Predicting Oil Movement in a Development System using Deep Latent Dynamics Models

Pavel Temirchev
PhD Student, Research Intern

November, 2018

Skoltech
Skolkovo Institute of Science and Technology

Contents

What is an oil reservoir?

Porosity

Porosity of the rock is defined as:

$$\phi = \frac{V_{pores}}{V_{bulk}}$$

What is an oil reservoir?

Permeability

Permeability is the ability of the rock to be permeable by fluids.
And is a coeficient in Darcy's law:

$$v = -\frac{k}{\mu}\nabla p$$

v - fluid velocity,
∇p - pressure gradient,
μ - fluid viscosity,
k - permeability

What is an oil reservoir?

Permeability

Multiphase Darcy's law:

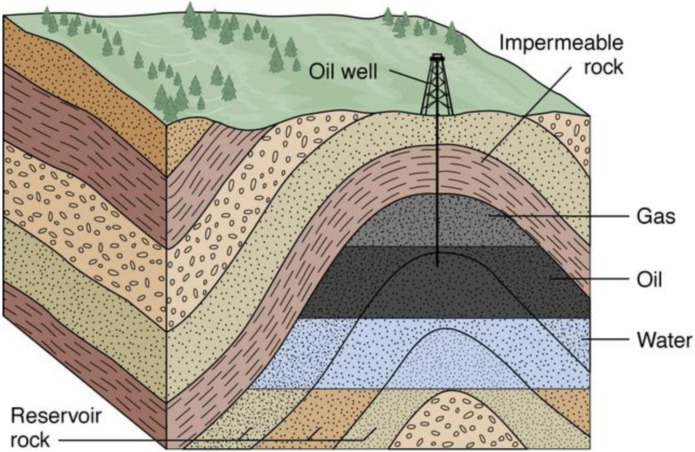
$$v_\alpha = -\frac{k_\alpha}{\mu_\alpha} \nabla p$$

α - one of the phases: oil, water, gas

$k_\alpha = k k_{r\alpha}$ - phase permeability

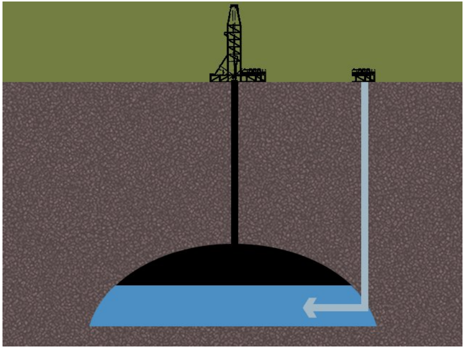
What is an oil reservoir?

- Porous and permeable rock
- Located at several hundreds of meters below Earth surface (1000–3000 meters)
- Saturated with oil, water and, possibly, natural gas



How to produce oil?

- Drill production wells
- Oil liberate gas if the pressure drops below the bubble point
- Inject water to maintain the reservoir pressure!



Reservoir modelling

Why to simulate:

- Easy to ruin the reservoir
- Expensive decisions
- Lots of iterative optimization problems:
  - Design optimization
  - History-matching
  - Uncertainty quantification

What to simulate:

- Oil, water and gas production from wells
- Pressure distribution in time $p(x, y, z, t)$
- Saturation distributions in time $s_\alpha(x, y, z, t)$

$$s_\alpha = \frac{V_\alpha}{V_{pores}}$$

$$\alpha \in \{oil, water, gas\}$$

Conventional reservoir simulation

Multiphase flow equations

Gas Equation:

$$\frac{\partial}{\partial t}\left[\phi\left[\frac{s_g}{B_g} + R_{so}\frac{s_o}{B_o}\right]\right] = \nabla\left[\left[\frac{k_g}{\mu_g B_g} + R_{so}\frac{k_o}{\mu_o B_o}\right]\nabla p\right] + q_g$$

Oil & Water Equations:

$$\frac{\partial}{\partial t}\left[\phi\frac{s_o}{B_o}\right] = \nabla\left[\frac{k_o}{\mu_o B_o}\nabla p\right] + q_o$$

$$\frac{\partial}{\partial t}\left[\phi\frac{s_w}{B_w}\right] = \nabla\left[\frac{k_w}{\mu_w B_w}\nabla p\right] + q_w$$

