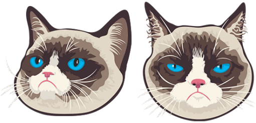
Speeding up wave propagation modeling
CheckPoint # 5

Team: GrumpyCat
Ivan Gerasimov, Olha Tsymboi

Moscow Institute of Physics and Technology

December 6, 2020



Introduction & Problem statement

Aim:

- To speed up acoustic equation solution using using neural networks.

Input:

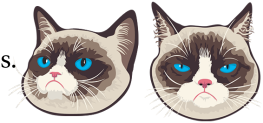
- time and space discretizations Δt, $\Delta x = \Delta z$
- time and space pints amout n_t, n_x, n_z
- impulse source time-series $q(t_i)$ and location x_s, z_s.
- special velocities $vp(x,z)$ at data points
- absorbing boundary conditions

Output:

- The solution of acoustic equation $u(x,z,t)$ at some point of time t.

Quality:

- RMSE error between normalized wavefields $\frac{u(x,z,t)}{\sigma(u)}$.
- Correlation coefficient between normalized wavefields.
- Execution time.



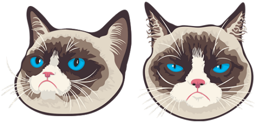
Model:

In the previous series

- Convolutional Auto-Encoder + L1Loss/MSE
- Convolutional Auto-Encoder + GRU Cell at the bottleneck + L1Loss/MSE
- UNet (UNet++)[**?**] + L1Loss/MSE
- Fully-Connected + Physic's Informed Loss [**?**]
- Convolutional + Physic's Informed Loss

Time is precious

- Avoid skip connections to reduce convolutions on large images
- Avoid point-wise prediction using FC layers
- Avoid extra encoder-decoder passes to predict full sequence



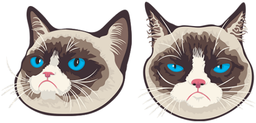
Model:

In the previous series

- Convolutional Auto-Encoder + L1Loss/MSE
- Convolutional Auto-Encoder + GRU Cell at the bottleneck + L1Loss/MSE
- UNet (UNet++)[**?**] + L1Loss/MSE
- Fully-Connected + Physic's Informed Loss [**?**]
- Convolutional + Physic's Informed Loss

Time is precious

- Avoid skip connections to reduce convolutions on large images
- Avoid point-wise prediction using FC layers
- Avoid extra encoder-decoder passes
  to predict full sequence



Model:

But still let us use Auto-Encoder as base-line

- Convolutional Auto-Encoder + L1Loss/MSE

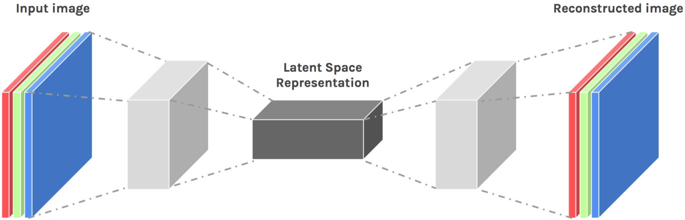
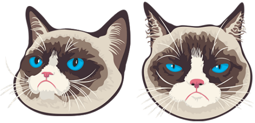


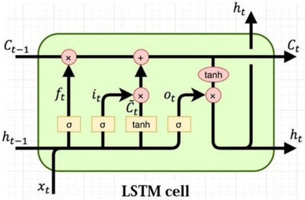
Figure: Convolutional Auto-Encoder

Input image: $u(x,z,t_i) + \frac{q(t_i)\Delta x^2}{\Delta t^2} \cdot \delta(x - x_s, z - z_s),\ vp(x,z)$



Model:

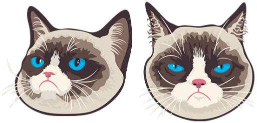
- Convolutional Auto-Encoder + LTSM Cell + L1Loss/MSE



