AtmoRep

Large scale representation learning of atmospheric dynamics

Christian Lessig, Ilaria Luise, Martin Schultz, Maike Sonnewald, Annesh Subramanian

address climate change

large scale machine learning

scientific insight
Representation learning

- Learn a domain-specific but task-independent neural network that is useful for a range of applications
Representation learning

- Learn a domain-specific but task-independent neural network that is useful for a range of applications
Representation learning

- Learn a domain-specific but task-independent neural network that is useful for a range of applications
Representation learning

- Learn a domain-specific but task-independent neural network that is useful for a range of applications
Representation learning

- Learn a domain-specific but task-independent neural network that is useful for a range of applications
  - Representation network provides transformation of network input to an effective feature space
Representation learning

- Learn a domain-specific but task-independent neural network that is useful for a range of applications
  - Representation network provides transformation of network input to an effective feature space
  - Self-supervised training on very large amounts of data (O (PB)) with very large networks (O (10^9) parameters)
Representation learning

- Learn a domain-specific but task-independent neural network that is useful for a range of applications
  - Representation network provides transformation of network input to an effective feature space
  - Self-supervised training on very large amounts of data (O (PB)) with very large networks (O (10⁹) parameters)
  - Useful for applications using tail network or fine-tuning
Can we perform representation learning in the Earth sciences?
Representation learning for the Earth sciences?

- Very large amounts of observational data
  - ERA5 reanalysis: 6+ PB
  - E.g. MetOp-SG: 8 x 864 GB/day
Representation learning for the Earth sciences?

- Very large amounts of observational data
  - ERA5 reanalysis: 6+ PB
  - E.g. MetOp-SG: 8 x 864 GB/day
Representation learning for the Earth sciences?

- Very large amounts of observational data
- No complete classical model for system and dynamics
  - Central issue for forecasting and climate projections
Representation learning for the Earth sciences?

- Very large amounts of observational data
- No complete classical model for system and dynamics
- Large networks learn statistical representations
Representation learning for Earth science?

Yann LeCun @ylecun - 10.09.22
Multiple interpretations of ambiguous percept must be associated with multiple values of an explanatory latent variable. By “latent”, I mean that they are not outputs but internal inputs. What are the mechanisms in the brain for exploring the set of plausible values?

Steve Stewart-Williams @SteveStuWill - 09.09.22
One of my all-time favorite illusions: The spinning dancer

If you look at the dancer on the left and the one in the middle, the one in the middle spins clockwise.

If you look at the dancer on the “right” and the one in the middle, the one in the middle spins counterclockwise.
Representation learning for the Earth sciences?

- Very large amounts of observational data
- No complete classical model for system and dynamics
- Large networks learn statistical representations
  - Chaoticity in atmospheric dynamics leads to ambiguity, i.e. there is often not one “correct answer”
Representation learning for the Earth sciences?

- Very large amounts of observational data
- No complete classical model for system and dynamics
- Large networks learn statistical representations
- Data transformations like PCA have proven very useful
AtmoRep

Large scale representation learning of atmospheric dynamics
AtmoRep

Historical observations


ERA5 reanalysis
Climate projections
Impact analysis
Forecasting
AtmoRep

Historical observations

ERA5 reanalysis
AtmoRep

Historical observations

ERA5 reanalysis

large scale machine learning

1950
1970
1990
2010

© Atmorep Collaboration, 2022
AtmoRep

Historical observations

ERA5 reanalysis

large scale machine learning

applications

Forecasting

Impact analysis

Climate projections

Downscaling

scientific insight
Data: ERA5 reanalysis
Data: ERA5 reanalysis
Data: ERA5 reanalysis
Data: ERA5 reanalysis
Data: ERA5 reanalysis

137 vertical layers
Data: ERA5 reanalysis

721x1440 horizontal grid (0.25 degree)

137 vertical layers
Data: ERA5 reanalysis

721x1440 horizontal grid (0.25 degree)

- vorticity
- divergence
- temperature
- geopotential
- ...