Pressure Equation:

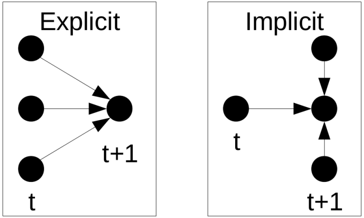
$$\nabla^2 p = \phi\frac{c_t}{\lambda_t}\frac{\partial p}{\partial t}$$

$$\lambda_t = \frac{k_o}{\mu_o} + \frac{k_g}{\mu_g} + \frac{k_w}{\mu_w}$$

Conventional reservoir simulation

Finite difference approximation

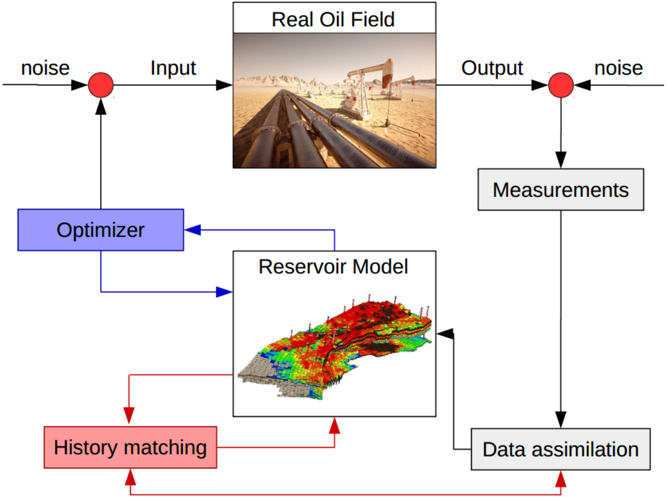
- Discretize in space and time dimensions (uniform grid-bloks and timesteps)
- Linearize
- Solve using either explicit or implicit scheme
- High accuracy
- High computational complexity – $O(n^3)$



Conventional reservoir simulation

Finite-Difference Reservoir Model:

- Accurate
- Bounded error
- Lots of options
- Slow
- RAM consumption
- Poor multithreading
- Lots of input data



Techniques to reduce the complexity

- Simplify the Finite Difference model
- Reduced Order Modelling - ROM (physics-aware, POD-based)
- Fully data-driven models

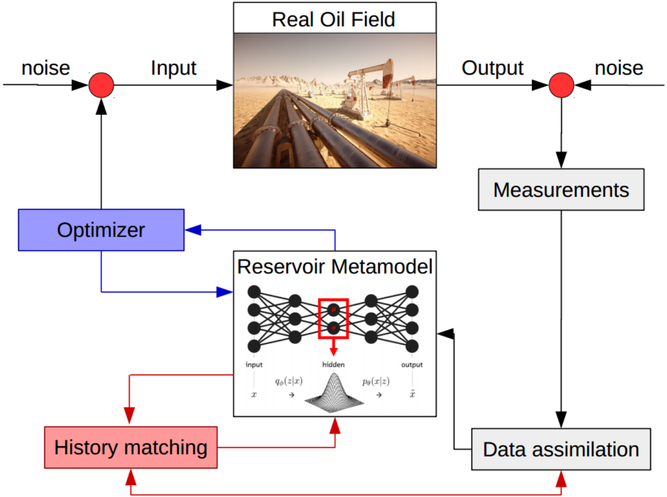
Techniques to reduce the complexity

- ~~Simplify the Finite Difference model~~
- Reduced Order Modelling - ROM (physics-aware, POD-based)
- Fully data-driven models

Techniques to reduce the complexity

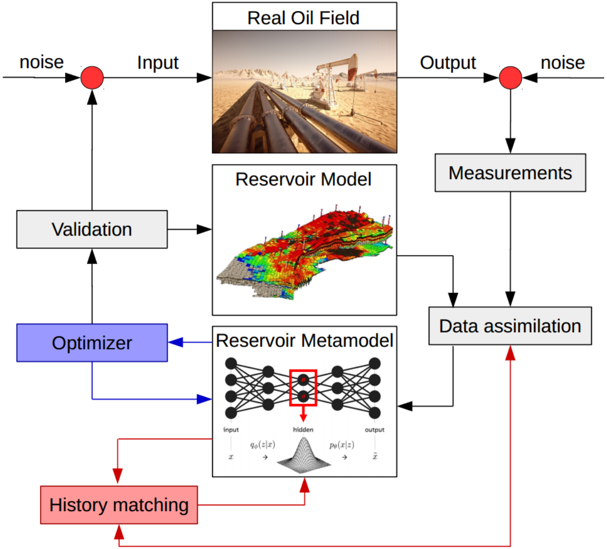
Data-Driven Reservoir Metamodel:

- Very fast
- Differentiable (sometime)
- Quite accurate
- Effective multithreading
- Unbounded error
- Subset of options



Techniques to reduce the complexity

Data-Driven Metamodel and **Finite-Difference Model Cooperation:**



Techniques to reduce the complexity

- ~~Simplify the Finite Difference model~~
- Reduced Order Modelling - ROM (physics-aware, POD-based)
- Fully data-driven models

POD-Galerkin approach

Proper Orthogonal Decomposition - POD

- POD = PCA $\approx$ SVD
- Object-Features vs. Snapshot formulation

$$X = (y_{00}, \dots, y_{it}, \dots) \in \mathbb{R}^{n \times N}$$

$$X = U \Sigma W$$

$$y \approx U^r \tilde{y}$$

POD-Galerkin approach

Simplified Equations

Simplified two-phase flow equations:

$$\nabla k \lambda \nabla p = q$$

$$\phi \frac{\partial s_w}{\partial t} + v(p) \nabla f_w(s_w) = q_w$$

Discretization:

$$Ap = b$$

$$\frac{\partial s_w}{\partial t} + Bf_w(s_w) = d$$

POD-Galerkin approach

Galerkin projection

Pressure:

$$ {U_p^r}^T A U_p^r \tilde{p} = {U_p^r}^T b $$

$$ \tilde{A}\tilde{p} = \tilde{b} $$

Saturation:

$$ \frac{\partial \tilde{s}_w}{\partial t} + {U_s^r}^T B f_w(U_s^r \tilde{s}_w) = \tilde{d} $$

- Still of complexity $O(n^3)$
- Authors propose to use DEIM (Discrete Empirical Interpolation Method), but...

POD-DRRNN approach

Deep Residual RNN

Generalized form of PDE:

$$\frac{dy}{dt} = Ay + F(y)$$

Residual:

$$r_{t+1} = y_{t+1} - y_t - \Delta t A y_{t+1} - \Delta t F(y_{t+1})$$

Deep Residual RNN:

$$y_{t+1}^{(1)} = y_{t+1}^{(0)} - w \cdot \sigma(U r_{t+1}^{(0)})$$

for $k > 1$:

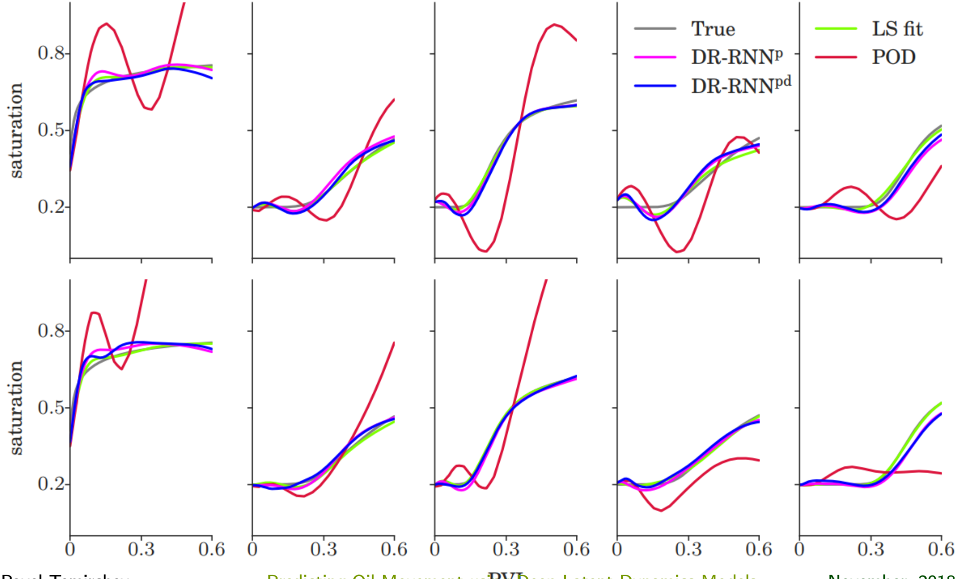
$$y_{t+1}^{(k)} = y_{t+1}^{(k-1)} - \frac{\eta_k}{\sqrt{G_k + \epsilon}} r_{t+1}^{(k)})$$

