^T (1 - x) \right\} \]

- O-PCA losses based on point-wise difference

- Neural style transfer based on high-level features extracted from deep convolutional neural network (CNN)
Artificial Neural Network

- Nonlinear function \( y = g(x) \) with multiple layers of neurons
  \[ h_l = f(W^T h_{l-1} + b) \]
- Linear function: \( W^T h_{l-1} + b \)
- Nonlinear activation function \( f(\cdot) \), e.g., ReLU, sigmoid

- Convolutional neural networks (CNN):
  - Convolutional layers
  - Suitable for image input
Neural Style Transfer Algorithm

\[
L_{\text{content}}(O, C) = \sum_l \frac{1}{N_l D_l} \|F_l(O) - F_l(C)\|_F^2 \\
L_{\text{style}}(O, S) = \sum_l \frac{1}{N_l^2} \|G_l(O) - G_l(S)\|_F^2
\]

- **Limitations of neural style transfer algorithm:**
  - Need to solve optimization online
  - Derivatives of the output image w.r.t input images not clear
Train a **transform net**, Johnson et al. (2016)
- Hour glass shape deep CNN with same input and output size
- Same loss function, optimize parameters in the transform net

\[
\text{Style: } G_t = F_t F_t^T / (N_t D_t)^2
\]
Fast Neural Style Transfer Algorithm

- Construct PCA with 1000 SGeMS realizations
- Train the model transform net on 3000 random PCA models
- Training takes 3 minutes on 1 GPU (NVIDIA Tesla K80)
- Final step: threshold at 0.5
Outline

- Background
- Deep-learning-based neural style transfer
- Convolutional neural network PCA (CNN-PCA) for geological parameterization
- Parameterization results
- History matching results
- Conclusions and future work
Unconditional Binary System

- Binary facies model:
- Training image size: 250 x 250
- Model size: 60 x 60, \( N_c = 3600 \)
- No hard data
- Goal: low-dimensional representation
- Reduced dimension: \( l = 70 \)
Unconditional Binary System

PCA Real. 1

O-PCA Real. 1

CNN-PCA Real. 1

PCA Real. 2

O-PCA Real. 2

CNN-PCA Real. 2
Conditional Binary System

- Binary facies model
- Training image size: 250 x 250
- Model size: 60 x 60, $N_c = 3600$
- Hard data at 16 well locations
- Reduced dimension: $l = 70$
- 200 new random realizations
- Additional hard-data loss

$$\arg\min_{O} \{L_{\text{content}}(O, C) + \gamma_1 L_{\text{style}}(O, S) + \gamma_2 L_{\text{data}}(O)\}$$
Conditional Binary System

- All 200 CNN-PCA models match all hard data
Outline

- Background
- Deep-learning-based neural style transfer
- Convolutional neural network PCA (CNN-PCA) for geological parameterization
- Parameterization results
- History matching results
- Conclusions and future work
History Match Conditional Model

- Oil-water, 60 x 60 grid
- 2 injectors, 2 producers, BHP controlled
- \( k_{\text{sand}} = 2000 \) md, \( k_{\text{mud}} = 0.02 \) md
- Data: production and injection rates for 1000 days every 100 days
- Number of data: \( N_d = 80 \)
- Goal: 30 RML posterior models
- Optimizer: PSO-MADS
Permeability Estimation

True model

One O-PCA prior model

O-PCA posterior model

One CNN-PCA prior model

CNN-PCA posterior model
PROD-1 Water Rate

O-PCA Prior models

O-PCA Posterior Models

CNN-PCA Prior models

CNN-PCA Posterior Models
Infill Well Prediction

- Two infill wells P3, P4
- Drilled at 1000 days
- Prediction to 2000 days
Permeability Estimation

True model

O-PCA #1

O-PCA #2

O-PCA #3

CNN-PCA #1

CNN-PCA #2

CNN-PCA #3
Infill Well PROD-4 Prediction

O-PCA Water Rate

CNN-PCA Water Rate
Field Prediction

O-PCA Oil Rate

CNN-PCA Oil Rate

15 curves

6 curves
Conclusions

- Developed CNN-PCA by combining deep-learning-based neural style transfer algorithm with PCA.
- CNN-PCA better preserves channel geometry compared to O-PCA.
- CNN-PCA history matching solutions provide more accurate prediction for infill wells.
Future Work

- Extend CNN-PCA to bimodal, three-facies and three dimensional reservoir models
- Apply CNN-PCA with gradient-based history matching
- Implement CNN-PCA treatment with ensemble-based history matching methods
Acknowledgments

- Stanford CS 231N course
  - Instructors: Justin Johnson, Serena Yeung, Fei-fei Li
  - Course project teammate: Yuanlin Wen (Google)

- Open source code:
  - O-PCA: Hai Xuan Vo (Chevron)
  - Neural style transfer (PyTorch): Hang Zhang
  - Fast neural style transfer (PyTorch): Abhisheek Kadian

- PSO-MADS: Obi Isebor

- Stanford CEES
Thank You!

Q & A
Backup Slides
Convolutional Layers

- Each neuron connects to a local region in previous layer
- Convolution: $N_l$ filters $w_l$ sliding through previous layers
- Convolutional layer $N_l$ channels:
  - Each channel is called a feature map of $D_l$ neurons
  - All channels form a feature matrix $F_l \in R^{N_l \times D_l}$
Convolutional Neural Network

- Neural network consists of mainly convolutional layers
- Nonlinear layers:
  - Activation layer, e.g., ReLU \( f(x) = \max(0,x) \)
  - Pooling layer: down sampling to reduce dimension
- Image classification:
  - Input: image of size \( H \times W \times 3 \)
  - Output: score for predefined image classes

VGG-16 Deep CNN (Simonyan and Zisserman, 2015)
Feature Matrix and Gram Matrix

- Content representation: feature matrix $F_l \in \mathbb{R}^{N_l \times D_l}$
- Style representation: Gram matrix $G_l = F_l F_l^T / (N_l D_l)^2$

* Images modified from Gatys et al. (2016)
CNN-PCA for Reparameterization

- Construct PCA with $N_r$ SGeMS realizations

$$m_{\text{pca}} = U_l \Lambda_l^{1/2} \xi_l + \bar{m} \quad \xi_l \sim N(0, I)$$

- Post-process $m_{\text{pca}}$ with fast neural style transfer algorithm