137 vertical layers
Data: ERA5 reanalysis

- vorticity
- divergence
- temperature
- geopotential
- ...

721x1440 horizontal grid (0.25 degree)

137 vertical layers

Time: hourly for 70 years
AtmoRep data

- Physical fields: vorticity, divergence, temperature, geopotential height, humidity
- Space: 721 x 1440 x 10 vertical layers
- Time: 24 time steps per day for 365 days for 70 years
AtmoRep data

vorticity
AtmoRep data

divergence
AtmoRep data

temperature
AtmoRep data

ggeopotential
AtmoRep data
AtmoRep data

- Vorticity
- Divergence
- Geopotential
- Temperature
AtmoRep data

Normalized spectra at 975 hPa

- vorticity
- divergence
- geopotential
- temperature
AtmoRep network architecture

◦ Transformer-based network architecture
  › Scales well to very large data-sets
  › Generative model (with decoder)
  › Attention maps provide (physical) interpretability
AtmoRep network architecture

- Transformer encoder-based network architecture
  - Scales well to very large data-sets
  - Generative model (with decoder)
  - Attention maps provide (physical) interpretability
- Network is local in space-time
AtmoRep network architecture
AtmoRep network architecture
AtmoRep network architecture
AtmoRep network architecture

- Transformer encoder-based network architecture
  - Scales well to very large data-sets
  - Generative model (with decoder)
  - Attention maps provide (physical) interpretability
- Network is local in space-time
  - Principal of dynamics are universally valid
  - Local particularities can be learned by providing time + space position as auxiliary information
What is a token?
What is a token?
What is a token?
What is a token?
What is a token?
What is a token?

- Token is small neighborhood in space-time
  - Small for token attention / interaction to be informative
  - Big enough so token has rich internal structure
What is a token?

- Token is small neighborhood in space-time
  - Small for token attention / interaction to be informative
  - Big enough so token has rich internal structure
- Token size is field-dependent
What is a token?

- Token is small neighborhood in space-time
  - Small for token attention / interaction to be informative
  - Big enough so token has rich internal structure
- Token size is field-dependent
- Multiple token sizes to provide multi-resolution structure and large contexts for neighborhood
Multiformer: respect the physical fields
Multiformer: respect the physical fields
Multiformer: respect the physical fields

Self attention
Multiformer: respect the physical fields
Multiformer: respect the physical fields
Multiformer: respect the physical fields

![Diagram of Multiformer model with Self-Attention and MLP layers for geopotential, temperature, divergence, and vorticity.
Multiformer: respect the physical fields
Multiformer: respect the physical fields
Multiformer: respect the physical fields
Multiformer: respect the physical fields

Self attention

Cross attention
Multiformer: respect the physical fields

- Attention maps:
Multiformer: respect the physical fields

- Attention maps:

  - Orography
  - Vorticity

  \[ \text{t-3} \quad \text{t-2} \quad \text{t-1} \quad \text{t} \]
Embedding of tokens

- Use non-trivial embedding network so that it models longer range effects and field interactions in a rich latent space
- Allow for tokens of different size in space-time
Embedding of tokens
Embedding of tokens
Embedding of tokens

- Use non-trivial embedding network so that multiformer can model longer range effects and field interactions in a rich latent space

- Allow for tokens of different size in space-time

=> Use small/medium-size standard transformer as embedding network
Embedding of tokens

- Allow for tokens of different size in space-time
Embedding of tokens

- Allow for tokens of different size in space-time
Embedding of tokens

- Allow for tokens of different size in space-time
Embedding of tokens

- Allow for tokens of different size in space-time
Embedding of tokens

- Allow for tokens of different size in space-time

Sub-tokens instead of pixels
Embedding of tokens

- Allow for tokens of different size in space-time

- Transformer takes an arbitrary number of tokens as input
  - Training yields consistent embedding
Training of embedding network