$$G_k = \gamma || r_{t+1}^{(k)} ||^2 + \xi G_{k-1}$$

Experimental setup

- Random permeability field
- Just one quadratic nonlinearity
- Fixed oil reservoir and well allocation scheme

Results for POD-Galerkin and POD-DRRNN



Techniques to reduce the complexity

- ~~Simplify the Finite Difference model~~
- Reduced Order Modelling - ROM (physics-aware, POD-based)
- Fully data-driven models

NEW NOTATION IS USED

Fully data-driven approach

Definition

Reservoir Metamodel – is a purely data-driven reservoir model aimed to approximate a Base Hydrodynamical Model (physical model).

Aim: create fast 3D metamodel of three-phase reservoir dynamics and production rates

Restriction: suited only for a subset of available simulation options

Simulation of a Development Unit

Objectives

1. Generate a **training set**
2. Create a metamodel of the **reservoir dynamics**
3. Create a metamodel of **production rates**

1. Training Set

Each object of the training set (scenario) consists of four parts:

Metadata vector m describing initial conditions and reservoir properties. $m \in \mathbb{R}^{61}$

1. Training Set

Each object of the training set (scenario) consists of four parts:

Metadata vector m describing initial conditions and reservoir properties. $m \in \mathbb{R}^{61}$

Control variable sequence of vectors u_t each describing production bottomhole pressure and injection rate at time t. $u_t \in \mathbb{R}^2$

1. Training Set

Each object of the training set (scenario) consists of four parts:

Metadata	vector m describing initial conditions and reservoir properties. $m \in \mathbb{R}^{61}$
Control variable	sequence of vectors u_t each describing production bottomhole pressure and injection rate at time t. $u_t \in \mathbb{R}^2$
Reservoir Dynamics	sequence of tensors f_t each containing pressure and saturation distributions at time t. $f_t \in \mathbb{R}^{3 \times n_x \times n_y \times n_z}$

1. Training Set

Each object of the training set (scenario) consists of four parts:

Metadata vector m describing initial conditions and reservoir properties. $m \in \mathbb{R}^{61}$

Control variable sequence of vectors u_t each describing production bottomhole pressure and injection rate at time t. $u_t \in \mathbb{R}^2$

Reservoir Dynamics sequence of tensors f_t each containing pressure and saturation distributions at time t. $f_t \in \mathbb{R}^{3 \times n_x \times n_y \times n_z}$

Production Rates sequence of vectors r_t each describing daily production rates of water, oil and gas. $r_t \in \mathbb{R}^3$

1. Training Set

Base Reservoir Model Run

- Metadata m and Control u are from Generative Model based on real laboratoty data

1. Training Set

Base Reservoir Model Run

- Metadata m and Control u are from Generative Model based on real laboratoty data
- Ground Truth Dynamics f and Production Rates r are from Base Finite-Difference Reservoir Simulator run on generated Metadata and Control.

1. Training Set

Base Reservoir Model Run

- Metadata m and Control u are from Generative Model based on real laboratoty data
- Ground Truth Dynamics f and Production Rates r are from Base Finite-Difference Reservoir Simulator run on generated Metadata and Control.
- Model 1/4th of a development unit (due to the symmetry)