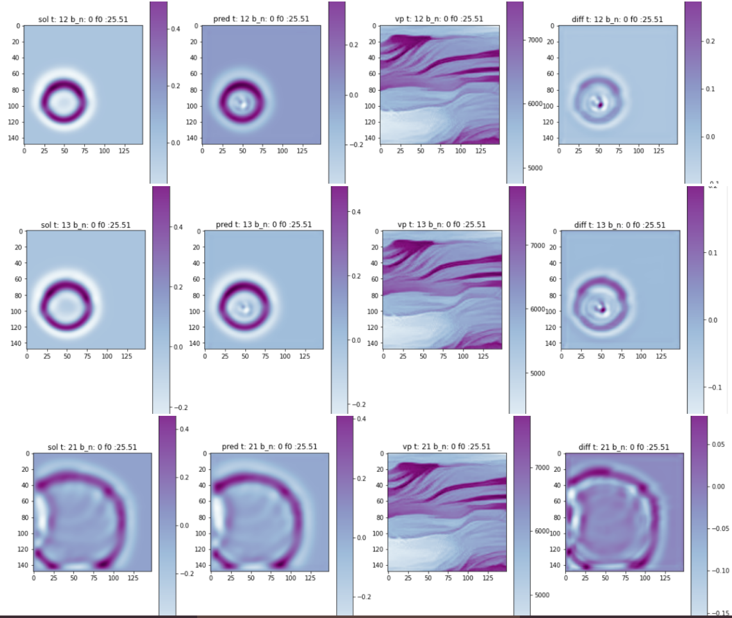
$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$
$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$
$$h_t = \tanh(C_t) * o_t$$

Figure: Convolutional LSTM Cell

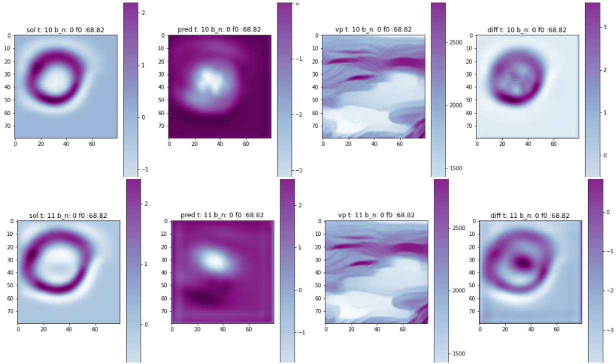
Input image: zeros($\mathrm{x}_{src}, z_{src}, t_i$) + $q(t_i)$
Hidden initial image:vp(x, z)

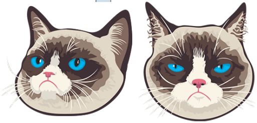


Results:


sol t: 12 b_n: 0 f0 :25.51
pred t: 12 b_n: 0 f0 :25.51
vp t: 12 b_n: 0 f0 :25.51
diff t: 12 b_n: 0 f0 :25.51
sol t: 13 b_n: 0 f0 :25.51
pred t: 13 b_n: 0 f0 :25.51
vp t: 13 b_n: 0 f0 :25.51
diff t: 13 b_n: 0 f0 :25.51
sol t: 21 b_n: 0 f0 :25.51
pred t: 21 b_n: 0 f0 :25.51
vp t: 21 b_n: 0 f0 :25.51
diff t: 21 b_n: 0 f0 :25.51
0 20 40 60 80 100 120 140
0 25 50 75 100 125
0.4 0.2 0.0 −0.2
7000 6000 5000
0.2 0.1 0.0 −0.1
0.05 0.00 −0.05 −0.10 −0.15

Results:


sol t: 10 b_n: 0 f0 :68.82
pred t: 10 b_n: 0 f0 :68.82
vp t: 10 b_n: 0 f0 :68.82
diff t: 10 b_n: 0 f0 :68.82
sol t: 11 b_n: 0 f0 :68.82
pred t: 11 b_n: 0 f0 :68.82
vp t: 11 b_n: 0 f0 :68.82
diff t: 11 b_n: 0 f0 :68.82
0
10
20
30
40
50
60
70
2500
2000
1500



References I

abbrv ref

Thank you!

Questions?