- Self-supervised training with variation of BERT masked language language model
Training of embedding network

- Self-supervised training with variation of BERT masked language model
  - Natural interpretation as forecasting / hindcasting / interpolation
Training of embedding network

- Self-supervised training with variation of BERT masked language language model
  - Natural interpretation as forecasting / hindcasting / interpolation
  - Performed on randomly cropped subset to obtain consistent embedding network for different sized tokens
AtmoRep training

Forecasting

BERT

BERT-large

Test loss
Statistical loss: respect the stochasticity

- Machine learning: Training on MSE/L$_2$ loss is problematic in terms of training dynamics
Statistical loss: respect the stochasticity

- Machine learning: Training on MSE/L$_2$ loss is problematic in terms of training dynamics
Statistical loss: respect the stochasticity

- Machine learning: Training on MSE/L\(_2\) loss is problematic in terms of training dynamics
Statistical loss: respect the stochasticity

- Respect the stochasticity in the dynamics
  - ML: Training on MSE/L₂ loss is problematic in terms of training dynamics
  - Training on just the mean is sub-optimal to learn a probabilistic/statistical representation of the dynamics and the system
Statistical loss: respect the stochasticity


Appendix for “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”

- 80% of the time: Replace the word with the [MASK] token, e.g., my dog is hairy → my dog is [MASK]
- 10% of the time: Replace the word with a random word, e.g., my dog is hairy → my dog is apple
- 10% of the time: Keep the word unchanged, e.g., my dog is hairy → my dog is hairy. The purpose of this is to bias the representation towards the actual observed word.

The advantage of this procedure is that the Transformer encoder does not know which words it will be asked to predict or which have been replaced by random words, so it is forced to keep a distributional contextual representation of every input token. Additionally, because random replacement only occurs for 1.5% of all tokens (i.e., 10% of 15%), this does not seem to harm the model’s language understanding capability. In Section C.2, we evaluate the impact this procedure.

Statistical loss: respect the stochasticity

- How to obtain better training dynamics and ensure probabilistic/statistical representation in network?
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity
Statistical loss: respect the stochasticity

training encourages statistical representation
Statistical loss: respect the stochasticity

- Machine learning: Training on MSE/L₂ loss is problematic in terms of training dynamics
Statistical loss: respect the stochasticity

- Machine learning: Training on MSE/L₂ loss is problematic in terms of training dynamics.
AtmoRep

- Large scale representation learning for atmospheric data
- Machine learning methodology needs to be completed and many open questions to be investigated
- Forecasting and projections
- How to integrate “raw“ observations?
- Coupled atmosphere+ocean system?
- ...

© Atmorep Collaboration, 2022
Network: attention maps


Statistical
Self-supervised training

- Task: Predict atmospheric state at time $t$ for a local neighborhood
- Use spatio-temporal neighborhood as data for prediction
  - 2-3 weeks for fine mesoscale predictions
  - Use data from hyperbolic cone around prediction target
  - All fields and all vertical layer (potentially masked for prediction task)
Transformers and attention


Transformers and attention
Transformers and attention
Transformers and attention
Transformers and attention
Transformers and attention
Fluid flow (vorticity)
Fluid flow (vorticity)

training with varying position and spherical eccentricity
Fluid flow (vorticity)
Fluid flow (vorticity)
Fluid flow

Prediction

Reference

Reference

Prediction 1

Prediction 2

© Atmorep Collaboration, 2022
Fluid flow
Fluid flow
Fluid flow

\[ \text{block}_1 = 1, \text{blocks}_1 = 1, \text{time} \text{(past steps)} \]

\[ \text{block}_1 = 1, \text{blocks}_1 = 1, \text{time} \text{(past steps)} \]
Fluid flow