1. Training Set

Base Reservoir Model Run

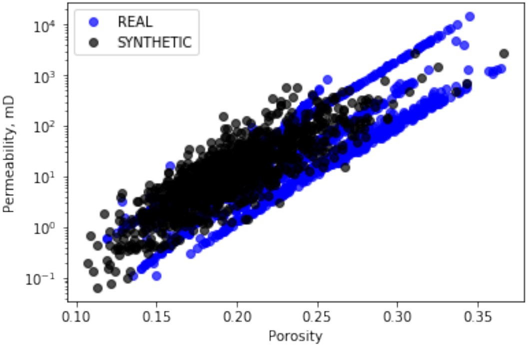
- Metadata m and Control u are from
  Generative Model based on real laboratoty data
- Ground Truth Dynamics f and Production Rates r are from
  Base Finite-Difference Reservoir Simulator run on generated Metadata and Control.
- Model 1/4th of a development unit (due to the symmetry)
- Computational grid resolution (n_x, n_y, n_z):
  $41 \times 60 \times 10$ (> 5000 scenarios)

1. Training Set

Generative Model for Metadata and Control

To generate diverse, but realistic Metadata and Contol variables we:

- Analyzed a lot of real laboratory data (provided by Gazpromneft-STC)
- Divided input properties into interdependent groups
- For each group fit parameters of distributions from the simple parametric families:
  - Normal
  - Log-Normal
  - Uniform



2. Reservoir Dynamics

Dynamics in a latent variable space

Our Aim is to approximate the function F:
$f_{t+1} = F(f_{0:t}, u_{0:t}, m)$

2. Reservoir Dynamics

Dynamics in a latent variable space

~~Our Aim is to approximate the function F:~~

$\cancel{f_{t+1} = F(f_{0:t}, u_{0:t}, m)}$

Too hard!

Instead:

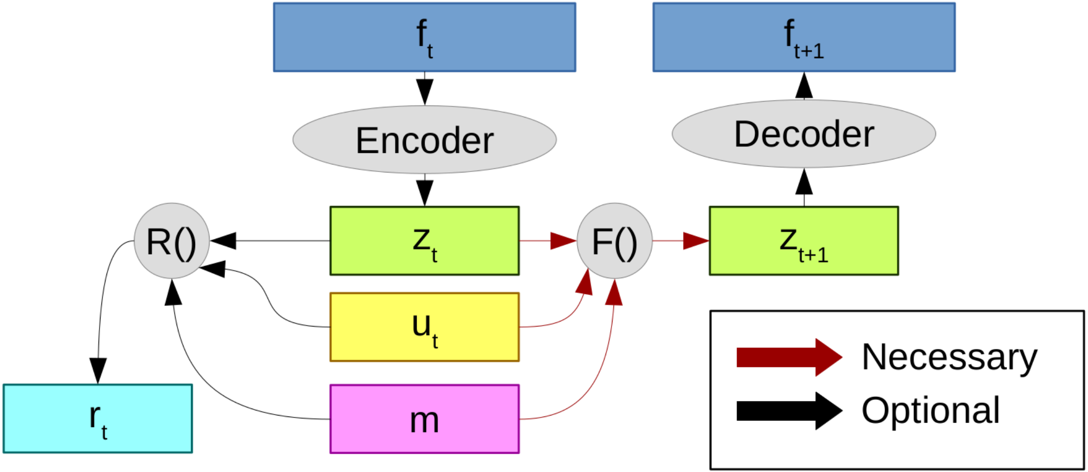
$$z_{t+1} = F_{latent}(z_{0:t}, u_{0:t}, m)$$

$$z_t = E(f_t); \qquad f_t = D(z_t)$$

$$dim(z) << dim(f)$$

2. Reservoir Dynamics

Dynamics in a latent variable space



2. Reservoir Dynamics

Minimization problem

End-to-End case:

$$\sum_{s} \sum_{t=0}^{T_s} ||f_t^s - \hat{f}_t^s||_2^2 \to \min_{\hat{f}}$$

where $\hat{f}$ - is a forecast of metamodel,
and f - is the Ground Truth Dynamics

Separate Training case:

Latent Dynamics loss:

$$\sum_{s} \sum_{t=0}^{T_s} ||z_t^s - \hat{z}_t^s||_2^2 \to \min_{\hat{z}}$$

Autoencoding loss:

$$\sum_{s} \sum_{t=0}^{T_s} ||f_t^s - D(E(f_t^s))||_2^2 \to \min_{E,D}$$

where $E(\cdot)$ and $D(\cdot)$ - are encoding and decoding models respectively

2. Reservoir Dynamics

Used autoencoding models:

- PCA: Principal Components Analysis (similar to POD)
- CVAE: Convolutional Conditional Variational Autoencoder

Used models of latent dynamics:

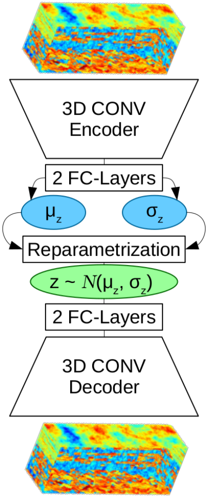
- Linear: Linear Regression
- MNN: Markovian Fully-Connected Neural Network
- GRU RNN: Gated Recurrent Neural Network

Conditional Variational Autoencoder

$$q_\phi(z|f_t^s, m) = \mathcal{N}\Big(\mu_z(f_t^s, m|\phi), \sigma_z(f_t^s, m|\phi)\Big)$$
$$p_\theta(f|z_t^s, m) = \mathcal{N}\Big(\mu_f(z_t^s, m|\theta), I\Big)$$
$$p_\theta(z|m) = \mathcal{N}(0, I)$$

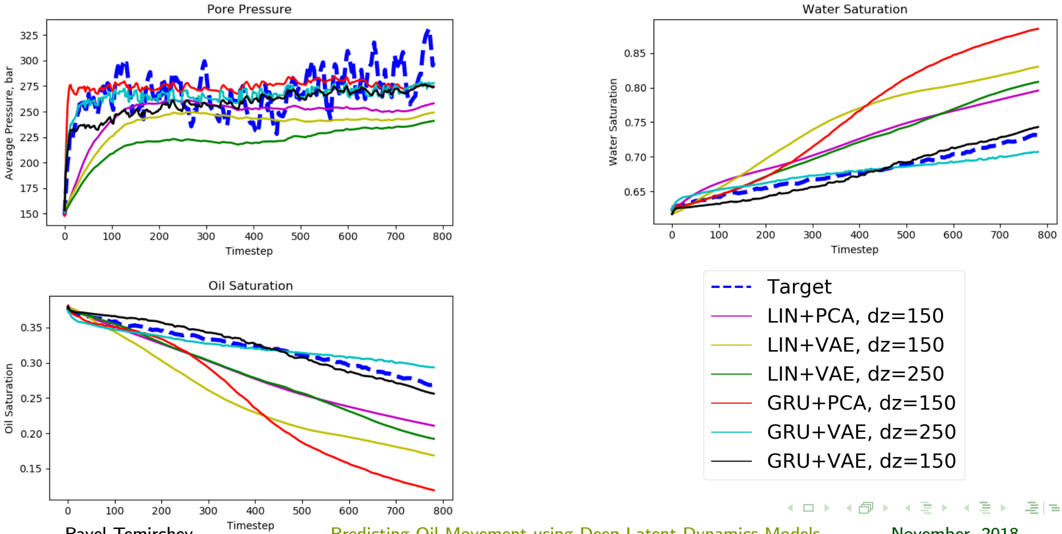
Evidence Lower Bound Objective:

$$\mathcal{L}(\theta, \phi, f_t^s, m) = -KL\Big(q_\phi(z|f_t^s, m)||p_\theta(z|m)\Big) + \ldots$$
$$+\mathbb{E}_{z\sim q_\phi}\Big[\log p_\theta(f_t^s|z, m)\Big]$$



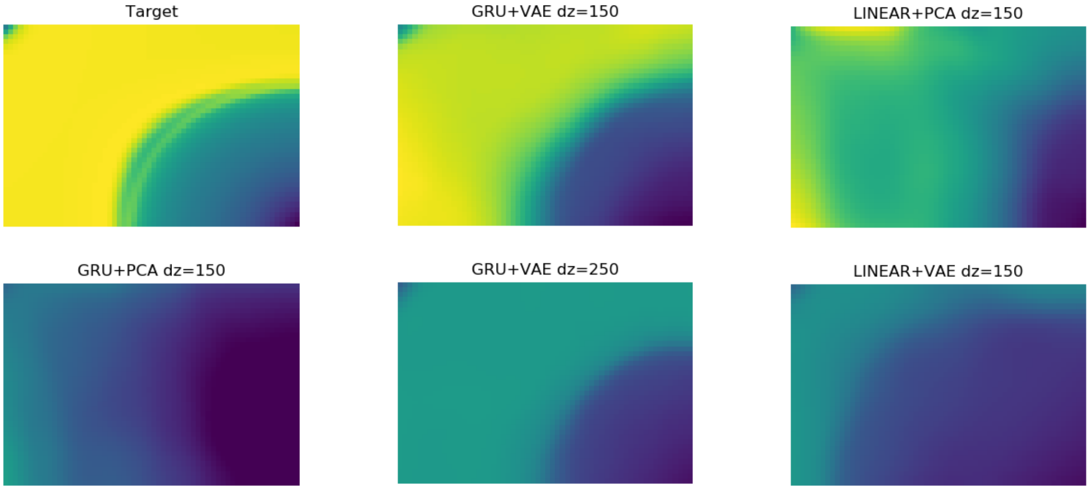
2. Reservoir Dynamics

Results: Mean values across tensor f in time



2. Reservoir Dynamics

Results: Horizontal slices of f (OIL SATURATION, 5600th day)



2. Reservoir Dynamics

Animated Dynamics

2. Reservoir Dynamics

Results: pointwise relative error across the validation set

Mean relative error for f_t in % and its standard deviation								
Dynamics	Encoding	d_z	Error %			S.T.D.		
			p	s_{oil}	s_{water}	p	s_{oil}	s_{water}
Linear	PCA	150	16.52	59.54	38.51	10.76	15.41	7.68
Linear	VAE	150	12.97	26.69	14.68	8.54	15.00	7.67
Linear	VAE	250	11.08	22.71	12.77	8.65	12.38	6.52
GRU	PCA	150	14.17	52.27	33.92	5.61	9.85	4.22
GRU	VAE	150	**8.72**	**12.65**	**7.37**	**4.39**	9.00	4.82
GRU	VAE	250	9.51	14.74	8.63	6.54	**8.51**	**4.19**

3. Production Rates
from the latent variable space

The Aim is to approximate the function R:

$$r_t = R(z_{0:t}, u_{0:t}, m)$$

Used models of production rates:

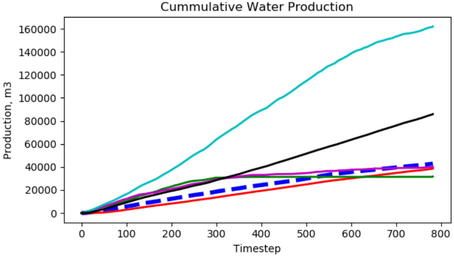
Linear: Linear Regression

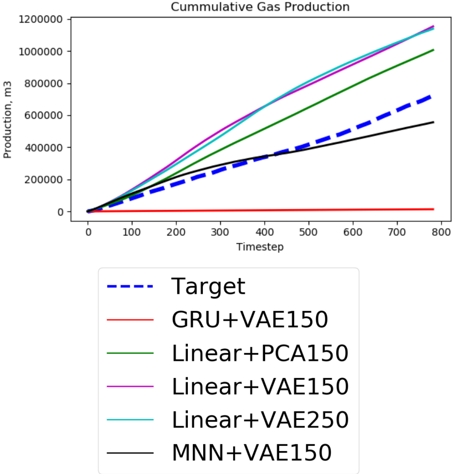
MNN: Markovian Fully-Connected Neural Network

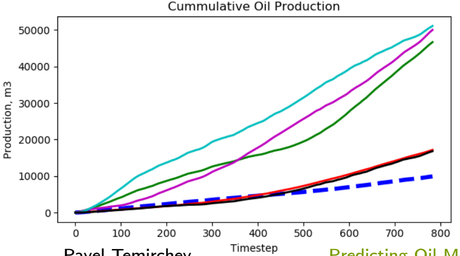
GRU RNN: Gated Recurrent Neural Network

3. Production Rates

Results: cummulative production







3. Production Rates

Results: relative error across the validation set

Mean relative error for r_t in %					
Production	Encoding	d_z	Error %		
			q_{water}	q_{oil}	q_{gas}
Linear	PCA	150	116.00	107.87	140.37
Linear	VAE	150	92.38	127.28	141.13
Linear	VAE	250	115.45	114.29	146.29
MNN	VAE	150	**57.77**	**57.29**	**138.74**
GRU	VAE	150	165.70	173.32	160.70

Results & Discussion

- 3D reservoir dynamics metamodelling from data is possible and efficient
- The methods may be transfered into other areas of science: climate forecasting, aerodynamics, etc.
- GRU RNN is able to capture complex dependencies from the reservoir model
- The useful information about reservoir state may be described by around 150 numbers
- Better production rates model is needed!