$t-2$, $t-1$, $t$, $t+1$
Fluid flow

Important for prediction
Fluid flow

Important for prediction
Fluid flow

important for prediction
Fluid flow

important for prediction
Fluid flow

important for prediction

\[ \text{block} = 1, \text{heads} \]
Fluid flow

important for prediction
Fluid flow

\[ \text{block} = 1, \text{heads} \]

\[
\begin{align*}
\text{time} & \quad \text{(past steps)} \\
t-2 & \quad t-1 & \quad t & \quad t+1 \\
\text{important for prediction}
\end{align*}
\]
Fluid flow

important for prediction

\[ \text{block} = 1 \text{, heads} \]

\[ \text{time (past steps)} \]

\[ \text{fluid flow block} = 1 \text{, heads} \]

\[ \text{time (past steps)} \]

\[ t-2 \quad t-1 \quad t \quad t+1 \]
Fluid flow

Fluid flow is important for prediction.
Fluid flow

Fluid flow block = 1, heads time (past steps)

important for prediction

irrelevant for prediction
Fluid flow
Fluid flow

t-2

t-1

t
Fluid flow

$t-2$

$t-1$

$t$
Fluid flow
Fluid flow
Fluid flow
Atmospheric vorticity (ERA5)
Atmospheric vorticity (ERA5)
Atmospheric vorticity (ERA5)
Atmospheric vorticity (ERA5)
Atmospheric vorticity (ERA5)
Atmospheric vorticity (ERA5)
Atmospheric vorticity (ERA5)
Atmospheric vorticity (ERA5)
Atmospheric vorticity (ERA5)
Prediction (relative error, 1979)
Prediction
Prediction

relative prediction error

Latitude

relative prediction error

Latitude
Prediction

relative prediction error

Latitude

1979
2019
Code

- Code is stabilizing
- Infrastructure code is implemented and working
- Parallelization shows very good (weak) scaling
- Multiformer code
  - Other users: Yi Deng, Bing
  - Sharing currently ad-hoc and by hand
  - There seems to be a clear demand
Data pipeline

- Local approach: work on local spatio-temporal neighborhoods
- Size of neighborhoods is (overall) pre-determined, center in space and time is sampled randomly with area preserving sampling (correction for spherical distortion)
- Embarrassingly parallel: different parallel tasks sample independently
- Data is stored in monthly files: allows good randomization while still having sufficiently large files
Training: 1 task (local) versus 8 tasks (booster)

blur=5
Data normalization

- All fields are scaled to zero mean and unit variance
  - Per grid point
  - Per month
- Computations completed (1979-2019) and on booster.
Data normalization: vorticity

uncorrected

corrected
Data normalization: divergence

uncorrected versus corrected
Data normalization

uncorrected

corrected
Multiformer

- Implemented and functional
  - Some minor features could be added if necessary, e.g. variable number of coupling heads
  - Details of attention coupling still under investigation (see below)

```python
cf.fields = {
    'vorticity' : [1, 768, ['divergence', 'orography'] ],
    'divergence' : [1, 256, ['vorticity', 'orography'] ],
    'orography'  : [0, 96, []] }
```
Training: vorticity vs. vorticity + divergence

blur=5

vort, dim_embed=512
vort+div+oro, dim_embed=512
Training: vorticity vs. vorticity + divergence

\[ \text{blur}=5 \]
Training: vorticity vs. vorticity + divergence

blur=5

vort, dim_embed=1024

vort+div+oro, dim_embed=768
Multiformer

orography

vorticity

prediction
Multiformer

attention map vorticity
Multiformer

attention map orography
Multiformer

orography

vorticity

prediction
Multiformer

attention map vorticity
Multiformer

attention map orography
Training: tail net and embedding dimension

![Graph showing test loss for different models with dimensions and tails specified.]