Further Work

- Metamodelling of a whole oil field, based on the proposed approach
- Control variables optimization via Model-Based Reinforcement Learning algorithms
- History matching (recovering metadata) via gradient optimization methods
- More simulation options

Acknowledgements & Collaborations

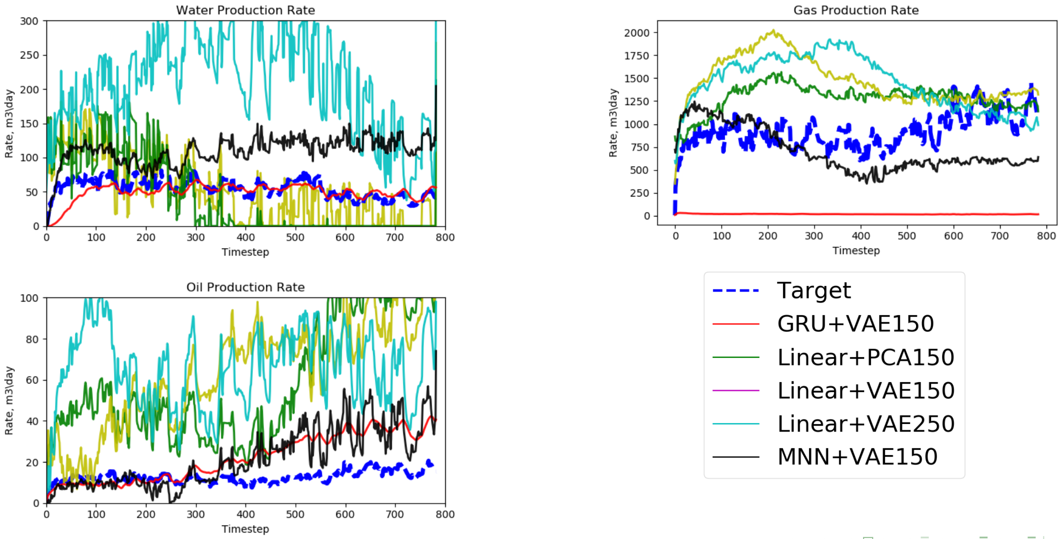
- In collaboration with **Gazpromneft-STC**
- Laboratory data and binary files parser were provided by *Gazpromneft-STC*
- Interpolation model and some dataset preprocessing were made by **Ruslan Kostoev** (*PhD student, CDISE*)
- Huge amount of expertise was provided by **Dmitry Koroteev**, **Evgeny Burnaev** and **Ivan Oseledets**

Minimization problem for Production Rates

$$\sum_{s} \sum_{t=0}^{T_s} ||r_t^s - \hat{r}_t^s||_2^2 \to \min_{\hat{r}}$$

where $\hat{r}$ - is a forecast of metamodel,
and r - is the Ground Truth Production Rates

Daily production rates results



Relative Error Equations

For Reservoir Dynamics:

$$error[l] = \frac{1}{N} \sum_{s} \sum_{t=0}^{T_s} \frac{||\hat{f}_t^s[l] - f_t^s[l]||_2^2}{||\frac{1}{2}(\hat{f}_t^s[l] + f_t^s[l]) + \epsilon||_2^2} \cdot 100\%$$

where $l \in \{pressure, oil\ saturation, water\ saturation\}$ is a parameter of interest.

For Production Rates:

$$error[l] = \frac{1}{N} \sum_{s} \sum_{t=0}^{T_s} \left| \frac{\hat{r}_t^s[l] - r_t^s[l]}{\frac{1}{2}(\hat{r}_t^s[l] + r_t^s[l]) + \epsilon} \right| \cdot 100\%$$

where $l \in \{water, oil, gas\}$ is a fluid of interest.