- dim_emebd=768, #tail=8
- dim_embed=768, #tail=2
Training: tail net and embedding dimension

blur=5

dim_emebd=768, #tail=8

dim_emebd=1024, #tail=2

dim_emebd=768, #tail=2
Attention mechanism

- Standard encoder-decoder coupling (keys, value are fed in)
  - Mixes fields (but the same happens in PDEs)
  - In principle more expressive
  - Focus on field can be kept through loss
Attention mechanism

- Standard encoder-decoder coupling (keys, value are fed in)
  - Mixes fields (but the same happens in PDEs)
  - In principle more expressive
  - Focus on field can be kept through loss
- Query coupling (queries influence keys)
  - Fields are kept separate but still influence each other
  - Facilitates physical interpretation (fields are “clean”)
  - Difficulties when token size differs
Embedding network

- Implemented and functional
- Exploits that a transformer can have a variable length input
- Class token as compressed representation
- BERT-style training
  - Tail network predicts masked token just using class token
  - Variable size of space-time neighborhood (per batch)
- Proper evaluation still open
Training: BERT versus forecasting

BERT (with fixed and variable sequence length)
Training: BERT versus forecasting

blur=1
Challenges

- Slow development due to wait times on batch system
  - Long (2h+) runs required to see differences between models
  - Unclear how to construct restricted but representative development mode (restriction of space + time is not effective)
- Limited compute time budget
Challenges

- blur=5, Europe and June only, 1 GPU
- blur=5, full globe, 1 GPU
- blur=5, full globe, 8 A100 GPUs (cont’d)
Challenges

◦ Slow development due to wait times on batch system
  › Long (2h+) runs required to see differences between models
  › Unclear how to construct restricted but representative development mode (restriction of space + time is not effective)

◦ Limited compute time budget

=> Apply for Google cloud credit? 2-4 weeks decision time
Challenges

◦ Comparisons between models are often inconclusive
◦ Architectural changes seem to have little effect
◦ No over-fitting but larger model also does not improve predictions very much: bottleneck currently unclear
Loss function

- “Never train the mean”
  - Using L2 and KL loss improves results by about 20%
Loss function

\[ \text{L2 + KL} \]
Loss function

- “Never train the mean”
  - Using L2 and KL loss improves results by about 20%
- Classification
  - Labels are considered as Kronecker probability distribution
  - Network predicts distribution and cross entropy loss for network training
Loss function
Loss function


Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. In ICLR.


Appendix for “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”

- 80% of the time: Replace the word with the [MASK] token, e.g., my dog is hairy → my dog is [MASK]

- 10% of the time: Replace the word with a random word, e.g., my dog is hairy → my dog is apple

- 10% of the time: Keep the word unchanged, e.g., my dog is hairy → my dog is hairy. The purpose of this is to bias the representation towards the actual observed word.

The advantage of this procedure is that the Transformer encoder does not know which words it will be asked to predict or which have been replaced by random words, so it is forced to keep a distributional contextual representation of every input token. Additionally, because random replacement only occurs for 1.5% of all tokens (i.e., 10% of 15%), this does not seem to harm the model’s language understanding capability. In Section C.2, we evaluate the impact this procedure.
Loss function
Loss function
Loss function
Loss function
Next steps

◦ BERT training for multiformer
  › Integrate forecasting
◦ Improve performance of data loading
◦ Probabilistic loss
Next steps

◦ Embedding network
  › Train for all fields
  › Scale to large neighborhood
◦ Separable space-time attention
◦ Larger space-time neighborhoods
Next steps

- Downstream applications
  - Down scaling, hurricane tracking, ...
- Train with absolute or relative error?
- Move towards scientific evaluation
  - Proper statistical analysis (e.g. of spatial distribution of error)
  - How to turn attention maps into a principled tool?
Publication plan

◦ Workshops
  › AGU, ECMWF, Neurips, Italian meteorological society

◦ Papers
  › ICLR?
  › Attention maps, BERT for space-time, ...
  › Science journal?
  › Downstream applications?